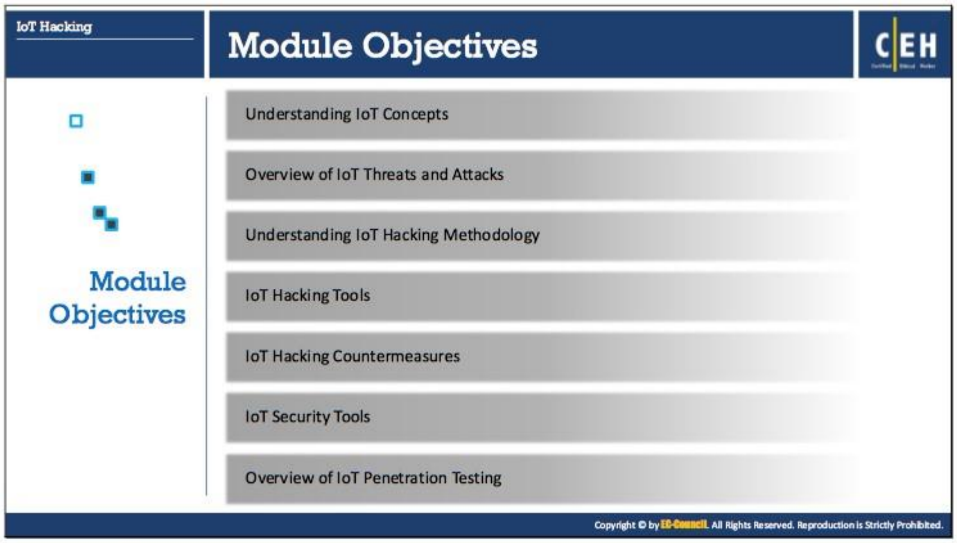
C|EH™
Certified Ethical Hacker

Module 18

# IoT Hacking

This page is intentionally left blank.



IoT Hacking

## Module Objectives

Module Objectives

- Understanding IoT Concepts
- Overview of IoT Threats and Attacks
- Understanding IoT Hacking Methodology
- IoT Hacking Tools
- IoT Hacking Countermeasures
- IoT Security Tools
- Overview of IoT Penetration Testing

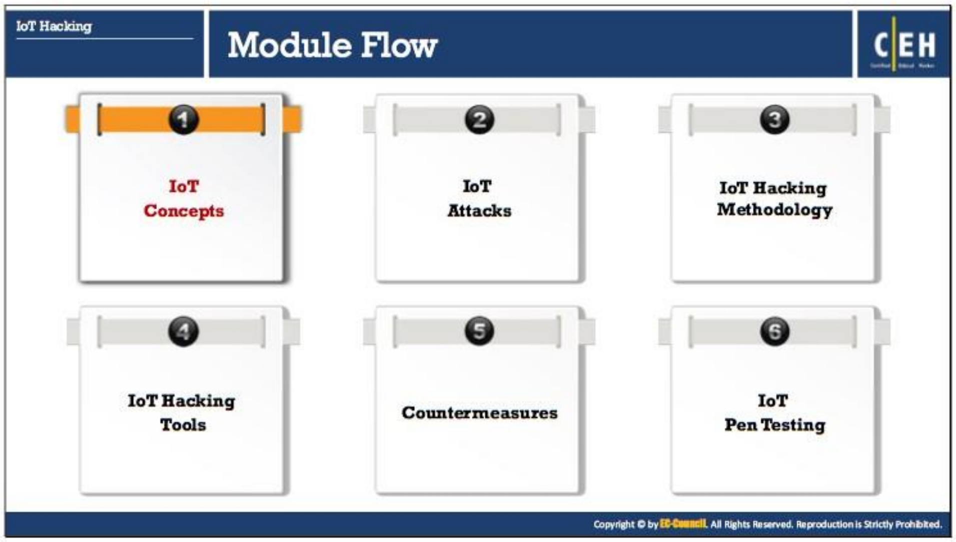
## Module Objectives

IoT has evolved from the convergence of wireless technology, micro-electromechanical systems, micro-services and Internet. IoT solutions are applied in different sectors of industry like healthcare, building management, agriculture, energy and transportation. Many organizations are driving the Internet of things transformation. IoT devices such as wearables, industrial appliances, connected electronic devices, smart grids, smart vehicles, etc. are becoming part of interconnected networks. These devices generate huge amount of data that is collected, analyzed, logged and stored on to the networks.

The Internet of things introduced a range of new technologies with associated capabilities into our daily lives. As IoT is an evolving technology, the immaturity of technologies and services provided by various vendors will have broad impact on the organizations leading to complex security issues. IoT security is difficult to ensure as the devices use simple processors and stripped down operating systems that may not support sophisticated security approaches. Organizations using these devices as part of their network need to protect both the devices and the information from attackers.

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

- Explain IoT concepts
- Understand different IoT threats and attacks
- Describe IoT hacking methodology
- Use different IoT hacking tools
- Apply countermeasures to prevent devices from IoT attacks
- Use different IoT security tools
- Perform IoT penetration testing



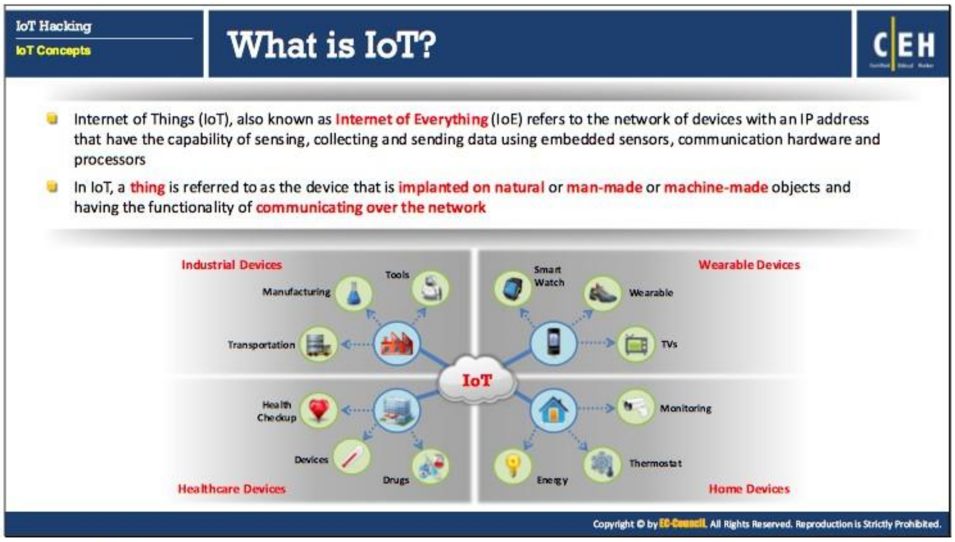
IoT Hacking

## Module Flow

1. IoT Concepts
2. IoT Attacks
3. IoT Hacking Methodology
4. IoT Hacking Tools
5. Countermeasures
6. IoT Pen Testing

## IoT Concepts

The Internet of Things (IoT) is an important and emerging topic in the field of technology, economics and in society in general. It is referred to as the web of connected devices, made possible from the intersection between machine-to-machine communications and big data analytics. The IoT is a future-facing development of the internet and abilities of physical devices that are eventually narrowing the gap between the virtual world and the physical world. This section deals with some of the important IoT concepts which one should be familiar with to understand the advanced topics covered later in this module.



IoT Hacking

IoT Concepts

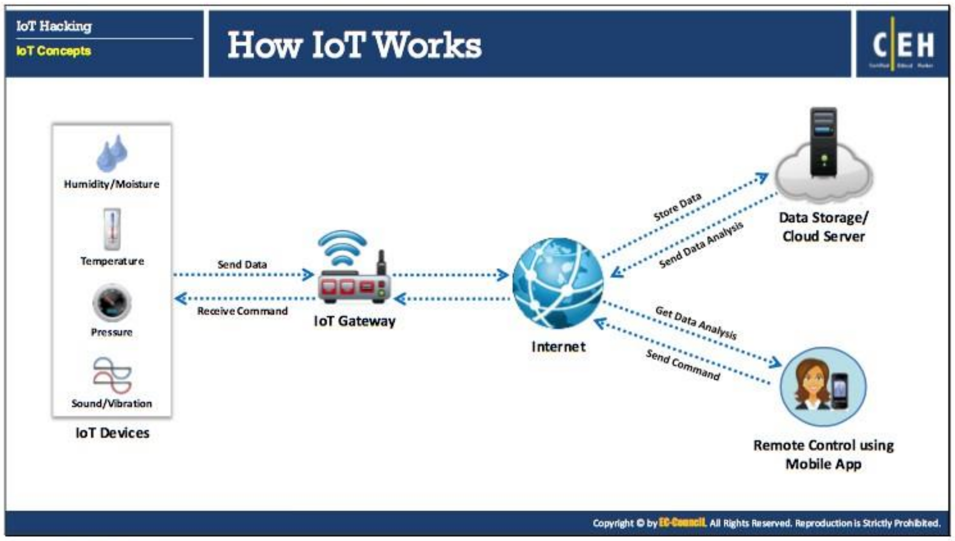
## What is IoT?

- Internet of Things (IoT), also known as **Internet of Everything** (IoE) refers to the network of devices with an IP address that have the capability of sensing, collecting and sending data using embedded sensors, communication hardware and processors
- In IoT, a **thing** is referred to as the device that is **implanted on natural** or **man-made** or **machine-made** objects and having the functionality of **communicating over the network**

### What is IoT?

Internet of Things (IoT), also known as Internet of Everything (IoE) refers to the computing devices that are web-enabled and have the capability of sensing, collecting and sending data using sensors, and the communication hardware and processors that are embedded within the device. In IoT, a thing is referred to as the device that is implanted on natural or man-made or machine-made objects and having the functionality of communicating over the network. IoT utilizes the existing emerging technology for sensing, networking, and robotics, therefore allowing the user to achieve deeper analysis, automation and integration within a system.

With the increase in the networking capabilities of machines and everyday appliances used in different sectors like offices, home, industries, transportation, buildings and wearable devices, they open up a world of opportunities for the betterment of business and better customer satisfaction. Some of the important key features of IoT are connectivity, sensors, artificial intelligence, small devices, and active engagement.



IoT Hacking
IoT Concepts

How IoT Works

## How IoT Works

IoT technology includes three primary systems such as IoT devices, gateway system, data storage system using Cloud and remote control using mobile apps. These systems together make the communication between two end points possible. Discussed below are some of the important components of IoT technology that play an essential role in the working of an IoT device:

- **Sensing Technology:** Sensors embedded in the devices sense a wide variety of information from their surroundings like temperature, gases, location, working of some industrial machine as well as sensing health data of a patient.
- **IoT Gateways:** Gateways are used to bridge the gap between the IoT device (internal network) and the end user (external network) and thus allowing them to connect and communicate with each other. The data collected by the sensors in IoT devices send the collected data to the concerned user or cloud through the gateway.
- **Cloud Server/Data Storage:** The collected data after travelling through the gateway arrives at the cloud, where it is stored and undergoes data analysis. The processed data is then transmitted to the user where he/she takes certain action based on the information received by him/her.
- **Remote Control using Mobile App:** The end user uses remote controls such as mobile phones, tabs, laptops, etc. installed with a mobile app to monitor, control, retrieve data, and take a specific action on IoT devices from a remote location.

**Example:**

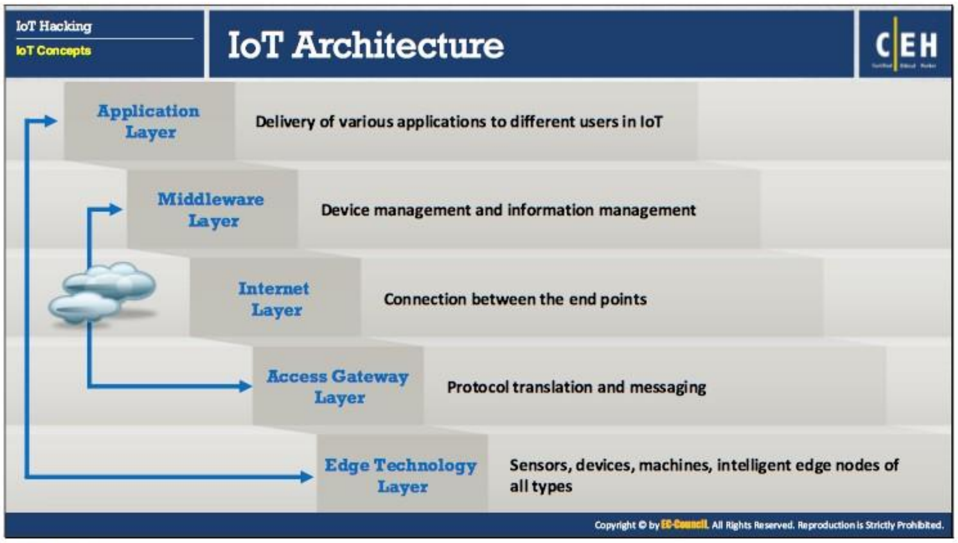
1. A smart security system installed in a home will be integrated with the gateway which in turn helps to connect the device to the Internet and the cloud infrastructure.

2. The data storage at the cloud has the information of each and every device connected to the network. The information possessed includes device's id, the present status of the device, who all accessed the device and for how many times. It also includes information like how long the device was accessed last time.

3. The connection with the cloud server is established through web services.

4. The user on the other side, who has the required app to access the device remotely on his mobile phone, interacts with it, which in turn makes him interact with the devices at home. Before accessing the device, he is asked to authenticate himself. If the credentials submitted by him match those saved in the cloud, he gets an access. Otherwise, his access is denied ensuring security. The cloud server identifies the device's id and sends a request associated with that device using gateways.

5. The security system that is currently recording the footage at home, if it senses any unusual activity, then it sends an alert to the cloud through the gateway, which matches the device's id and the user associated with it and finally the end user gets an alert.



IoT Hacking
IoT Concepts

## IoT Architecture

| | |
|---|---|
| Application Layer | Delivery of various applications to different users in IoT |
| Middleware Layer | Device management and information management |
| Internet Layer | Connection between the end points |
| Access Gateway Layer | Protocol translation and messaging |
| Edge Technology Layer | Sensors, devices, machines, intelligent edge nodes of all types |

## IoT Architecture

The Internet of Things architecture includes several layers starting from the Application layer at the top to the Edge Technology Layer at the bottom. These layers are designed in such a way that they can meet the requirements of various sectors like societies, industries, enterprises, governments, etc. The functions performed by each layer in the architecture are given below:

- **Edge Technology Layer**

  This layer consists of all the hardware parts like sensors, RFID tags, readers or other soft sensors and the device itself. These entities are the primary part of the data sensors that are deployed in the field for monitoring or sensing various phenomena. This layer plays an important part in data collection, connecting devices within the network and with the server.

- **Access Gateway Layer**

  This layer helps to bridge the gap between two end points like a device and a client. The very first data handling also takes place in this layer. It carries out message routing, message identification and subscribing.

- **Internet Layer**

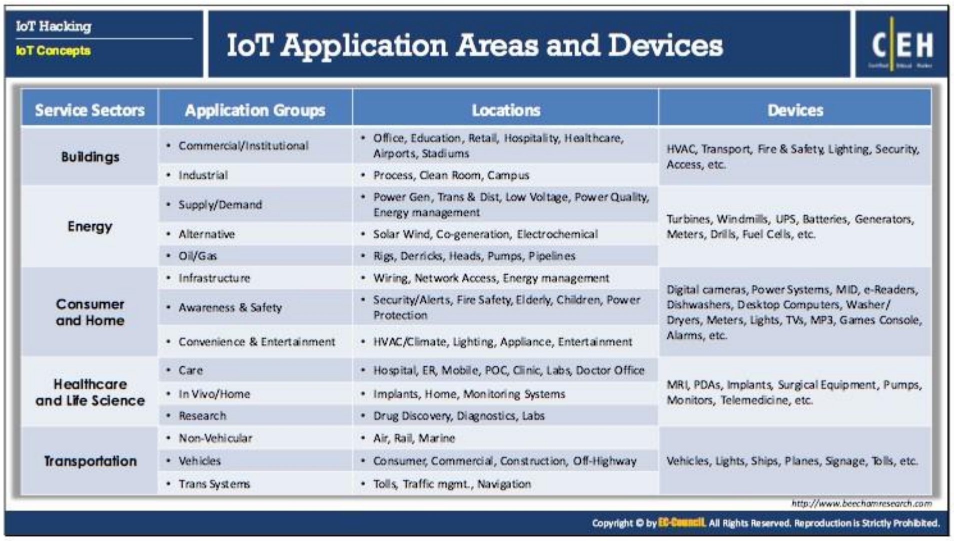
  This is the crucial layer as it serves as the main component in carrying out the communication between two end points such as device-to-device, device-to-cloud, device-to-gateway and back-end data-sharing.

- **Middleware Layer**

  This is one of the most critical layers that operates in two-way mode. As the name suggests this layer sits in the middle of the application layer and the hardware layer, thus behaving as an interface between these two layers. It is responsible for important functions such as data management, device management and various issues like data analysis, data aggregation, data filtering, device information discovery and access control.

- **Application Layer**

  This layer placed at the top of the stack, is responsible for the delivery of services to the respective users from different sectors like building, industrial, manufacturing, automobile, security, healthcare, etc.

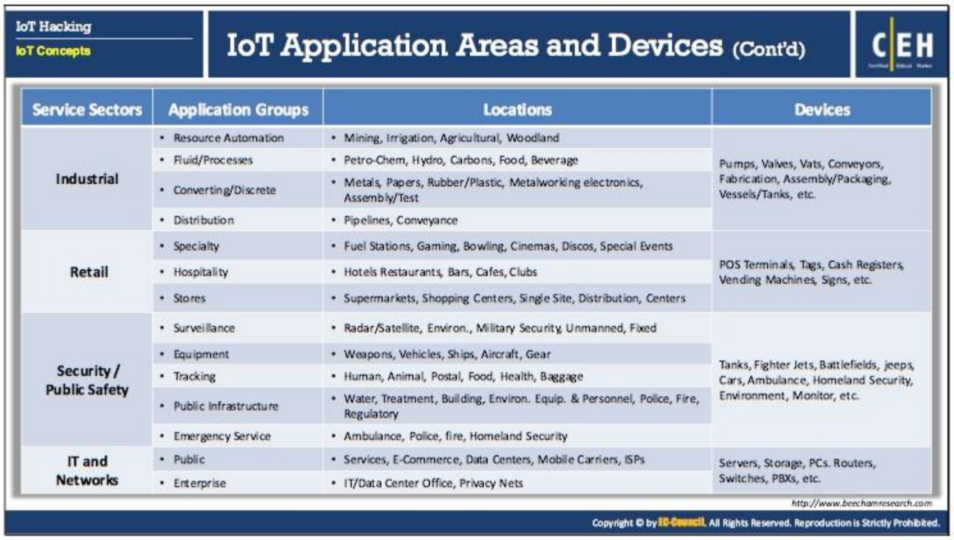


IoT Hacking
IoT Concepts

## IoT Application Areas and Devices

| Service Sectors | Application Groups | Locations | Devices |
|---|---|---|---|
| Buildings | • Commercial/Institutional | • Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums | HVAC, Transport, Fire & Safety, Lighting, Security, Access, etc. |
| | • Industrial | • Process, Clean Room, Campus | |
| Energy | • Supply/Demand | • Power Gen, Trans & Dist, Low Voltage, Power Quality, Energy management | Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc. |
| | • Alternative | • Solar Wind, Co-generation, Electrochemical | |
| | • Oil/Gas | • Rigs, Derricks, Heads, Pumps, Pipelines | |
| Consumer and Home | • Infrastructure | • Wiring, Network Access, Energy management | Digital cameras, Power Systems, MID, e-Readers, Dishwashers, Desktop Computers, Washer/ Dryers, Meters, Lights, TVs, MP3, Games Console, Alarms, etc. |
| | • Awareness & Safety | • Security/Alerts, Fire Safety, Elderly, Children, Power Protection | |
| | • Convenience & Entertainment | • HVAC/Climate, Lighting, Appliance, Entertainment | |
| Healthcare and Life Science | • Care | • Hospital, ER, Mobile, POC, Clinic, Labs, Doctor Office | MRI, PDAs, Implants, Surgical Equipment, Pumps, Monitors, Telemedicine, etc. |
| | • In Vivo/Home | • Implants, Home, Monitoring Systems | |
| | • Research | • Drug Discovery, Diagnostics, Labs | |
| Transportation | • Non-Vehicular | • Air, Rail, Marine | Vehicles, Lights, Ships, Planes, Signage, Tolls, etc. |
| | • Vehicles | • Consumer, Commercial, Construction, Off-Highway | |
| | • Trans Systems | • Tolls, Traffic mgmt., Navigation | |

http://www.beechamresearch.com



IoT Hacking
IoT Concepts

## IoT Application Areas and Devices (Cont'd)

| Service Sectors | Application Groups | Locations | Devices |
|---|---|---|---|
| Industrial | • Resource Automation | • Mining, Irrigation, Agricultural, Woodland | Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc. |
| | • Fluid/Processes | • Petro-Chem, Hydro, Carbons, Food, Beverage | |
| | • Converting/Discrete | • Metals, Papers, Rubber/Plastic, Metalworking electronics, Assembly/Test | |
| | • Distribution | • Pipelines, Conveyance | |
| Retail | • Specialty | • Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events | POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc. |
| | • Hospitality | • Hotels Restaurants, Bars, Cafes, Clubs | |
| | • Stores | • Supermarkets, Shopping Centers, Single Site, Distribution, Centers | |
| Security / Public Safety | • Surveillance | • Radar/Satellite, Environ., Military Security, Unmanned, Fixed | Tanks, Fighter Jets, Battlefields, jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc. |
| | • Equipment | • Weapons, Vehicles, Ships, Aircraft, Gear | |
| | • Tracking | • Human, Animal, Postal, Food, Health, Baggage | |
| | • Public Infrastructure | • Water, Treatment, Building, Environ. Equip. & Personnel, Police, Fire, Regulatory | |
| | • Emergency Service | • Ambulance, Police, fire, Homeland Security | |
| IT and Networks | • Public | • Services, E-Commerce, Data Centers, Mobile Carriers, ISPs | Servers, Storage, PCs. Routers, Switches, PBXs, etc. |
| | • Enterprise | • IT/Data Center Office, Privacy Nets | |

http://www.beechamresearch.com

### IoT Application Areas and Devices

Internet of Things devices have a wide range of applications. They are used in almost every sector of the society to assist in various things to simplify routine work, personal tasks and thus, improving the standard of living. IoT technology is included in smart homes and buildings, healthcare devices, industrial appliances, transportation, security devices, retail sector, etc.

Some of the applications of IoT devices are as follows:

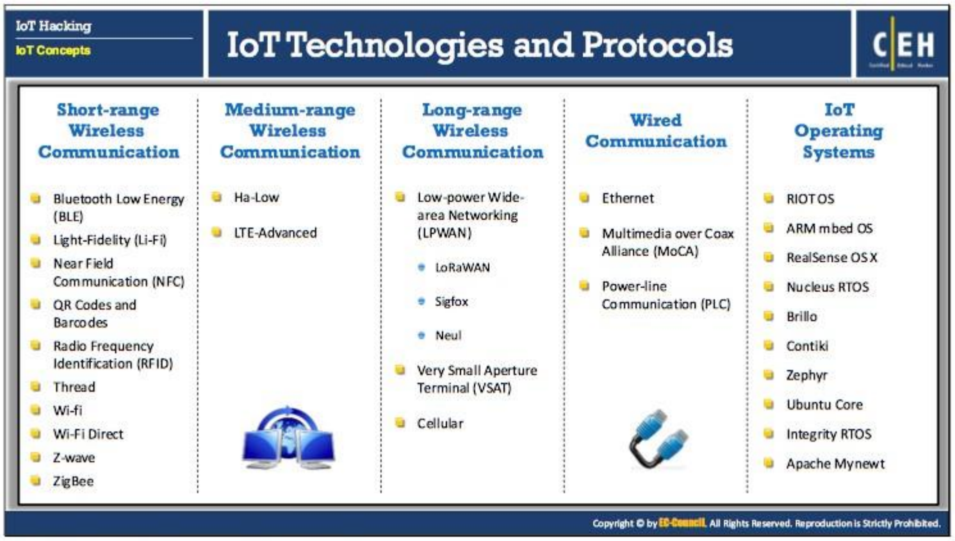
- The smart devices that are connected to the Internet, providing different services to end users include, thermostat, lighting system and security systems and several other systems that reside in buildings.
- In healthcare and life science sectors, devices like wearable devices, health monitoring devices like implanted heart pacemakers, ECG, EKG, surgical equipment, telemedicine, etc.
- The Industrial Internet of Things (IIoT) is capturing new growth through three approaches: Increasing production that boosts revenues, using intelligent technology that is entirely changing the way goods are made and creation of new hybrid business models.
- Similarly, using IoT technology transportation sector follows the concept of vehicle-to-vehicle, vehicle-to-roadside and vehicle-to-pedestrian communication, thus improving the traffic conditions, navigation system and parking schemes.
- IoT in retail is majorly used in payments, advertisements and tracking or monitoring products, thus protecting them from theft and loss, thereby increasing revenue.
- In IT and networks IoT devices mainly consist of various office machines like printers, faxing machines, copiers as well as monitoring of PBXs, thus, improving communication between endpoints and providing an ease of sending data across long distances.

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

| Service Sectors | Application Groups | Locations | Devices |
|---|---|---|---|
| **Buildings** | Commercial/ Institutional | Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums | HVAC, Transport, Fire & Safety, Lighting, Security, Access, etc. |
| | Industrial | Process, Clean Room, Campus | |
| **Energy** | Supply/ Demand | Power Gen, Trans & Dist, Low Voltage, Power Quality, Energy management | Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc. |
| | Alternative | Solar Wind, Co-generation, Electrochemical | |
| | Oil/Gas | Rigs, Derricks, Heads, Pumps, Pipelines | |
| **Consumer and Home** | Infrastructure | Wiring, Network Access, Energy management | Digital cameras, Power Systems, MID, e-Readers, Dishwashers, Desktop Computers, Washer/ Dryers, Meters, Lights, TVs, MP3, Games Console, Alarms, etc. |
| | Awareness & Safety | Security/Alerts, Fire Safety, Elderly, Children, Power Protection | |
| | Convenience & Entertainment | HVAC/Climate, Lighting, Appliance, Entertainment | |
| **Healthcare** | Care | Hospital, ER, Mobile, POC, Clinic, Labs, Doctor Office | MRI, PDAs, Implants, Surgical Equipment, |

| | | | |
|---|---|---|---|
| **and Life Science** | In Vivo/Home | Implants, Home, Monitoring Systems | Pumps, Monitors, Telemedicine, etc. |
| | Research | Drug Discovery, Diagnostics, Labs | |
| **Transportation** | Non-Vehicular | Air, Rail, Marine | Vehicles, Lights, Ships, Planes, Signage, Tolls, etc. |
| | Vehicles | Consumer, Commercial, Construction, Off-Highway | |
| | Trans Systems | Tolls, Traffic mgmt., Navigation | |
| **Industrial** | Resource Automation | Mining, Irrigation, Agricultural, Woodland | Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc. |
| | Fluid/ Processes | Petro-Chem, Hydro, Carbons, Food, Beverage | |
| | Converting/ Discrete | Metals, Papers, Rubber/Plastic, Metalworking electronics, Assembly/Test | |
| | Distribution | Pipelines, Conveyance | |
| **Retail** | Specialty | Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events | POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc. |
| | Hospitality | Hotels Restaurants, Bars, Cafes, Clubs | |
| | Stores | Supermarkets, Shopping Centers, Single Site, Distribution, Centers | |
| **Security / Public Safety** | Surveillance | Radar/Satellite, Environ., Military Security, Unmanned, Fixed | Tanks, Fighter Jets, Battlefields, jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc. |
| | Equipment | Weapons, Vehicles, Ships, Aircraft, Gear | |
| | Tracking | Human, Animal, Postal, Food, Health, Baggage | |
| | Public Infrastructure | Water, Treatment, Building, Environ. Equip. & Personnel, Police, Fire, Regulatory | |
| | Emergency Service | Ambulance, Police, fire, Homeland Security | |
| **IT and Networks** | Public | Services, E-Commerce, Data Centers, Mobile Carriers, ISPs | Servers, Storage, PCs. Routers, Switches, PBXs, etc. |
| | Enterprise | IT/Data Center Office, Privacy Nets | |

TABLE 18.1: IoT application areas and devices



IoT Hacking
IoT Concepts

## IoT Technologies and Protocols

**Short-range Wireless Communication**

- Bluetooth Low Energy (BLE)
- Light-Fidelity (Li-Fi)
- Near Field Communication (NFC)
- QR Codes and Barcodes
- Radio Frequency Identification (RFID)
- Thread
- Wi-fi
- Wi-Fi Direct
- Z-wave
- ZigBee

**Medium-range Wireless Communication**

- Ha-Low
- LTE-Advanced

**Long-range Wireless Communication**

- Low-power Wide-area Networking (LPWAN)
  - LoRaWAN
  - Sigfox
  - Neul
- Very Small Aperture Terminal (VSAT)
- Cellular

**Wired Communication**

- Ethernet
- Multimedia over Coax Alliance (MoCA)
- Power-line Communication (PLC)

**IoT Operating Systems**

- RIOT OS
- ARM mbed OS
- RealSense OS X
- Nucleus RTOS
- Brillo
- Contiki
- Zephyr
- Ubuntu Core
- Integrity RTOS
- Apache Mynewt

## IoT Technologies and Protocols

The IoT includes a wide range of new technologies and skills. The challenging problem in the IoT space is the immaturity of technologies with associated services and that of the vendors providing them. They lay a key challenge for the organizations exploiting the IoT. For a successful communication between two endpoints, IoT primarily exploits standard and networking protocols.

The major communication technologies and protocols with respect to the range between a source and the destination are as follow:

**Short Range Wireless Communication**

- **Bluetooth Low Energy (BLE):** Bluetooth LE or Bluetooth Smart is a wireless personal area network. This technology is designed to provide applications in various sectors like healthcare, security, entertainment, fitness, etc.
- **Light-Fidelity (Li-Fi):** Li-Fi is like Wi-Fi with only two differences: mode of communication and the speed. Li-Fi is a Visible Light Communications (VLC) system that uses common household light bulbs for data transfer at a very high speed of 224Gbps.
- **Near-field Communication (NFC):** NFC is a type of short range communication that uses magnetic field induction to enable communication between two electronic devices. It is basically used in connectionless mobile payment, social networking and in identification of documents or some product.
- **QR Codes and Barcodes:** These codes are machine readable tags that contains information about the product or item to which they are attached. Quick Response code or QR code is a two-dimensional code that stores product's information and it can be

scanned using smart phones whereas Barcode comes in both, one dimensional (1D) and two-dimensional (2D) code.

- **Radio Frequency Identification (RFID):** RFID stores data in tags that are read using electromagnetic fields. RFID is used in many sectors like industrial, offices, companies, automobile, pharmaceuticals, livestock and pets.
- **Thread:** Thread is an IPv6 based networking protocol for IoT devices. Its main aim is home automation, so that the devices can communicate with each other on local wireless networks.
- **Wi-Fi:** Wi-Fi is a technology that is widely used in wireless local area networking or LAN. Presently, the most common Wi-Fi standard that is used in homes or companies is 802.11n which offers a maximum speed of 600 Mbps and range of approximately 50 meters.
- **Wi-Fi Direct:** It is used for peer-to-peer communication without the need of a wireless access point. The Wi-Fi direct devices start communication only after deciding which device will act as an access point. s.
- **Z-Wave:** Z-Wave is a low power, short-range communication designed primarily for home automation. It provides a simple and reliable way to wirelessly monitor and control household devices like HVAC, thermostat, garage, home cinema etc.
- **Zig-Bee:** It is another short-range communication protocol based on IEEE 203.15.4 standard.

  Zig-Bee is for the devices that transfer data infrequently at low data-rate in a restricted area and within a range of 10-100 meters.

**Medium Range Wireless Communication**

- **HaLow:** It is another variant of Wi-Fi standard that provides extended range, making it useful for communications in rural areas. It offers low data rates, thus reducing power and cost for transmission.
- **LTE- Advanced:** LTE-Advanced is a standard for mobile communication that provides enhancement to LTE thus focusing on providing higher capacity in terms of data rate, extended range, efficiency and performance.

**Long Range Wireless Communication**

- **LPWAN:** Low Power Wide Area Networking (LPWAN) is a type of wireless telecommunication network, designed in such a way so as to provide long-range communications between two end points. Available LPWAN protocols and technologies include:
  - **LoRaWAN:** Low Power Wide Area Network (LoRaWAN) is used to support applications such as mobile, industrial machine-to-machine and secure two-way communications for IoT devices, smart cities and healthcare applications.
  - **Sigfox:** It is used in devices that have small battery life and need to transfer low level of data.

- **Neul:** It is used in a tiny part of the TV white space spectrum to deliver high quality, high power, high coverage and low-cost networks.

- **Very Small Aperture Terminal (VSAT):** VSAT is a communication protocol that is used for data transfer using small dish antennas for both broadband data and narrowband data.

- **Cellular:** Cellular is a type of communication protocol that is used for communication over a longer distance. It is used to send high-quality data but with a cost of being expensive and high consumption of power.

**Wired Communication**

- **Ethernet:** Ethernet is the most commonly used type of network protocol today. It is a type of LAN (Local Area Network) which refers to a wired connection of computers in a small building, office or on a campus.

- **Multimedia over Coax Alliance (MoCA):** MoCA is a type of network protocol that provides a high definition video of home and content related to it over the existing coaxial cable.

- **Power-line Communication (PLC):** It is type of protocol where electrical wires are used to transmit power and data from one end point to another end point. PLC is required for applications in different areas like home automation, industrial devices and for broadband over power lines (BPL).

**IoT Operating Systems**

IoT devices consist of both hardware and software components. Hardware components include end devices and gateways whereas software part includes operating systems. Due to increase in production of hardware components (gateways, sensor nodes, etc.), traditional IoT devices that previously used to run without an OS, started adopting new OS implementations that are specially programmed for IoT devices. These operating systems provide connectivity, usability and interoperability to the devices.

Given below are some of the operating systems used by IoT devices:

- **RIOT OS**: It has less resource requirement and uses energy efficiently. It has an ability of running on embedded systems, actuator boards, sensors, etc.

- **ARM mbed OS**: It is mostly used for low-powered devices like wearable devices.

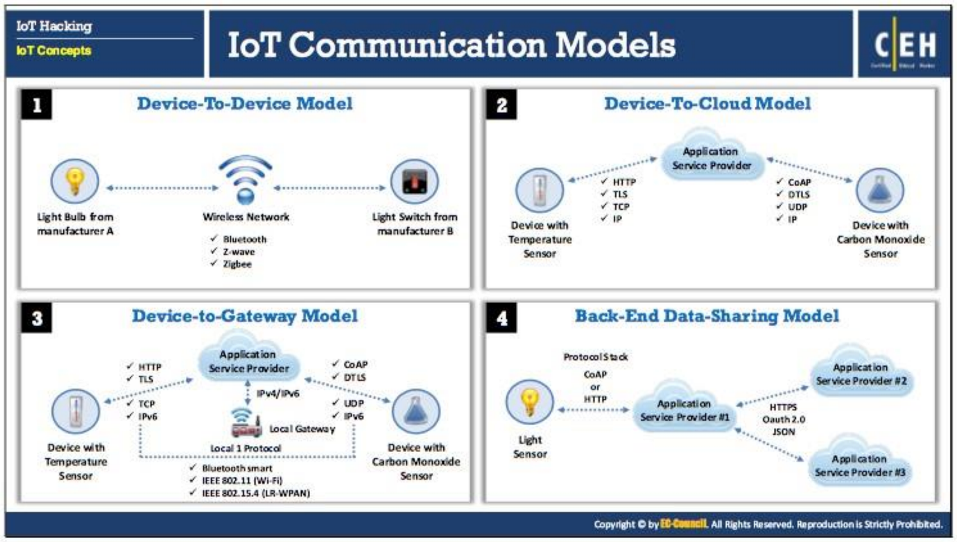
- **RealSense OS X**: It is used in Intel's depth sensing technology. Therefore, it is implemented in cameras, sensors, etc.

- **Nucleus RTOS**: Primarily used in aerospace, medical and industrial applications.

- **Brillo**: It is an android based embedded OS, used in low-end devices such as thermostats.

- **Contiki**: It is used in low-power wireless devices such as street lighting, sound monitoring systems, etc.

- **Zephyr**: It is used in low power and resource constrained devices.
- **Ubuntu Core**: Also known as Snappy, it is used in robots, drones, edge gateways, etc.
- **Integrity RTOS**: Primarily used in aerospace or defense, industrial, automotive and medical sectors.
- **Apache Mynewt**: It supports devices that work on Bluetooth Low Energy protocol.



IoT Hacking
IoT Concepts

## IoT Communication Models

CEH
1
Device-To-Device Model
Light Bulb from manufacturer A
Wireless Network
✓ Bluetooth
✓ Z-wave
✓ Zigbee
Light Switch from manufacturer B
2
Device-To-Cloud Model
Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IP
Device with Temperature Sensor
✓ CoAP
✓ DTLS
✓ UDP
✓ IP
Device with Carbon Monoxide Sensor
3
Device-to-Gateway Model
Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IPv6
IPv4/IPv6
✓ CoAP
✓ DTLS
✓ UDP
✓ IPv6
Local Gateway
Device with Temperature Sensor
Local 1 Protocol
✓ Bluetooth smart
✓ IEEE 802.11 (Wi-Fi)
✓ IEEE 802.15.4 (LR-WPAN)
Device with Carbon Monoxide Sensor
4
Back-End Data-Sharing Model
Protocol Stack
CoAP
or
HTTP
Light Sensor
Application Service Provider #1
HTTPS
Oauth 2.0
JSON
Application Service Provider #2
Application Service Provider #3

## IoT Communication Models

IoT technology uses different technical communication models each having its own characteristics. These models highlight the flexibility in the way these IoT devices can communicate with each other or with the client. Discussed below are four communication models and key characteristics associated with each model:

- **Device-to-Device Communication Model**

  In this type of communication, devices that are connected interact with each other through the internet but mostly they use protocols like ZigBee, Z-Wave or Bluetooth. Device-to-Device communication is most commonly used in the smart home devices like a thermostat, Light Bulb, Door-locks, CCTV cameras, Fridge, etc. where these devices transfer small data packets to each other at a low data rate. This model is also popular in communication between wearable devices. For example, an ECG/EKG device attached to the body of a patient will be paired to his/her smartphone and will send him/her notifications in an emergency.

- **Device-to-Cloud Communication Model**

  In this type of communication, devices communicate with the cloud directly rather than directly communicating with the client in order to send or receive the data or commands. It uses communication protocols such as Wi-Fi or Ethernet and sometimes uses Cellular as well.

  A case for Wi-Fi based Device-to-Cloud communication would be a CCTV camera which can be accessed on the smartphone from a remote location. In this scenario the device (here CCTV camera) cannot directly communicate with the client, rather it first sends data to the cloud and then if the client inputs correct credentials, he is then allowed to access the cloud which in turn allows him to access the device at his home.

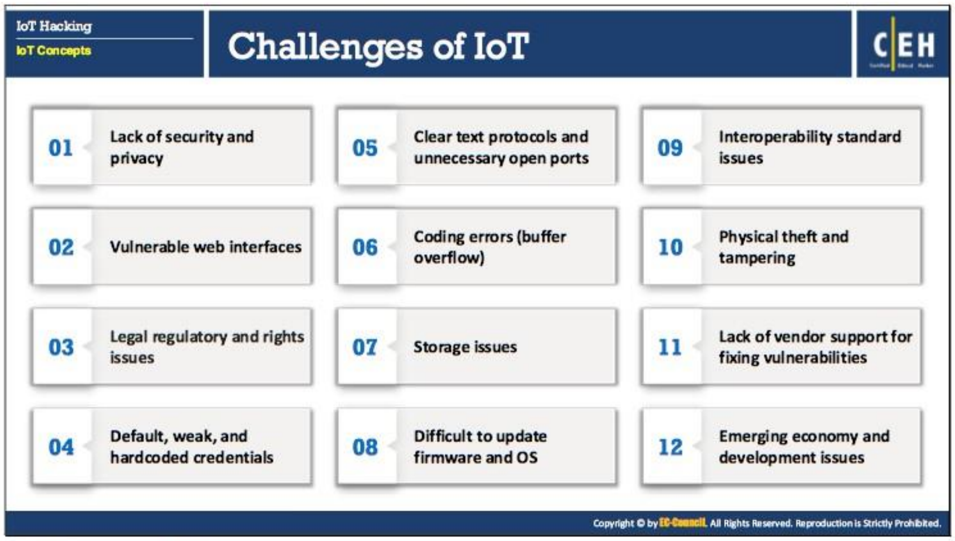
- **Device-to-Gateway Communication Model**

  In the Device-to-Gateway communication, Internet of Things device communicates with an intermediate device called a Gateway, which in turn communicates with the cloud service. This gateway device could be a Smartphone or a Hub that is acting as an intermediate point, also provides security features and data or protocol translation. The protocols generally used in this mode of communication are ZigBee and Z-Wave.

  If the application layer gateway is the smartphone, then it might take the form of an app that interacts with an IoT device and with the cloud. This device might be a smart TV that connects to the cloud service through a mobile phone app.

- **Back-End Data-Sharing Communication Model**

  This type of communication model extends the device-to-cloud communication type in which the data from the IoT devices can be accessed by authorized third parties. Here devices upload their data onto the cloud which is later accessed or analyzed by the third parties. An example of this model would be an analyzer of a company that analyzes the yearly or monthly energy consumption of a company. Later the analysis can be used to bring down the cost of company's expenditure on energy by following certain energy harvesting or saving techniques.



IoT Hacking
IoT Concepts

Challenges of IoT

01 Lack of security and privacy

02 Vulnerable web interfaces

03 Legal regulatory and rights issues

04 Default, weak, and hardcoded credentials

05 Clear text protocols and unnecessary open ports

06 Coding errors (buffer overflow)

07 Storage issues

08 Difficult to update firmware and OS

09 Interoperability standard issues

10 Physical theft and tampering

11 Lack of vendor support for fixing vulnerabilities

12 Emerging economy and development issues

## Challenges of IoT

IoT technology is growing so quickly that it has become ubiquitous. With lots of applications and features but a lack of basic security policies, IoT devices today are easy prey for hackers. In addition, the upgrades in IoT devices have introduced new security flaws that can easily be exploited by hackers. To overcome this big issue, manufacturing companies should consider security as the top most priority, starting with planning, design and all the way up to deployment, implementation, management and maintenance.

Discussed below are some of the challenges of IoT devices that make them vulnerable to many threats:

- **Lack of Security and Privacy:** Most of the IoT devices today such as household devices, industrial devices, healthcare devices, automobiles and so on are connected to the internet and these devices contain important and confidential data. These devices lack even basic security and privacy policies so that the hackers can exploit this lack to carry out some malicious activity.
- **Vulnerable Web Interfaces:** Many IoT devices come with embedded web server technology that makes them vulnerable to attacks.
- **Legal Regulatory and Rights Issue:** Due to the interconnection of IoT devices certain security issues are raised with no existing legal laws that address these issues.
- **Default, Weak and Hardcoded Credentials:** One of the most common reasons for cyber-attacks on IoT is its authentication system. These devices usually come with default and weak credentials, which can easily be exploited by a hacker to gain unauthorized access to the device.

- **Clear Text Protocols and Unnecessary Open Ports:** IoT devices lack encryption techniques during transmission of data which at times makes them use certain protocols that transmit data in clear text in addition to having open ports.
- **Coding Errors (Buffer overflow):** Most of the IoT devices today have embedded web services that are subjected to the same vulnerabilities that are commonly exploited on web services platforms. As a result, updating such functionality may give rise to issues like buffer overflows, SQL injection, etc. within technology infrastructure.
- **Storage Issues:** IoT devices generally come with smaller data storage capacity, but the data collected and transmitted by the devices is limitless. Therefore, this gives rise to data storage, management and protection issues.
- **Difficult to Update Firmware and OS:** Upgrading firmware is an essential step towards countering vulnerabilities in a device but this upgrading may break a device's functionality. For this reason, the developers or the manufacturing companies may hesitate or even refuse to get product support or make adjustments during the development phase of their products.
- **Interoperability Standard Issues:** One of the biggest obstacles for IoT devices is the Interoperability issue which is a key to the viability and long-term growth of the entire IoT ecosystem. The issues that arise due to lack of interoperability in IoT devices are the inability of manufacturers to test APIs using common methods and mechanisms, their inability to secure devices using software from third parties and their inability to manage and monitor devices using a common layer.
- **Physical Theft and Tampering:** Physical attacks on the IoT devices include tampering with the devices to inject malicious code or files to make the device work the way attacker wants or make hardware modifications to the devices. Counterfeiting the devices may also be the issue when proper physical protection is not there to shield the devices.
- **Lack of Vendor Support for Fixing Vulnerabilities:** The firmware of the devices has to be upgraded in order to protect the devices against certain vulnerabilities but vendors are hesitant or they usually refuse to get a third-party access to their devices.
- **Emerging Economy and Development Issues:** With wide spread opportunities of IoT devices in every field, it adds multiple layers of complexity for the policy makers. The new environment of these devices adds a new dimension for the policy makers who would have to design new blueprints and policies for IoT devices.

IoT Hacking
IoT Concepts

## Threat vs Opportunity

CEH

If **MISCONFIGURED** and **MISAPPREHEND**, IoT poses an unprecedented risk to personal data, privacy and safety

If **APPREHENDED** and **PROTECTED**, IoT will boost transmissions, communications, delivery of services and standard of living

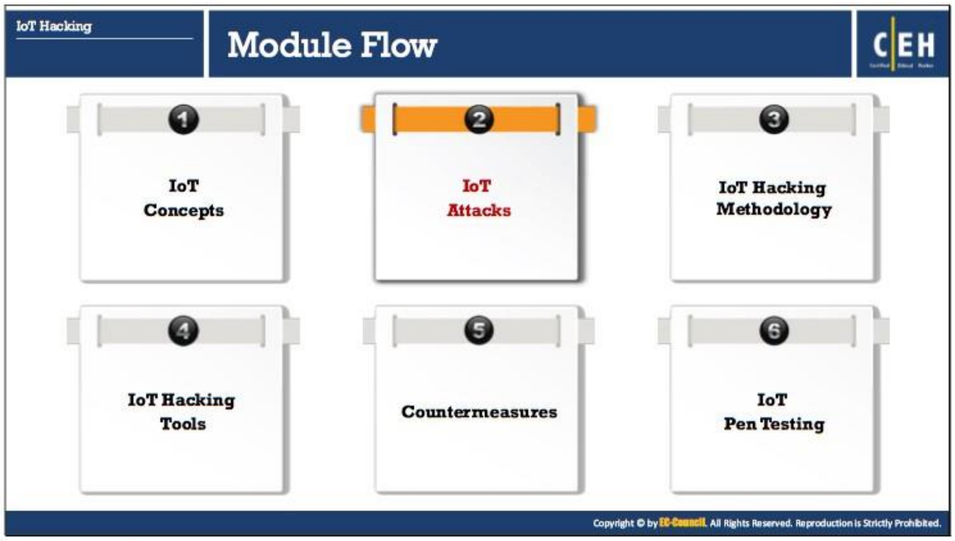
### Threat vs Opportunity

If **MISCONFIGURED** and **MISAPPREHENDED**, IoT poses an unprecedented risk to personal data, privacy and safety. If **APPREHENDED** and **PROTECTED**, IoT will boost transmissions, communications, delivery of services and standard of living.

The threats of IoT can be listed down in three primary categories: Security, Privacy and Safety. All these categories are interrelated as they deal with the same device and its connectivity. The importance of these categories can be understood as the IoT devices are fast becoming more pervasive in our lives than the smart phones and will have access to the most confidential or sensitive personnel information such as health records, financial records and social security numbers.

For instance, when it comes to smart mobiles or tablets we have only a couple of concerns while if we possess any IoT device then the concerns would quickly multiply in number. Therefore, considering what IoTs can access, security, privacy and safety are of paramount importance.

If these three categories of threat are prioritized and a number of required techniques are followed in order to overcome these issues then it will result in enhanced and secure communication between two endpoints, fewer cyber-attacks on the devices, good user experience and will also result in cost savings and efficiency gains.



IoT Hacking

## Module Flow

1 IoT Concepts

2 IoT Attacks

3 IoT Hacking Methodology

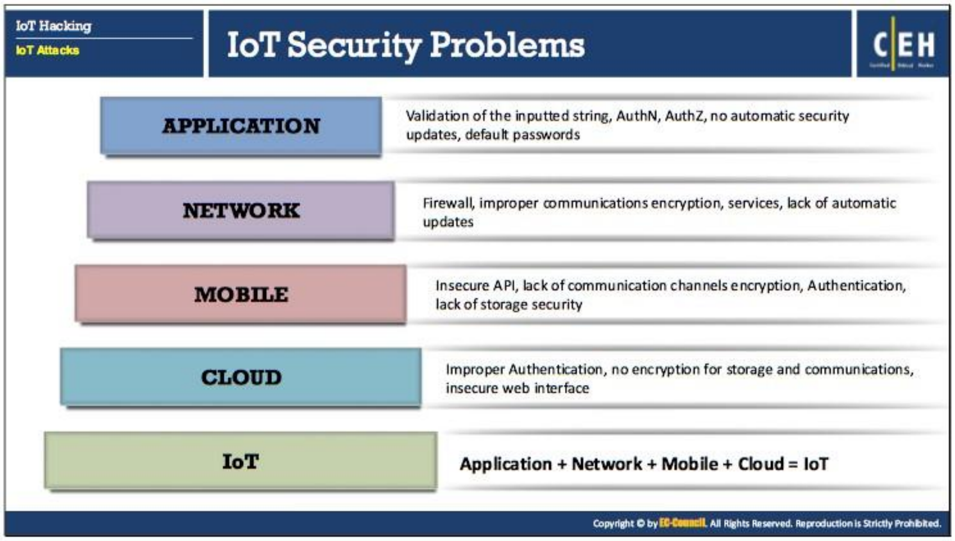
4 IoT Hacking Tools

5 Countermeasures

6 IoT Pen Testing

## IoT Attacks

Attackers implement various techniques to launch attacks on e target IoT devices or networks. This section discusses top IoT threats with the basic types of IoT attack vectors and techniques that include DDoS attack, attacks on HVAC systems, rolling code attack, BlueBorne attack, jamming attack, etc.



IoT Hacking
IoT Attacks

## IoT Security Problems

**APPLICATION** — Validation of the inputted string, AuthN, AuthZ, no automatic security updates, default passwords

**NETWORK** — Firewall, improper communications encryption, services, lack of automatic updates

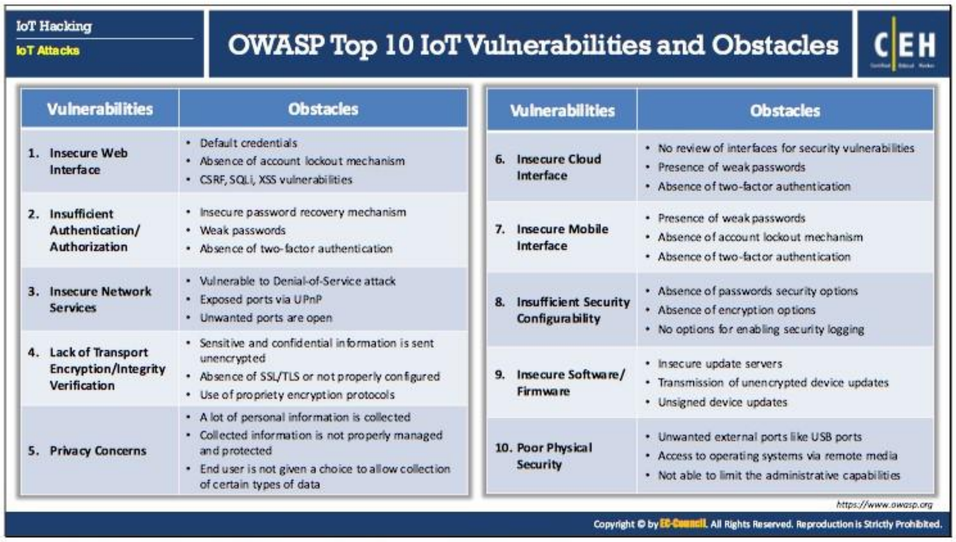
**MOBILE** — Insecure API, lack of communication channels encryption, Authentication, lack of storage security

**CLOUD** — Improper Authentication, no encryption for storage and communications, insecure web interface

**IoT** — **Application + Network + Mobile + Cloud = IoT**

### IoT Security Problems

Potential vulnerabilities in the IoT system can result in major problems for the organizations. Most of the IoT devices come with security issues such as absence of proper authentication mechanism or use of default credentials, absence of lock-out mechanism, absence of strong encryption scheme, absence of proper key management systems, improper physical security, etc.



IoT Hacking
IoT Attacks

OWASP Top 10 IoT Vulnerabilities and Obstacles

| Vulnerabilities | Obstacles | Vulnerabilities | Obstacles |
|---|---|---|---|
| 1. Insecure Web Interface | • Default credentials<br>• Absence of account lockout mechanism<br>• CSRF, SQLi, XSS vulnerabilities | 6. Insecure Cloud Interface | • No review of interfaces for security vulnerabilities<br>• Presence of weak passwords<br>• Absence of two-factor authentication |
| 2. Insufficient Authentication/ Authorization | • Insecure password recovery mechanism<br>• Weak passwords<br>• Absence of two-factor authentication | 7. Insecure Mobile Interface | • Presence of weak passwords<br>• Absence of account lockout mechanism<br>• Absence of two-factor authentication |
| 3. Insecure Network Services | • Vulnerable to Denial-of-Service attack<br>• Exposed ports via UPnP<br>• Unwanted ports are open | 8. Insufficient Security Configurability | • Absence of passwords security options<br>• Absence of encryption options<br>• No options for enabling security logging |
| 4. Lack of Transport Encryption/Integrity Verification | • Sensitive and confidential information is sent unencrypted<br>• Absence of SSL/TLS or not properly configured<br>• Use of propriety encryption protocols | 9. Insecure Software/ Firmware | • Insecure update servers<br>• Transmission of unencrypted device updates<br>• Unsigned device updates |
| 5. Privacy Concerns | • A lot of personal information is collected<br>• Collected information is not properly managed and protected<br>• End user is not given a choice to allow collection of certain types of data | 10. Poor Physical Security | • Unwanted external ports like USB ports<br>• Access to operating systems via remote media<br>• Not able to limit the administrative capabilities |

*https://www.owasp.org*

## OWASP Top 10 IoT Vulnerabilities and Obstacles

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

The OWASP top 10 IoT vulnerabilities are listed below:

- **Insecure Web Interface**

  Insecure web interface occurs when certain issues arise such as weak credentials, lack of account lockout mechanism and account enumeration. These issues result in loss of data, loss of privacy, lack of accountability, denial of access and complete device access takeover.

- **Insufficient Authentication/Authorization**

  Insufficient authentication refers to using weak credentials such as an insecure or weak password which offers poor security, thus allowing a hacker to gain access to the user account, and causing loss of data, loss of accountability and denying user to access the account.

- **Insecure Network Services**

  Insecure network services are prone to various attacks like buffer overflow attacks, attacks that cause denial-of-service scenario, thus leaving the device inaccessible to the user. An attacker uses various automated tools such as port scanners and fuzzers to detect the open ports and exploit them to gain unauthorized access to the services.

- **Lack of Transport Encryption/Integrity Verification**

  Due to lack of message encryption techniques in the transmission of data, the data can be easily intercepted and viewed which can result in information loss and based on the exposed data, the IoT device or user accounts can be compromised.

- **Privacy Concerns**

  IoT devices generate some private and confidential data but due to lack of proper protection schemes, it leads to privacy concerns, which makes it is easy to discover and review the data that is being produced, sent, and collected.

- **Insecure Cloud Interface**

  An insecure cloud interface is available when easy to guess credentials are used for a user account. A hacker exploits the insufficient authentication mechanism and lack of proper transport encryption to access data or control the user account.

- **Insecure Mobile Interface**

  An insecure mobile interface is present if the credentials are easy to guess and account enumeration is possible. Insecure mobile interfaces are easy to find by simply reviewing the connection to the wireless network and identifying if SSL is in use.

- **Insufficient Security Configurability**

  This kind of issue arises when the device user lacks the ability to change the security controls in an IoT device which can increase the device's vulnerability, thus making it an easy target for hackers to exploit.

- **Insecure Software/Firmware**

  Due to the lack of ability of a device to update itself when vulnerabilities or security loopholes are discovered, there exists a major security concern. An attacker via unencrypted connections, can capture an update file and perform a malicious activity such as tampering with the file content via DNS hijacking.

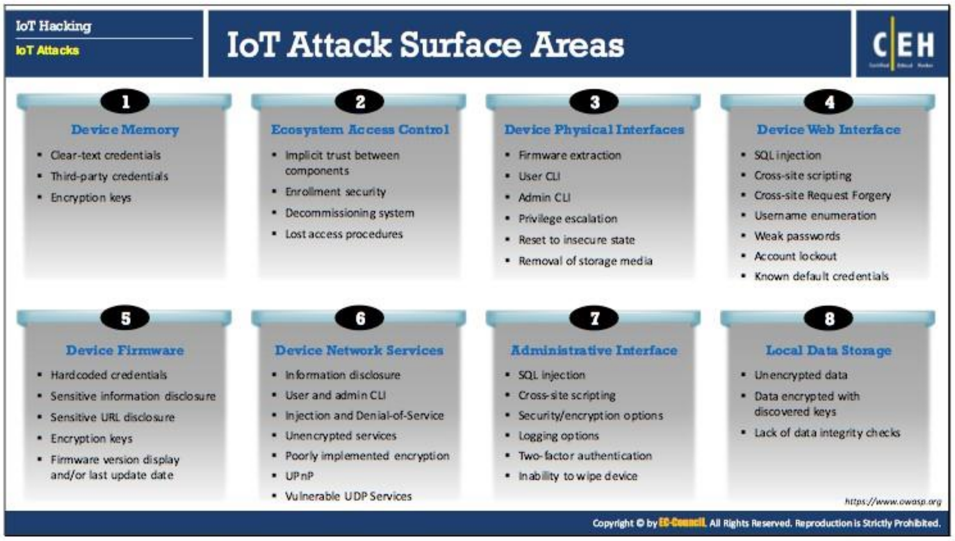
- **Poor Physical Security**

  Physical security concerns arise if an attacker physically accesses the device and accesses the data stored in it. Such issues also take place if the device is accessed via a USB port or SD card or via some other hardware device.

| Vulnerabilities | Obstacles |
|---|---|
| **1. Insecure Web Interface** | ▪ Default credentials<br>▪ Absence of account lockout mechanism<br>▪ CSRF, SQLi, XSS vulnerabilities |
| **2. Insufficient Authentication/ Authorization** | ▪ Insecure password recovery mechanism<br>▪ Weak Passwords<br>▪ Absence of two-factor authentication |

| | |
|---|---|
| **3. Insecure Network Services** | ▪ Vulnerable to Denial-of-Service attack<br>▪ Exposed ports via UPnP<br>▪ Unwanted ports are open |
| **4. Lack of Transport Encryption/Integrity Verification** | ▪ Sensitive and confidential information is sent unencrypted<br>▪ Absence of SSL/TLS or not properly configured<br>▪ Use of propriety encryption protocols |
| **5. Privacy Concerns** | ▪ A lot of personal information is collected<br>▪ Collected information is not properly managed and protected<br>▪ End user is not given a choice to allow collection of certain types of data |
| **6. Insecure Cloud Interface** | ▪ No review of interfaces for security vulnerabilities<br>▪ Presence of weak passwords<br>▪ Absence of two-factor authentication |
| **7. Insecure Mobile Interface** | ▪ Presence of weak passwords<br>▪ Absence of account lockout mechanism<br>▪ Absence of two-factor authentication |
| **8. Insufficient Security Configurability** | ▪ Absence of passwords security options<br>▪ Absence of encryption options<br>▪ No options for enabling security logging |
| **9. Insecure Software / Firmware** | ▪ Insecure update servers<br>▪ Transmission of unencrypted device updates<br>▪ Unsigned device updates |
| **10. Poor Physical Security** | ▪ Unwanted external ports like USB ports<br>▪ Access to operating systems via remote media<br>▪ Not able to limit the administrative capabilities |

TABLE 18.2: OWASP top 10 IoT vulnerabilities and obstacles



IoT Hacking
IoT Attacks

## IoT Attack Surface Areas

**1 Device Memory**
- Clear-text credentials
- Third-party credentials
- Encryption keys

**2 Ecosystem Access Control**
- Implicit trust between components
- Enrollment security
- Decommissioning system
- Lost access procedures

**3 Device Physical Interfaces**
- Firmware extraction
- User CLI
- Admin CLI
- Privilege escalation
- Reset to insecure state
- Removal of storage media

**4 Device Web Interface**
- SQL injection
- Cross-site scripting
- Cross-site Request Forgery
- Username enumeration
- Weak passwords
- Account lockout
- Known default credentials

**5 Device Firmware**
- Hardcoded credentials
- Sensitive information disclosure
- Sensitive URL disclosure
- Encryption keys
- Firmware version display and/or last update date

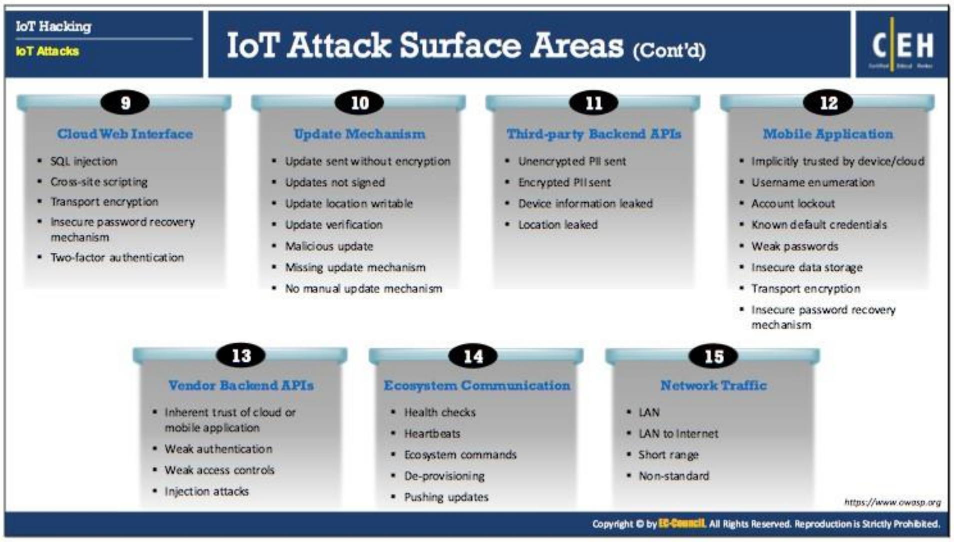
**6 Device Network Services**
- Information disclosure
- User and admin CLI
- Injection and Denial-of-Service
- Unencrypted services
- Poorly implemented encryption
- UPnP
- Vulnerable UDP Services

**7 Administrative Interface**
- SQL injection
- Cross-site scripting
- Security/encryption options
- Logging options
- Two-factor authentication
- Inability to wipe device

**8 Local Data Storage**
- Unencrypted data
- Data encrypted with discovered keys
- Lack of data integrity checks

https://www.owasp.org

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



IoT Hacking
IoT Attacks

## IoT Attack Surface Areas (Cont'd)

**9 Cloud Web Interface**
- SQL injection
- Cross-site scripting
- Transport encryption
- Insecure password recovery mechanism
- Two-factor authentication

**10 Update Mechanism**
- Update sent without encryption
- Updates not signed
- Update location writable
- Update verification
- Malicious update
- Missing update mechanism
- No manual update mechanism

**11 Third-party Backend APIs**
- Unencrypted PII sent
- Encrypted PII sent
- Device information leaked
- Location leaked

**12 Mobile Application**
- Implicitly trusted by device/cloud
- Username enumeration
- Account lockout
- Known default credentials
- Weak passwords
- Insecure data storage
- Transport encryption
- Insecure password recovery mechanism

**13 Vendor Backend APIs**
- Inherent trust of cloud or mobile application
- Weak authentication
- Weak access controls
- Injection attacks

**14 Ecosystem Communication**
- Health checks
- Heartbeats
- Ecosystem commands
- De-provisioning
- Pushing updates

**15 Network Traffic**
- LAN
- LAN to Internet
- Short range
- Non-standard

https://www.owasp.org

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

## IoT Attack Surface Areas

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

The OWASP IoT attack surface areas are given below:

- **Device Memory**

  This is one of the most important components in the IoT ecosystem. A device's memory is necessary in order to store important information about certain events. Discussed below are some of the vulnerabilities present in this component:

  - **Clear-text Credentials**

    **Vulnerability**: Unencrypted credentials or clear-text credentials may lead to credentials and information leak from a device.

    **Consideration**: In order to keep the device and its information secure, the credentials that are used for accessing some device and even the communication between two endpoints should be carried out in an encrypted form so that it cannot be easily accessed to compromise it or get an unauthorized access to the platform.

  - **Third-party Credentials**

    **Vulnerability**: Using third party credentials a device can be accessed and exploited.

    **Consideration**: Only certain functionalities should be exempted to access for the third parties and the credentials used by third parties should be encrypted using a strong encryption mechanism so that even if the hacker obtains them, he/she should not be able to decrypt them to gain an access to the device.

  - **Encryption Keys**

    **Vulnerability**: Encryption keys can be obtained by the hackers, using which they can get an unauthorized access to the device.

    **Consideration**: Proper key management system must be used to protect the encryption keys from hackers. Encryption keys should not be stored with the data that they decrypt otherwise if the machine on which both are located is compromised then so are the keys.

- **Ecosystem Access Control**

  - **Implicit Trust between Components**

    **Vulnerability**: Implicit trust can result in trusting malicious component that in turn can result in malfunctions of all the components.

    **Consideration**: Before interaction each component should authenticate itself with other component. If trust relationships are acquired, there should be strong mechanisms and procedures to ensure that the trust cannot be abused.

  - **Enrolment Security**

    **Vulnerability**: Enrolling the device in the absence of certain restrictions or authentication mechanisms can result in putting onboard a malicious device that can put the network's security at risk.

**Consideration**: Each device should authenticate itself before getting enrolled.

- **Decommissioning System**

  **Vulnerability**: Any single device may put the whole system at risk by compromising it.

  **Consideration**: The compromised devices should be handled carefully by analyzing the problem and developing methods to counter the problem. Certain techniques should also be adopted in order to prepare the system if some unwanted situation arrives, like clearing data and resetting the device from the cloud, debugging and decommissioning the system, etc.

- **Lost Access Procedures**

  **Vulnerability**: Lack of defining proper purpose of each device and its access level may result in the situation known as right escalation.

  **Consideration**: Proposing proper method where each device has the ability to be configured and what functionalities it can perform. ACL at device and the network levels should be implemented that would eventually decrease the security gaps and improve the control over the devices.

- **Device Physical Interfaces**

  - **Firmware Extraction**

    **Vulnerability**: Hidden vulnerabilities in the system can be exposed if the firmware is allowed to be accessed.

    **Consideration**: The security consideration for this would be to use the firmware in an encrypted form.

  - **User CLI**

    **Vulnerability**: If the user is allowed to access all the parts of a device or has an administrator level rights, it can put the device security at high risk.

    **Consideration**: Preferred approach would be to limit users' access to the core part of the device and certain changes in the devices should be allowed to be made.

  - **Admin CLI**

    **Vulnerability**: Access to user console or the admin console in order to perform administrative tasks or to access the data received by the device may expose it to exploitation and may compromise it.

    **Consideration**: The security consideration for such vulnerability is not to expose the console access to the devices for purposes like debugging etc. The administrative rights should be limited and for the live devices debugging ports should be blocked.

  - **Privilege Escalation**

    **Vulnerability**: Physical access to the device, if it is not configured properly may result in elevated access to the system resource which is usually not allowed for a user. This may result in exploiting the device functions.

**Consideration**: Consideration for this would be to design the firmware in such a way that the user cannot access that part of the device which he/she is not supposed to access.

- **Reset to Insecure State**

  **Vulnerability**: In case of physical access to the device, there is a possibility to reset the storage memory of the device to an unwanted or undesired state.

  **Consideration**: Firmware needs to be designed such a way that the access to resetting the device should be denied.

- **Removal of Storage Media**

  **Vulnerability**: Access to the device physically may lead to access to the storage media which can further expose firmware, data stored in the device and the credentials.

  **Consideration**: Additional security at the hardware level or the hardware encryption should be implemented.

- **Device Web Interface**

  - **SQL Injection**

    **Vulnerability**: SQL injection is a type of code injection technique where malicious code is injected in the application in order to extract and modify the database content.

    **Consideration**: Strong mitigation strategy against SQL injection includes use of prepared statements with parameterized queries.

  - **Cross-site Scripting**

    **Vulnerability**: Cross Site Scripting or XSS is a type of attack found in web applications, using which an attacker can inject malicious code into the application to get an unauthorized access to the web application.

    **Considerations**: Carefully monitoring and validating all the inputs that are assumed to be insecure and not trusting data coming from unknown source.

  - **Cross-site Request Forgery**

    **Vulnerability**: Cross-site Request Forgery is a type of attack where a malicious web site, blog, instant message or program causes a user's web browser to behave abnormally on a trusted site for which the user is authenticated at that moment.

    **Considerations**: Adoption of additional authentication data into requests that allow the web application to detect requests from unauthorized locations.

  - **Username Enumeration**

    **Vulnerability**: User Enumeration is a kind of technique where an attacker finds out whether some username is already existing or not with the help of forgot password form. Once a set of existing or valid usernames are obtained, they can be used to get further access to their accounts.

**Considerations**: Applications should specify their own usernames and they should not be predictable and CAPTCHA can also be used to avoid user enumeration up to a certain extent.

- **Weak Passwords**

  **Vulnerability**: Weak or easy to guess passwords can be easily brute forced by an attacker to access user's personal and confidential data.

  **Considerations**: Strong passwords having lower case, upper case, alpha and numeric characters should be used. One should also avoid using dictionary words as their password as they are easy to crack.

- **Account Lockout**

  **Vulnerability**: Account lockout mechanism is used in order to prevent the system from brute force password guessing attack. Absence of lockout mechanism can allow an attacker to brute force the password and gain an access to the user's account and access his/her private data.

  **Considerations**: Proper lockout mechanism should be implemented which locks out an account of individual after 3-5 unsuccessful login attempts for a certain period of time.

- **Known Default Credentials**

  **Vulnerability**: If default credentials are not changed, they can be easily cracked and the device can go in wrong hands.

  **Considerations**: Users should change the credentials of any device they buy in order to prevent it from unauthorized access.

- **Device Firmware**

  - **Hardcoded Credentials**

    **Vulnerability**: Most of the devices that are bought by the customer, come with default credentials that are set by the manufacturing companies and users usually do not reset the default credentials that make them vulnerable to unauthorized access. After successfully compromising such devices, hackers can turn them into a bot.

    **Consideration**: IoT device users need to change/reset the default credentials in order to get an additional layer of security against attacks.

  - **Sensitive Information/URL Disclosure**

    **Vulnerability**: Leak of sensitive or confidential data via URLs may make the devices exposed to attacks.

**Consideration**: All the information transmitted through URLs must be properly encrypted. Firmware should be designed in such a way that the information stored in the device is in strong encrypted form.

- **Encryption Keys**

  **Vulnerability**: Access to encryption keys may result in its decryption and obtaining of confidential data.

  **Consideration**: The encryption keys that are used for decrypting the data should not be available directly from the device's memory; rather it should be stored in the cloud and sent across the network to the device when required.

- **Firmware Version Display and/or Last Update Date**

  **Vulnerability**: The sensitive information about the device (such as credentials, controls keys, update information, etc.) should not be visible to all, i.e., it should be encrypted and should not be shared among devices.

  **Consideration**: Preferred approach would be to use separate control keys whereas the same credentials or the control keys should not be shared across the network. Different security keys for different devices provides an additional level of security against threats.

- **Device Network Services**

  - **Information Disclosure**

    **Vulnerability**: Leak of sensitive or confidential data may make the devices exposed to attacks.

    **Consideration**: Firmware should be designed in such a way that the information stored in the device is in strong encrypted form.

  - **Denial-of-Service**

    **Vulnerability**: Any denial of service attack may impact the services offered by the cloud.

    **Consideration**: Intrusion detection mechanism should be deployed in order to handle different denial of service attacks.

  - **UPnP**

    **Vulnerability**: Unwanted ports like Universal Plug and Play (UPnP) comes enabled by default in the devices, putting device security at risk as it allows malware to enter and destroy the device and the local network.

    **Consideration**: Manufacturer should design the devices such a way that these types of vulnerable ports should by default become disabled.

- **Vulnerable UDP Services**

  **Vulnerability**: Vulnerable UDP services can put the security of the system at high risk. Certain unwanted or malicious files can be transferred using such services which can even destroy the system and important data.

  **Consideration**: Firmware should be designed in such a way that certain risky services should by default come disabled.

- User and admin CLI
- Injection and Unencrypted services
- Poorly implemented encryption

- **Administrative Interface**
  - SQL Injection
  - Cross-site Scripting and Cross-site Request Forgery
  - Username Enumeration and Known Default Credentials
  - Weak Passwords and Account Lockout
  - Security/encryption and Logging options
  - Two-factor authentication
  - Inability to wipe device
- **Local Data Storage**
  - **Unencrypted Data**

    **Vulnerability**: Clear text or unencrypted communications in a network are prone to attacks like data interception.

    **Consideration**: Strong encryption mechanisms that encrypts data should be adopted so that it cannot go in wrong hands and cannot be misused.

  - **Data Encrypted with Discovered Keys**

    **Vulnerability**: Can lead to ransomware attack where an attacker who has encrypted the data and has the keys, can ask for the ransom in order to decrypt the data.

    **Consideration**: Update the device on a regular basis and avoid opening email from unknown source as it might contain a malicious attachment.

  - **Lack of Data Integrity Checks**

    **Vulnerability**: Weak encryption mechanisms may result in the data interception and loss of important information.

    **Consideration**: Security consideration for such issue would be to use strong encryption techniques like Transport Layer Security (TLS).

- **Cloud Web Interface**
  - **Transport Encryption**

    **Vulnerability**: Transport encryption is an essential step towards a device's security, lack of which can result in loss of important information, loss of privacy and compromise of device as well.

    **Consideration**: Proper transport encryption techniques should be implemented in order to keep the data encrypted and protected when in transit.
  - SQL Injection
  - Cross-site Scripting and Cross-site Request Forgery
  - Username Enumeration and Known Default Credentials
  - Weak Passwords and Account Lockout
  - Insecure password recovery mechanism
  - Two-factor authentication
- **Update Mechanism**
  - **Update Sent without Encryption**

    **Vulnerability**: Unavailability of secure update transferring mechanism opens the door for cyber-attacks.

    **Consideration**: Tested and strong encryption mechanisms should be incorporated for secure transmission of updates to the devices.
  - **Updates Not Signed**

    **Vulnerability**: Updates that are not signed from a trusted or reliable source might contain malicious files which can harm the device or the system.

    **Consideration**: Verify whether the updates to be installed are signed and are from a trusted source. If they are not, avoid installing them.
  - **Update Verification**

    **Vulnerability**: Update verification mechanism verifies the updates that will be installed in the device. If this option is not turned on, then the system would not be able to distinguish between the malicious and genuine updates that can eventually harm the device.

    **Consideration**: Keep the update verification option turned on so that if some malicious update or an update from an unknown source pops up, it will be discarded.
  - **Malicious Update**

    **Vulnerability**: Provides unauthorized access to attackers, using which he/she can perform malicious activities using the device.

**Consideration**: Verify if the update is from trusted source; if it is not, it should be discarded.

- **Missing Update Mechanism**

  **Vulnerability**: Usually updates remove system vulnerabilities, thus preventing various attacks. Missing update mechanism can make the device or the system prone to various online and offline attacks.

  **Consideration**: Make sure any device you buy has an update mechanism installed in it, or if it is already there make sure it is turned on.

- **No Manual Update Mechanism**

  **Vulnerability**: Some updates are not automatically installed in your system; you must install them manually. Therefore, absence of a manual update mechanism can make your device vulnerable to certain attacks. Updates usually include various security patches to update the device's software and remove all the existing vulnerabilities.

  **Consideration**: Make sure that the device you buy has a manual update mechanism present in it and gives you the liberty of manually updating the device for updates which are not installed automatically.

- **Third-party Backend APIs**

  - **Unencrypted PII Sent**

    **Vulnerability**: Sending of unencrypted Personally Identifiable Information (PII) has the potential of identifying a specific individual. It contains important information that can distinguish one person from another. Therefore, if the hackers get an access to this information they can carry out malicious activities like identity stealing accessing the device illegitimately.

    **Consideration**: PII should be kept, and sent in encrypted form, so that the hackers won't be able to see the information in clear text or will be unable to decrypt it.

  - **Device Information Leaked**

    **Vulnerability**: Lack of information storage security policies can lead to information leak, the consequences of which could be loss of sensitive and confidential data that in turn could help the hackers get an unauthorized access to the device.

    **Consideration**: Firmware should incorporate certain security policies that keep personal as well as the device information protected.

  - **Location Leaked**

    Vulnerability: Leak of location of a device could result in physically accessing the device and the information possessed by it or compromising the device.

Consideration: Firmware should make sure that the sensitive information such as location, identity, device banner, etc. should be kept in encrypted form so that it becomes inaccessible to attacker through debugging or physical level.

- **Mobile Application**
  - **Implicitly Trusted by Device or Cloud**

    **Vulnerability**: Trusting each device connected to the network or the cloud without having any doubts about it leads to high risks. For example, a device connected to the network might be a fake one or an infected one, connection to which the whole network can get infected.

    **Consideration**: Implementation of trust policies is a perfect step in order to counter this problem. Policies should be such that a device or the cloud should be properly analyzed (based on identity, location, not infected, etc.) before it is considered as trusted.

  - **Username Enumeration**

    **Vulnerability**: Some web applications have a security loophole where they reveal that whether an entered username exists on the system or not. Exploiting this vulnerability, an attacker can guess and find out the username and then using brute force attack can gain an access with that username to the device.

    **Consideration**: Best practice to overcome this issue would be to design a system where the usernames cannot be easily found out and after certain failed attempts the application should stop responding or providing service to the user for a certain period of time and this time should keep on increasing with the increase in number of failed attempts.

  - **Account Lockout**

    **Vulnerability**: Unavailability of account lockout mechanism after a certain period of inactivity on a system can result in unauthorized access to the device by hackers.

    **Consideration**: Account lockout mechanism should be incorporated in the device that locks the user out after a defined period of time so that no illegitimate user can access the account and obtain important information.

  - **Known Default Credentials or Weak Passwords**

    **Vulnerability**: Lack of proper authentication mechanism or known default usernames and passwords may result in increasing credentials leak which can put the device at risk.

    **Consideration**: From the cloud side, the authentication mechanism should be used. And rather than transferring the credentials to the cloud every time, a mechanism such as token should be used and keeping the token's life span to a short period of time (few minutes) can automatically increase the security level.

- **Insecure Data Storage**

  **Vulnerability**: Unsecured data storage can lead to a leak or exposure of sensitive or confidential data.

  **Consideration**: Firmware should be designed in such a way that all the data storage layers of IoT should be properly protected. Some storage layers in memory are NoSQL, RDBMS and Big Data Hadoop.

- Transport Encryption, Insecure password recovery mechanism and Two-factor authentication

- **Vendor Backend API's**

  - **Inherent Trust of Cloud or Mobile Application**

    **Vulnerability**: Trusting each mobile application or cloud without having any doubts about it leads to high risks. For example, a device using a malicious mobile application which may be fake, or infected, could result in the whole network being infected.

    **Consideration**- Implementation of trust policies is a perfect step to counter this problem. Policies should be such that a mobile application or the cloud should be properly analyzed (based on identity, script, not infected, etc.) before it is considered as trusted.

  - **Weak Authentication**

    **Vulnerability**: As the security is entirely dependent on the strength of authentication mechanism and credentials used, the weak authentication mechanism results in security issues of the device such as loss of privacy, unauthorized access, change of device's settings and infecting different components of the device.

    **Consideration**: Two factor or multi factor authentication mechanism should be used to increase the device's security level.

  - **Weak Access Controls**

    **Vulnerability**: Lack of defining proper purpose of each device and its access level may result in the situation known as right escalation.

    **Consideration**: Proposing a proper method where each device has the ability to be configured and what functionalities it can perform. ACL at device and the network level should be implemented that would eventually decrease the security gaps and improve the control over the devices.

- **Ecosystem Communication**

  - **Health Checks**

    **Vulnerability**: Any vulnerability present in a health care device can be exploited by an attacker and can put the patient's life at risk. Vulnerable medical devices are also connected to many monitors and sensors, therefore making them potential entry points to the larger network of a hospital.

**Consideration**: Manufacturing companies, rather than increasing other features in the healthcare devices should increase the security features, thus making it impossible for attackers to bypass the device's security.

- **Heartbeats**

  **Vulnerability**: Security flaws in the pacemaker or the features which make it accessible from a remote location, can be exploited by the potential hacker which can even result in killing the patient.

  **Consideration**: Manufacturing companies should put more emphasis on medical device security and secure the devices from potential attacks.

- **Ecosystem Commands**

  **Vulnerability**: Lack of verification of any command coming from the system exposes it to exploits or attacks.

  **Consideration**: Commands that alter the system or update the device's configuration should have additional verification systems to check whether the command is coming from a genuine source or not.

- **De-provisioning**

  **Vulnerability**: Devices that are not in use but still connected to the Internet are another welcoming factor for various attacks on the device and the network.

  **Consideration**: Unused devices should be detached or terminated from the Internet. Removal of access to certain devices is also an effective solution to this problem.

- **Pushing Updates**

  **Vulnerability**: Malicious updates from the attackers through an attachment in the email or through compromised third parties can impact the system security badly by installing unwanted or malicious files which can lead to data loss, inability to access the device or ransom demands to get access back to the device.

  **Consideration**: Device users should be more cautious while opening some links or attachments that seem suspicious or coming from some unknown source.

- **Network Traffic**

  - **LAN**

    **Vulnerability**: Absence of robust security or configured security, lack of secure locations and lack of network monitoring can result in the problems like connection interception, jamming signal attacks, man-in-the-middle attacks, etc.

    **Consideration**: Before deploying LAN, it should be kept in mind that the location is secure and on the software level firewall should be deployed to keep hackers away from the network.

- **LAN to Internet**

  **Vulnerability**: Not having proper security infrastructure (Firewall, anti-virus, DMZ), lack of proper network monitoring and insecure location of deployment, can result in various attacks from internal or external networks.

  **Consideration**: The very first thing while deploying LAN is the location. Ensure that it is secure and proper security policies and practices are followed to enhance the network's security making it difficult for the attacker to breach the network security.

- **Short Range**

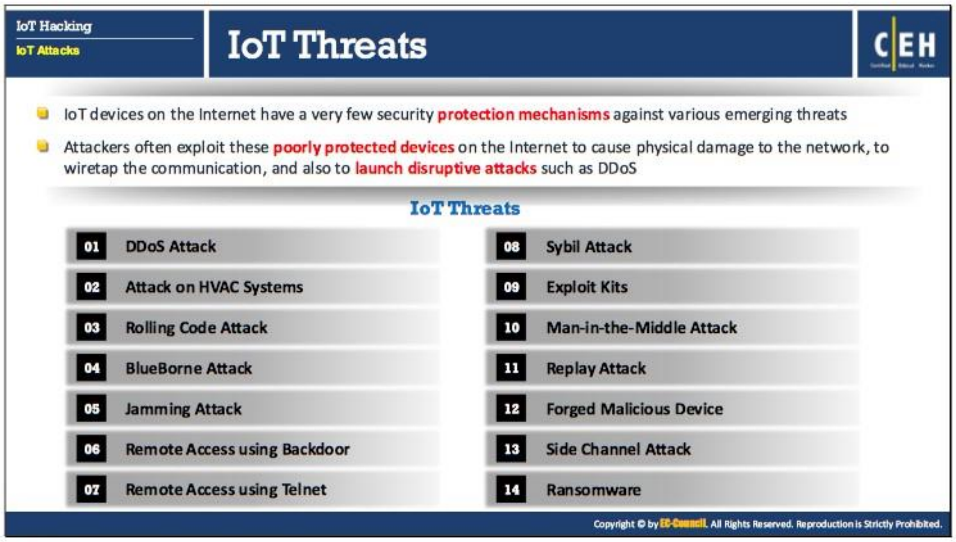
  **Vulnerability**: Short range devices like Bluetooth devices are vulnerable to various attacks, if the frequency on which it is working on, is known by the intruder. They can easily see all the personal or sensitive information present in your device.

  **Consideration**: In order to make the short-range communication secure, a good security design should be implemented that hardens the device's security.

- **Non-standard**

  **Vulnerability**: Non-standardized network traffic might contain malicious files that could harm the network's security and the devices.

  **Consideration**: Each piece of network traffic passing through should be standardized and should be checked before leaving or coming into the network.



IoT Hacking
**IoT Attacks**

## IoT Threats

- IoT devices on the Internet have a very few security **protection mechanisms** against various emerging threats
- Attackers often exploit these **poorly protected devices** on the Internet to cause physical damage to the network, to wiretap the communication, and also to **launch disruptive attacks** such as DDoS

**IoT Threats**

- 01 **DDoS Attack**
- 02 **Attack on HVAC Systems**
- 03 **Rolling Code Attack**
- 04 **BlueBorne Attack**
- 05 **Jamming Attack**
- 06 **Remote Access using Backdoor**
- 07 **Remote Access using Telnet**
- 08 **Sybil Attack**
- 09 **Exploit Kits**
- 10 **Man-in-the-Middle Attack**
- 11 **Replay Attack**
- 12 **Forged Malicious Device**
- 13 **Side Channel Attack**
- 14 **Ransomware**

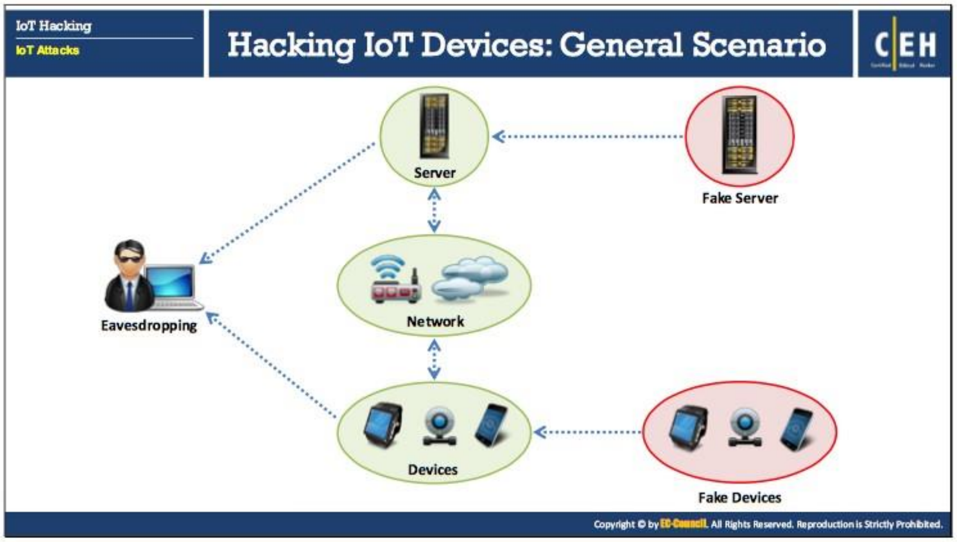
### IoT Threats

IoT devices on the Internet have a very few security protection mechanisms against various emerging threats. These devices are infected by malware or malicious code at an alarming rate. Attackers often exploit these poorly protected devices on the Internet to cause physical damage to the network, to wiretap the communication, and also to launch disruptive attacks such as DDoS.

Listed below are some of IoT attacks:

- **DDoS Attack**: Attacker converts the devices into an army of botnet to target a specific system or server, making it unavailable to provide services.
- **Exploiting HVAC**: HVAC system vulnerabilities are exploited by attackers to steal confidential information such as user credentials and to perform further attacks on the target network.
- **Rolling Code**: An attacker jams and sniffs the signal to obtain the code transferred to the vehicle's receiver and uses it to unlock and steal the vehicle.
- **BlueBorne Attack**: Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the device.
- **Jamming Attack**: Attacker jams the signal between the sender and the receiver with malicious traffic that makes the two endpoints unable to communicate with each other.
- **Remote Access using Backdoor**: Attackers exploit vulnerabilities in the IoT device to turn the device into a backdoor and gain access to an organization's network.

- **Remote Access using Telnet**: Attackers exploit an open telnet port to obtain information that is shared between the connected devices, including their software and hardware models.
- **Sybil Attack**: Attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Exploit Kits**: A malicious script used by the attackers to exploit poorly patched vulnerabilities in an IoT device.
- **Man-in-the-Middle Attack**: An attacker pretends to be a legitimate sender who intercepts all the communication between the sender and receiver and hijacks the communication.
- **Replay Attack**: Attackers intercept legitimate messages from a valid communication and continuously send the intercepted message to the target device to perform a denial-of-service attack or crash the target device.
- **Forged Malicious Device**: Attackers replace authentic IoT devices with malicious devices, if they have physical access to the network.
- **Side Channel Attack**: Attackers perform side channel attacks by extracting information about encryption keys by observing the emission of signals i.e. "side channels" from IoT devices.
- **Ransomware Attack**: Ransomware is a type of malware that uses encryption to block user's access to his/her device either by locking the screen or by locking a user's files.



IoT Hacking
IoT Attacks

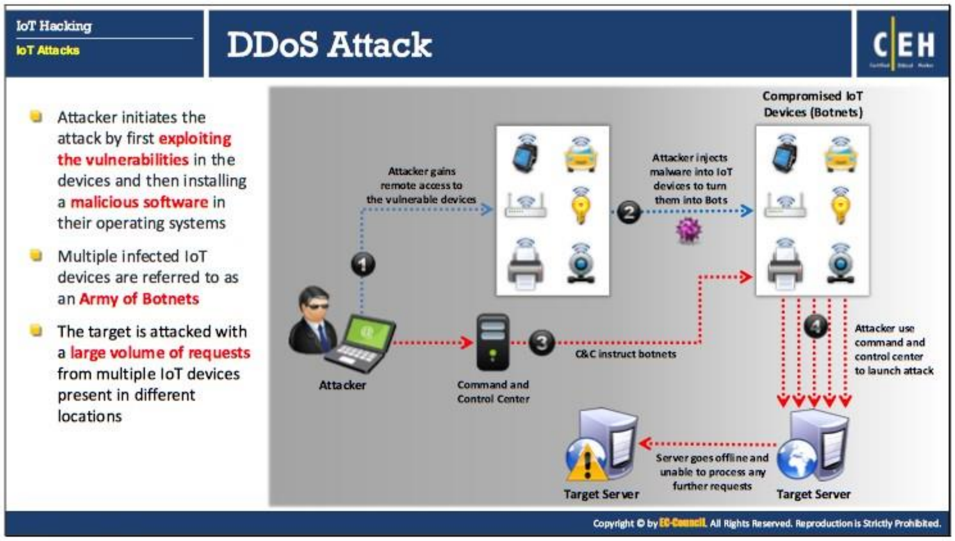
Hacking IoT Devices: General Scenario

## Hacking IoT Devices: General Scenario

The Internet of Things (IoT) includes different technologies such as embedded sensors, microprocessors and power management devices. Security consideration changes from device to device and application to application. The greater the amount of confidential data we send across the network, the greater the risk arises of data theft, data manipulation, data tampering as well as attacks on routers and servers.

Improper security infrastructure might lead to the following unwanted scenarios:

- An Eavesdropper intercepts communication between two endpoints and sniffs the confidential information that is sent across. He/she can misuse that information for his/her own benefit.
- A Fake Server can be used to send some unwanted commands in order to trigger some events which are not planned. For example, some physical resource (water, coal, oil, electricity) can be sent to some unknown and unplanned destination and so on.
- A Fake Device can inject some malicious script into the system to make it work the way the device wants. This may cause the system to behave inappropriately and dangerously.



IoT Hacking
**IoT Attacks**

## DDoS Attack

- Attacker initiates the attack by first **exploiting the vulnerabilities** in the devices and then installing a **malicious software** in their operating systems
- Multiple infected IoT devices are referred to as an **Army of Botnets**
- The target is attacked with a **large volume of requests** from multiple IoT devices present in different locations

Compromised IoT Devices (Botnets)

1 Attacker gains remote access to the vulnerable devices

2 Attacker injects malware into IoT devices to turn them into Bots

3 C&C instruct botnets

4 Attacker use command and control center to launch attack

Attacker

Command and Control Center

Server goes offline and unable to process any further requests

Target Server

Target Server

### DDoS Attack

DDoS, a Distributed Denial-of-Service attack is a type of attack where multiple infected systems are used to pound a single online system or service that makes the server useless, slow and unavailable for a legitimate user for a short period of time. The attacker initiates the attack by first exploiting the vulnerabilities in the devices and then installing malicious software in their operating systems. These multiple compromised devices are referred to as an Army of Botnets.

Once an attacker decides his target, he instructs the botnets or zombie agents to send requests to the target server that he is attacking. The target is attacked with a large volume of requests from multiple IoT devices present in different locations. As a result, the target's system is flooded with more number of requests than it can handle. Therefore, it either goes offline, or suffers a loss in performance or shuts down completely.

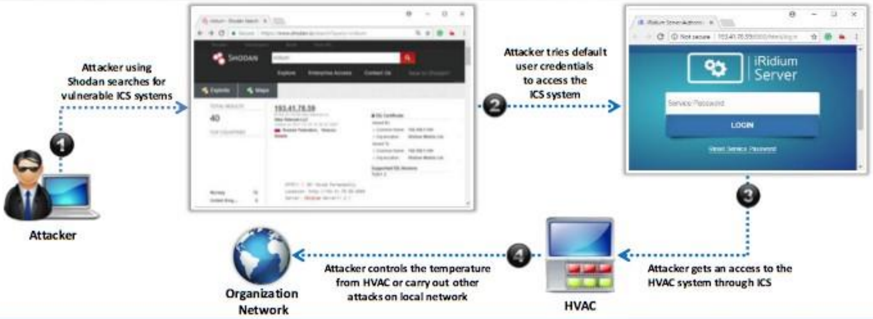
**Given below are the steps followed by an attacker to perform DDoS attack on IoT devices:**

- Attacker gains remote access to vulnerable devices
- After gaining access, he/she injects malware into the IoT devices to turn them into Botnets
- Attacker uses command and control center to instruct botnets and to send multiple requests to the target server resulting in DDoS attack
- Target server goes offline and becomes unavailable to process any further requests

IoT Hacking
IoT Attacks

## Exploit HVAC

- Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms, giving attackers a gateway to hack corporate systems
- HVAC systems have many security vulnerabilities that are exploited by attackers to steal login credentials, gain access to HVAC system and perform further attack on the organization's network

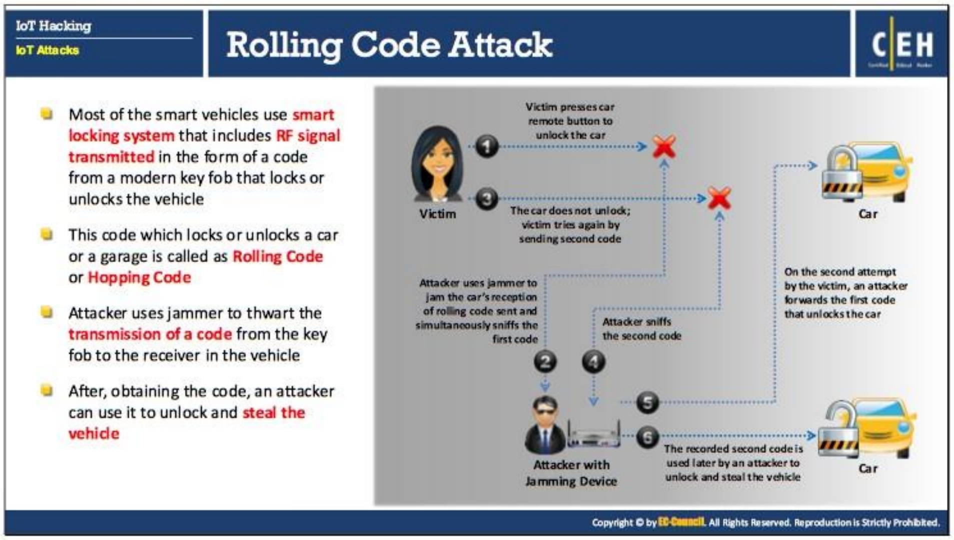

Attacker using Shodan searches for vulnerable ICS systems
1
Attacker
Attacker tries default user credentials to access the ICS system
2
iRidium Server
LOGIN
3
Attacker gets an access to the HVAC system through ICS
HVAC
4
Attacker controls the temperature from HVAC or carry out other attacks on local network
Organization Network

### Exploit HVAC

Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms, giving attackers a gateway to hack corporate systems. HVAC systems have many security vulnerabilities that are exploited by attackers to steal login credentials, gain access to HVAC system and perform further attack on the organization's network. HVAC systems are generally connected to the networks of various industries, government sectors, hospitals, etc. These systems provide remote access rights to HVAC vendors and third parties to support remote administration of the system such as remotely monitoring energy consumption and temperatures at various places. In addition, many HVAC companies provide common login name and password to different organizations. Attackers take advantage of this to obtain remote access to corporate networks and steal confidential information of the organization.

**Steps followed by an attacker to exploit HVAC systems:**

- Attacker using **Shodan** (*https://www.shodan.io*) and searches for vulnerable ICS systems
- Based on the vulnerable ICS systems found, the attacker then tries for default user credentials using online tools such as*https://madifi.de/defpass/index.php*
- Attacker tries default user credentials to access the ICS system
- After gaining access to the ICS system, the attacker tries to get access to the HVAC system remotely through ICS system
- After gaining access to the HVAC system, an attacker can control the temperature from HVAC or carry out other attacks on the local network



IoT Hacking
IoT Attacks

## Rolling Code Attack

- Most of the smart vehicles use smart locking system that includes RF signal transmitted in the form of a code from a modern key fob that locks or unlocks the vehicle
- This code which locks or unlocks a car or a garage is called as Rolling Code or Hopping Code
- Attacker uses jammer to thwart the transmission of a code from the key fob to the receiver in the vehicle
- After, obtaining the code, an attacker can use it to unlock and steal the vehicle

Victim

1 Victim presses car remote button to unlock the car

2 Attacker uses jammer to jam the car's reception of rolling code sent and simultaneously sniffs the first code

3 The car does not unlock; victim tries again by sending second code

4 Attacker sniffs the second code

5 On the second attempt by the victim, an attacker forwards the first code that unlocks the car

6 The recorded second code is used later by an attacker to unlock and steal the vehicle

Attacker with Jamming Device

Car

Car

### Rolling Code Attack

Most of the smart vehicles use smart locking systems that include an RF signal transmitted in the form of a code from a modern key fob that locks or unlocks the vehicle. Here a code is sent to the vehicle which is different for every other use and is only used once, that means if a vehicle receives a same code again it rejects it.

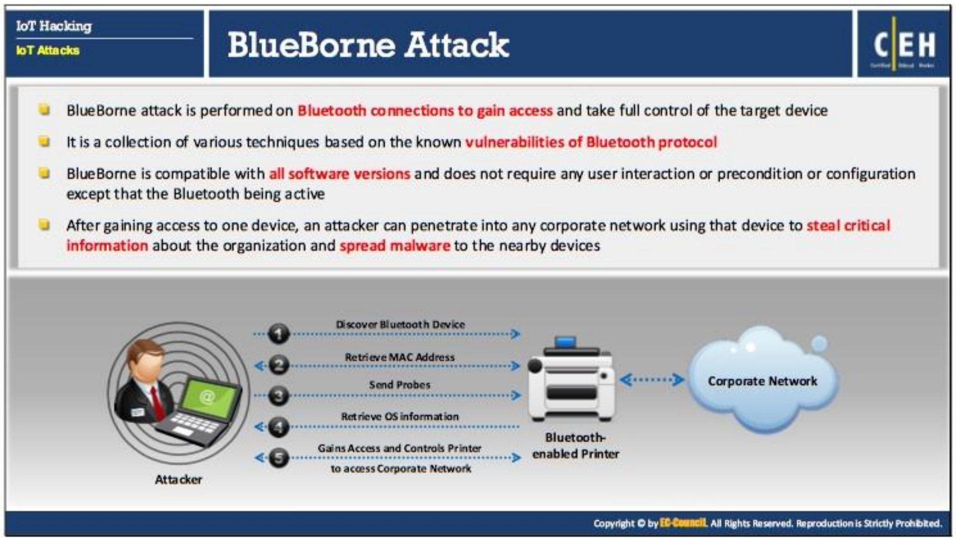
This code which locks or unlocks a car or a garage is called as Rolling Code or Hopping Code. It is used in keyless entry system to prevent replay attacks. An eavesdropper can capture the code transmitted and later use it to unlock the garage or the vehicle.

To obtain the Rolling Code, the attacker simultaneously thwarts the transmission of a signal from the key fob to the receiver in the vehicle. This attack is performed using a jamming device that both jams the signal and sniffs the code and the attacker later uses that code to unlock the vehicle or the garage door.

For example, given below are the steps followed by an attacker to perform rolling code attack:

- Victim presses car remote button and tries to unlock the car
- Attacker uses a jammer that jams the car's reception of rolling code sent by the victim and simultaneously sniffs the first code
- The car does not unlock; victim tries again by sending a second code
- Attacker sniffs the second code
- On the second attempt by the victim, an attacker forwards the first code that unlocks the car
- The recorded second code is used later by an attacker to unlock and steal the vehicle

Attackers can make use of tools such as rfcat-rolljam, RFCrack, etc. to perform this attack.



IoT Hacking
**IoT Attacks**

## BlueBorne Attack

- BlueBorne attack is performed on **Bluetooth connections to gain access** and take full control of the target device
- It is a collection of various techniques based on the known **vulnerabilities of Bluetooth protocol**
- BlueBorne is compatible with **all software versions** and does not require any user interaction or precondition or configuration except that the Bluetooth being active
- After gaining access to one device, an attacker can penetrate into any corporate network using that device to **steal critical information** about the organization and **spread malware** to the nearby devices

CEH
Attacker
1 Discover Bluetooth Device
2 Retrieve MAC Address
3 Send Probes
4 Retrieve OS information
5 Gains Access and Controls Printer to access Corporate Network
Bluetooth-enabled Printer
Corporate Network

### BlueBorne Attack

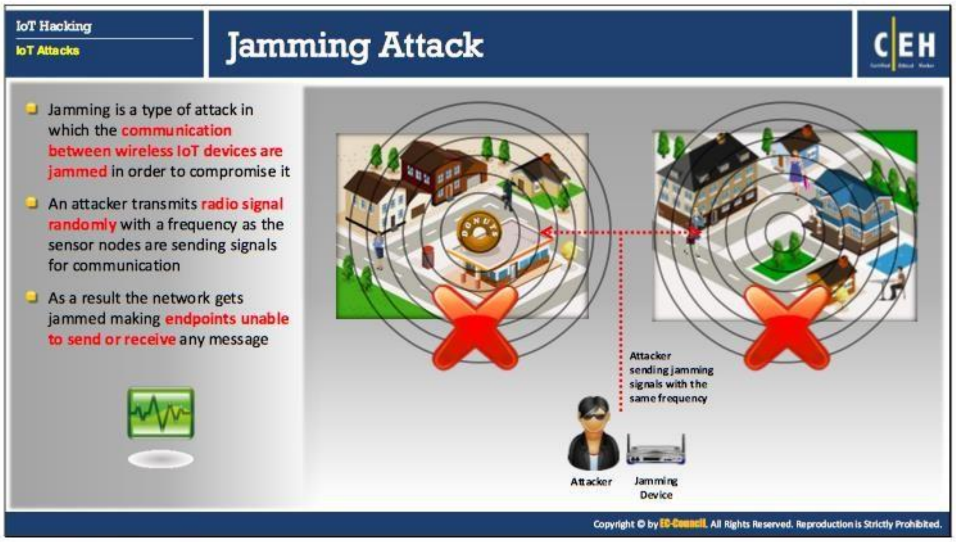
BlueBorne attack is performed on Bluetooth connections to gain access and take full control of the target device. Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the devices. BlueBorne is a collection of various techniques based on the known vulnerabilities of the Bluetooth protocol. This attack can be performed on multiple IoT devices including those running operating systems such as Android, Linux, Windows and older versions of iOS. In all the operating systems, the Bluetooth process has high privileges. After gaining access to one device, an attacker can penetrate into any corporate network using that device to steal critical information of the organization and spread malware to the nearby devices.

BlueBorne is compatible with all software versions and does not require any user interaction or precondition or configuration except that the Bluetooth be active. This attack establishes a connection with the target Bluetooth-enabled device without even pairing with the device. Using this attack, an attacker can discover Bluetooth-enabled devices, even though they are not in an active discovery mode. Once the attacker identifies any nearby device, he/she tries to extract MAC address and OS information to perform further exploitation on the target OS. Based on the vulnerabilities present in the Bluetooth protocol, attackers can even perform remote code execution and man-in-the-middle attack on the target device. This attack can be performed on various IoT devices such as smart TVs, phones, watches, car audio systems, printers etc.

**Steps to perform BlueBorne attack:**

- Attacker discovers active Bluetooth-enabled devices around him/her. All the Bluetooth-enabled devices can be located even if they are not in discoverable mode

- After locating any nearby device, the attacker obtains MAC address of the device
- Now, the attacker sends continuous probes to the target device to determine the operating system
- After identifying the OS, the attacker exploits the vulnerabilities in the Bluetooth protocol to gain access to the target device
- Now the attacker can perform remote code execution or man-in-the-middle attack and take full control of the device.



IoT Hacking
IoT Attacks

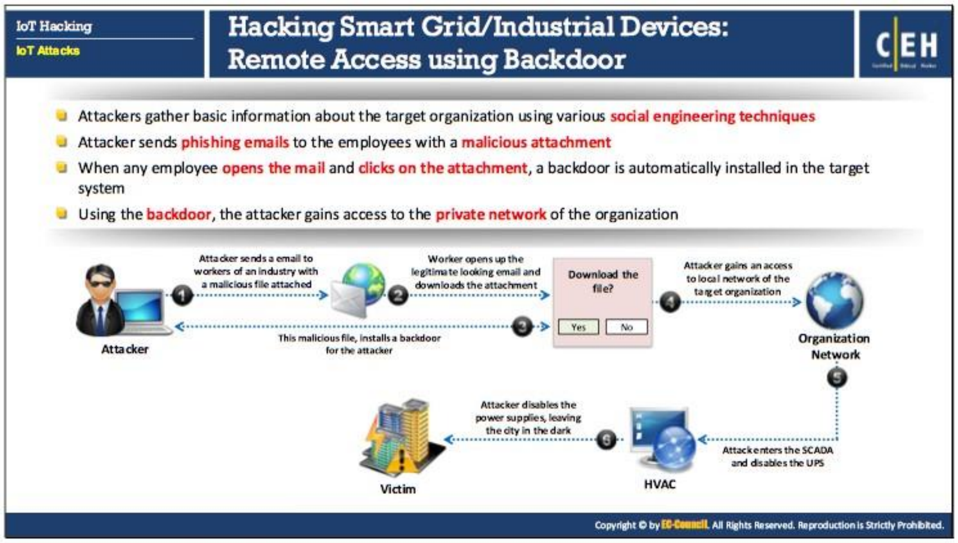
## Jamming Attack

- Jamming is a type of attack in which the **communication between wireless IoT devices are jammed** in order to compromise it
- An attacker transmits **radio signal randomly** with a frequency as the sensor nodes are sending signals for communication
- As a result the network gets jammed making **endpoints unable to send or receive** any message

## Jamming Attack

Jamming is a type of attack in which the communication between wireless IoT devices are jammed in order to compromise it. During this attack, an overwhelming volume of malicious traffic is sent which results in DoS attack to authorized users thus, obstructing legitimate traffic and making the endpoints unable to communicate with each other. Every wireless device and the wireless network is prone to this attack.

Attackers use special types of hardware and transmit radio signals randomly with the same frequency on which the device is communicating. The signals or the traffic generated by the jamming device appears to be noise for the wireless devices which causes them to hold their transmissions until the noise subsides. This, results in a DoS attack that jams the network and devices are unable to send or receive any data.



IoT Hacking

IoT Attacks

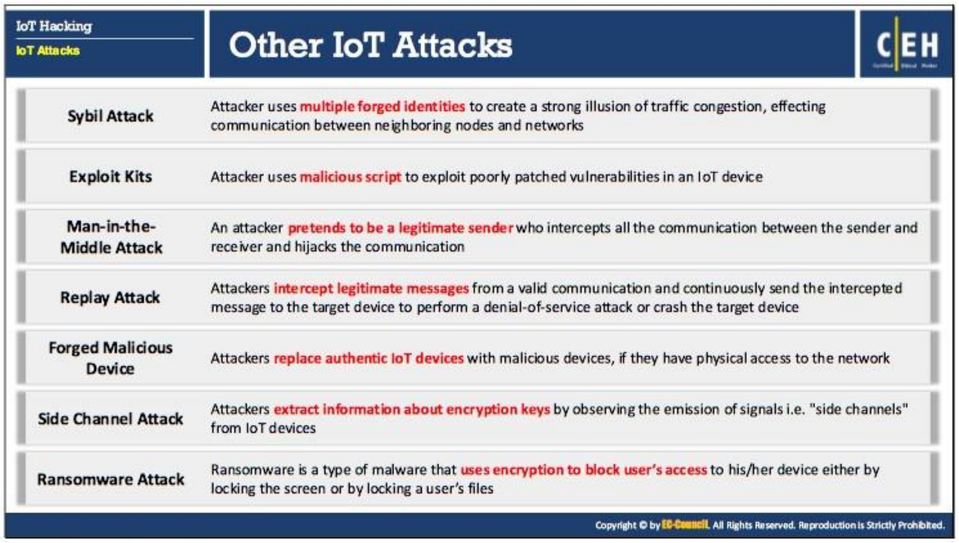
## Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

- Attackers gather basic information about the target organization using various **social engineering techniques**
- Attacker sends **phishing emails** to the employees with a **malicious attachment**
- When any employee **opens the mail** and **clicks on the attachment**, a backdoor is automatically installed in the target system
- Using the **backdoor**, the attacker gains access to the **private network** of the organization

Attacker
1 Attacker sends a email to workers of an industry with a malicious file attached
2 Worker opens up the legitimate looking email and downloads the attachment
Download the file?
Yes
No
3 This malicious file, installs a backdoor for the attacker
4 Attacker gains an access to local network of the target organization
Organization Network
5 Attack enters the SCADA and disables the UPS
HVAC
6 Attacker disables the power supplies, leaving the city in the dark
Victim

## Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

Attackers gather basic information about the target organization using various social engineering techniques. After obtaining information such as email IDs of the employees, an attacker sends phishing emails to the employees with a malicious attachment (e.g. Word document). When any employee of the target organization opens the email, and clicks on the attachment, a backdoor is automatically installed in the target system. Using the backdoor, the attacker gains access to the private network of the organization. For example, consider an attack on a power grid. In such type of attack, after gaining access to the private network, an attacker can access SCADA (Supervisory Control and Data Acquisition network) that controls the grid. After gaining access to the SCADA network, the attacker replaces the legitimate firmware with a malicious firmware to process commands sent by the attacker. Finally, the attacker can disable the power supply to any particular place by sending malicious commands to the substation control systems from SCADA.



IoT Hacking
IoT Attacks

## Other IoT Attacks

CEH

| | |
|---|---|
| **Sybil Attack** | Attacker uses **multiple forged identities** to create a strong illusion of traffic congestion, effecting communication between neighboring nodes and networks |
| **Exploit Kits** | Attacker uses **malicious script** to exploit poorly patched vulnerabilities in an IoT device |
| **Man-in-the-Middle Attack** | An attacker **pretends to be a legitimate sender** who intercepts all the communication between the sender and receiver and hijacks the communication |
| **Replay Attack** | Attackers **intercept legitimate messages** from a valid communication and continuously send the intercepted message to the target device to perform a denial-of-service attack or crash the target device |
| **Forged Malicious Device** | Attackers **replace authentic IoT devices** with malicious devices, if they have physical access to the network |
| **Side Channel Attack** | Attackers **extract information about encryption keys** by observing the emission of signals i.e. "side channels" from IoT devices |
| **Ransomware Attack** | Ransomware is a type of malware that **uses encryption to block user's access** to his/her device either by locking the screen or by locking a user's files |

## Other IOT Attacks

### Sybil Attack

Vehicular communications play an important role in safe transportation by exchanging important safety messages and traffic updates, but even the VANETs is not safe from the attackers' reach. An attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks. Sybil attacks in VANETs (Vehicular Ad hoc Networks) are regarded as the most serious attacks which puts a great impact on network's performance. This type of attack impairs the potential applications in VANETs by creating a strong illusion of traffic congestion. To perform this type of attack, a vehicle declares to be present at different locations at a same time.

For example, a node that spoofs itself as other nodes and launches an attack is called Sybil node 'X'. It is created by forming new identity or stealing existing legal identity. In proper communication, the other nodes 'A' and 'B' should only communicate with each other. But, in this scenario node 'X' comes in between as a known internal node and attacks the network. The node 'X' tries to communicate with other normal neighboring nodes (A and B) using multiple forged identities. Thus, it creates a huge chaos and security risks in the network.

### Exploit Kits

Exploit kit is a malicious script used by the attackers to exploit poorly patched vulnerabilities in an IoT device. These kits are designed in such a way that whenever there are new vulnerabilities, new ways of exploitation and add on functions will be added to the device automatically. After detecting vulnerabilities, these kits send the exact exploit to install malware, which can execute and corrupt the device. These exploit kits pose a dangerous threat as they go undetected in IoT environments affecting IoT devices and its infrastructure, forcing them to behave unexpectedly.

### Man-in-the-Middle Attack

In Man-in-the-middle attack, the attacker pretends to be a legitimate sender, intercepts all the communication between the sender and receiver and hijacks the communication. IoT devices are generally connected to a network and act as a gateway to all sensitive and personal information. Therefore, any malicious user can pose to be a legitimate sender and send malicious requests to the device to gain control on the device. IoT devices such as IP enabled cameras, routers, modems, and Internet gateways have cryptographic vulnerabilities that lead to man-in-the-middle attacks.

### Replay Attack

In replay attack, attackers intercept legitimate messages from a valid communication and continuously send the intercepted message to the target device to perform a denial-of-service attack or delay it in order to manipulate the message or crash the target device. For example, consider a replay attack that regenerates the signal that is used to control some IoT device like front door. Front door uses lock that is opened using simple infrared signals. Simply the attacker records the IR modulation pattern, reproduces the signal and performs replay attack on the door to unlock it.

### Forged Malicious Device

Attackers replace authentic IoT devices with malicious devices, if they have physical access to the network. It is very difficult to discover such attacks because the forged device resembles the legitimate one. The forged devices contain backdoors that are used by the attackers to perform various malicious activities in the network.

### Side Channel Attack

Attackers perform side channel attack by extracting information about encryption keys by observing the emission of signals i.e. "side channels" from IoT devices. All devices emit these signals that provide the information about internal computing process either by power consumption or electromagnetic emanations. Attackers carefully observe side channel emissions to acquire all the knowledge about varying power consumption to access and duplicate the encryption key non-evasively. The main advantage of this attack is that it is easy and requires less time for accessing encryption keys. Leaked Information from the vulnerable devices helps the attackers to exploit other side channel techniques such as performing power consuming attack and time-based attack.

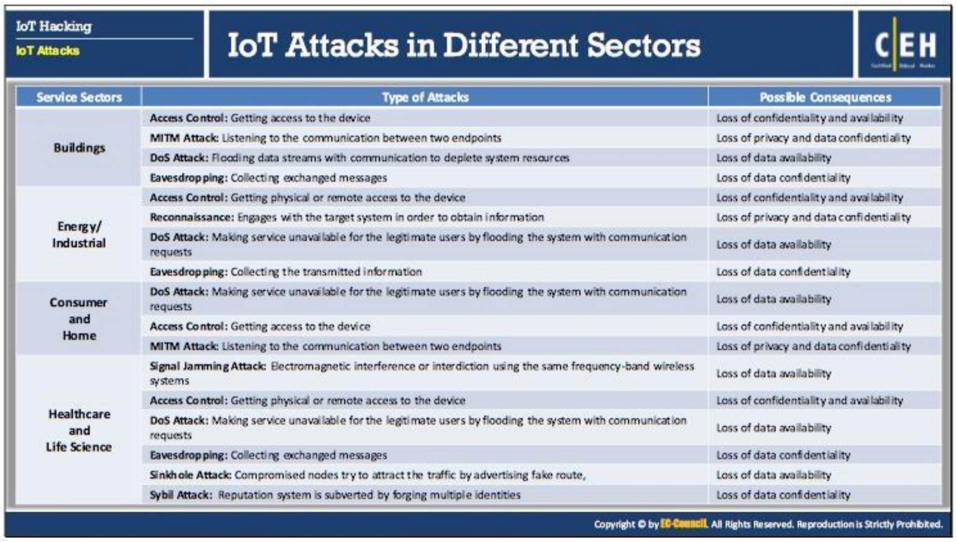
### Ransomware Attack

Ransomware is a type of malware that uses encryption to block user's access to his/her device either by locking the screen or by locking a user's files and it stays blocked until a ransom is paid that allows a user again to access his/her device.

A user can encounter this problem numerous ways. It can be mistakenly downloaded with some other malware or software or some files and sometimes through malicious advertisement (malvertisement).

Discussed below are some phases of Ransomware:

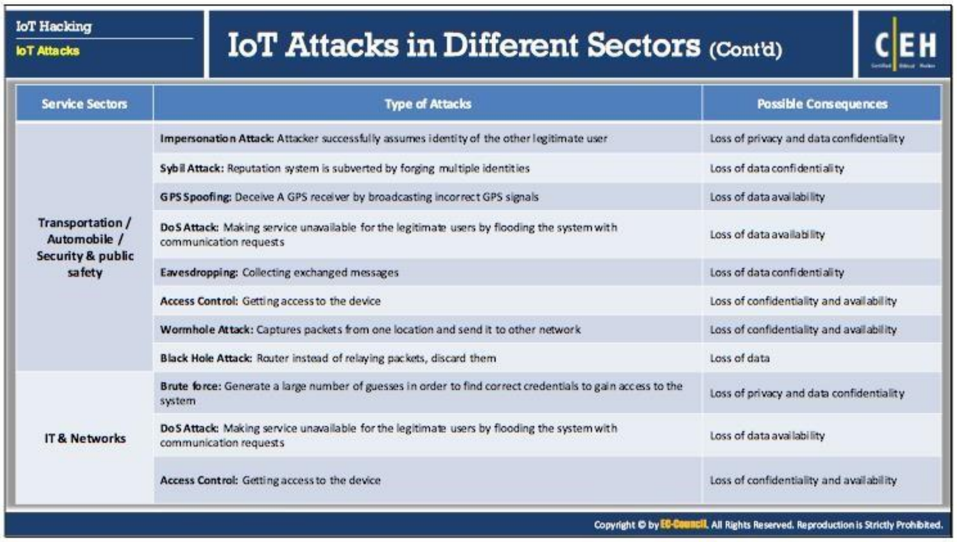
- **Phase 1**: Victim receives an email from the attacker that appears to be from a legitimate sender. This email contains an attachment of a malicious file.
- **Phase 2**:
  - User opens the mail and clicks the malicious file. Malware is downloaded and launches legitimate child processes like PowerShell, VSSasdmin encryption mechanism, cmd.exe. As a result, the device gets connected to an attacker's Command and Control (C&C) server.
  - The personal files on victim's device get encrypted.
- **Phase 3**: Notification of ransomware is delivered to the victim's device and he/she is asked to pay a ransom in the form of money or bitcoins to gain access to his/her files.



IoT Hacking
IoT Attacks

## IoT Attacks in Different Sectors

| Service Sectors | Type of Attacks | Possible Consequences |
|---|---|---|
| Buildings | **Access Control:** Getting access to the device | Loss of confidentiality and availability |
| | **MITM Attack:** Listening to the communication between two endpoints | Loss of privacy and data confidentiality |
| | **DoS Attack:** Flooding data streams with communication to deplete system resources | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| Energy/ Industrial | **Access Control:** Getting physical or remote access to the device | Loss of confidentiality and availability |
| | **Reconnaissance:** Engages with the target system in order to obtain information | Loss of privacy and data confidentiality |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting the transmitted information | Loss of data confidentiality |
| Consumer and Home | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Access Control:** Getting access to the device | Loss of confidentiality and availability |
| | **MITM Attack:** Listening to the communication between two endpoints | Loss of privacy and data confidentiality |
| Healthcare and Life Science | **Signal Jamming Attack:** Electromagnetic interference or interdiction using the same frequency-band wireless systems | Loss of data availability |
| | **Access Control:** Getting physical or remote access to the device | Loss of confidentiality and availability |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| | **Sinkhole Attack:** Compromised nodes try to attract the traffic by advertising fake route, | Loss of data availability |
| | **Sybil Attack:** Reputation system is subverted by forging multiple identities | Loss of data confidentiality |



IoT Hacking
IoT Attacks

## IoT Attacks in Different Sectors (Cont'd)

| Service Sectors | Type of Attacks | Possible Consequences |
|---|---|---|
| Transportation / Automobile / Security & public safety | **Impersonation Attack:** Attacker successfully assumes identity of the other legitimate user | Loss of privacy and data confidentiality |
| | **Sybil Attack:** Reputation system is subverted by forging multiple identities | Loss of data confidentiality |
| | **GPS Spoofing:** Deceive A GPS receiver by broadcasting incorrect GPS signals | Loss of data availability |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| | **Access Control:** Getting access to the device | Loss of confidentiality and availability |
| | **Wormhole Attack:** Captures packets from one location and send it to other network | Loss of confidentiality and availability |
| | **Black Hole Attack:** Router instead of relaying packets, discard them | Loss of data |
| IT & Networks | **Brute force:** Generate a large number of guesses in order to find correct credentials to gain access to the system | Loss of privacy and data confidentiality |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Access Control:** Getting access to the device | Loss of confidentiality and availability |

### IoT Attacks in Different Sectors

The internet of things (IoT) technology is making progress in each and every sector of society including industries, healthcare, agriculture, smart city, security, transportation, etc. But due to the implementation of decentralized approach in IoT technology, organizations focus less on the security of the devices. Therefore, rather than segmenting the IoT technology into different parts, suppliers focus more on spotting the vulnerabilities and exploiting them.

These vulnerabilities present in the IoT devices can be exploited by the attackers to launch various attacks such as DoS attack, jamming attack, MITM attack, Sybil attack, etc. and gather data which results in loss of privacy and confidentiality.

Different IoT sectors and the attacks associated with each of them is listed below:

| **Service Sectors** | **Type of Attacks** | **Possible Consequences** |
|---|---|---|
| **Buildings** | **Access Control:** Getting access to the device | Loss of confidentiality and availability |
| | **MITM Attack:** Listening to the communication between two endpoints | Loss of privacy and data confidentiality |
| | **DoS Attack:** Flooding data streams with communication to deplete system resources | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| **Energy/ Industrial** | **Access Control:** Getting physical or remote access to the device | Loss of confidentiality and availability |
| | **Reconnaissance:** Engages with the target system in order to obtain information | Loss of privacy and data confidentiality |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting the transmitted information | Loss of data confidentiality |
| **Consumer and Home** | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Access Control: G**etting access to the device | Loss of confidentiality and availability |
| | **MITM Attack:** Listening to the communication between two endpoints. | Loss of privacy and data confidentiality |
| **Healthcare and Life Science** | **Signal Jamming Attack:** Electromagnetic interference or interdiction using the same frequency-band wireless systems | Loss of data availability |
| | **Access Control:** Getting physical or remote access to the device | Loss of confidentiality and availability |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| | **Sinkhole Attack:** Compromised nodes try to attract the traffic by advertising fake route, | Loss of data availability |

| | | |
|---|---|---|
| | **Sybil Attack:** Reputation system is subverted by forging multiple identities | Loss of data confidentiality |
| **Transportation / Automobile / Security & public safety** | **Impersonation Attack:** Attacker successfully assumes identity of the other legitimate user | Loss of privacy and data confidentiality |
| | **Sybil Attack:** Reputation system is subverted by forging multiple identities | Loss of data confidentiality |
| | **GPS Spoofing:** Deceive A GPS receiver by broadcasting incorrect GPS signals. | Loss of data availability |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Eavesdropping:** Collecting exchanged messages | Loss of data confidentiality |
| | **Access Control:** Getting access to the device | Loss of confidentiality and availability |
| | **Wormhole Attack:** Captures packets from one location and send it to other network | Loss of confidentiality and availability |
| | **Black Hole Attack:** Router discards packets instead of relaying them. | Loss of data |
| **IT & Networks** | **Brute force:** Generate a large number of guesses in order to find correct credentials to gain access to the system | Loss of privacy and data confidentiality |
| | **DoS Attack:** Making service unavailable for the legitimate users by flooding the system with communication requests | Loss of data availability |
| | **Access Control:** Getting access to the device | Loss of confidentiality and availability |

TABLE 18.3: IoT application areas and attacks

IoT Hacking
IoT Attacks

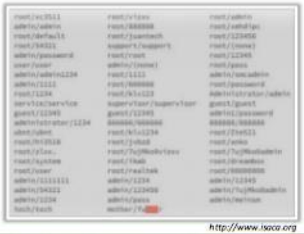
## Case Study: Dyn Attack

- Mirai is a **piece of malware** that deliberately finds Internet of Things (IoT) devices to infect
- Once infected, Mirai adds the **infected IoT** to a **botnet**
- Mirai was built for two main purposes:
  - Find and **infect other IoT devices** to further grow the botnet
  - Participate in DDoS attacks based upon commands received from a remote **C&C infrastructure**

### Stage 1: Infect the Device

- **Continuously scan** for IoT devices that are accessible over the Internet
  - It primarily scans for ports 22, 23, 5747, etc. that are open, and can easily be configured to scan for others
- Once connected to an IoT, it attempts to login using list of **username/password** combinations included in the malware, gain access, and **infect the device**
- Infected device then **scans other networks** looking for more IoT devices and **launches DDoS attacks**

List of username/password combinations included in the malware



```
root/xc3511          root/vizxv                root/admin
admin/admin          root/888888               root/xmhdipc
root/default         root/juantech             root/123456
root/54321           support/support           root/(none)
admin/password       root/root                 root/12345
user/user            admin/(none)              root/pass
admin/admin1234      root/1111                 admin/smcadmin
admin/1111           root/666666               root/password
root/1234            root/klv123               Administrator/admin
service/service      supervisor/supervisor     guest/guest
guest/12345          guest/12345               admin1/password
administrator/1234   666666/666666             888888/888888
ubnt/ubnt            root/klv1234              root/Zte521
root/hi3518          root/jvbzd                root/anko
root/zlxx.           root/7ujMko0vizxv         root/7ujMko0admin
root/system          root/ikwb                 root/dreambox
root/user            root/realtek              root/00000000
admin/1111111        admin/1234                admin/12345
admin/54321          admin/123456              admin/7ujMko0admin
admin/1234           admin/pass                admin/meinsm
tech/tech            mother/fu[redacted]
```

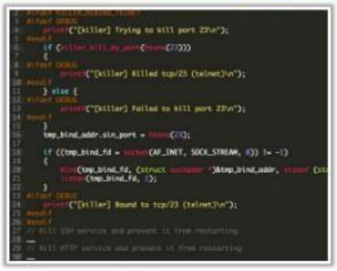
http://www.isaca.org

IoT Hacking
IoT Attacks

## Case Study: Dyn Attack (Cont'd)

### Stage 2: Protect Itself

- Kills other **process running** on the IoT device such as SSH, Telnet, HTTP to prevent the owner from **gaining remote access** to the IoT device while infected
- **Rebooting the IoT device** can remove the malware, but it can quickly become infected again

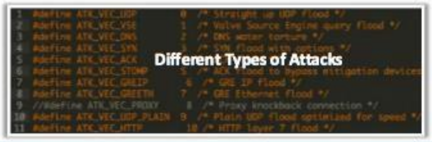


```
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
    tmp_bind_addr.sin_port = htons(23);

    if ((tmp_bind_fd = socket(AF_INET, SOCK_STREAM, 0)) != -1)
    {
        bind(tmp_bind_fd, (struct sockaddr *)&tmp_bind_addr, sizeof (st
        listen(tmp_bind_fd, 1);
    }
#ifdef DEBUG
    printf("[killer] Bound to tcp/23 (telnet)\n");
#endif
#endif
// Kill SSH service and prevent it from restarting
…
// Kill HTTP service and prevent it from restarting
…
```

### Stage 3: Launch Attack

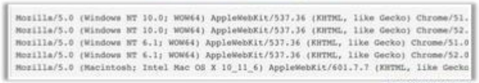
- Mirai infected IoT device **launch different types of attacks** as a part of the malware
- When attacking **using HTTP GET Floods**, Mirai bots uses a list of default user-agents



Different Types of Attacks

```
#define ATK_VEC_UDP        0  /* Straight up UDP flood */
#define ATK_VEC_VSE        1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS        2  /* DNS water torture */
#define ATK_VEC_SYN        3  /* SYN flood with options */
#define ATK_VEC_ACK        4  /* ACK flood */
#define ATK_VEC_STOMP      5  /* ACK flood to bypass mitigation devices
#define ATK_VEC_GREIP      6  /* GRE IP flood */
#define ATK_VEC_GREETH     7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY      8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN  9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP       10 /* HTTP layer 7 flood */
```

**List of Default User-agents**



```
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko
```

http://www.isaca.org

## Case Study: Dyn Attack

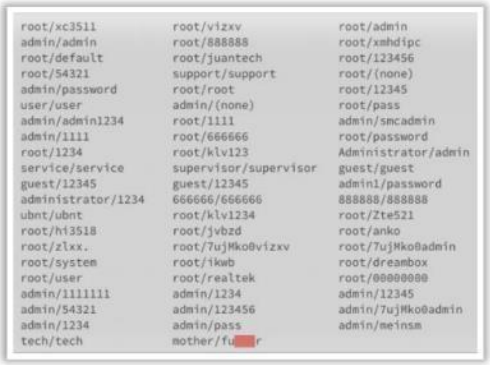
Source: *http://www.isaca.org*

Mirai is a piece of malware that deliberately finds Internet of Things (IoT) devices to infect. Once infected, Mirai adds the infected IoT to a botnet. Mirai was built for two main purposes: find and infect other IoT devices to further grow the botnet and participate in DDoS attacks based upon commands received from a remote C&C infrastructure. Mirai was responsible for the 1+ Tbps attack on OVH and DYN in October 2016.

## How Does It Work

- **Stage 1: Infect the Device**

  The attack starts by exploiting weak default security on many IoT devices. The malware operates by continuously scanning for IoT devices that are accessible over the Internet. It primarily scans for ports 22, 23, 5747, etc. that are open, and can easily be configured to scan for others. Once connected to an IoT, Mirai attempts to login using a list of username/password combinations included in the malware, gain access, and infect the device. The infected device then scans other networks looking for more IoT devices and launches DDoS attacks.

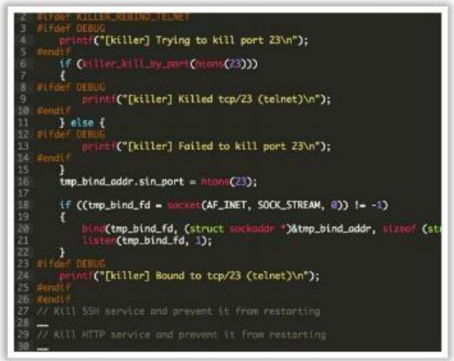


```
root/xc3511            root/vizxv              root/admin
admin/admin            root/888888             root/xmhdipc
root/default           root/juantech           root/123456
root/54321             support/support         root/(none)
admin/password         root/root               root/12345
user/user              admin/(none)            root/pass
admin/admin1234        root/1111               admin/smcadmin
admin/1111             root/666666             root/password
root/1234              root/klv123             Administrator/admin
service/service        supervisor/supervisor   guest/guest
guest/12345            guest/12345             admin1/password
administrator/1234     666666/666666           888888/888888
ubnt/ubnt              root/klv1234            root/Zte521
root/hi3518            root/jvbzd              root/anko
root/zlxx.             root/7ujMko0vizxv       root/7ujMko0admin
root/system            root/ikwb               root/dreambox
root/user              root/realtek            root/00000000
admin/1111111          admin/1234              admin/12345
admin/54321            admin/123456            admin/7ujMko0admin
admin/1234             admin/pass              admin/meinsm
tech/tech              mother/fu[redacted]r
```

FIGURE 18.1: List of username/passwords included in Mirai

- **Stage 2: Protect Itself**

  Mirai kills other processes running on the IoT device like SSH, Telnet, HTTP. It does this to prevent the owner from gaining remote access to the IoT device while infected. Rebooting the IoT device can remove the malware, but it can quickly become infected again.



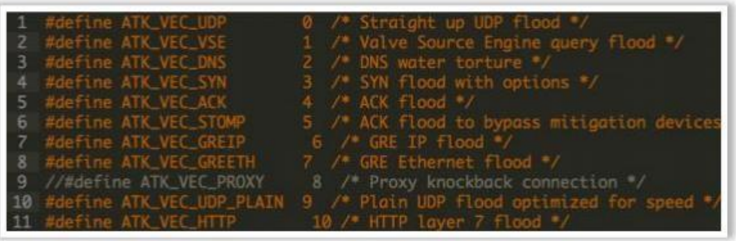
```c
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
    tmp_bind_addr.sin_port = htons(23);

    if ((tmp_bind_fd = socket(AF_INET, SOCK_STREAM, 0)) != -1)
    {
        bind(tmp_bind_fd, (struct sockaddr *)&tmp_bind_addr, sizeof (st
        listen(tmp_bind_fd, 1);
    }
#ifdef DEBUG
    printf("[killer] Bound to tcp/23 (telnet)\n");
#endif
#endif
// Kill SSH service and prevent it from restarting
…
// Kill HTTP service and prevent it from restarting
…
```

FIGURE 18.2: Stage 2 of Mirai attack

- **Stage 3: Launch Attack**

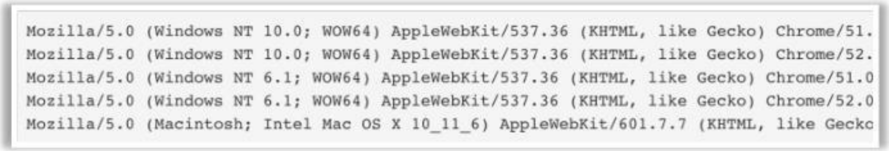
  After successfully infecting devices, Mirai infected IoT device launch different types of attacks as a part of the malware.



```
#define ATK_VEC_UDP        0  /* Straight up UDP flood */
#define ATK_VEC_VSE        1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS        2  /* DNS water torture */
#define ATK_VEC_SYN        3  /* SYN flood with options */
#define ATK_VEC_ACK        4  /* ACK flood */
#define ATK_VEC_STOMP      5  /* ACK flood to bypass mitigation devices
#define ATK_VEC_GREIP       6  /* GRE IP flood */
#define ATK_VEC_GREETH     7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY     8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN  9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP        10 /* HTTP layer 7 flood */
```

FIGURE 18.3: Different types of attacks

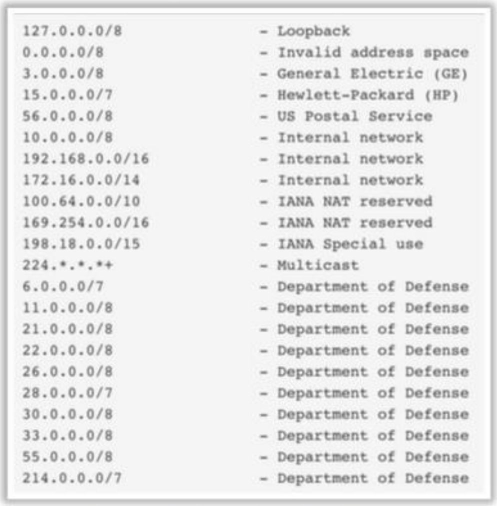
When attacking using HTTP GET floods, Mirai bots will use a list of default user-agents.



```
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko
```

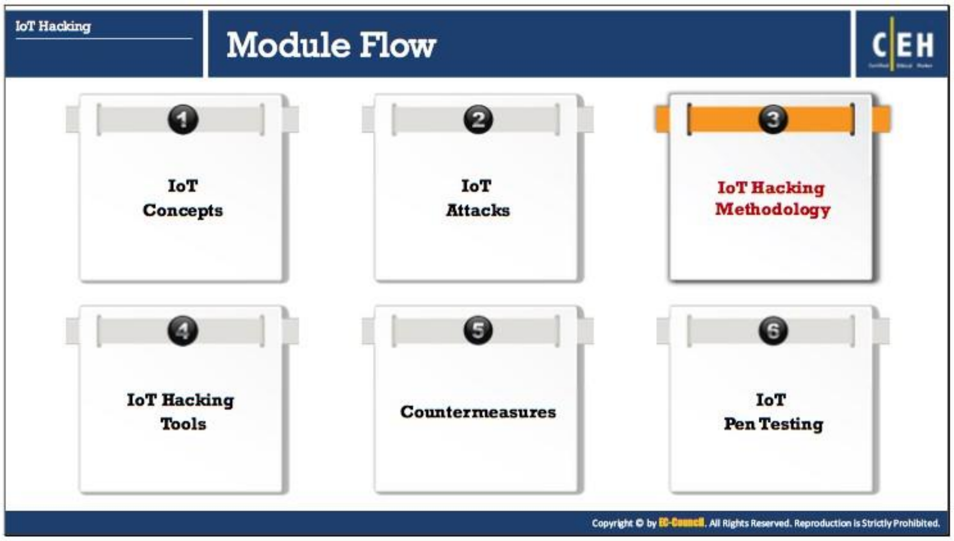
FIGURE 18.4: List of default user-agents used by Mirai Bots

One unique attribute of the malware was that it included a list of known networks in the U.S. Mirai bots are told to avoid these networks when scanning for other vulnerable IoT devices



```
127.0.0.0/8        - Loopback
0.0.0.0/8          - Invalid address space
3.0.0.0/8          - General Electric (GE)
15.0.0.0/7         - Hewlett-Packard (HP)
56.0.0.0/8         - US Postal Service
10.0.0.0/8         - Internal network
192.168.0.0/16     - Internal network
172.16.0.0/14      - Internal network
100.64.0.0/10      - IANA NAT reserved
169.254.0.0/16     - IANA NAT reserved
198.18.0.0/15      - IANA Special use
224.*.*.*+         - Multicast
6.0.0.0/7          - Department of Defense
11.0.0.0/8         - Department of Defense
21.0.0.0/8         - Department of Defense
22.0.0.0/8         - Department of Defense
26.0.0.0/8         - Department of Defense
28.0.0.0/7         - Department of Defense
30.0.0.0/8         - Department of Defense
33.0.0.0/8         - Department of Defense
55.0.0.0/8         - Department of Defense
214.0.0.0/7        - Department of Defense
```

FIGURE 18.5: List of known networks in US



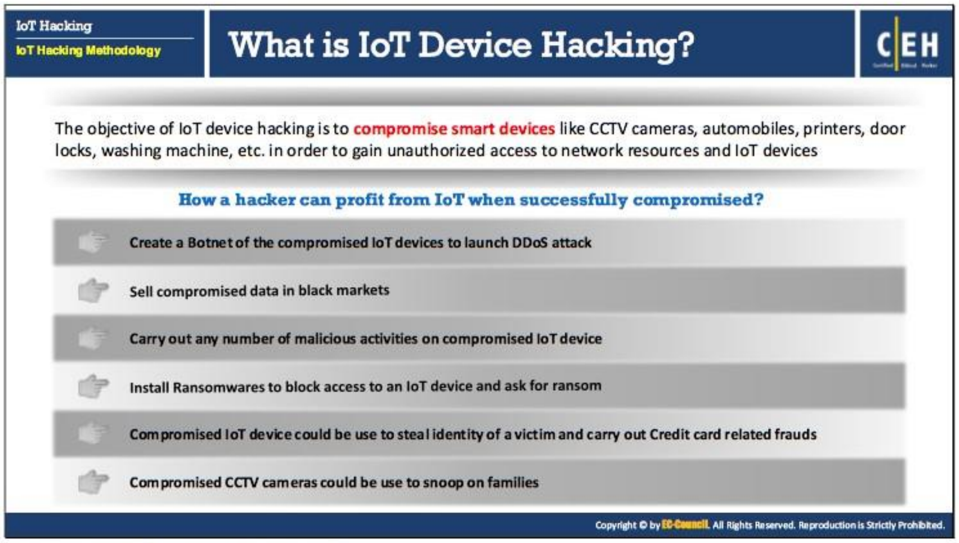
IoT Hacking

## Module Flow

1. IoT Concepts
2. IoT Attacks
3. IoT Hacking Methodology
4. IoT Hacking Tools
5. Countermeasures
6. IoT Pen Testing

## IoT Hacking Methodology

Using the IoT hacking methodology, an attacker acquires information through techniques such as gathering information, identifying attack surface area, and vulnerability scanning, and uses it to hack the target device and network. This section will focus on the tools and techniques used by the attacker to achieve his/her goal of hacking the target IoT device.



IoT Hacking
IoT Hacking Methodology

## What is IoT Device Hacking?

The objective of IoT device hacking is to **compromise smart devices** like CCTV cameras, automobiles, printers, door locks, washing machine, etc. in order to gain unauthorized access to network resources and IoT devices

**How a hacker can profit from IoT when successfully compromised?**

- Create a Botnet of the compromised IoT devices to launch DDoS attack
- Sell compromised data in black markets
- Carry out any number of malicious activities on compromised IoT device
- Install Ransomwares to block access to an IoT device and ask for ransom
- Compromised IoT device could be use to steal identity of a victim and carry out Credit card related frauds
- Compromised CCTV cameras could be use to snoop on families

## What is IoT Device Hacking?

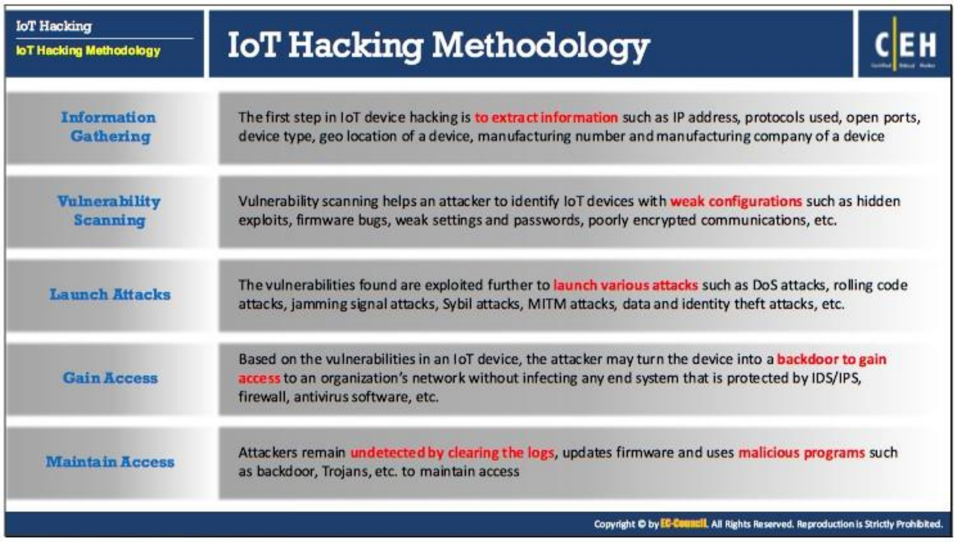
Due to the significant growth of paradigm of Internet of Things, more and more devices are coming into our lives every day. From the automation of home to the healthcare application, IoT is everywhere. But with the promise to make our lives easier and more comfortable, we cannot underestimate the risk of cyber-attacks. These IoT devices lack basic security thus making them prone to various cyber-attacks.

The objective of a hacker in exploiting IoT device is to get an unauthorized access to the user's device and data. A hacker can use compromised IoT devices in order to build up an army of Botnets which in turn is used to launch a DDoS attack.

**How a hacker gains profit from IoT when successfully compromised?**

Today all your data, location, email accounts, financial information, pictures reside on your smart devices or IoT devices which is like a treasure trove of data for hackers. With the increase in selling and buying of IoT devices in the market, they are outnumbering people. The number of IoT devices is expected to reach 29.5 billion in 2020.

Due to lack of security policies, smart devices become an easy target for a hacker who can compromise the device to spy on user activities, misuse the sensitive information (such as patient's health record, etc.), install ransomware to block access to the device, monitor a victim's activities using CCTV cameras, carry on credit card related frauds, get an access to the user's home or making the device a part of an army of Botnets to carry out DDoS attacks.



IoT Hacking
IoT Hacking Methodology

## IoT Hacking Methodology

| | |
|---|---|
| **Information Gathering** | The first step in IoT device hacking is **to extract information** such as IP address, protocols used, open ports, device type, geo location of a device, manufacturing number and manufacturing company of a device |
| **Vulnerability Scanning** | Vulnerability scanning helps an attacker to identify IoT devices with **weak configurations** such as hidden exploits, firmware bugs, weak settings and passwords, poorly encrypted communications, etc. |
| **Launch Attacks** | The vulnerabilities found are exploited further to **launch various attacks** such as DoS attacks, rolling code attacks, jamming signal attacks, Sybil attacks, MITM attacks, data and identity theft attacks, etc. |
| **Gain Access** | Based on the vulnerabilities in an IoT device, the attacker may turn the device into a **backdoor to gain access** to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc. |
| **Maintain Access** | Attackers remain **undetected by clearing the logs**, updates firmware and uses **malicious programs** such as backdoor, Trojans, etc. to maintain access |

## IoT Hacking Methodology

The following are different phases in hacking IoT devices:

- Information Gathering
- Vulnerability Scanning
- Launch Attacks
- Gain Access
- Maintain Access

IoT Hacking
IoT Hacking Methodology

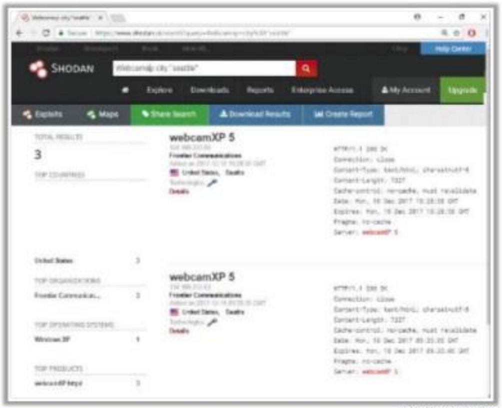
## Information Gathering using Shodan

- Shodan provides an information about all the **internet connected devices** such as routers, traffic lights, CCTV cameras, servers, smart home devices, etc.
- Attackers can make use of this tool to gather information such as **IP address, hostname**, ISP, **device's location** and **the banner of the target IoT device**
- Attackers can gather information on a target device using filters given below:
  - Search for webcams using geolocation:

    `webcamxp country:"US"`
  - Search using city:

    `Webcamxp city:"seattle"`
  - Find webcam using longitude and latitude:

    `Webcamxp geo:" -50.81,201.80"`



```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 7327
Cache-control: no-cache, must revalidate
Date: Mon, 18 Dec 2017 [illegible] GMT
Expires: Mon, 18 Dec 2017 [illegible] GMT
Pragma: no-cache
Server: webcamXP 5
```

https://www.shodan.io

### Information Gathering

The first and the foremost step in IoT device hacking is to extract information such as IP address, protocols used (Zigbee, BLE, 5G, IPv6LoWPAN, etc.), open ports, device type, Geo location of a device, manufacturing number and manufacturing company of a device. In this step, an attacker also identifies the hardware design, its infrastructure and the main components embedded on a target device that is present online. Attackers make use of tools such as Shodan, Censys, Thingful, etc. to perform information gathering or reconnaissance on a target device. The devices that are unavailable in the network but within the communication area can also be detected by using sniffers such as Foren6, Z-Wave Sniffer, CloudShark, Wireshark, etc.

### Information Gathering using Shodan

Source: *https://www.shodan.io*

Shodan is a search engine that provides information about all the internet connected devices such as routers, traffic lights, CCTV cameras, servers, smart home devices, industrial devices, etc. Attackers can make use of this tool to gather information such as IP address, hostname, ISP, device's location and the banner of the target IoT device.

Attackers can gather information on a target device using filters given below:

- **Search for webcams using geolocation:**

  `webcamxp country:"US"` (Obtains all the webcamxp webcams present in US.)
- **Search using city:**

  `Webcamxp city:"seattle"` (Obtains existing webcamxp webcams in Seattle.)

- **Find webcam using longitude and latitude:**

  `Webcamxp geo:" -50.81,201.80"` (Obtains a specific webcam present at the geolocation "-50.81,201.80" in the city Boston and country US.)

Additional filters used by the attackers to obtain target information:

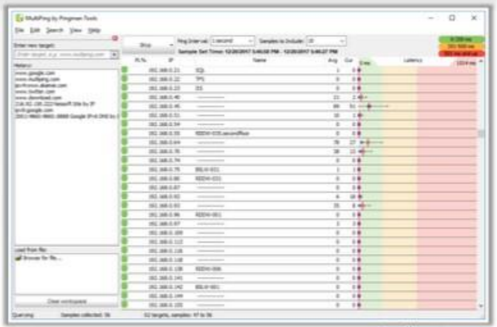
- **Net:** Search on the basis of the IP address or CIDR
- **OS**: Search on the basis of the operating system used by the devices
- **Port**: Find all open ports
- **Before/after**: Provides result within a certain timeframe

IoT Hacking
**IoT Hacking Methodology**

## Information Gathering using MultiPing

- An attacker can use MultiPing to **find IP address of any IoT device** in the target network
- After obtaining the IP address of an IoT device, the attacker can perform further scanning to **identify vulnerabilities** present in that device

- Steps to perform scanning to identify the IP address of any IoT device:
  - Open **MultiPing** application and click **File → Add Address Range**
  - Select router's gateway IP address from the **Initial Address to add** drop-down field
  - Set the **Number of addresses** to **"255"**, and click the **OK** button
  - MultiPing will cycle through every possible IP on the range you selected and it begins testing every IP address that responds to its ping
  - Each row in **MultiPing Window** is a device on the network. From the list, the attacker can identify the IP address of the target IoT device
  - To find the target device faster, set the **ping interval to 1**



*https://www.pingman.com*

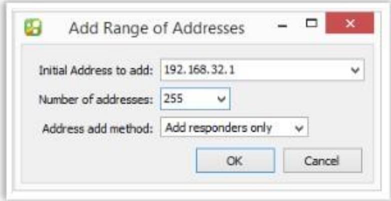
### Information Gathering using MultiPing

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

An attacker can use the MultiPing tool to find IP address of any IoT device in the target network. After obtaining the IP address of an IoT device, the attacker can perform further scanning to identify vulnerabilities present in that device.

Steps to perform scanning to identify IP address of any IoT device:

- Open **MultiPing** application and click **File → Add Address Range**
- In the Add Range of Address popup window:
  - Select router's gateway IP address from the **Initial Address to Add** drop-down field
  - Set the **Number of addresses** to **"255"**
  - Click the **OK**



Add Range of Addresses

Initial Address to add: 192.168.32.1

Number of addresses: 255

Address add method: Add responders only

FIGURE 18.6: Adding range of IP addresses in MultiPing

- MultiPing will cycle through every possible IP on the range you selected and it begins testing every IP address that responds to its ping

- Each row on MultiPing Window is a device on the network. From the list, the attacker can identify the IP address of the target IoT device
- To find the target device faster, set the ping interval to 1

IoT Hacking
**IoT Hacking Methodology**

## Vulnerability Scanning using Nmap

CEH

- Attackers use **vulnerability scanning tools** such as Nmap to identify all the IoT devices connected to the network along with their **open ports** and **services**

**Scanning for Vulnerabilities using Nmap**

To scan for a particular IP address

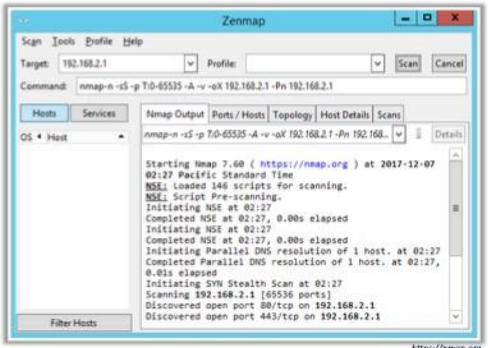
```
nmap -n -Pn -sS -pT:0-65535 -v -A -oX
<Name> <IP>
```

To check for open TCP and UDP services and ports

```
nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v
-A -oX <Name> <IP>
```

To identify the IPv6 capabilities of a device

```
nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535
-v -A -oX <Name> <IP>
```



Target: 192.168.2.1

Command: nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168.2.1

nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168...

```
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-07
02:27 Pacific Standard Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:27
Completed Parallel DNS resolution of 1 host. at 02:27,
0.01s elapsed
Initiating SYN Stealth Scan at 02:27
Scanning 192.168.2.1 [65536 ports]
Discovered open port 80/tcp on 192.168.2.1
Discovered open port 443/tcp on 192.168.2.1
```

https://nmap.org

## Vulnerability Scanning

Once the attackers gather information about a target device, they search for the attack surfaces of a device (identify the vulnerabilities) which they can attack. Vulnerability scanning allows an attacker to find the total number of vulnerabilities present in the firmware, infrastructure and system components of an IoT device that is accessible. After identifying the attack surface area, the attacker will scan for vulnerabilities in that area to identify an attack vector and perform further exploitation on the device.

Vulnerability scanning helps an attacker to identify IoT devices with weak configurations such as hidden exploits, firmware bugs, weak settings and passwords, poorly encrypted communications, etc. On the other hand, it also assists security professionals in securing the IoT devices in the network by determining the security loopholes or vulnerabilities in the current security mechanisms before the attackers can exploit them.

### Vulnerability Scanning using Nmap

Attackers use vulnerability-scanning tools such as Nmap to identify the IoT devices connected to the network along with their open ports and services. Nmap generates raw IP packets in different ways to identify live hosts or devices on the network, services offered by them, their operating systems, type of packet filters used, etc.

Attackers use the following Nmap command to scan a particular IP address:

```
nmap -n -Pn -sS -pT:0-65535 -v -A -oX <Name><IP>
```

To perform complete scan of the IoT device that checks for both TCP and UDP services and ports:

```
nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>
```

To identify the IPv6 capabilities of a device,

```
nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>
```

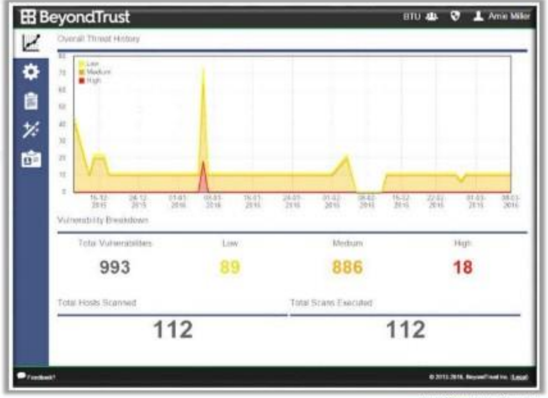
IoT Hacking
**IoT Hacking Methodology**

## Vulnerability Scanning using RIoT Vulnerability Scanner

- Retina IoT vulnerability scanner **identify at-risk IoT devices**, such as IP cameras, DVRs, printers, routers, etc.
- This tool gives you an attacker's view of all the IoT devices and their **associated vulnerabilities**

**Features**

- Identify vulnerable IoT devices
- Check for default or hard-coded passwords
- Perform external scans of up to 256 IP addresses
- Generates reports of IoT vulnerabilities and their remediation



Overall Threat History

Vulnerability Breakdown

| Total Vulnerabilities | Low | Medium | High |
|---|---|---|---|
| 993 | 89 | 886 | 18 |

| Total Hosts Scanned | Total Scans Executed |
|---|---|
| 112 | 112 |

https://www.beyondtrust.com

### Vulnerability Scanning using RIoT Vulnerability Scanner

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

Retina IoT vulnerability scanner identify at-risk IoT devices, such as IP cameras, DVRs, printers, routers, etc. This tool gives you an attacker's view of all the IoT devices and their associated vulnerabilities. Utilizing precise information such as server banner and header data, RIoT will pinpoint the make and model of a particular IoT device. It also performs tests whether that device is using default or hard-coded credentials for Telnet, SSH or basic HTTP authentication, which are the preferred attack vectors that botnets initially use to breach a system. Using this tool an attacker can simply specify a target IP or IP range to identify vulnerabilities.

Features of RIoT vulnerability scanner:

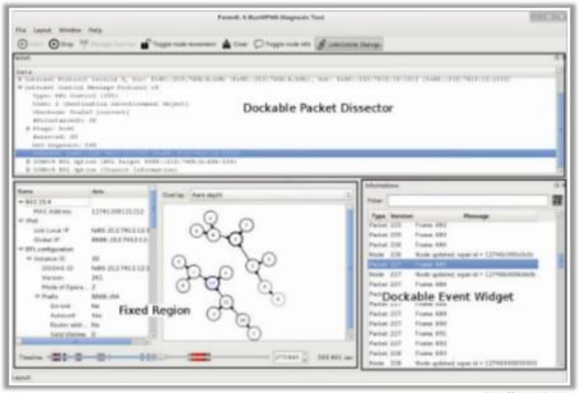
- Identify vulnerable IoT devices
- Check for default or hard-coded passwords
- Perform external scans of up to 256 IPs
- Generates reports of IoT vulnerabilities and their remediation

IoT Hacking

IoT Hacking Methodology

## Sniffing using Foren6

- Attackers use tools like Foren6 to **sniff the traffic** of IoT devices
- Foren6 uses sniffers **to capture 6LoWPAN traffic** and renders the network state it in a graphical user interface
- Foren6 captures all **RPL-related information** and identifies abnormal behaviors
- It combines multiple sniffers and **captures live packets** from deployed networks in a non-intrusive manner



http://cetic.github.io

### Sniffing using Foren6

*Source: http://cetic.github.io*

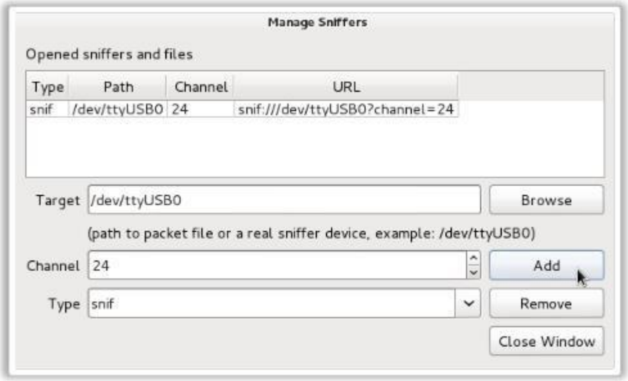
Attackers use tools like Foren6 to sniff the traffic of IoT devices. Foren6 is a non-intrusive 6LoWPAN network analysis tool. It leverages passive sniffer devices to reconstruct a visual and textual representation of network information to support real-world Internet of Things applications.

Foren6 uses sniffers to capture 6LoWPAN traffic and renders the network state in a graphical user interface. It detects routing problems. The Routing Protocol for 6LoWPAN Networks, RPL, is an emerging IETF standard. Foren6 captures all RPL-related information and identifies abnormal behaviors. It combines multiple sniffers and captures live packets from deployed networks in a non-intrusive manner.

For example, the basic steps to analyze a real 6LoWPAN network using a Contiki-based sniffer module.

- Open **Foren6**, after installation
- Now, Open the **'Manage Sources'** dialog by clicking the **Manage Sources** button in the Toolbar or from the '**File**' menu
- In this dialog, remove any existing entries in the top section by selecting each individual element and hitting the '**Remove**' button
- Next, add a new source by specifying the three fields as shown below:
    - **Target**: Type the path to the USB device (example: /dev/ttyUSB0)
    - **Channel**: The integer value of the Channel you want to snif (1 to 26)
    - **Type**: Select snif

- Click the **Add** button when the above information is entered
- If the device is found by the application, it will appear in the list of available devices. If your device exists but you get an error at this point, is it likely that the user running Foren6 does not have permission to access that serial device. Then, launch the foren6 application as root.
- Hit the **Close** button to return to the main window
- Click the **Start** button (which will now be enabled) to launch a packet capture



Manage Sniffers

Opened sniffers and files

| Type | Path | Channel | URL |
|---|---|---|---|
| snif | /dev/ttyUSB0 | 24 | snif:///dev/ttyUSB0?channel=24 |

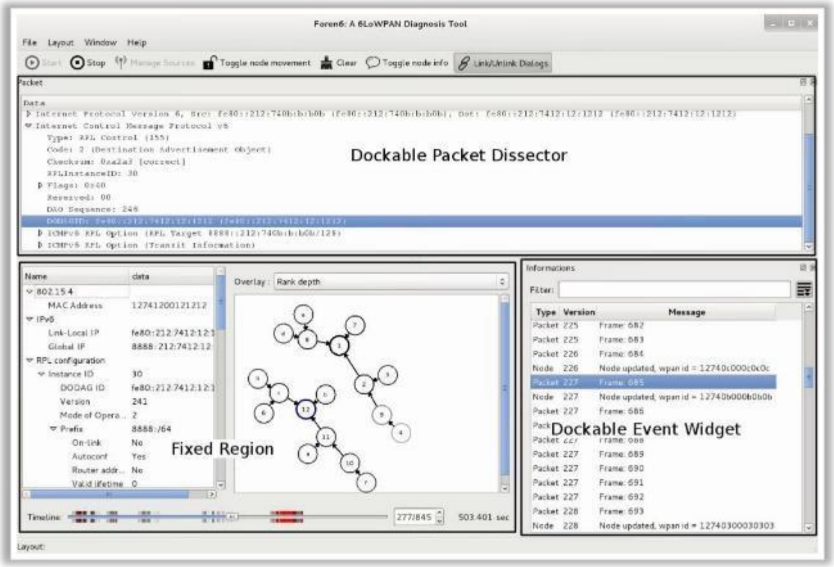
Target /dev/ttyUSB0

(path to packet file or a real sniffer device, example: /dev/ttyUSB0)

Channel 24

Type snif

FIGURE 18.7: Sniffing using Foren6



```
Data
▷ Internet Protocol Version 6, Src: fe80::212:740b:b:b0b (fe80::212:740b:b:b0b), Dst: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
▽ Internet Control Message Protocol v6
    Type: RPL Control (155)
    Code: 2 (Destination Advertisement Object)
    Checksum: 0xa2a3 [correct]
    RPLInstanceID: 30
  ▷ Flags: 0x40
    Reserved: 00
    DAO Sequence: 248
    DODAGID: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
  ▷ ICMPv6 RPL Option (RPL Target 8888::212:740b:b:b0b/128)
  ▷ ICMPv6 RPL Option (Transit Information)
```

| Name | data |
|---|---|
| 802.15.4 | |
| MAC Address | 12741200121212 |
| IPv6 | |
| Link-Local IP | fe80::212:7412:12:1 |
| Global IP | 8888::212:7412:12 |
| RPL configuration | |
| Instance ID | 30 |
| DODAG ID | fe80::212:7412:12:1 |
| Version | 241 |
| Mode of Opera... | 2 |
| Prefix | 8888::/64 |
| On-link | No |
| Autoconf | Yes |
| Router addr... | No |
| Valid lifetime | 0 |

Overlay: Rank depth

277/845 503.401 sec

```
Type    Version    Message
Packet  225        Frame: 682
Packet  225        Frame: 683
Packet  226        Frame: 684
Node    226        Node updated, wpan id = 12740c000c0c0c
Packet  227        Frame: 685
Node    227        Node updated, wpan id = 12740b000b0b0b
Packet  227        Frame: 686
Pack
Packet  227        Frame: 688
Packet  227        Frame: 689
Packet  227        Frame: 690
Packet  227        Frame: 691
Packet  227        Frame: 692
Packet  228        Frame: 693
Node    228        Node updated, wpan id = 12740300030303
```

FIGURE 18.8: Screenshot of Foren6 showing various result panes

### Launch Attacks

In vulnerability scanning phase, attackers try to find out the vulnerabilities present in the target device. The vulnerabilities found are then exploited further to launch various attacks such as DDoS attacks, rolling code attacks, jamming signal attacks, Sybil attacks, MITM attacks, data and identity theft attacks, etc. For example, an attacker can make use of RFCrack tool to perform rolling code attack, replay attack and jamming attack on a device. Similarly, an attacker may also use tools such as KillerBee to attack ZigBee and IEEE 802.15.4 networks.

IoT Hacking
IoT Hacking Methodology

## Rolling Code Attack using RFCrack

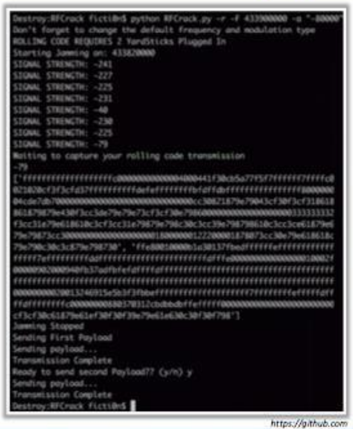
- Attackers use RFCrack tool to obtain the **rolling code** sent by the victim to **unlock the vehicle** and later use the same code for unlocking and stealing the vehicle
- RFCrack is used for **testing RF communications** between any physical device that communicates over sub **Ghz frequencies**
- Some of the commands used by an attacker to perform rolling code attack, are given below:
  - Live Replay:

    ```
    python RFCrack.py -i
    ```
  - Rolling Code:

    ```
    python RFCrack.py -r -M MOD_2FSK -F 314350000
    ```
  - Adjust RSSI Range:

    ```
    python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F
    314350000
    ```
  - Jamming:

    ```
    python RFCrack.py -j -F 314000000
    ```



```
Destroy:RFCrack ficti0n$ python RFCrack.py -r -F 433900000 -u "-80000"
Don't forget to change the default frequency and modulation type
ROLLING CODE REQUIRES 2 YardSticks Plugged In
Starting Jamming on: 433820000
SIGNAL STRENGTH: -241
SIGNAL STRENGTH: -227
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -231
SIGNAL STRENGTH: -40
SIGNAL STRENGTH: -230
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -79
Waiting to capture your rolling code transmission
-79
['ffffffffffffffffffffc000000000000004000441f30cb5a77f5f7fffffff7ffffc0
021020cf3f3cfd37ffffffffffffaefeffffffffbfdffdbfffffffffffffffff800000
04cde7db70000000000000000000000000000000cc30821879e79043cf30f3cf318618
8618798 79e430f3cc3de79e79e73cf3cf30e7986000000000000000000000001333333332
f3cc31e79e61861 0c3cf3cc31e79879e798c30c3cc39e798798610c3cc3ce61879e6
79e79873cc300000000000000000000000001800000012220000187887 3cc30e79e618618c
79e798c30c3c879e798730', 'ffe0001 0000b1a30137fbedffffffefffffffffffff
fffff7effffffffffaffffffffffffffffffffffffdfffe0000000000000000001000 2f
0000000200003 40fb37adfbfefdfffffdffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
0000000002001324 6915e5b3f3fbbeffffffffffffffffffffffff7ffffffffefffffdff
ffdffffffffc0000000000 80378312cbdbbdbffefffff0000000000000000000000000
cf3cf30c61879e61ef30f30f39e79e61e630c30f30f798']
Jamming Stopped
Sending First Payload
Sending payload...
Transmission Complete
Ready to send second Payload?? (y/n) y
Sending payload...
Transmission Complete
Destroy:RFCrack ficti0n$ 
```

https://github.com

**Rolling Code Attack using RFCrack**

Source: *https://github.com*

Attackers use the RFCrack tool to obtain the rolling code sent by the victim to unlock a vehicle and later use the same code for unlocking and stealing the vehicle. RFCrack is used for testing RF communications between any physical device that communicates over sub Ghz frequencies. It is used along with the combination of hardware such as yardsticks to jam, replay and sniff the signal coming from the sender.

Features of RFCrack:

- Perform replay attacks (-i -F)
- Send Saved Payloads (-s –u)
- Perform Rolling code bypass attacks (-r -F -M)
- Perform jamming (-j -F)
- Scanning incrementally through frequencies (-b -v -F)
- Scanning common frequencies (-k)

Commands used by an attacker to perform rolling code attack, are given below:

- Live Replay:

  ```
  python RFCrack.py -i
  ```
- Rolling Code:

  ```
  python RFCrack.py -r -M MOD_2FSK -F 314350000
  ```
- Adjust RSSI Range:

```
python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F 314350000
```

- Jamming:

```
python RFCrack.py -j -F 314000000
```

- Scan common freq:

```
python RFCrack.py -k
```

- Scan with your list:

```
python RFCrack.py -k -f 433000000 314000000 390000000
```

- Incremental Scan:

```
python RFCrack.py -b -v 5000000
```

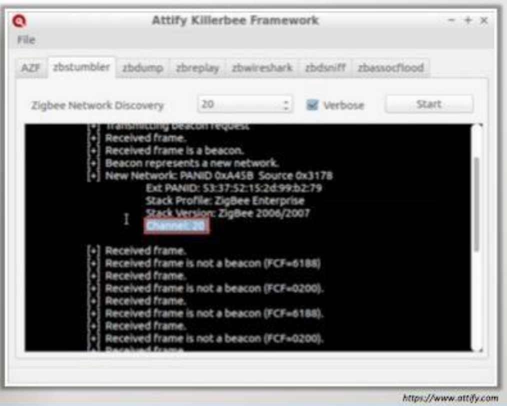
- Send Saved Payload:

```
python RFCrack.py -s -u ./files/test.cap -F 315000000 -M
MOD_ASK_OOK
```

IoT Hacking
**IoT Hacking Methodology**

## Hacking Zigbee Devices with Attify Zigbee Framework

- Most of the IoT devices use ZigBee protocol for **short range wireless communication**
- Attackers find **vulnerabilities in ZigBee** based IoT and smart devices and exploit them using tools like Attify ZigBee Framework
- ZigBee protocol makes use of **16 different channels** for all communications
- Attackers use **Zbstumbler** from Attify Zigbee framework to identify the channel used by the target device
- An attacker can perform replay attack by **capturing** and **replaying the same packets** to observe the behavior of the device



Zigbee Network Discovery: 20 ☑ Verbose

```
[+] Transmitting beacon request.
[+] Received frame.
[+] Received frame is a beacon.
[+] Beacon represents a new network.
[+] New Network: PANID 0xA45B  Source 0x3178
        Ext PANID: 53:37:52:15:2d:99:b2:79
        Stack Profile: ZigBee Enterprise
        Stack Version: ZigBee 2006/2007
        Channel: 20

[+] Received frame.
[+] Received frame is not a beacon (FCF=6188)
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
[+] Received frame is not a beacon (FCF=6188).
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
```

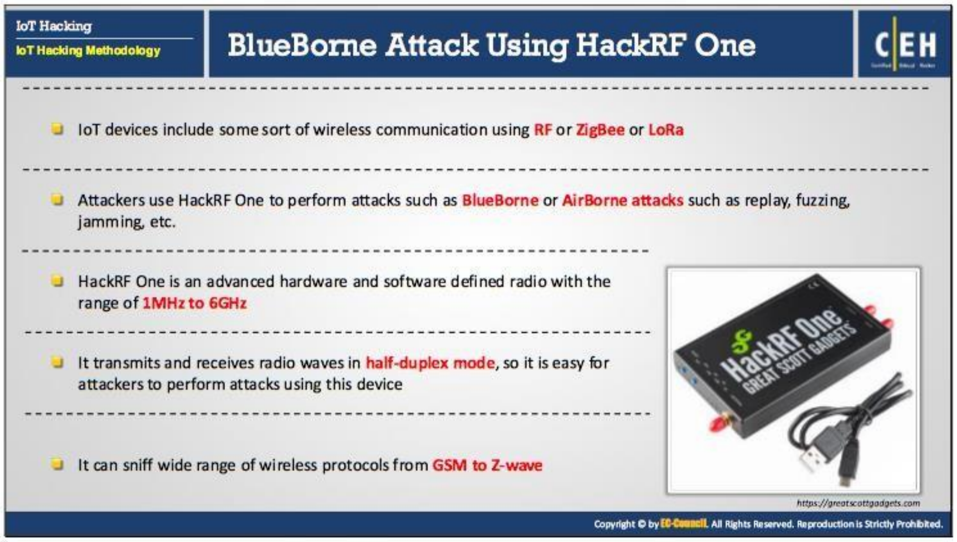
*https://www.attify.com*

### Hacking Zigbee Devices with Attify Zigbee Framework

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

Most of the IoT devices use ZigBee protocol for short range wireless communication. Attackers find vulnerabilities in ZigBee based IoT and smart devices and exploit them using tools such as Attify ZigBee Framework. Attackers take advantage of the vulnerabilities in these devices to sniff confidential information in transit and in some cases, take control of the device itself.

Attify ZigBee framework consists of a set of tools used to perform ZigBee penetration testing. ZigBee protocol makes use of 16 different channels for all communications. Attackers use Zbstumbler from Attify Zigbee framework to identify the channel used by the target device. Once the attacker identifies the channel of the target device, he/she starts capturing the packets that are being transmitted from or/and to the device. At this stage, an attacker can simply perform replay attack by capturing and replaying the same packets to observe the behavior of the device. After observing the behavior of the device, the attacker can perform further exploitation on the device.



IoT Hacking
IoT Hacking Methodology

## BlueBorne Attack Using HackRF One

- IoT devices include some sort of wireless communication using **RF** or **ZigBee** or **LoRa**
- Attackers use HackRF One to perform attacks such as **BlueBorne** or **AirBorne attacks** such as replay, fuzzing, jamming, etc.
- HackRF One is an advanced hardware and software defined radio with the range of **1MHz to 6GHz**
- It transmits and receives radio waves in **half-duplex mode**, so it is easy for attackers to perform attacks using this device
- It can sniff wide range of wireless protocols from **GSM to Z-wave**

https://greatscottgadgets.com

### BlueBorne Attack using HackRF One

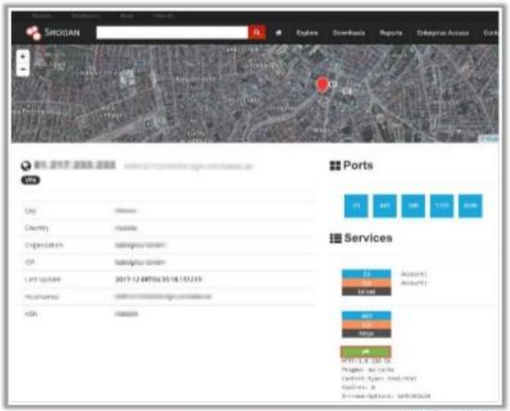
Source: *https://greatscottgadgets.com*

IoT devices include wireless communication using RF or ZigBee or LoRa. Attackers use HackRF One to perform attacks such as BlueBorne or AirBorne attacks such as replay, fuzzing, jamming, etc. HackRF One is an advanced hardware and software defined radio with the range of 1MHz to 6GHz. It transmits and receives radio waves in half-duplex mode, so it is easy for attackers to perform attacks using this device. It can sniff a wide range of wireless protocols from GSM to Z-wave.

IoT Hacking
**IoT Hacking Methodology**

## Gaining Remote Access using Telnet

- Attackers perform **port scanning** to learn about **open ports** and services on the target IoT device
- Many embedded system applications in IoT devices such as industrial control system, routers, VoIP phones, televisions, etc. implement remote access capabilities using Telnet
- If an attacker identifies that the **Telnet port is open**, he/she exploits this vulnerability to **gain remote access** to the device
- Attackers use tools such as **Shodan** and **Censys** to gain remote access to the target device



https://www.shodan.io

## Gain Remote Access

Vulnerabilities identified in the vulnerability scanning phase allow an attacker to remotely gain access, command and control the attack while evading detection from various security products. Based on the vulnerabilities in an IoT device, the attacker may turn the device into a backdoor to gain access to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc. After gaining remote access, attackers use these devices as a platform to launch attacks on other devices in the network.

### Gaining Remote Access using Telnet

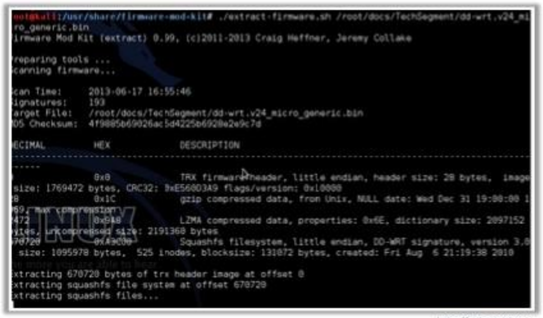
Attackers perform port scanning to learn about open ports and services on the target IoT device. If an attacker identifies that the Telnet port is open, he/she exploits this vulnerability to gain remote access to the device. Many embedded system applications in IoT devices such as industrial control systems, routers, VoIP phones, televisions, etc. implement remote access capabilities using Telnet. These applications include a Telnet server for remote access.

Once attacker identifies an open telnet port, he/she can learn what information is shared between the connected devices, including their software and hardware models. Then the attacker performs further attacks by exploiting their specific vulnerabilities. First, the attacker identifies whether authentication is required or not. If not required, he/she directly obtains unauthorized access to explore the data stored in the device. If authentication is required, then the attacker tries all the default credentials such as root/root, system/system, etc. or performs brute force attack to obtain passwords for the administrator or common user accounts. For example, an attacker can use tools such as Shodan, Censys, etc. to gain remote access to the target device.

IoT Hacking
IoT Hacking Methodology

## Maintain Access by Exploiting Firmware

- Attackers **exploit the firmware** installed on the IoT device to **maintain access** on the device
- After gaining remote access, attackers explore the file system to **access the firmware** on the device
- Attackers use tools such as **Firmware Mod Kit** to reconstruct the malicious firmware from the legitimate firmware
- The Firmware Mod Kit allows for easy **deconstruction** and **reconstruction** of firmware images for various embedded devices



```
oot@kali:/usr/share/firmware-mod-kit# ./extract-firmware.sh /root/docs/TechSegment/dd-wrt.v24_mi
ro_generic.bin
irmware Mod Kit (extract) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake

reparing tools ...
canning firmware...

can Time:     2013-06-17 16:55:46
ignatures:    193
arget File:   /root/docs/TechSegment/dd-wrt.v24_micro_generic.bin
D5 Checksum:  4f9885b69026ac5d4225b6920e2e9c7d

ECIMAL        HEX             DESCRIPTION
-------------------------------------------------------------------------------------------
-----
              0x0             TRX firmware header, little endian, header size: 28 bytes,  image
size: 1769472 bytes, CRC32: 0xE568D3A9 flags/version: 0x10000
8             0x1C            gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00 1
69, max compression
472           0x948           LZMA compressed data, properties: 0x6E, dictionary size: 2097152
ytes, uncompressed size: 2191368 bytes
70720         0xA3C00         Squashfs filesystem, little endian, DD-WRT signature, version 3.0
 size: 1095978 bytes,  525 inodes, blocksize: 131072 bytes, created: Fri Aug  6 21:19:38 2010

xtracting 670720 bytes of trx header image at offset 0
xtracting squashfs file system at offset 670720
xtracting squashfs files...
```

*https://code.google.com*

### Maintain Access

Once the attacker gains access to the device, the attacker uses various techniques to maintain access and perform further exploitation. Attackers remain undetected by clearing the logs, updating firmware and using malicious programs such as backdoor, Trojans, etc. to maintain access. Attackers use tools such as Firmware Mod Kit, Firmalyzer Enterprise, Firmware Analysis Toolkit, etc. to exploit firmware.

**Maintain Access by Exploiting Firmware**

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

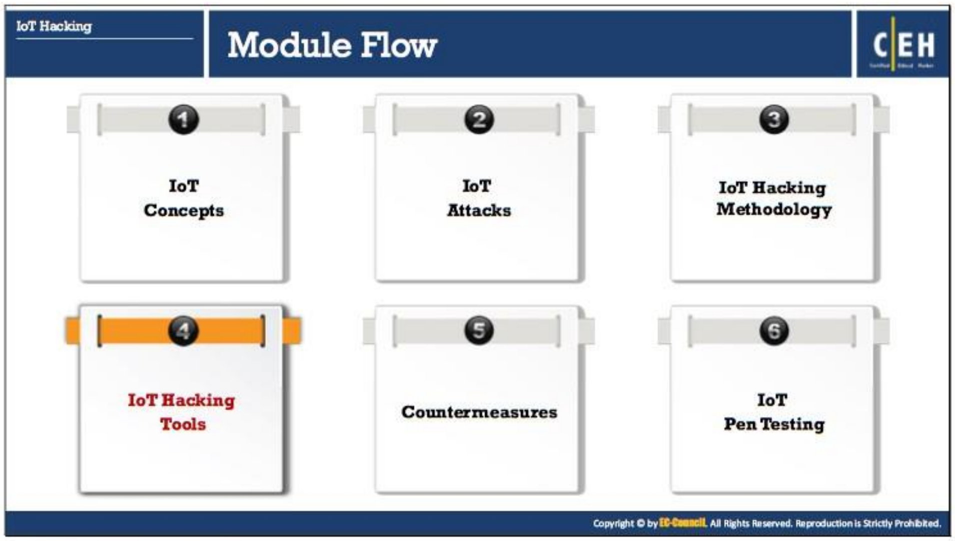
The Firmware Mod Kit allows for easy deconstruction and reconstruction of firmware images for various embedded devices. While it primarily targets Linux based routers, it is compatible with most firmware that makes use of common firmware formats and file systems such as TRX/uImage and SquashFS/CramFS.

The Firmware Mod Kit is a collection of tools, utilities, and shell scripts. The utilities can be used directly, or the shell scripts can be used to automate and combine common firmware operations (e.g. extract and rebuild). The Firmware Mod Kit does the following:

- Extract a firmware image into its component parts
- User makes desired modification to the firmware's file system or web UI (webif)
- Rebuild firmware
- Flash modified firmware onto device and brick it

The core scripts to facilitate firmware operations are listed below.

| Primary Scripts | Secondary Scripts |
|---|---|
| extract-firmware.sh | ddwrt-gui-extract.sh |
| Firmware extraction script | Extracts Web GUI files from extracted DD-WRT firmware |
| build-firmware.sh | ddwrt-gui-rebuild.sh |
| Firmware re-building script | Restores modified Web GUI files to extracted DD-WRT firmware |



IoT Hacking

## Module Flow

1. IoT Concepts
2. IoT Attacks
3. IoT Hacking Methodology
4. IoT Hacking Tools
5. Countermeasures
6. IoT Pen Testing

## IoT Hacking Tools

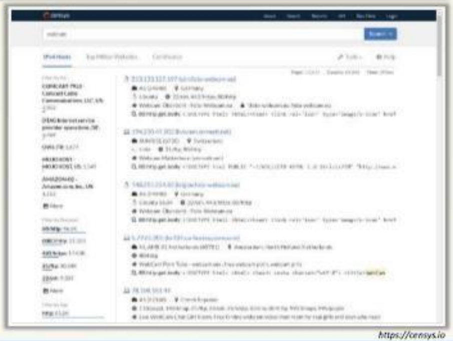
Attackers use IoT hacking tools to gather information about the devices connected to the network, their open ports and services, identify attack surface area and associated vulnerabilities to perform further exploitation on the device and the organization's network. This section deals with various IoT hacking tools.

IoT Hacking
IoT Hacking Tools

## Information Gathering Tools

**Censys**

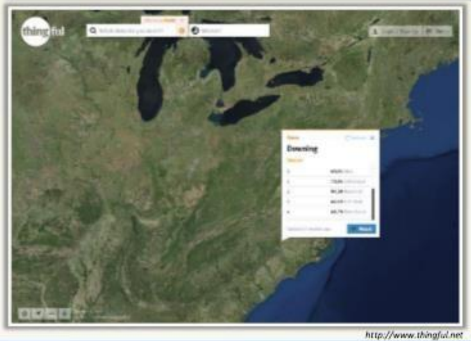
Censys allows an attacker to **continually monitor** every **reachable server** and device on the Internet



https://censys.io

**Thingful**

Thingful is a search engine for the Internet of Things to find and **use open IoT data** from around the world



http://www.thingful.net

### Information Gathering Tools

Attackers use information gathering tools such as Shodan and Censys to gather basic information about the target device and network. Using these tools attackers obtain information such as live devices connected to the network, their make, open ports and services, their physical location, etc.

- **Censys**

  Source: *https://censys.io*

  Censys is a public search engine and data processing facility backed by data collected from ongoing Internet-wide scans. Censys supports full-text searches on protocol banners and queries a wide range of derived fields. It can identify specific vulnerable devices and networks and generate statistical reports on broad usage patterns and trends. Censys continually monitors every reachable server and device on the Internet, and analyze them in real time. Censys allows a pen-tester to understand your network attack surface and discover new threats and assess their global impact. Censys collects data on hosts and websites through daily ZMap and ZGrab scans of the IPv4 address space, in turn maintaining a database of how hosts and websites are configured.

- **Thingful**

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

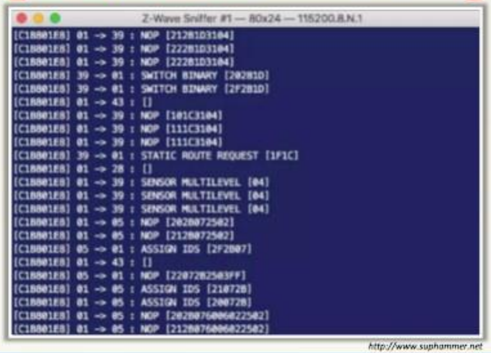
  Thingful is a search engine for the Internet of Things to find and use open IoT data from around the world. It helps organizations make better decisions with external IoT data. It collects real-time IoT data across dozens of verticals, including weather, environment, smart city, energy, and transport. Thingful's data pipes make it quick and easy to find and use the IoT data.

IoT Hacking
IoT Hacking Tools

## Sniffing Tools

**Z-Wave Sniffer**

Z-Wave sniffer is used to **sniff the traffic**, perform **real-time monitoring** and **capture packets** from all Z-Wave networks



```
[C18801E8] 01 -> 39 : NOP [21281D3104]
[C18801E8] 01 -> 39 : NOP [22281D3104]
[C18801E8] 01 -> 39 : NOP [22281D3104]
[C18801E8] 39 -> 01 : SWITCH BINARY [20281D]
[C18801E8] 39 -> 01 : SWITCH BINARY [2F281D]
[C18801E8] 01 -> 43 : []
[C18801E8] 01 -> 39 : NOP [101C3104]
[C18801E8] 01 -> 39 : NOP [111C3104]
[C18801E8] 01 -> 39 : NOP [111C3104]
[C18801E8] 39 -> 01 : STATIC ROUTE REQUEST [1F1C]
[C18801E8] 01 -> 28 : []
[C18801E8] 01 -> 39 : SENSOR MULTILEVEL [04]
[C18801E8] 01 -> 39 : SENSOR MULTILEVEL [04]
[C18801E8] 01 -> 39 : SENSOR MULTILEVEL [04]
[C18801E8] 01 -> 05 : NOP [2028072502]
[C18801E8] 01 -> 05 : NOP [2128072502]
[C18801E8] 05 -> 01 : ASSIGN IDS [2F2007]
[C18801E8] 01 -> 43 : []
[C18801E8] 05 -> 01 : NOP [22072B2503FF]
[C18801E8] 01 -> 05 : ASSIGN IDS [21072B]
[C18801E8] 01 -> 05 : ASSIGN IDS [20072B]
[C18801E8] 01 -> 05 : NOP [2028076006022502]
[C18801E8] 01 -> 05 : NOP [2128076006022502]
```

*http://www.suphammer.net*

**CloudShark**
*https://www.cloudshark.org*

**Ubiqua Protocol Analyzer**
*https://www.ubilogix.com*

**Perytons Protocol Analyzers**
*http://www.perytons.com*

**Wireshark**
*https://www.wireshark.org*

**Tcpdump**
*http://www.tcpdump.org*

### Sniffing Tools

System administrators use automated tools to monitor their network and devices connected to the network, but attackers misuse these tools to sniff network data. Listed below are some of the tools that an attacker can use for sniffing traffic generated by IoT devices.

- **Z-Wave Sniffer**

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

  Z-Wave sniffer is hardware tool used to sniff traffic generated by smart devices connected in the network.

  **Features**:

  - Perform real-time monitoring
  - Captures packets from all Z-Wave networks
  - Provides upgradable firmware
  - Supports Windows, MAC OS and Linux
  - Works with all Z-Wave controllers (Including Fibaro, Homeseer, Tridium Niagara, Z-Way, SmartThings, Vera or any other Z-Wave controller)

**Listed below are some of the additional tools used to sniff traffic generated by IoT devices:**

- CloudShark (*https://www.cloudshark.org*)
- Ubiqua Protocol Analyzer (*https://www.ubilogix.com*)
- Perytons Protocol Analyzers (*http://www.perytons.com*)
- Wireshark (*https://www.wireshark.org*)

- Tcpdump (*http://www.tcpdump.org*)
- Open Sniffer (*https://www.sewio.net*)
- APIMOTE IEEE 802.15.4/ZIGBEE SNIFFING HARDWARE (*http://www.riverloopsecurity.com*)
- Ubertooth (*https://github.com*)

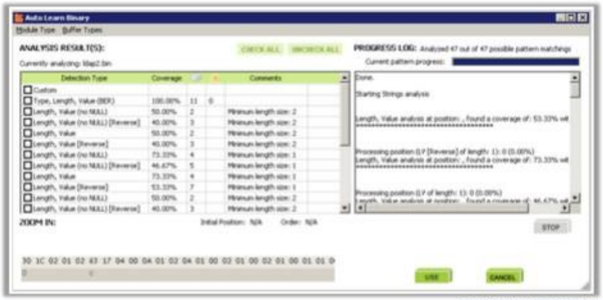
IoT Hacking
**IoT Hacking Tools**

## Vulnerability Scanning Tools

CEH

**beSTORM**

beSTORM is a **smart fuzzer** to find **buffer overflow vulnerabilities** by automating and documenting the process of delivering corrupted input and watching for an unexpected response from the application



ANALYSIS RESULT(S):

Currently analyzing: ldap2.bin

| Detection Type | Coverage | | | Comments |
|---|---|---|---|---|
| □ Custom | | | | |
| □ Type, Length, Value (BER) | 100.00% | 11 | 0 | |
| □ Length, Value (no NULL) | 50.00% | 2 | | Minimum length size: 2 |
| □ Length, Value (no NULL) [Reverse] | 40.00% | 3 | | Minimum length size: 2 |
| □ Length, Value | 50.00% | 2 | | Minimum length size: 2 |
| □ Length, Value [Reverse] | 40.00% | 3 | | Minimum length size: 2 |
| □ Length, Value (no NULL) | 73.33% | 4 | | Minimum length size: 1 |
| □ Length, Value (no NULL) [Reverse] | 46.67% | 5 | | Minimum length size: 1 |
| □ Length, Value | 73.33% | 4 | | Minimum length size: 1 |
| □ Length, Value [Reverse] | 53.33% | 7 | | Minimum length size: 1 |
| □ Length, Value (no NULL) | 50.00% | 2 | | Minimum length size: 2 |
| □ Length, Value (no NULL) [Reverse] | 40.00% | 3 | | Minimum length size: 2 |

ZOOM IN:

Initial Position: N/A Order: N/A

```
30 1C 02 01 02 43 17 04 00 0A 01 02 04 01 00 02 01 00 02 01 00 01 01 0
0 8
```

PROGRESS LOG: Analyzed 47 out of 47 possible pattern matchings

Current pattern progress:

```
Done.

Starting Strings analysis

Length, Value analysis at position: , found a coverage of: 53.33% wit
**********************************


Processing position (LV [Reverse] of length: 1): 0 (0.00%)
Length, Value analysis at position: , found a coverage of: 73.33% wit
**********************************


Processing position (LV of length: 1): 0 (0.00%)
Length, Value analysis at position: , found a coverage of: 46.67% wit
```

*https://www.beyondsecurity.com*

**Rapid7 Metaspoilt PRO**
*https://www.rapid7.com*

**IoTsploit**
*https://iotsploit.co*

**IoTSeeker**
*https://information.rapid7.com*

**Bitdefender Home Scanner**
*https://play.google.com*

**IoTInspector**
*https://www.incapsula.com*

### Vulnerability Scanning Tools

Vulnerability scanning allows an attacker to identify vulnerabilities in IoT devices and their network and to further determine how they can be exploited. These tools assist network security professionals in overcoming the identified weaknesses in the device and network by suggesting various remediation techniques to protect the organization's network.

- **beSTORM**

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

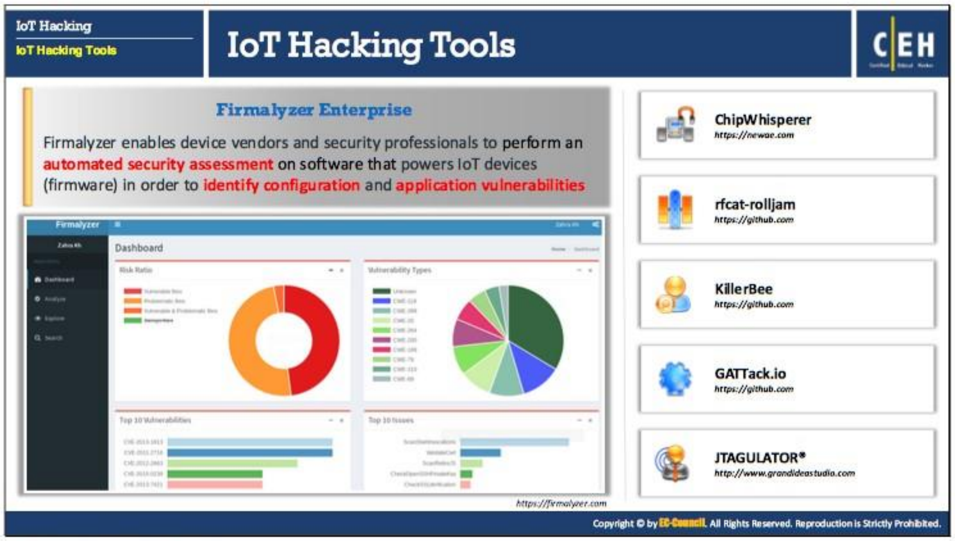
  beSTORM is a smart fuzzer to find buffer overflow vulnerabilities by automating and documenting the process of delivering corrupted input and watching for unexpected response from the application. It supports multi-protocol environment and address breaches by testing over 50 protocols while providing automated binary and textual analysis, advanced debugging and stack tracing.

  By applying automated protocol based fuzzing techniques, beSTORM is an automated black-box auditing tool. It tries virtually every attack combination intelligently, starting with the most likely scenarios and detects application anomalies, which indicate a successful attack. This way security holes can be found in the application far faster, without brute force testing and almost without any user intervention. beSTORM is equipped with the ability to use multiple processors or multiple machines to parallelize the audit and substantially reduce the testing duration.

**Listed below are some of the additional vulnerability scanners for IoT devices:**

- Rapid7 Metaspoilt PRO (*https://www.rapid7.com*)
- IoTsploit (*https://iotsploit.co*)

- IoTSeeker (*https://information.rapid7.com*)
- Bitdefender Home Scanner (*https://www.bitdefender.com*)
- IoTInspector (*http://www.iot-inspector.com*)



IoT Hacking Tools

**Firmalyzer Enterprise**

Firmalyzer enables device vendors and security professionals to perform an **automated security assessment** on software that powers IoT devices (firmware) in order to **identify configuration** and **application vulnerabilities**

*https://firmalyzer.com*

ChipWhisperer
*https://newae.com*

rfcat-rolljam
*https://github.com*

KillerBee
*https://github.com*

GATTack.io
*https://github.com*

JTAGULATOR®
*http://www.grandideastudio.com*

## IoT Hacking Tools

Listed below are some of the IoT hacking tools used by attackers to exploit target IoT devices and network to perform various attacks such as DDoS, jamming, BlueBorne, etc.

- **Firmalyzer Enterprise**

  Source: *https://firmalyzer.com*

  Firmalyzer enables device vendors and security professionals to perform automated security assessment on software that powers IoT devices (firmware) in order to identify configuration and application vulnerabilities. This tool notifies users about the vulnerabilities discovered and assists to mitigate those in a timely manner.

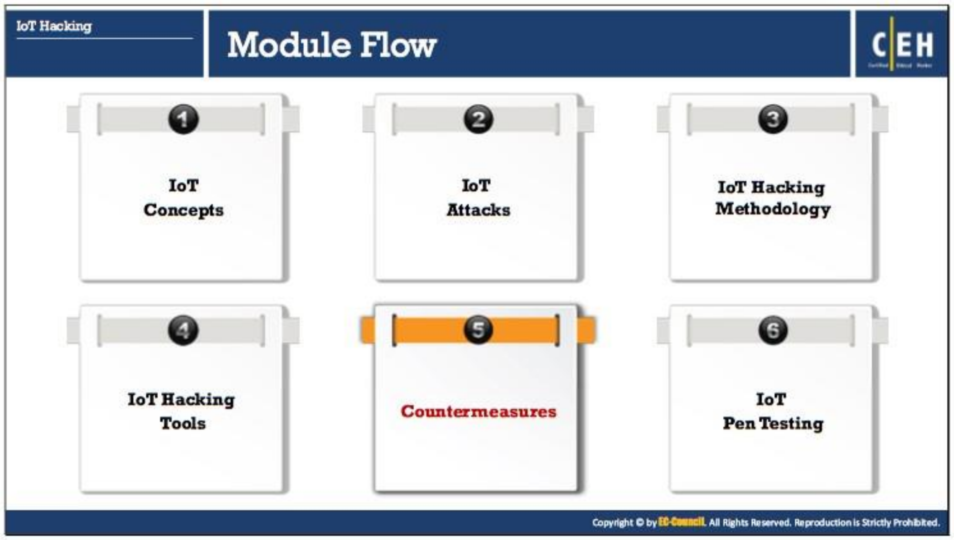
  **Features**:

  - Explore files inside firmware and view their details and content
  - Find configuration issues and ways to fix them
  - Discover vulnerable apps in firmware (PHP, Java, JavaScript)
  - Can identify a large number of components and their versions used in IoT devices including 3rd party applications and libraries
  - Shows known vulnerabilities of the detected components including complete details
  - Can identify cryptographic issues such as problematic certificates or hard-coded private keys
  - Shows complete details of each file inside the firmware

  - Assists compliance and due diligence activities by searching copyright notes in each file content inside a firmware
  - Provides the ability to define advanced queries based on different factors and performs an advanced search

**Listed below are some of the additional tools to perform IoT hacking:**

- ChipWhisperer (*https://newae.com*)
- rfcat-rolljam (*https://github.com*)
- KillerBee (*https://github.com*)
- GATTack.io (*https://github.com*)
- JTAGULATOR® (*http://www.grandideastudio.com*)
- Firmware Analysis Toolkit (*https://github.com*)



IoT Hacking

## Module Flow

1 IoT Concepts

2 IoT Attacks

3 IoT Hacking Methodology

4 IoT Hacking Tools

5 Countermeasures

6 IoT Pen Testing

## Countermeasures

This section discusses various IoT security measures and IoT security tools to be used to prevent, protect and recover from various types of attacks on IoT devices and its networks. Following these counter-measures organizations can implement proper security mechanisms to protect the confidential information transmitted between the devices and corporate network.

IoT Hacking
Countermeasures

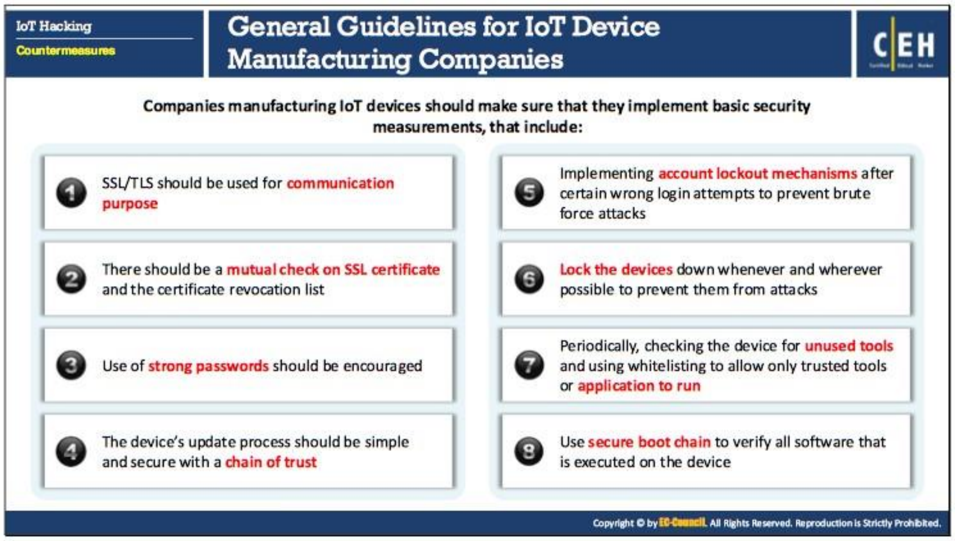
## How to Defend Against IoT Hacking

1. Disable the "guest" and "demo" user accounts if enabled
2. Use the "Lock Out" feature to lock out accounts for excessive invalid login attempts
3. Implement strong authentication mechanisms
4. Locate control system networks and devices behind firewalls, and isolate them from the business network
5. Implement IPS and IDS in the network
6. Implement end-to-end encryption and use Public Key Infrastructure (PKI)
7. Use VPN architecture for secure communication
8. Deploy security as a unified, integrated system
9. Allow only trusted IP addresses to access device from the Internet
10. Disable telnet (port 23)
11. Disable UPnP port on routers
12. Prevent the devices against physical tampering
13. Patch vulnerabilities and update the device firmware regularly
14. Monitor traffic on port 48101 as the infected devices attempt to spread the malicious file using port 48101

### How to Defend Against IoT Hacking

- Disable the "guest" and "demo" user accounts if enabled
- Use the "Lock Out" feature to lock out accounts for excessive invalid login attempts
- Implement strong authentication mechanism
- Locate control system networks and devices behind firewalls, and isolate them from the business network
- Implement IPS and IDS in the network
- Implement end-to-end encryption and use Public Key Infrastructure (PKI)
- Use VPN architecture for secure communication
- Deploy security as a unified, integrated system
- Allow only trusted IP addresses to access the device from the Internet
- Disable telnet (port 23)
- Disable UPnP port on routers
- Prevent the devices against physical tampering
- Patch vulnerabilities and update the device firmware regularly
- Monitor traffic on port 48101 as the infected devices attempt to spread the malicious file using port 48101.

- Position of mobile nodes should be verified with an aim to referring one physical node with one vehicle identity only, that means one vehicle cannot have two or more than two identities
- Data privacy should be implemented. Therefore, the user's account or the identity should be kept protected and hidden from other users
- Data authentication should be done to confirm the identity of the original source node
- Maintain the data confidentiality using symmetric key encryption



IoT Hacking
Countermeasures

**General Guidelines for IoT Device Manufacturing Companies**

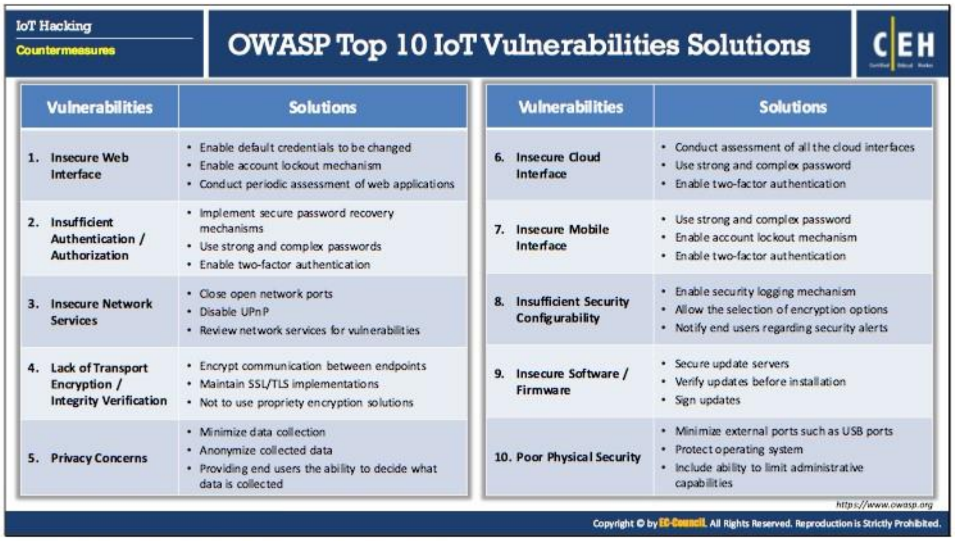
**Companies manufacturing IoT devices should make sure that they implement basic security measurements, that include:**

1. SSL/TLS should be used for **communication purpose**
2. There should be a **mutual check on SSL certificate** and the certificate revocation list
3. Use of **strong passwords** should be encouraged
4. The device's update process should be simple and secure with a **chain of trust**
5. Implementing **account lockout mechanisms** after certain wrong login attempts to prevent brute force attacks
6. **Lock the devices** down whenever and wherever possible to prevent them from attacks
7. Periodically, checking the device for **unused tools** and using whitelisting to allow only trusted tools or **application to run**
8. Use **secure boot chain** to verify all software that is executed on the device

## General Guidelines for IoT Device Manufacturing Companies

Companies manufacturing IoT devices should make sure that they implement basic security measurements that include:

- SSL/TLS should be used for communication purpose
- There should be a mutual check on SSL certificates and the certificate revocation list
- Use of strong passwords should be encouraged
- The device's update process should be simple, secured with a chain of trust
- Implementing account lockout mechanisms after certain wrong login attempts to prevent brute force attacks
- Lock the devices down whenever and wherever possible to prevent them from attacks
- Periodically checking the device for unused tools and using whitelisting to allow only trusted tools or application to run
- Use secure boot chain to verify all software that is executed on the device



IoT Hacking
Countermeasures

OWASP Top 10 IoT Vulnerabilities Solutions

| Vulnerabilities | Solutions |
|---|---|
| 1. Insecure Web Interface | • Enable default credentials to be changed<br>• Enable account lockout mechanism<br>• Conduct periodic assessment of web applications |
| 2. Insufficient Authentication / Authorization | • Implement secure password recovery mechanisms<br>• Use strong and complex passwords<br>• Enable two-factor authentication |
| 3. Insecure Network Services | • Close open network ports<br>• Disable UPnP<br>• Review network services for vulnerabilities |
| 4. Lack of Transport Encryption / Integrity Verification | • Encrypt communication between endpoints<br>• Maintain SSL/TLS implementations<br>• Not to use propriety encryption solutions |
| 5. Privacy Concerns | • Minimize data collection<br>• Anonymize collected data<br>• Providing end users the ability to decide what data is collected |
| 6. Insecure Cloud Interface | • Conduct assessment of all the cloud interfaces<br>• Use strong and complex password<br>• Enable two-factor authentication |
| 7. Insecure Mobile Interface | • Use strong and complex password<br>• Enable account lockout mechanism<br>• Enable two-factor authentication |
| 8. Insufficient Security Configurability | • Enable security logging mechanism<br>• Allow the selection of encryption options<br>• Notify end users regarding security alerts |
| 9. Insecure Software / Firmware | • Secure update servers<br>• Verify updates before installation<br>• Sign updates |
| 10. Poor Physical Security | • Minimize external ports such as USB ports<br>• Protect operating system<br>• Include ability to limit administrative capabilities |

*https://www.owasp.org*

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

## OWASP Top 10 IoT Vulnerabilities Solutions

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

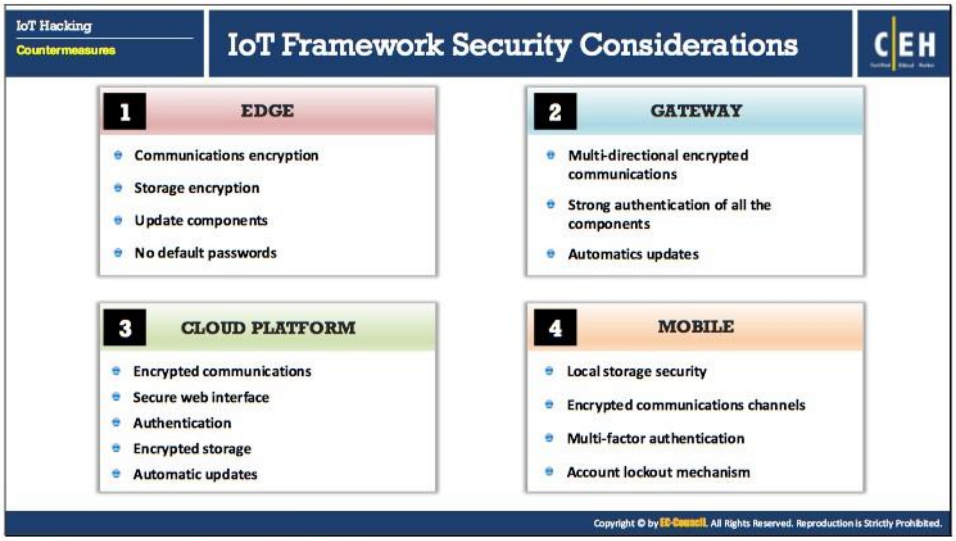
IoT technology has been developed rapidly without undertaking appropriate consideration for the security of the devices. Due to the security vulnerabilities present in the IoT devices, risks related to potential cyberattacks, stealing of confidential information, privacy invasion, etc. are increasing rapidly. It is necessary for the developers or the security professionals to test the devices for various vulnerabilities, before integrating the IoT system and products into an infrastructure.

The OWASP top 10 security vulnerabilities and solutions associated with each vulnerability are given below:

| Vulnerabilities | Solutions |
|---|---|
| **1. Insecure Web Interface** | ▪ Enable default credentials to be changed<br>▪ Enable account lockout mechanism<br>▪ Conduct periodic assessment of web applications |
| **2. Insufficient Authentication / Authorization** | ▪ Implement secure password recovery mechanisms<br>▪ Use strong and complex passwords<br>▪ Enable two-factor authentication |
| **3. Insecure Network Services** | ▪ Close open network ports<br>▪ Disable UPnP<br>▪ Review network services for vulnerabilities |

| | |
|---|---|
| **4. Lack of Transport Encryption / Integrity Verification** | ▪ Encrypt communication between endpoints<br>▪ Maintain SSL/TLS implementations<br>▪ Not to use propriety encryption solutions |
| **5. Privacy Concerns** | ▪ Minimize data collection<br>▪ Anonymize collected data<br>▪ Providing end users the ability to decide what data is collected |
| **6. Insecure Cloud Interface** | ▪ Conduct assessment of all the cloud interfaces<br>▪ Use strong and complex password<br>▪ Enable two-factor authentication |
| **7. Insecure Mobile Interface** | ▪ Use strong and complex password<br>▪ Enable account lockout mechanism<br>▪ Enable two-factor authentication |
| **8. Insufficient Security Configurability** | ▪ Enable security logging mechanism<br>▪ Allow the selection of encryption options<br>▪ Notify end users regarding security alerts |
| **9. Insecure Software / Firmware** | ▪ Secure update servers<br>▪ Verify updates before installation<br>▪ Sign updates |
| **10. Poor Physical Security** | ▪ Minimize external ports such as USB ports<br>▪ Protect operating system<br>▪ Include ability to limit administrative capabilities |

TABLE 18.4: OWASP top 10 IoT vulnerabilities and solutions



IoT Hacking
Countermeasures

IoT Framework Security Considerations

**1 EDGE**

- Communications encryption
- Storage encryption
- Update components
- No default passwords

**2 GATEWAY**

- Multi-directional encrypted communications
- Strong authentication of all the components
- Automatics updates

**3 CLOUD PLATFORM**

- Encrypted communications
- Secure web interface
- Authentication
- Encrypted storage
- Automatic updates

**4 MOBILE**

- Local storage security
- Encrypted communications channels
- Multi-factor authentication
- Account lockout mechanism

## IoT Framework Security Considerations

For designing secure and protected IoT devices, security issues should be properly considered. One of the most important consideration is the development of a secure IoT framework for building the device. Ideally a framework should be designed in such a way that it provides a default security so that the developers do not have to think about it later.

Security evaluation criteria for the IoT framework is broken down into four parts. Each part has its own security related concerns that are discussed in the evaluation criteria for each part. The security evaluation criteria for the IoT devices are discussed below:

- **Edge**

  The edge is the main physical device in the IoT ecosystem that interacts with its surroundings and contains various components like sensors, actuators, operating systems, hardware and network and communication capabilities. It is heterogeneous and can be deployed anywhere and in any condition. Therefore, an ideal framework for edge would be such that it provides cross platform components so that it can be deployed and work in any physical condition possible.

  Other framework consideration for edge would be proper communications and storage encryption, no default credentials, strong passwords, use latest up to date components and so on.

- **Gateway**

  The gateway acts as a first step for an edge into the world of Internet as it connects the smart devices to the cloud components. It is referred to as communication aggregator that allows communication with a secure and trusted local network as well as a secure connection with an untrusted public network. It also provides a layer of security to all the devices connected to it. As it serves as an aggregation point for an edge, the gateway has a very important security role in the ecosystem.

An ideal framework for the gateway should incorporate strong encryption techniques for secure communications between endpoints. Also, the authentication mechanism for the edge components should be as strong as any other component in the framework. Where ever possible the gateway should be designed in such a way that it authenticates multi-directionally to carry out trusted communication between the edge and the cloud. Automatic updates should also be provided to the device for countering vulnerabilities.

- **Cloud Platform**

  In an IoT ecosystem, the cloud component is referred to as the main central aggregation and data management point. Access to the cloud is restricted. The cloud component is usually at higher risk as it the central point of data aggregation for most of the data in the ecosystem. It also includes command and control (C2) component which is a centralized computer that issues various commands for the distribution of extensions and updates.

  A secure framework for the cloud component should include encrypted communications, strong authentication credentials, secure web interface, encrypted storage, automatic updates and so on.

- **Mobile**

  In an IoT ecosystem, the mobile interface plays an important part particularly where the data needs to be collected and managed. Using mobile interfaces, users can access and interact with the edge in their home or workplace from miles away. Some mobile applications provide the users only limited data from specific edge devices while others allow the complete manipulation of the edge components. Proper attention should be given to the mobile interface as they are prone to various cyber-attacks.

  An ideal framework for the mobile interface should include proper authentication mechanism for the user, account lockout mechanism after a certain number of failed attempts, local storage security, encrypted communication channels and the security of the data transmitted over the channel.

IoT Hacking
Countermeasures

## IoT Security Tools

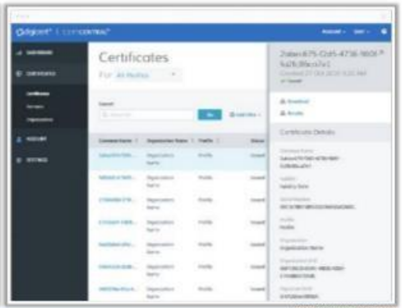
### SeaCat.io

SeaCat.io is a **security-first SaaS technology** to operate IoT products in a reliable, scalable and secure manner

### DigiCert IoT Security Solution

DigiCert IoT Security Solutions **protect private data** and home networks while preventing unauthorized access using **PKI-based security solutions** for IoT devices

https://www.teskalabs.com



https://www.digicert.com

IoT Hacking
Countermeasures

## IoT Security Tools (Cont'd)

Pulse: IoT Security Platform
https://www.pwnieexpress.com

Symantec IoT Security
https://www.symantec.com

darktarce
https://www.darktrace.com

Cisco IoT Threat Defense
https://www.cisco.com

Cisco Umbrella
https://umbrella.cisco.com

Google Cloud IoT
https://cloud.google.com

net-Shield
https://github.com

Noddos
https://www.noddos.io

AWS IoT Device Defender
https://aws.amazon.com

Bayshore Industrial Cyber Protection Platform
https://www.bayshorenetworks.com

Trustwave Endpoint Protection Suite
https://www.trustwave.com

NSFOCUS ADS
https://nsfocusglobal.com

Norton Core
https://us.norton.com

zvelo IoT Security Solution
https://zvelo.com

Carwall
https://karambasecurity.com

### IoT Security Tools

The Internet of Things is not only the devices connected to the Internet but also it is very complex, rapidly growing technology. To understand and analyze various risk factors, proper security solutions must be incorporated to protect the IoT devices. Use of IoT security tools helps organizations to vastly limit security vulnerabilities, thereby protecting the IoT devices and networks from different kinds of attacks.

- **SeaCat.io**

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

  SeaCat.io is a security-first SaaS technology to operate IoT products in a reliable, scalable and secure manner. It provides protection to end users, business, and data.

  **Features**:

  - Manage connected products from a central place
  - Access your remote devices using various tools
  - Monitor connected devices and automate updates to fix bugs
  - Protect users with an authorized cryptography and comply with regulations
  - Complementary SSL VPN for all your devices
  - Hide APIs from a direct Internet exposure and know who is using them and how
  - Ensure devices are malware-free and prevent hackers from controlling and making them part of a botnet
  - Automated alerting service helps in reacting to any incident proactively and addressing possible intrusion attempts or downtimes
  - Integrate with existing SIEM, identity management, log collectors, intrusion detection systems, etc.
  - Supports protocols such as HTTP, MQTT, SSH, VNC, TCP/IP, Syslog, LDAP, CEF, etc.
  - Store, view, analyze, search and tail log events in realtime

- **DigiCert IoT Security Solution**

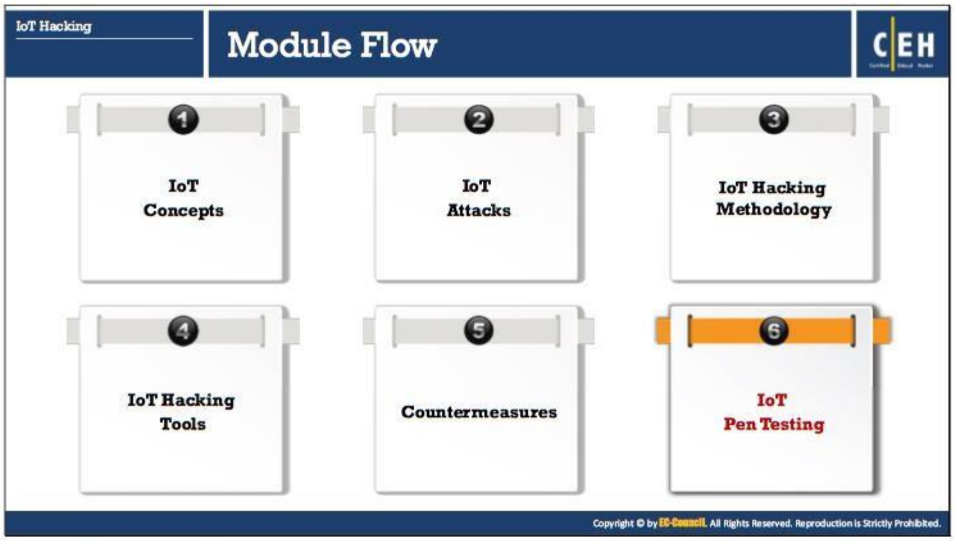
  Source: *https://www.digicert.com*

  DigiCert Home and Consumer IoT Security Solutions protect private data and home networks while preventing unauthorized access using PKI-based security solutions for consumer IoT devices. Home IoT products offer many conveniences but there are massive amounts of private consumer data being transferred to and from these services, leaving it vulnerable to attack if left unsecured. Security across an entire IoT home demands proper device authentication and data encryption to ensure that all connections are trusted and communications are protected. Properly implemented Public Key Infrastructure (PKI) creates a foundation for systems, devices, applications, and users to interact safely with consumer IoT products.

**Listed below are some of the additional IoT security tools and solutions:**

- Pulse: IoT Security Platform (*https://www.pwnieexpress.com*)
- Symantec IoT Security (*https://www.symantec.com*)
- darktarce (*https://www.darktrace.com*)
- Cisco IoT Threat Defense (*https://www.cisco.com*)

- Cisco Umbrella (*https://umbrella.cisco.com*)
- Google Cloud IoT (*https://cloud.google.com*)
- net-Shield (*https://github.com*)
- Noddos (*https://www.noddos.io*)
- AWS IoT Device Defender (*https://aws.amazon.com*)
- Bayshore Industrial Cyber Protection Platform (*https://www.bayshorenetworks.com*)
- Trustwave Endpoint Protection Suite (*https://www.trustwave.com*)
- NSFOCUS ADS (*https://nsfocusglobal.com*)
- Norton Core (*https://us.norton.com*)
- zvelo IoT Security Solution (*https://zvelo.com*)
- Carwall (*https://karambasecurity.com*)
- SecBee (*https://github.com*)
- libsecurity (*https://developer.ibm.com*)
- Bullguard IoT Scanner (*https://iotscanner.bullguard.com*)
- Mirai Vulnerability Scanner (*https://www.incapsula.com*)
- Kaspersky IoT Scanner (*https://play.google.com*)
- BlueBorne Vulnerability Scanner (*https://play.google.com*)

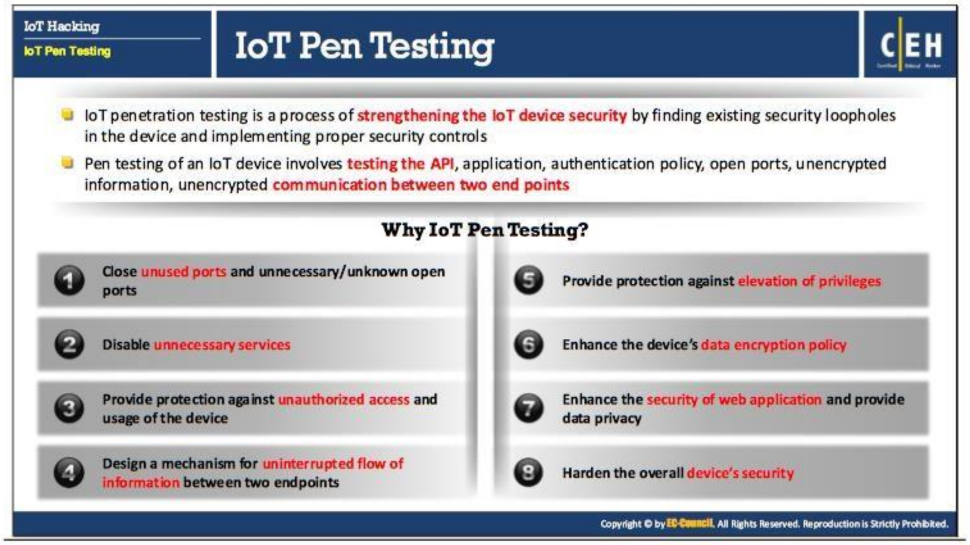


IoT Hacking

## Module Flow

CEH

1. **IoT Concepts**
2. **IoT Attacks**
3. **IoT Hacking Methodology**
4. **IoT Hacking Tools**
5. **Countermeasures**
6. **IoT Pen Testing**



IoT Hacking
IoT Pen Testing

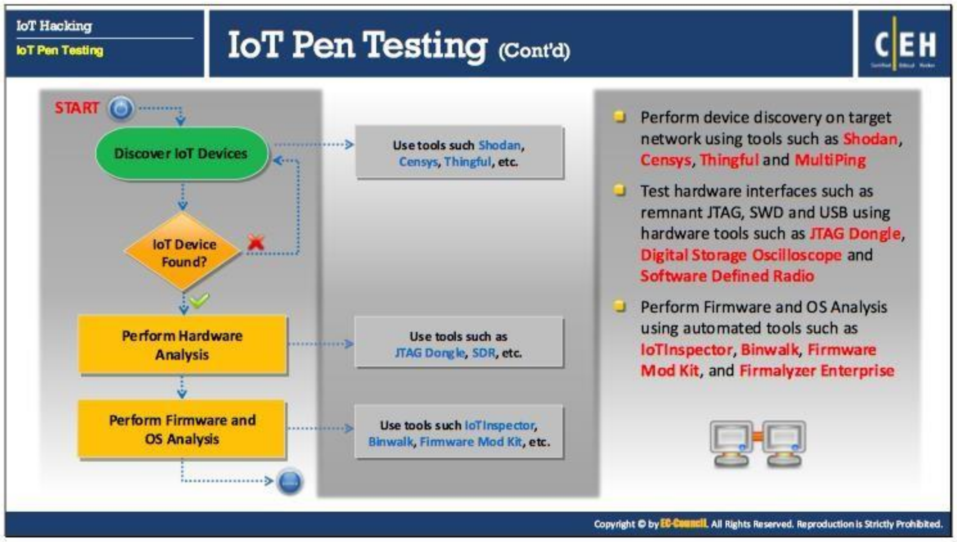
## IoT Pen Testing

CEH

- IoT penetration testing is a process of **strengthening the IoT device security** by finding existing security loopholes in the device and implementing proper security controls
- Pen testing of an IoT device involves **testing the API**, application, authentication policy, open ports, unencrypted information, unencrypted **communication between two end points**

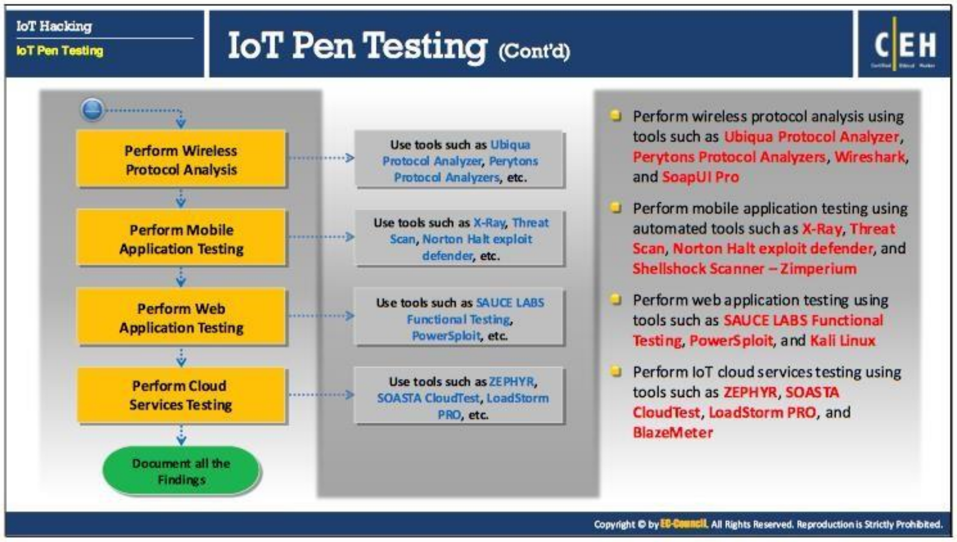
### Why IoT Pen Testing?

1. Close **unused ports** and unnecessary/unknown open ports
2. Disable **unnecessary services**
3. Provide protection against **unauthorized access** and usage of the device
4. Design a mechanism for **uninterrupted flow of information** between two endpoints
5. Provide protection against **elevation of privileges**
6. Enhance the device's **data encryption policy**
7. Enhance the **security of web application** and provide data privacy
8. Harden the overall **device's security**


IoT Hacking
IoT Pen Testing
CEH
START
Discover IoT Devices
IoT Device Found?
Perform Hardware Analysis
Perform Firmware and OS Analysis
Use tools such Shodan, Censys, Thingful, etc.
Use tools such as JTAG Dongle, SDR, etc.
Use tools such IoTInspector, Binwalk, Firmware Mod Kit, etc.
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

## IoT Pen Testing (Cont'd)

- Perform device discovery on target network using tools such as Shodan, Censys, Thingful and MultiPing
- Test hardware interfaces such as remnant JTAG, SWD and USB using hardware tools such as JTAG Dongle, Digital Storage Oscilloscope and Software Defined Radio
- Perform Firmware and OS Analysis using automated tools such as IoTInspector, Binwalk, Firmware Mod Kit, and Firmalyzer Enterprise


IoT Hacking
IoT Pen Testing
CEH
Perform Wireless Protocol Analysis
Perform Mobile Application Testing
Perform Web Application Testing
Perform Cloud Services Testing
Document all the Findings
Use tools such as Ubiqua Protocol Analyzer, Perytons Protocol Analyzers, etc.
Use tools such as X-Ray, Threat Scan, Norton Halt exploit defender, etc.
Use tools such as SAUCE LABS Functional Testing, PowerSploit, etc.
Use tools such as ZEPHYR, SOASTA CloudTest, LoadStorm PRO, etc.
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

## IoT Pen Testing (Cont'd)

- Perform wireless protocol analysis using tools such as Ubiqua Protocol Analyzer, Perytons Protocol Analyzers, Wireshark, and SoapUI Pro
- Perform mobile application testing using automated tools such as X-Ray, Threat Scan, Norton Halt exploit defender, and Shellshock Scanner – Zimperium
- Perform web application testing using tools such as SAUCE LABS Functional Testing, PowerSploit, and Kali Linux
- Perform IoT cloud services testing using tools such as ZEPHYR, SOASTA CloudTest, LoadStorm PRO, and BlazeMeter

## IoT Pen Testing

The emerging technology, Internet of Things (IoT) introduced a complex environment for both developers and security professionals. To secure IoT an in-depth understanding of the underlying complex environment, a thorough research of components, and assessment plan plays a major role. IoT pen testing is complex because of different architectures, operating systems, and communication protocols used. IoT penetration testing helps organizations to identify the vulnerabilities in the IoT device architecture, operating system, firmware and network using various software and hardware tools and penetration testing techniques. This section describes the steps involved in pen-testing the IoT devices and various tools used to accomplish this task.

IoT penetration testing is a process of strengthening the IoT device security by finding existing security loopholes in the device and implementing proper security controls. Penetration testing of an IoT device involves testing the API, application, authentication policy, open ports, unencrypted information, and unencrypted communication between two end points.

For hardware, a pen-tester must dissemble the device to test the security of embedded components, storage chips, firmware and configurations of the device and reverse the executable files to discover the vulnerabilities present in them. Once done with the penetration testing, document all your findings at every stage of testing.

IoT penetration testing will help pen-testers to:

- Close unused ports and unnecessary/unknown open ports
- Disable unnecessary services
- Provide protection against unauthorized access and usage of the device
- Design a mechanism for uninterrupted flow of information between two endpoints
- Reduce the risk of compromise and protect the devices from various attacks
- Provide protection against elevation of privileges
- Enhance the device's data encryption policy
- Enhance the security of web application
- Harden the overall device's security

If the device has more open ports it becomes easier for an attacker to connect to it. The first thing an attacker does is test the device for vulnerabilities such as open ports, services and protocols used by the device using which it can be attacked. Designers may install and configure some unwanted services on the device that leave the services with default settings, making it highly vulnerable and an easy target for the attackers. This can cause unauthorized access and launch further attacks such as man-in-the-middle attacks, jamming attacks, DoS attacks, replay attacks, etc. Attackers might also perform banner grabbing to identify the device and find potential entry points or vulnerabilities to the enter the device's network.

Therefore, close all the unused/unnecessary open ports, unwanted services, and configure the device in such a way that it hides the display of the banner. Also create inbound and outbound firewall rules to block all the unwanted ports from allowing any connections from outside the network.

Follow the steps given below to conduct penetration testing of IoT devices:

- **Step 1: Discover IoT Devices**

  The first step of IoT penetration testing is to detect IoT devices connected to the target network. You can attempt to detect IoT devices that are accessible in the target network using tools such as Shodan, Censys, Thingful, and MultiPing.

- **Step 2: Hardware Analysis**

  Perform hardware analysis by evaluating physical and hardware components to check whether they are sufficient to prevent attacks such as altering the platform's components and their usual execution flow. For example, interfaces such as remnant JTAG, SWD and USB are tested as they are often used for interacting with the underlying hardware using hardware tools such as JTAG Dongle, Digital Storage Oscilloscope and Software Defined Radio. Online Tools such as FCC ID Search can be used to obtain details on the wireless operating frequency (including strength), photos of the device, user manuals for the device, and SAR reports on the wireless emissions.

- **Step 3: Firmware and OS Analysis**

  Perform firmware and OS analysis to check the vulnerabilities in the device's firmware and underlying operating system. You can perform firmware analysis to verify the built-in security of the device's firmware and its updating process, such as cryptographically signed firmware updates and using proper authentication mechanisms in the devices to verify digital signatures. You can perform operating system analysis to test the booting process, code execution, application code dumps, various data confidentiality mechanisms and whether the critical application data is properly erased from memory or not.

  You can use tools listed below to perform firmware and OS analysis to identify vulnerabilities:

  - IoTInspector (*http://www.iot-inspector.com*)
  - Binwalk (*https://github.com*)
  - Firmware Mod Kit (*https://github.com*)
  - Firmalyzer Enterprise (*https://firmalyzer.com*)
  - Firmware Analysis Toolkit (*https://github.com*)
  - psad - Intrusion Detection with iptables Logs (*https://github.com*)

- **Step 4: Wireless Protocol Analysis**

  Perform wireless protocol analysis to verify the security and configuration of wireless communication protocols used for local device communication, such as ZigBee, 6LoWPAN and Bluetooth LE. This step includes verifying the role of the device, cryptographic algorithms used for encryption and authentication. You can perform attacks such as replay, man-in-the-middle, unauthorized network access and finally verify the protocol stack using fuzz testing.

You can use tools listed below to perform wireless protocol analysis.

- Ubiqua Protocol Analyzer (*https://www.ubilogix.com*)
- Perytons Protocol Analyzers (*http://www.perytons.com*)
- Wireshark (*https://www.wireshark.org*)
- SoapUI Pro (*https://github.com*)
- Attify Zigbee Framework (*https://github.com*)
- Z3sec (*https://github.com*)

- **Step 5: Mobile Application Testing**

  Perform mobile application penetration testing on IoT devices to test various components of mobile applications such as data protection mechanisms used at storage level, transport level, authentication, authorization, session management and data validation.

  You can use tools listed below to test and identify vulnerabilities in IoT mobile applications:

  - X-Ray (*https://labs.duo.com*)
  - Threat Scan (*http://free.kaspersky.com*)
  - Norton Halt exploit defender (*https://community.norton.com*)
  - Shellshock Scanner - Zimperium (*https://www.zimperium.com*)
  - Hackode (*http://www.ravikumarpurbey.com*)
  - BlueBorne Vulnerability Scanner by Armis (*https://www.armis.com*)
  - EternalBlue Vulnerability Scanner (*https://ebvscanner.firebaseapp.com*)

- **Step 6: Web Application Testing**

  Perform web application penetration testing to evaluate the security of web application used to interact with the IoT devices. The process detects various security weaknesses, flaws, or vulnerabilities, and presents them to the system owner, along with an assessment of potential impacts and possible solutions. IoT web application are tested for vulnerabilities such as input data validation, buffer overflow conditions, SQL injection, bypassing authentication, code execution, etc.

  You can use tools listed below to test IoT web applications for vulnerabilities:

  - SAUCE LABS Functional Testing (*https://saucelabs.com*)
  - PowerSploit (*https://github.com*)
  - Kali Linux (*https://www.kali.org*)
  - WAFNinja (*https://github.com*)
  - Arachni (*http://www.arachni-scanner.com*)

- **Step 7: Cloud Services Testing**

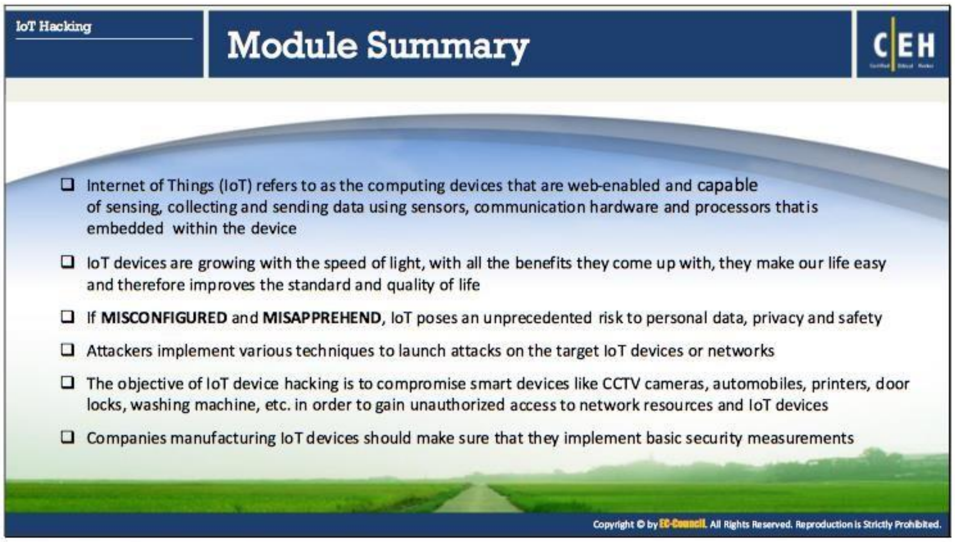
  IoT networks, applications and devices exchange data with multiple back-end platforms. These platforms such as external cloud services must be tested to check whether an attacker is able to gain unauthorized access to obtain critical information.

  You can use tools to test various IoT cloud services for vulnerabilities:

  - ZEPHYR (*https://www.getzephyr.com*)
  - SOASTA CloudTest (*https://www.soasta.com*)
  - LoadStorm PRO (*https://loadstorm.com*)
  - BlazeMeter (*https://www.blazemeter.com*)
  - Nexpose (*https://www.rapid7.com*)

- **Step 8: Document all the Findings**

  After performing all the tests, document all findings and tests conducted. Analyze the target's security and plan respective countermeasures to cover any security gaps.



IoT Hacking

## Module Summary

- ❑ Internet of Things (IoT) refers to as the computing devices that are web-enabled and capable of sensing, collecting and sending data using sensors, communication hardware and processors that is embedded within the device
- ❑ IoT devices are growing with the speed of light, with all the benefits they come up with, they make our life easy and therefore improves the standard and quality of life
- ❑ If **MISCONFIGURED** and **MISAPPREHEND**, IoT poses an unprecedented risk to personal data, privacy and safety
- ❑ Attackers implement various techniques to launch attacks on the target IoT devices or networks
- ❑ The objective of IoT device hacking is to compromise smart devices like CCTV cameras, automobiles, printers, door locks, washing machine, etc. in order to gain unauthorized access to network resources and IoT devices
- ❑ Companies manufacturing IoT devices should make sure that they implement basic security measurements

## Module Summary

This module familiarized you with various topics related with IoT, such as its concepts, threats, attacks, hacking methodology, hacking tools, countermeasures, security tools, IoT pentesting, and pentesting tools. The next module discusses Cloud Computing and the means to secure it.