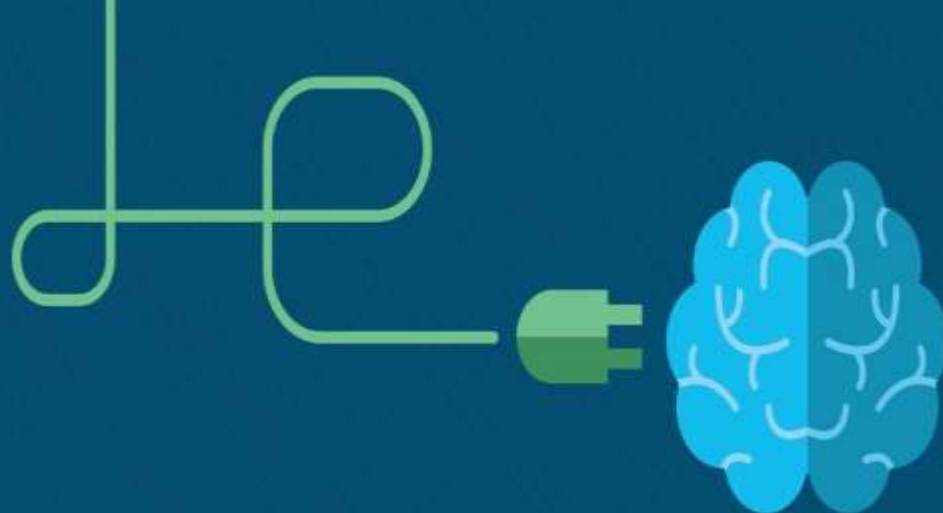


CS433: Internet of Things

NCS463: Internet of Things

Dr. Ahmed Shalaby

<http://bu.edu.eg/staff/ahmedshalaby14>



Chapter 1: The IoT Under Attack

IoT Security 1.0 v2.0



Chapter 1 - Sections & Objectives

- 1.1 Explain the need for IoT security in several IoT environments.
 - Explain why security should be a focus of the IoT
 - Explain how the unique security risks of the IoT differ from standard IT security.
- 1.2 Evaluate potential risks in various IoT use cases.
 - Explain the unique security requirements of the IoT in the smart home.
 - Explain the unique security requirements of the IoT in healthcare.

1.1 Unsecured Connected Things

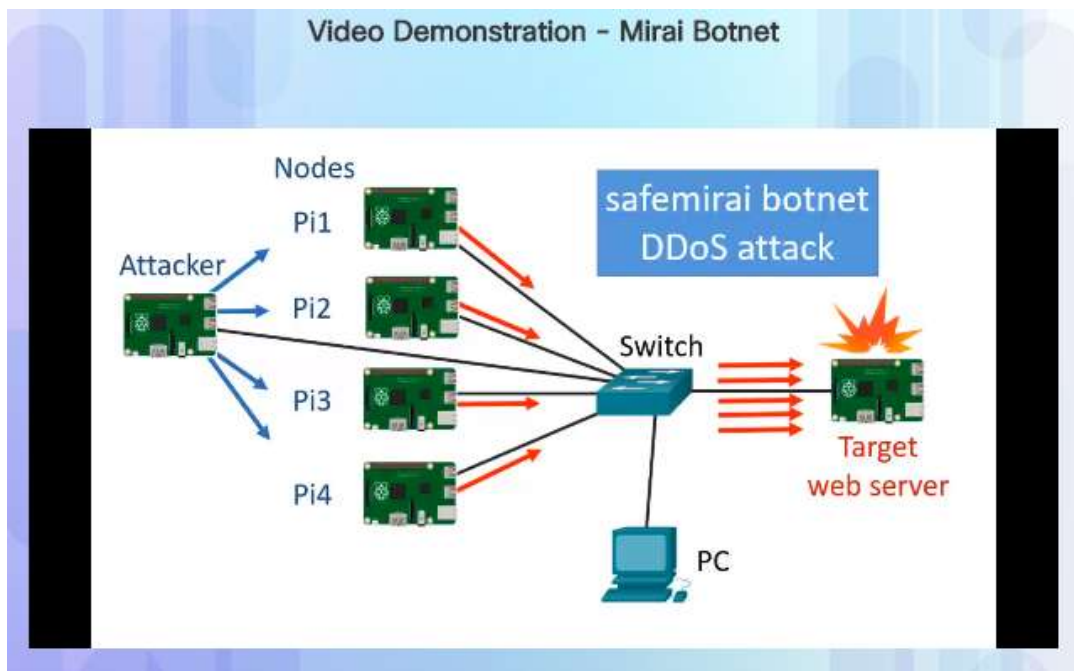
Anatomy of an IoT Attack (Video)

```
55
56 <iframe id="null" class="null">
57
58 class MedisploitModule < Msf::Exploit::Remote
59   Rank = NormalRanking
60
61   include Msf::Exploit::Remote::HttpServer::HTML
62
63   def initialize(info = {})
64     super(update_info(info,
65       'Name' => 'MS05-054 Internet Discover JavaScript OnLoad Handler Remote
66       'Description' => %q{
67         This bug is triggered when the browser handles a JavaScript 'onLoad' handler in
68         conjunction with an improperly initialized 'window()' JavaScript function.
69         This exploit results in a call to an address lower than the heap. The javascript
70         prompt() places our shellcode near where the call operand points to. We call
71         prompt() multiple times in separate iframes to place our return address.
72         We hide the prompts in a popup window behind the main window. We spray the he
```

- IoT has expanded the opportunities for threat actors to act against our networks.
- IoT devices are increasingly being compromised.
- IoT devices are used in a wide variety of attacks because they lack critical device protections such as strong passwords, up-to-date operating systems, and segmented networks.

Unsecured Connected Things

Video Demonstration – Mirai Botnet



- Mirai is malware that targets IoT devices configured with default login information.
- Closed-circuit television (CCTV) cameras make up the majority of Mirai's targets.
- Using a brute force dictionary attack, Mirai runs through a list of default username/passwords.
- October 2016 - Services of Dyn, a (DNS) provider, was attacked, causing Internet outages for millions of users in the United States and Europe.

The Unique IoT Risk

IT and OT in the Manufacturing Sector

- Two distinct networking domains in organizations:
 - **Information Technology (IT)** – Includes devices in the data center, in the cloud, bring your own device (BYOD), and thousands of sensors and actuators connected in the field.
 - **Operational Technology (OT)** – Includes industrial control systems, supervisory control and data acquisition systems, and all the devices that connect to these systems.
- Historically, OT kept the plant running smoothly and IT managed business applications from the front office.
- World of manufacturing is changing:
 - IT and OT operations managers use IT tools to sift through the reams of operational data and make real-time decisions.
 - IT teams can also use this data to do innovative things such as improving the supply chain and reducing downtime.



IT and OT Convergence (Video)

The Unique IoT Risk

Consumer Technology

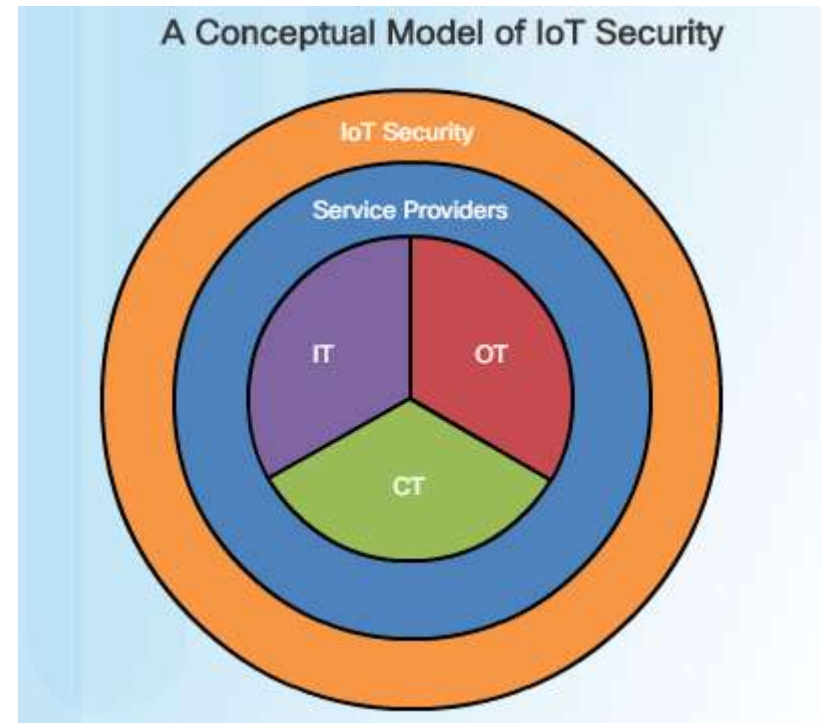
- CT includes connected devices in the home, wearable technology, smart cars, and more.
- Increased number of devices used to communicate.
- In 2016, Internet traffic from CT devices was 61% of all IP traffic. Of all the CT traffic, 81% of it was video traffic.



The Unique IoT Risk

IoT Security Model

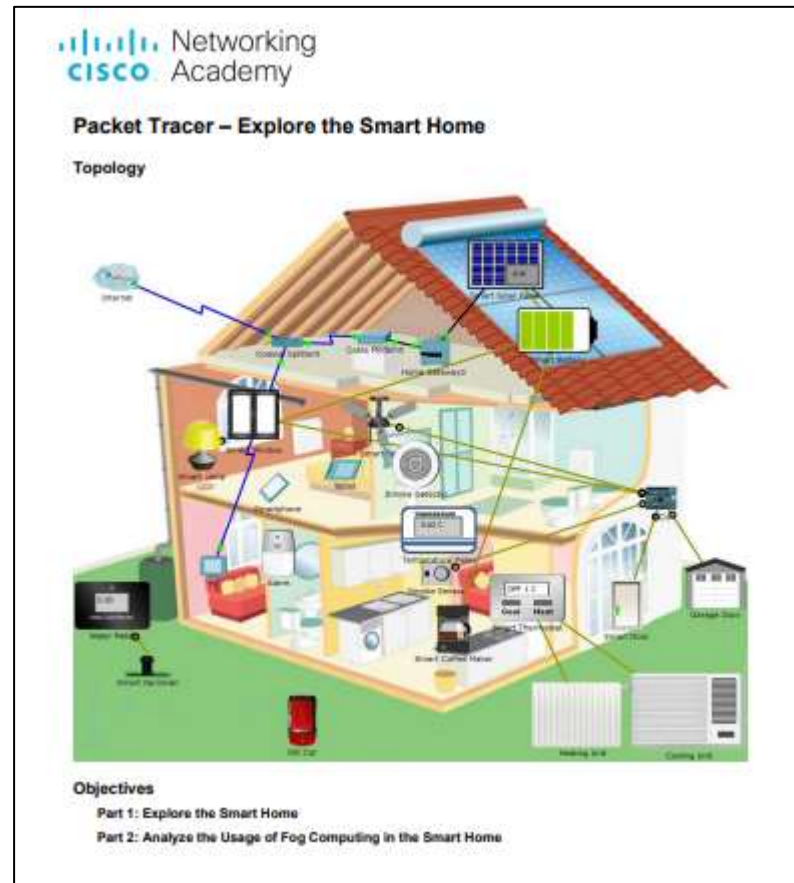
- Whether the IoT device belongs to IT, OT, CT, or some combination of the three, strong security is required.
- Service providers are organizations that connect our devices to the Internet.
 - They are in a position to offer services to address the IoT security needs of their clients.



1.2 IoT Use Case: Smart Home

Packet Tracer – Explore the Smart Home

- A smart home is an example of how the IoT is transforming the way we live, work, and play.
- Smart home devices include lights, thermostats, security systems, smoke and fire detection, appliances, TVs, doors, windows, and anything that can be remotely monitored and controlled.



1.2 IoT Use Case: Healthcare

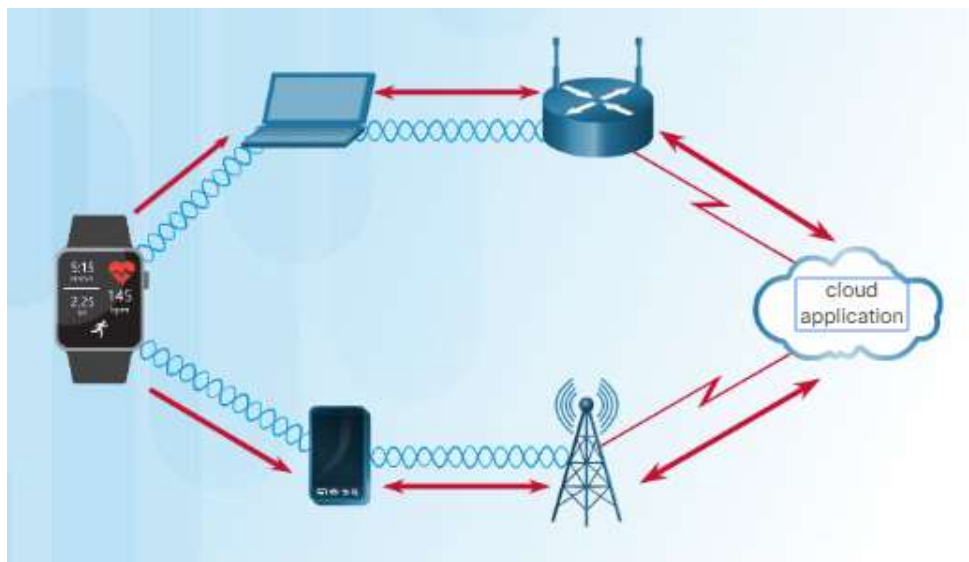
IoT Use Case: Healthcare - Video



- This discussion will provide an explanation of IoT healthcare use cases, vulnerabilities, risks, and mitigation.

IoT Use Case: Healthcare

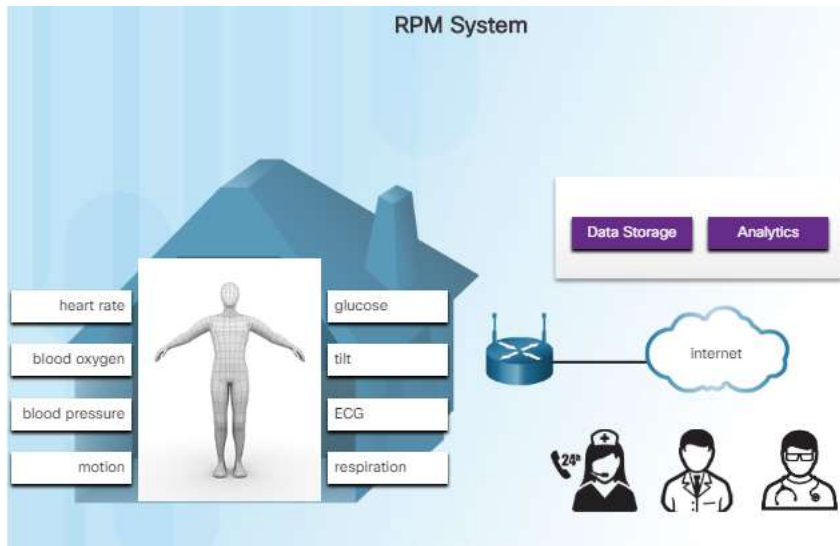
Personal Fitness Devices



- Fitness devices are among the most popular commercial IoT products:
 - Some communicate with a cloud application.
 - Some use a Bluetooth connection to a phone and a cellular data or Wi-Fi connection to the internet and cloud.
 - They come in many forms, such as a wrist watch, headband, helmet, or head phones.
 - They usually consist of a sensor that can detect your heart rate and an accelerometer that detects motion in the form of steps.
 - A cloud application enables storage of personal fitness data, an analysis dashboard, and a wide range of configuration settings.

IoT Use Case: Healthcare

IoT Healthcare Monitoring



- Healthcare Monitoring is one function of IoT devices:
 - It involves the collection and evaluation of patient data over a period of time.
 - Its real-time remote patient monitoring (RPM) has been enabled by IoT.
 - Patients can be monitored at home.
 - Monitoring devices worn by a patient and connected to the Internet.
 - Gateway combines signals from the sensors and securely submits the data to the cloud.
 - An RPM system is shown. A patient is wearing sensors that form a body sensor network (BSN).

IoT Use Case: Healthcare

IoT in the Hospital

- As many as 20 medical devices can be found in a single hospital room.
- IoT provides various functionalities to connected medical devices.
 - Monitoring and submitting patient data.
 - Sensors can be used to track the location of IoT devices and monitor device operation in order to detect problems and help prevent device failure.
 - Therapeutic devices use actuators that are controlled by software to regulate the administration of drugs, fluids, and oxygen.
 - Adds great efficiency to operations, but also adds challenges for IT departments and data security professionals.



Hacking a Pacemaker

- Security vulnerabilities have been found in connected medical devices such as drug infusion and insulin pumps, Bluetooth-enabled defibrillators, refrigeration units that are used to store drugs and blood, and many other devices.
- August 2017 - US government Food and Drug Administration (FDA) approved a software update that patched a security flaw in radio frequency-enabled implantable cardiac pacemakers.
 - Pacemakers include an embedded microprocessor and firmware that is vulnerable to remote attacks over radio frequency (RF).
 - Firmware update could be made over RF without requiring removal or replacement of the device.
 - Estimated that 465,000 devices were affected.



IoT Use Case: Healthcare

Vulnerabilities



- Healthcare networks' vulnerabilities range from weak or nonexistent authentication, unsecured embedded server processes, and unnecessarily vulnerable applications that could be compromised due to user error.
- Healthcare personnel may use web clients running on medical devices and systems to surf the web and read email, making these vulnerable to the same attacks as any computer.
- Many medical devices have a long period of use, the computers used to operate them run old and unpatched operating systems.
- Medical devices are poorly regulated and frequently not designed according to hardware and software security standards.

IoT Use Case: Healthcare

Risks

- Vulnerabilities in connected healthcare devices result in many risks: they can be manipulated, interrupted, or disabled, resulting in patient injury or death.
- Poor device security can allow a threat actor to access data that is stored on a connected healthcare device, or the device can provide access to data stored on the network.
- Personally-identifiable information (PII) about patients can be stolen or manipulated.
- Government regulations regarding the handling of PII can result in severe penalties to healthcare organizations.



IoT Use Case: Healthcare

Mitigation

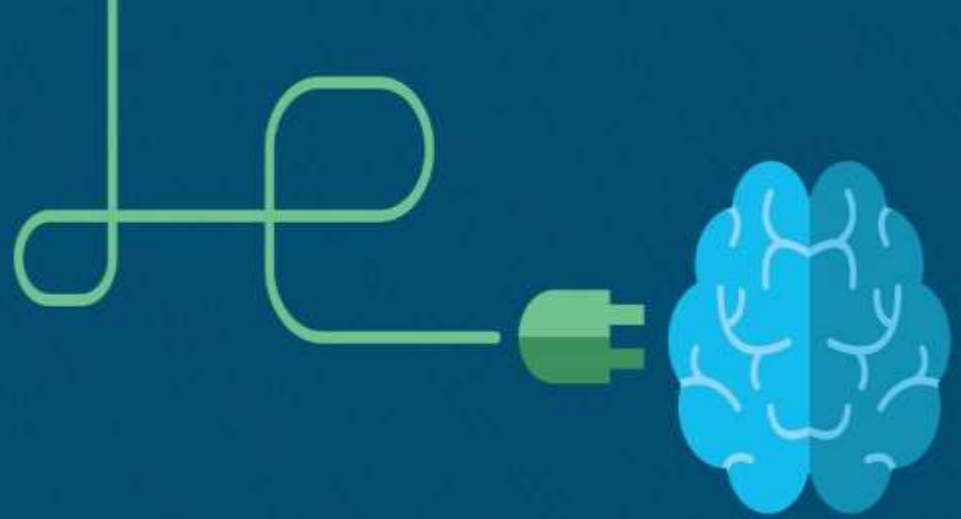
- Device manufacturers: Need to design and build their devices with security in mind throughout the development lifecycle.
- Healthcare administrators: Ensure the devices purchased are secure and that device security has been adequately configured.
- IT personnel: Provide a reliable means for updating and patching network-attached devices.
- Network architectures: Isolate data and control networks from maintenance, vendor, and asset location functionalities.
- Healthcare personnel: Training to build security awareness and create institutional values that embrace security.



Chapter Summary

Summary

- IoT devices are increasingly being compromised and used in a wide variety of attacks because they often lack critical device protections such as strong passwords, up-to-date operating systems, and segmented networks.
- By converging IT and OT, operations managers use IT tools to sift through the reams of operational data and make real-time decisions. IT teams can also use this data to do innovative things such as improving the supply chain and reducing downtime.
- The smart home is an example of how the IoT is transforming the way we live, work, and play.
- A cloud application enables storage of personal fitness data, an analysis dashboard, and a wide range of configuration settings.
- Patients who do not require direct observation by healthcare professionals in a clinical setting can be monitored at home. Monitoring devices can be worn by a patient and connected to the internet.
- The use of the IoT in health care adds great efficiency to operations, but also adds challenges for IT departments and data security professionals



Chapter 2: IoT Systems and Architectures

IoT Security 1.0

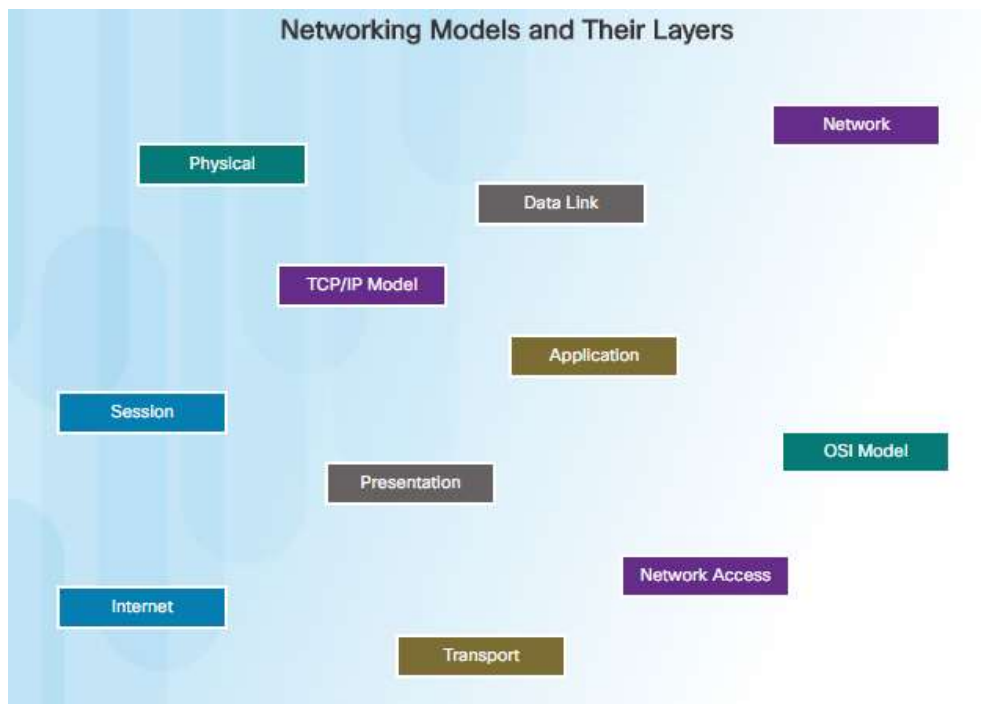


Chapter 2 - Sections & Objectives

- 2.1 Use Industry standard models to explain IoT systems.
 - Explain the value of IoT industry standards.
 - Explain the value of IoT industry standard models.
- 2.2 Use a common model to explain IoT Security.
 - Explain how a layered security model is useful in understanding IoT security requirements.
- 2.3 Create an IoT threat model.
 - Explain the cybersecurity job roles.
 - Explain how threat models are constructed.

Networking Models

OSI and TCP/IP Models



- Layered models are used to illustrate how data communication occurs from end to end.
- Benefits to using a layered model to explain protocols and operations:
 - They assist in protocol design.
 - They foster competition because products from different vendors can work together.
 - They prevent technology or capability changes in one layer from affecting other layers above and below.
 - They provide a common language to describe networking functions and capabilities.
- OSI - Open Systems Interconnection.
- TCP/IP - Transport Control Protocol/Internet Protocol

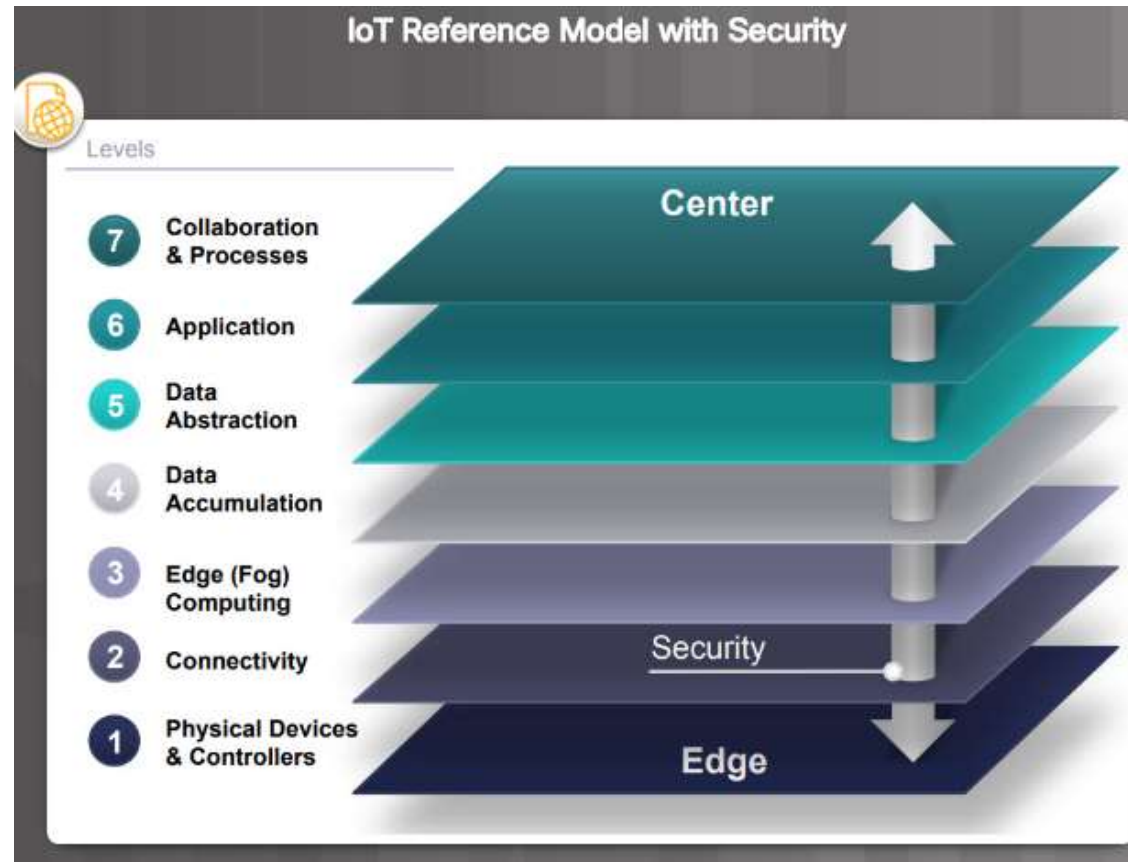
2.1 IoT Models

IoT Reference Model

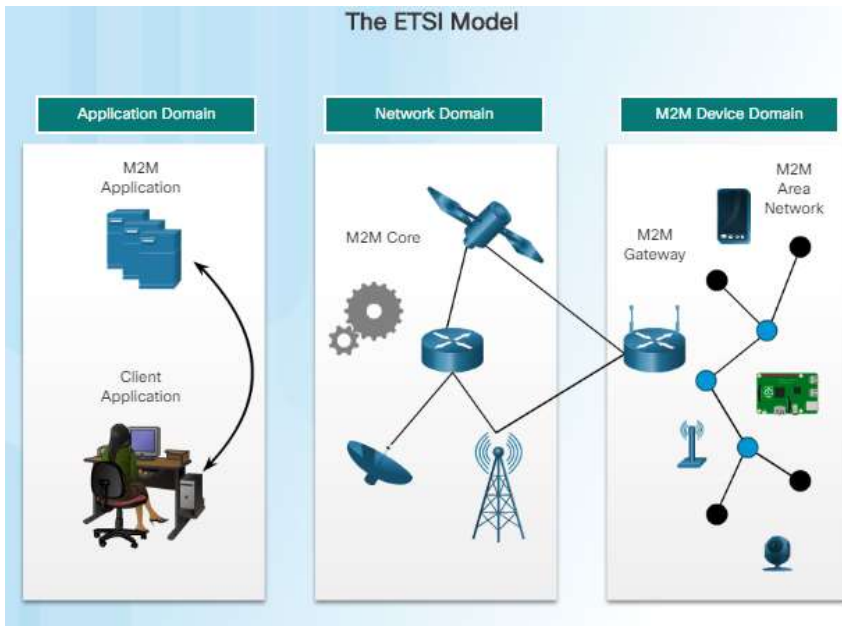
IoT Reference Model		
Level		Description
7	Collaboration & Processes (Involving people and business processes)	Transcends multiple applications to include the communication and collaboration required between people and business processes.
6	Application (Reporting, analytics, control)	Information interpretation based on the nature of the device data and business needs.
5	Data Abstraction (Aggregation and access)	Focused on rendering the data and its storage in ways to enable application development.
4	Data Accumulation (Storage)	Data in motion is converted to data at rest. The data is also transformed so that it can be consumed by upper levels.
3	Edge (Fog) Computing (Data element analysis and transformation)	Converts data into information that is suitable for storage and higher level processing.
2	Connectivity (Communication and processing units)	Responsible for reliable and timely data transmission between devices and the network, across networks, and between the network and data processing in Level 3.
1	Physical Devices & Controllers (The "Things" of IoT)	Includes a wide range of endpoint devices that send and receive information.

Security in the IoT Reference Model

- Security measures include:
 - Securing the hardware and software of each device or system connected to the IoT network.
 - Providing security for all the processes that occur at each level in the network.
 - Securing the movement of data and communications between each level.



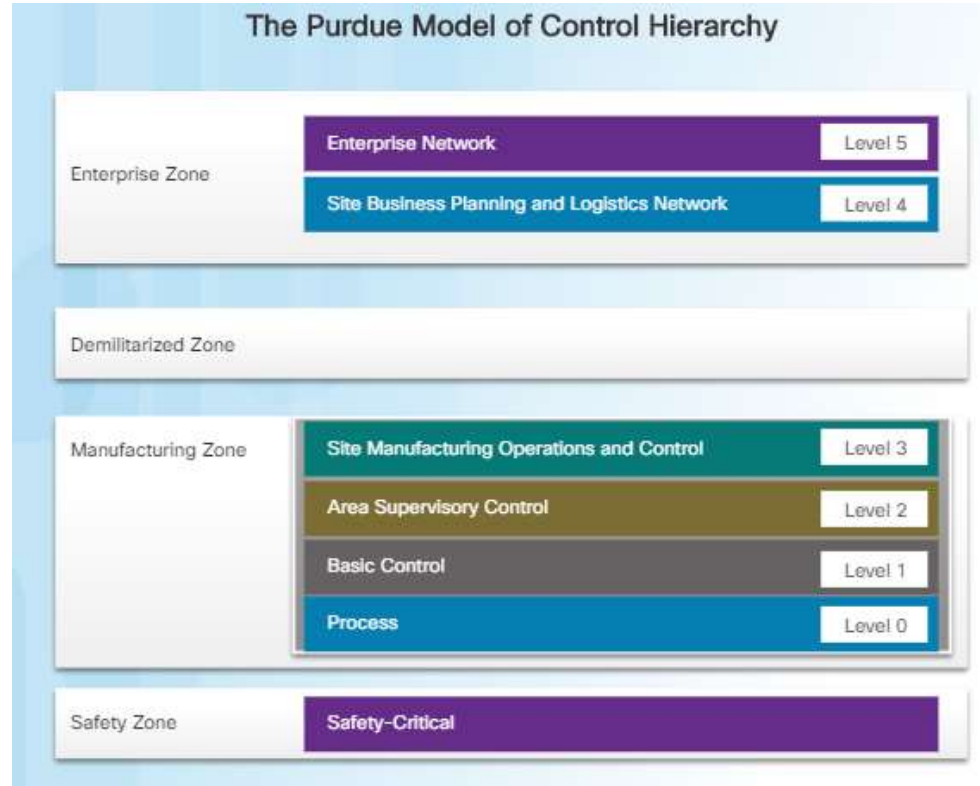
ETSI M2M Standardized Architecture



- 2008 - European Telecommunications Standards Institute (ETSI) created an architecture for machine-to-machine (M2M) communications.
- Purpose of the model is to provide a common framework for understanding the placement of various standards and protocols in an IoT system.
- Model includes three domains:
 - **Application Domain** - management functions can occur such as data analytics, connectivity management, smart energy management, fleet management, or others.
 - **Network Domain** - where data exits on the local network and is transported to the Application Domain using wired and wireless protocols.
 - **M2M Device Domain** - where end devices, such as sensors, actuators, and controllers, connect to the network through M2M gateways.

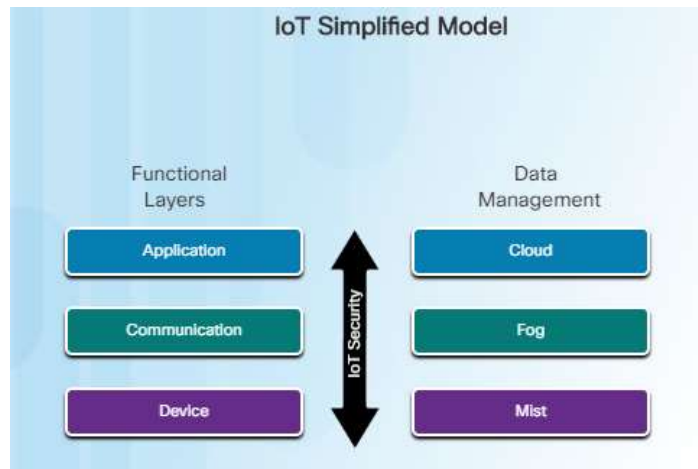
Other IoT Models

- **Purdue Model for Control Hierarchy** - Used in the manufacturing industry, segments devices and equipment into hierarchical functions.
- **Industrial Internet Reference Architecture (IIRA)** - Standards-based framework used by system architects to design industrial systems.
- **Internet of Things - Architecture (IoT-A)** - More formally known as the Architectural Reference Model (ARM) for the Internet of Things.



2.2 IoT Security Layers

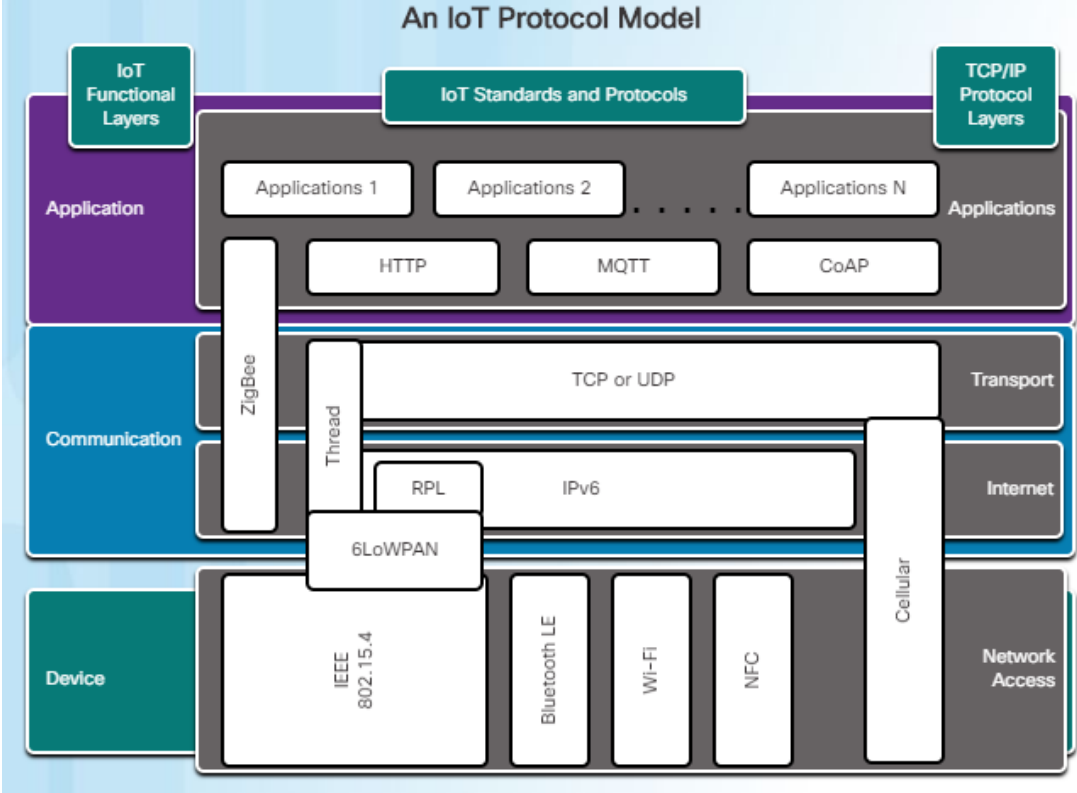
A Simple IoT Model



- Domains - Application, Communication, and Device layers.
 - **Device** layer of an irrigation system might include individual sprinkler heads, moisture sensors, temperature sensors, and actuators.
 - **Communication** layer, these devices might all be connected to a local irrigation control panel that monitors the state of the system.
 - **Application** layer, the control panel may be connected to a remote data center where all the control panels for multiple irrigation systems are aggregated.
- For data management, interested in when and where data is processed.
 - **Mist** layer, close to the ground where things are connected to the network.
 - **Fog** layer on a local device that has more power, such as irrigation system's control panel.
 - Can a supervisor remotely override the autonomous actions of the control panel using a mobile or desktop application in the **Cloud**?

IoT Security Layers

IoT Security Model



- This course uses a combination of the functional layers of the IoT simplified model overlaid with the TCP/IP model.
- **Application**
 - ZigBee, Hypertext Transfer Protocol (HTTP/HTTPS), Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP)
- **Communication**
 - Thread, Transport Control Protocol (TCP), UDP, RPL, IPv6
- **Device**
 - 6LoWPAN, IEEE 802.15.4, Bluetooth Low Energy (BLE), Wi-Fi, Near Field Communication (NFC), Cellular

2.3 NICE and IoT Systems

NICE Cybersecurity Workforce Framework

- National Institute for Standards and Technology (NIST) published the National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework in November 2017.
 - Excellent reference for learning how to identify, recruit, develop, and retain cybersecurity talent.
 - Publication defines work roles that include the necessary knowledge, skills, and abilities (KSAs) required as well as the tasks performed by someone in the work role.
- The work roles are divided into seven categories. For this course, we are interested in the **Securely Provision** category and the **Protect and Defend** category.



NICE and IoT Systems

Securely Provision

- **Securely Provision** work roles are responsible for conceptualizing, designing, procuring, and implementing secure information technology (IT) systems.
 - In this course, the Risk Management specialty area is the focus.
 - Includes all the processes necessary to assure that existing and new IT systems meet the organization's cybersecurity and risk requirements.
- **Security Control Assessor** - People in this role conduct comprehensive assessments of the management, operational, and technical security controls to determine their overall effectiveness.



NICE and IoT Systems

Protect and Defend

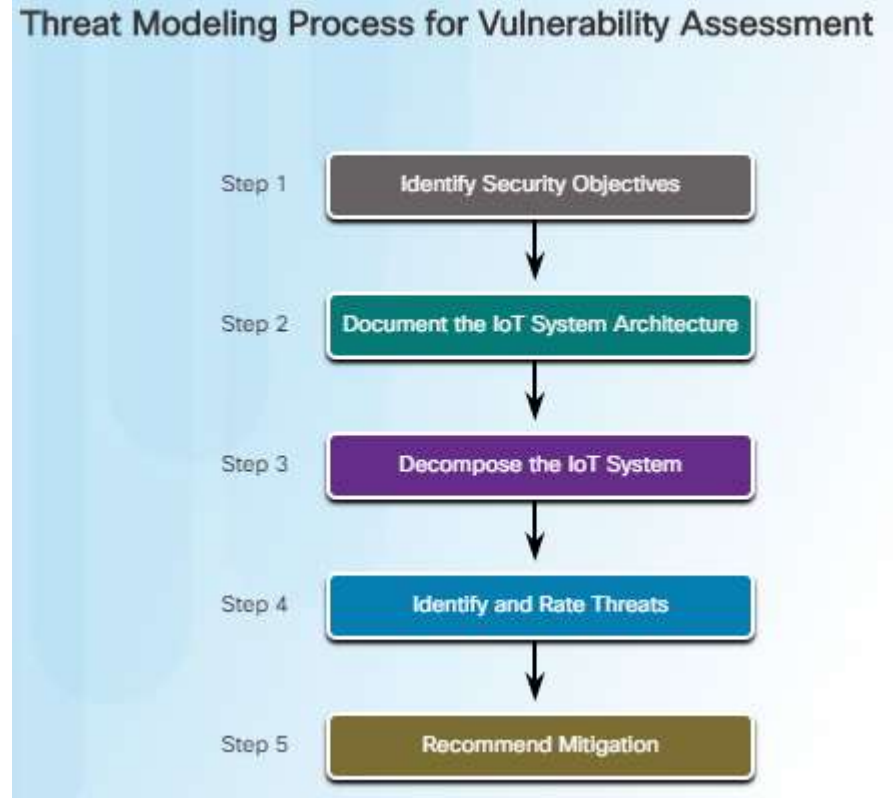
- **Protect and Defend** work roles are identifying, analyzing, and mitigating threats to IT systems.
 - Vulnerability Assessment and Management specialty area is the focus for this course.
 - Includes conducting assessments of threats and vulnerabilities; determining deviations from acceptable configurations or policies; assessing the level of risk; and developing or recommending appropriate mitigation countermeasures.
- **Vulnerability Assessment Analyst** - perform assessments of IT systems and identify where those systems deviate from acceptable configurations or policy.



Threat Model Analysis

Threat Model Analysis for an IoT System

- Threat modeling - tool used to conduct tasks for risk management and vulnerability assessments.
 - Structured approach for analyzing the security and vulnerability of a system, whether that system be a device's hardware, software, or the networks used to communicate with other devices.
- This course uses an adaption of Microsoft's Threat Model Analysis and applies it to an IoT system.



Step 1: Identify Security Objectives



- Use the following categories to determine the security objectives for the IoT system:
 - **Identity** - Document the controls that are in place to ensure that evidence is collected on the identity of users accessing and using the IoT system.
 - **Financial** - Document the financial risks of the various aspects of the IoT system so that management can determine which level of risk is acceptable. For example, the financial risk of losing a controller in the network would not be as tolerable as losing one of the sensors that report to the controller.
 - **Reputation** - Document possible impact on the organization's reputation if the IoT system is attacked. For example, the reputation of a company that sells web cameras would be impacted if their product became part of a worldwide distributed denial of service (DDoS) attack.
 - **Privacy and Regulation** - Document the impact of privacy concerns and regulation requirements. For example, the data from a temperature sensor in an irrigation system may not have any privacy or regulatory concerns.
 - **Availability Guarantees** - Document the expected availability and guaranteed uptime of the IoT system. For example, the tolerance for downtime to an industrial control system (ICS) may be very low and require the implementation of significant security measures and system redundancies.
 - **Safety** - Document the potential impacts to physical welfare of people and physical damage to equipment and facilities.

Step 2: Document the IoT System Architecture

- Create documents that describe the IoT system architecture including:
 - Components of the IoT system at the application, communication, and device layers
 - The flow of data between components and between layers
 - The technologies, protocols, and standards used to implement the IoT system



Step 3: Decompose the IoT System



- Dive deeper into individual components and features that impact the security objectives of the IoT system.
- Create a security profile that will help you identify threats and vulnerabilities in the design, implementation, and deployment of the IoT system.
- During this step, gather information about the IoT system using the following tasks:
 - **Identify trust boundaries** between trusted components and untrusted components.
 - **Identify data flow** between devices, the communications network, and the applications.
 - **Identify entry points** where data is input into the system.
 - **Identify sensitive data** within the IoT system where secure resources are stored and manipulated.
 - **Document the security profile** to include approaches to input validation, authentication, authorization, configuration, and any other areas of the IoT system that are vulnerable.

Step 4: Identify and Rate Threats



- A threat is a potential danger to any asset such as data or components of the IoT system.
 - Threat actors are people or entities who exploit vulnerabilities.
 - A vulnerability is a weakness in the IoT system or its design that could be exploited by a threat.
 - Vulnerabilities combine to make up an attack surface. Attack surfaces describe different points where a threat actor could get into a system, and where they could get data out of the system.
- In this course, use two tools to identify and rate the threats and vulnerabilities.
 - STRIDE is a vulnerability assessment tool used to identify the threats. STRIDE is an acronym that stands for the following categories of threats: **Spooffing Identity, Tampering with Data, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege**
 - DREAD is a risk assessment tool and is used to rate the threats discovered in the STRIDE process. DREAD is an acronym that stands for the variables used to quantify, compare, and prioritize the amount of risk in each threat:
 - **DREAD Risk Rating = (Damage + Reproducibility + Exploitability + Affected Users + Discoverability)/5**

Threat Model Analysis

Step 5: Recommend Mitigations Techniques and Technologies

- After identifying and rating the threats, determine the mitigation techniques for each threat and select the most appropriate technology that would reduce or eliminate the threat.
- During this evaluation, keep in mind what makes sense from a business perspective, including existing policies within your organization.



Chapter Summary

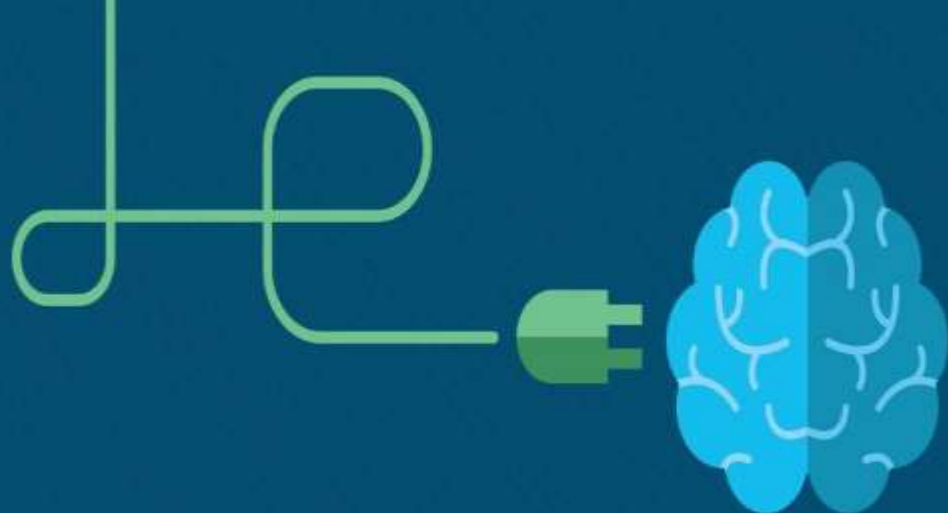
Summary

- There are many benefits to using a layered model to explain protocols and operations.
 - Assist in protocol design, foster competition, prevent technology or capability changes in one layer from affecting other layers above and below, and provide a common language to describe networking functions and capabilities.
- The intent of the IoT reference model is to provide common terminology and help clarify how information flows and is processed for a unified IoT industry.
 - Security must permeate throughout all the levels in the IoT Reference Model.
 - ETSI model includes three domains: M2M device, network, and application.
 - Other IoT models include the Purdue Model for Control Hierarch, IIRA, and IoT-A.
- IoT security layers in a simplified IoT model consist of device, network, and application layers. To better understand the placement of the more common protocols and standards used in IoT systems, a combination of the layers can be overlaid with TCP/IP.

Chapter Summary

Summary (Cont.)

- NIST's NICE publication is an excellent reference for learning how to identify, recruit, develop, and retain cybersecurity talent.
 - Work Category - risk management: includes all the processes necessary to assure that existing and new IT systems meet the organization's cybersecurity and risk requirements.
 - Work Category - vulnerability assessment and management: includes conducting assessments of threats and vulnerabilities; determining deviations from acceptable configurations or policies; assessing the level of risk; and developing or recommending appropriate mitigation countermeasures.
- Threat model analysis: Step 1 is to identify security objectives, step 2 is to document the IoT system architecture, step 3 is to decompose the IoT system, step 4 is to identify and rate threats, and step 5 is to recommend mitigation. In step 4, STRIDE is used to identify threats and DREAD is used to rate threats.



Chapter 3: The IoT Device Layer Attack Surface

IoT Security 1.0 v2.0



Chapter 3 - Sections & Objectives

- 3.1 Explain the operation of IoT device hardware and firmware.
 - Explain the operation of IoT device hardware components.
 - Explain the operation of IoT device software components.
- 3.2 Perform threat modeling activities to evaluate IoT device hardware and firmware.
 - Perform threat modeling activities to evaluate IoT device hardware.
 - Perform threat modeling activities to evaluate IoT device firmware.
- 3.3 Recommend measures to mitigate threats to IoT devices.
 - Recommend measures to mitigate threats at the device layer.
 - Recommend measures to mitigate protocol security threats on an IoT Device.

3.1 IoT Device Hardware Components

OWASP Hardware Vulnerability Components

Open Web Application Security Project has a list of vulnerabilities for each Attack Surface

- **Hardware Sensors**
 - Environment manipulation
 - Tampering
 - Damage
- **Device Memory**
 - Default username and password
 - Sensitive data
 - Plaintext usernames and passwords
 - Encryption keys
- **Device Physical Interfaces**
 - Removal of storage media
 - Reset to insecure state
 - Device ID/Serial number
 - Serial interface connections
 - User and Administrative access
 - Privilege escalation



- **Device Firmware**
 - Backdoor Accounts
 - Hardcoded credentials
 - Encryption keys
 - Firmware version display
 - Firmware version last update date
 - Vulnerable services
 - Security related function API exposure
- **Firmware Update Mechanism**
 - Update sent without encryption
 - Updates not signed
 - Update location writable
 - Update verification and authentication
 - Malicious update
 - Missing update mechanism
 - No manual update mechanism

IoT Device Hardware Components

Constrained Devices

- IoT is made up of constrained devices which usually have very limited power, memory, and processing cycles. Communication capabilities are limited and unlikely that encryption is implemented (one of the OWASP vulnerabilities)

Name	Data Size (RAM)	Code Size (Flash Storage)
Class 0, C0	< 10 Kilobytes	< 100 Kilobytes
Class 1, C1	~ 10 Kilobytes	~ 100 Kilobytes
Class 2, C2	~ 50 Kilobytes	~ 250 Kilobytes

- Constrained devices in this course:**

- Smart Sensors** - Center of IoT devices. Capable of communicating with a monitoring system by using a microprocessor and have the ability to self-diagnose a problem.
- Embedded Devices** - Contain a computing system designed for a special purpose, typically designed to run a single application. Products may provide internet connectivity and are considered smart or intelligent.
- Prototyping** - Raspberry Pi and Arduino are prototyping devices for embedded systems. The Raspberry Pi needs a complete operating system to operate. The Arduino is a single-board microcontroller that can be configured by writing program code to instruct it do various functions. The program is then compiled and sent to the Arduino's non-volatile flash memory.

IoT CPU Types

- Major CPU types used in the IoT are ARM, MIPS and x86.
- Two categories: Reduced Instruction Set Computing (RISC) and Complex Instruction Set Computing (CISC).
- **RISC processors**
 - Have fewer transistors than CISC processors.
 - Dominate the mobile computing market.
 - Fewer transistors translate to lower cost, less power consumed, and less heat produced.
 - The two main providers of RISC processors include:
 - **ARM** (Advanced RISC Machine) – This is an architecture generally licensed to other companies to design their own processor. Both 32-bit and 64-bit architectures. Raspberry Pi is an ARM processor.
 - **MIPS** (Microprocessor without Interlocked Pipeline Stages) - Used for many processors in embedded systems as well as networking, mobile, and IoT devices. Both 32-bit and 64-bit implementations.



IoT Device Hardware Components

IoT CPU Types (Cont.)



- **CISC processors**
 - Ability to perform several operations with a single instruction. More transistors are necessary to store the more complex instructions which create more heat, require more power, and add to the cost of the processor. The use of complex instructions reduces the size of the program code which provides a benefit in loading and storing applications.
 - Primary providers are Intel and Advanced Micro Devices (AMD). Both Intel and AMD have been actively pursuing the IoT market by attempting to reduce power consumption and heat in their processors.
- **Heterogeneous Computing**
 - Involves using more than one kind of processor with different capabilities. A common approach used by several manufactures employs the Graphics Processing Unit (GPU) to perform complex mathematical calculations or to handle encryption and decryption tasks.
- **big.LITTLE Computing**
 - ARM's big.LITTLE technology uses processors (cores) with differing processing capabilities and power requirements. The big processor provides the most compute performance, and has higher power requirements. Using big.LITTLE can extend battery life in devices that are in remote locations.

IoT Device Hardware Components

Memory

- IoT devices have various types of memory used for storing data, firmware, and processing.
- Common memory types and uses are:
 - **SD Card** – Used to store data necessary for IoT operation or to store collected data. Must be protected from removal.
 - **Non-Volatile Memory** – EPROM (erasable programmable read-only memory) and EEPROM (electrically erasable programmable read-only memory). Retain the information stored even when power is off. Used to store firmware, the bootloader, and other critical information required for the IoT device to operate. An attacker may be able to read the communication between the memory and the microcontroller.
 - **Volatile Memory** – SRAM (Static Random Access Memory) and DRAM (Dynamic Random Access Memory) are used to hold the operating code and provide temporary storage while the device is running. After the device is powered down, all data in memory is lost.



IoT Device Hardware Components

Physical Ports

- IoT devices may have ports such as USB and Ethernet.
 - Standard procedures for the protection of USB and Ethernet ports should be employed.
 - Those ports could be used to extract information from the device by connecting another computer system to the IoT device.
- The following is a list of some of the other communication ports that may be available:
 - **Universal Asynchronous Receiver-Transmitter (UART)** – This interface could be used to communicate with other hardware peripheral devices. There are typically three pins necessary for serial communication with UART: Tx (Transmit), Rx (Receive), and Ground. There is the possibility that data may be transmitted on these pins providing an attacker with the ability to capture the data. When these pins are not needed they should be disabled within the configuration if possible.



IoT Device Hardware Components

Physical Ports (Cont.)

- The following is a list of some of the other communication ports that may be available:
 - **Inter-Integrated Circuit (I2C)** – Serial data protocol used for short distance communication, often between chips on the same board. I2C may be used to communicate between the microcontroller and EEPROM chips to store data or program code. An attacker could potentially corrupt data or extract data that is being transferred.
 - **Serial Peripheral Interface (SPI)** – Short distance serial protocol that uses a four-wire serial bus. Allows for multiple devices to be connected to the same wires and is capable of full duplex communication. Used for communicating with devices on the same board. Used for communicating with EEPROM, flash, or other devices located as much as a few feet away. Extracting sensitive information is a very real possibility.
 - **Joint Test Action Group (JTAG)** – JTAG is not a communication protocol, but rather a protocol to be used for testing and debugging. Providing access to the JTAG port could allow an attacker to reverse engineer the logic for the microcontroller. An attacker could also extract the firmware and possibly even load malicious firmware on the device.



IoT Device Software Components

Embedded Systems

- **Embedded systems** are designed for a specific function within a larger system.
- Example: Home security devices
 - All operations are controlled by a microcontroller designed specifically for that purpose.
 - Microcontroller can be programmed for the sensors unique to the installation.
 - Sensors may include smoke, motion, gas, and temperature sensors that provide data to the microcontroller which will trigger an alarm if something exceeds the thresholds set for the particular sensor.
 - Microcontroller may have the ability to display information on a screen or communicate with other computer equipment for monitoring.



IoT Device Software Components

Embedded Systems (Cont.)



- Some embedded systems use microprocessors.
 - A microprocessor and microcontroller may have the same CPU embedded.
 - The microcontroller-based system is self-contained and could include flash memory, RAM, serial communications, and other peripherals within the integrated circuit.
- Embedded systems may use an embedded operating system or be programmed directly using the machine code for the CPU. In some cases, stripped down versions of Linux are used.
- Debugging the programming for embedded systems is handled differently than typical PC software debugging.
 - On a PC, the software is developed on the same processor that the program will run on. This provides the use of built-in tools within the development environment to debug for potential program errors.
 - On an embedded system, the software is built outside the environment in which it will be operating. In order to debug embedded system software, the systems make use of the JTAG port to track down software issues.

IoT Device Software Components

Compiled or Interpreted Code

- Developers have a choice as to the environment when developing application software.
 - Emulation environments available for the PC for developing applications targeted to other platforms. When developing for mobile devices, an emulator is available so that screen layout would mirror how the application will look and perform on the mobile device
- Developers also have a choice as to the type of programming language.
 - **Compiled Code**
 - Source code is written in a format that is readable with a text editor and then converted (compiled) into machine code that is read and executed by the processor.
 - The developer must complete the compilation process before the program is useable. If changes are necessary, the text code is changed and then recompiled prior to being used.
 - Examples include C, C++, Rust and Visual Basic.
 - **Interpreted Code**
 - Each instruction is executed one after another. The interpreter translates the instruction into a form of machine code that can be performed by the processor. If an error occurs, the program will stop at that point and corrections can be made.
 - Examples: Python, JavaScript, Perl, and PHP.



IoT Device Software Components

Compiled or Interpreted Code (Cont.)

- Depending on the operating system used, it is also possible to use scripts to perform various tasks.
 - Linux uses shell scripts to perform certain repetitive tasks.
 - Windows uses PowerShell.
 - Both operate in an interpretive fashion.
- Interpreted code is easy to modify by an attacker because it is stored in a text format.
- Compiled code could be altered by an attacker using a debugger and replacing machine code instructions with malicious code. With compiled code, it is possible to digitally sign the binary executable to verify that it has not been altered.



IoT Device Software Components

Debug/Boot Mode

- Common for systems to offer a special debug/boot mode in case the system encounters a problem when starting up.
- Sometimes the debug/boot mode can be accessed using a keystroke combination.
- This is also possible in the case where attackers have access to the device board. They may be able to use the JTAG port.
- When operating in debug/boot mode, there is a possibility authentication could be bypassed.
- If attackers can gain access to the debug/boot mode, it would be possible for them to make other changes to the system or even install a backdoor. This would provide access to the system, if the system is available on a network.



Common IoT Operating Systems

- IoT devices typically use a trimmed down version of an operating system.
- Developers can choose from open source and commercial options.
- Busybox** - open source and uses a Linux kernel.
 - Provides a set of programs that can be executed from the command line
 - Developer should disable the unnecessary programs during compilation. Example: Telnet
- Android Embedded** - lightweight Linux version primarily used in mobile devices, but can be used for IoT devices.
 - Designed to reduce power consumption and works with the common processors used in IoT devices.
- Commercial options** - products such as VxWorks, Windows 10 IoT, and ARM Mbed are available.

```
Busybox Command Line

Type 'busybox' to see the list of available commands.
/* busybox
Busybox v1.20.0 (2012-04-22 12:29:58 CEST) multi-call binary.
Copyright (C) 1998-2011 Erik Andersen, Rob Landley, Denys Vlasenko
and others. Licensed under GPLv2.
See source distribution for full notice.

Usage: busybox [function] [arguments]...
or: busybox --list[-full]
or: busybox --install [-s] [DIR]
or: function [arguments]...

Busybox is a multi-call binary that combines many common unix
utilities into a single executable. Most people will create a
link to busybox for each function they wish to use and Busybox
will act like whatever it was invoked as.

Currently defined functions:
[, [[, acpid, add-shell, addgroup, adduser, adjtimex, arp, arping, ash,
awk, base64, basename, beep, blkid, blockdev, bootchartd, brctl,
bunzip2, bzip2, cal, cat, catv, chat, chatter, chgrp, chmod,
chown, chpasswd, chpst, chroot, chrt, chvt, cksum, clear, cmp, comm,
conspy, cp, cpio, crond, crontab, cryptpw, ctyback, cut, date, dc, dd,
<output omitted>
/*

https://busybox.net/faq\_busybox.html
```

3.2 Hardware Security

Physical Vulnerabilities of Constrained Devices

- Constrained devices are often placed in remote locations where physical security may be difficult to implement.
- Potential vulnerabilities could include:
 - Theft of the device.
 - Physical damage to the device.
 - Disabling the device, removing power source.
 - Disabling communication, disconnecting cables or other means of disruption.
- Provide some type of video surveillance where possible.
- Provide a tamper proof or tamper resistant type of housing.



Hardware Security

Physical Device Security

- A device such as a sensor could be moved, causing it to lose calibration.
- Many smart sensors have the ability to trigger an alarm when they are not in proper adjustment.
- Devices that have storage, such as an SD card, could potentially have data stolen or destroyed by an attacker.
- Standard surveillance and security protocols should be implemented as a first layer of defense.
- Physical device security also includes insuring that you always have access to the device. Consider implementing battery backup to power IoT devices in case of power outages.



Hardware Vulnerabilities

- Hardware vulnerabilities are very common with many IoT devices.
- Wired magazine published an article in August 2017 outlining how to gain access to firmware on numerous IoT devices using eMMC flash and a \$10 SD card reader.
 - By soldering five wires to the eMMC flash chip and using a standard SD card reader the attacker was able to retrieve the firmware, operating system, and software on the chip and then save them to a PC. After the software is copied it can be examined for code vulnerabilities. Hacks like this one illustrate the need for implementing physical device security.
- Hardware vulnerabilities include:
 - Shell access via the UART connection to an IoT doorbell.
 - UART hack - a smart refrigerator provided access to a root shell when the system was rebooted.

Hardware Vulnerabilities (Cont.)

- Other devices with hardware-based vulnerabilities that have been exploited include:
 - Blu-Ray players
 - Cameras
 - Home automation devices
 - Media players
 - Music players
 - NAS devices
 - Printers
 - Televisions
 - VoIP hardware
 - Medical devices
 - Networking devices
 - Android TV devices



Firmware Vulnerabilities

- IoT devices require firmware to run.
- Firmware is basically embedded software that contains a minimal operating system and related programs to control the IoT device.
- IoT device firmware can contain security vulnerabilities that are discovered after their release. Firmware-related vulnerabilities for IoT devices are similar to those of other computers or networking devices.



Firmware Vulnerabilities (Cont.)

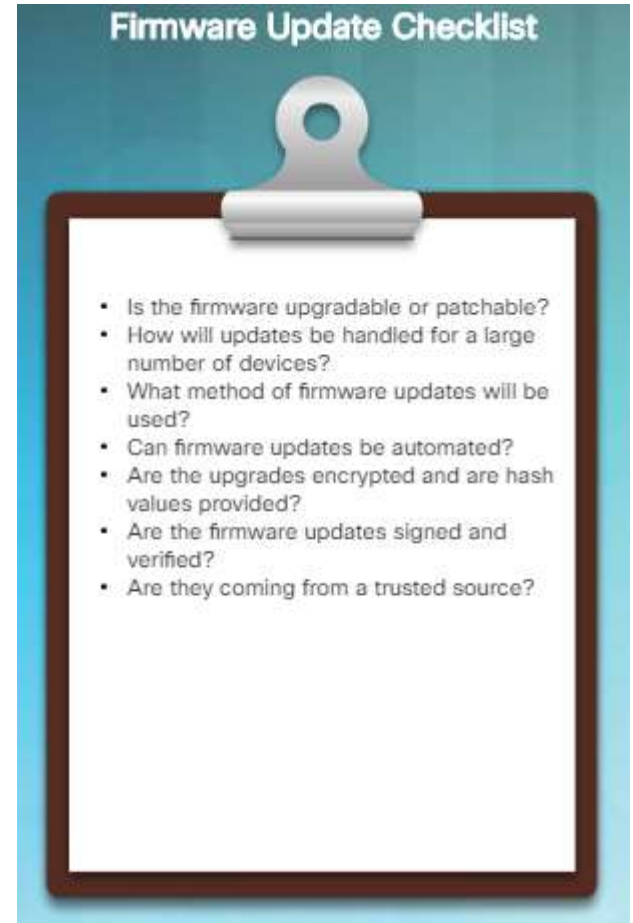
- These vulnerabilities include the following:
 - **Default Login Credentials**
 - Most IoT attacks occur because the default login credentials were not changed.
 - Important that usernames and passwords are changed to meet strong criteria before connecting any IoT device to the internet.
 - **Distributed Denial of Service (DDoS) attacks**
 - DDOS attacks require botnets of infected systems from around the world.
 - After the weak login information is known, a hacker can write an automated script to log into remote IoT devices and copy the infected software to their systems.
 - **Out-of-Date Firmware**
 - If a hacker is trying to target a specific IoT device or group of devices, they will usually look to see if the firmware is out of date or look for any exploits that have yet to be addressed with a patch.

Firmware Vulnerabilities (Cont.)

- These vulnerabilities include the following:
 - **Buffer Overflow Attacks**
 - Buffer overflow attacks can occur with vulnerable software when the programmer does not account for the size of the input that a user might enter.
 - A buffer overflow attack could cause corrupt data, a denial of service, or could allow malicious code to run on the target system.
 - **Backdoor Installation**
 - The installation of a backdoor usually occurs after the attacker gains remote access to the IoT device.
 - On a Linux-based operating system, the attacker could run the netcat command in the background and execute malicious commands on this system remotely from anywhere in the world.
 - In addition, network diagnostic and testing tools are sometimes left behind in the firmware by the IoT device manufacturer. These tools can make the devices more exploitable if unauthorized entry occurs.

Firmware Update Issues

- Updating IoT firmware and installing patches to fix security vulnerabilities are critical components of network security.
- IoT security has not kept up with the growth rate of IoT devices.
 - In some cases, patches do not exist for security vulnerabilities for devices.
 - In other cases, the device may not even be updatable or patchable.
- IoT devices in an organization might number in the thousands or tens of thousands.
 - Installing updates and patches on this number of devices presents its own challenges.
 - It is important to verify that all upgrades and patches come from a verified source.



Firmware Vulnerabilities

Firmware Update Solutions

- Necessary to keep a database of all IoT devices and firmware information.
- Firmware should be updated as soon as new releases come out because it is likely these updates are addressing security vulnerabilities.
- A plan should exist for checking the manufacturer's website for updates on a regular basis.
- Important to monitor or subscribe to security vulnerability services.
- The best solution is to have an automatic system for updating firmware and installing security patches on IoT devices within an organization. Important that any firmware updates or patches are digitally signed and verified before installing.



Firmware Vulnerabilities

Rooting an OS

- Rooting an IoT device - attacker followed a process that successfully granted him root access.
- Root access provides attacker complete control over that device.
- The JTAG and UART interfaces are common attack vectors for gaining root access to the device.
 - After obtaining access, can read the device's memory and modify the firmware.
 - After access to the firmware, can look for vulnerabilities and introduce new security holes.

```
root root 12288 фев 9 14:42 bin
root root 4096 фев 9 14:44 boot
root root 4096 авг 18 2017 cdrom
root root 4120 фев 27 19:04 dev
root root 12288 фев 9 14:44 etc
root root 4096 сен 14 19:23 home
root root 32 авг 18 2017 initrd.img
root root 4096 янв 25 19:11 lib
root root 4096 янв 25 19:11 lib64
root root 16384 авг 18 2017 lost+found
root root 4096 окт 1 13:22 media
root root 4096 июн 29 2017 mnt
```

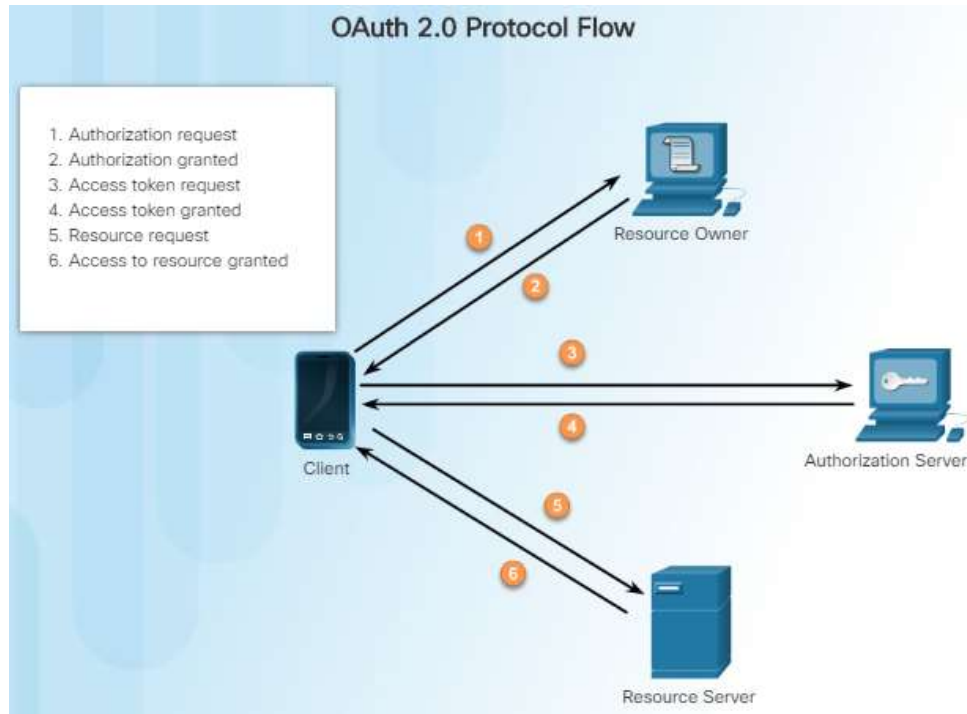
3.3 Network Access Control Concepts

Access Control Models

- A security analyst should be familiar with different basic access control models to have a better understanding of how threat actors can break the access controls.
 - **Mandatory access control (MAC)** - Applies the strictest access control and is typically used in military or mission critical applications. Assigns security level labels to information and provides users with access based on their security level clearance.
 - **Discretionary access control (DAC)** - It allows users to control access to their data as owners of that data.
 - **Non-Discretionary access control** - Access decisions are based on an individual's roles and responsibilities within the organization, also known as role-based access control (RBAC).
 - **Attribute-based access control (ABAC)** - Allows access based on attributes of the object (resource) to be accessed, the subject (user) accessing the resource, and environmental factors regarding how the object is to be accessed, such as time of day.
- **Principle of least privilege** - users should be granted the minimum amount of access required to perform their work function.
- **Privilege escalation exploit** - vulnerabilities in servers or access control systems are exploited to grant an unauthorized user, or software process, higher levels of privilege than they should have.

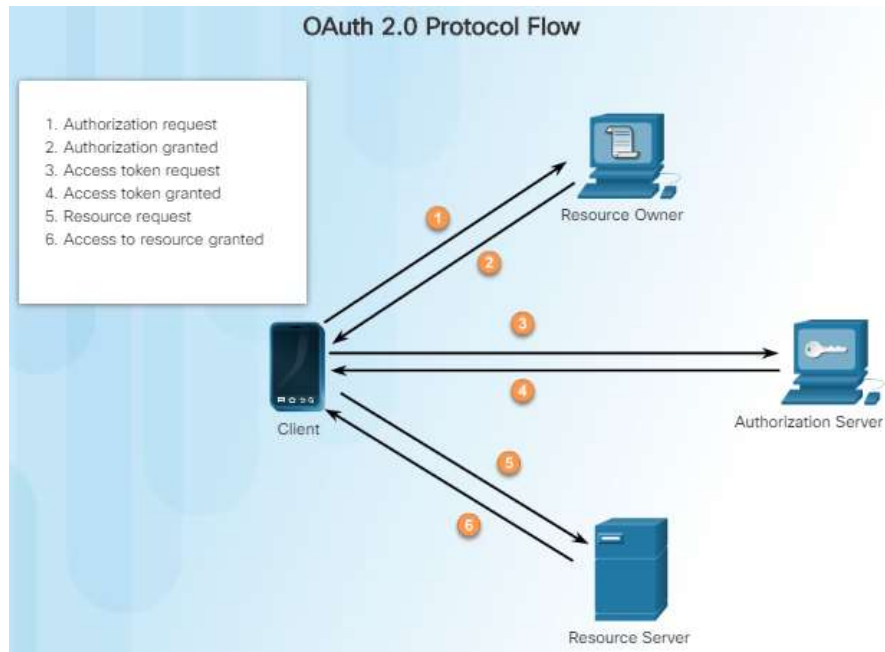
Network Access Control Concepts

OAuth 2.0 Authorization Framework



- Identity and access management (IAM) is the security principle that defines who can access what resources and the privileges they have when they obtain access.
- OAuth 2.0 Authorization Framework is a standardized protocol for internet-based authentication and authorization.
 - Used for access control of IoT devices to make them more secure by having a server handle the authorization of resources.
 - See next slide for details about OAuth 2.0 Authorization process.

OAuth 2.0 Authorization Framework (Cont.)



1. The client makes an authorization request to the resource owner.
2. The resource owner sends back an authorization grant to the client.
3. The client sends the authorization grant to the authorization server, requests an access token, and tries to authenticate.
4. If the authentication was successful, the authorization server validates the authorization grant and sends an access token back to the client.
5. The client sends the access token to the resource server to make a resource request.
6. After validating the access token, the resource server allows access to the requested resource.

IoT Device Identity Management



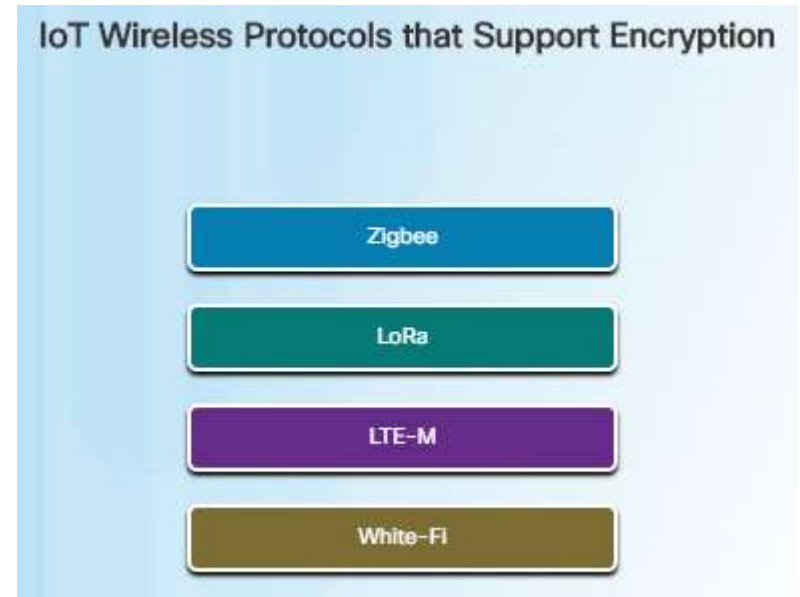
- Identity management also refers to the identification of these wide range of devices, as well as managing their access to data.
- There are a wide range of IoT devices that communicate with other devices. Not uncommon for sensitive information to be transmitted and the access to this data should be controlled.
- IoT device identity management should handle IoT device access to other information from other resources in addition to handling access to that device's resources.
- As the number of IoT devices continues to grow exponentially, the relationships between devices also grows exponentially.
- Identity Resource Management (IRM) - helps organizations manage a larger number of identities and relationships while keeping resources secure.

Encryption

Data and Password Security

- Encryption - mechanism that is used to ensure data confidentiality.
- Encrypting - applying an algorithm to data that will make it unreadable to those who are not authorized to see the information.
- Passwords should always be encrypted.
- Encryption for IoT data is critical because information being transmitted could contain sensitive information.
- IoT devices are vulnerable because many older IoT devices currently in production do not support encryption.

- IoT devices usually require wireless communication which makes it easier to intercept data transmissions if there is no encryption.



Encryption

Encryption in Constrained Systems



- Most IoT devices do not have the processing power or resources necessary for the more robust encryption algorithms.
- Lightweight encryption algorithms can be used.
 - These algorithms could be implemented in software or hardware.
 - Currently, there is no standard, and many IoT devices do not support encryption at all.
- The National Institute of Standards and Technology (NIST) has recently started the “lightweight cryptography initiative”. Its goal is to develop a standard cryptographic algorithm that can be used in small IoT devices with minimal resources.

Encryption

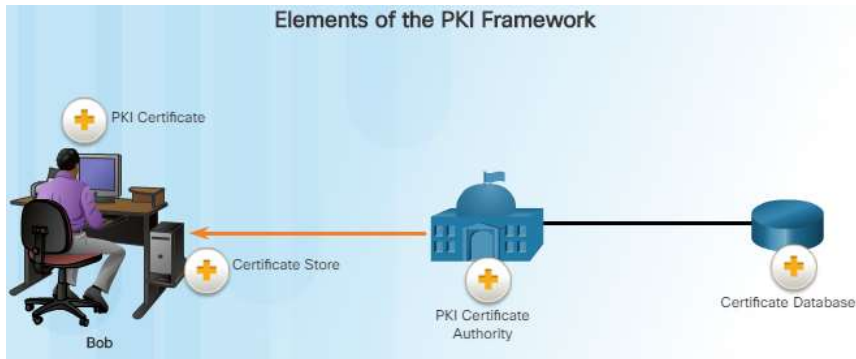
Public Key Cryptography



- **Cryptography** - based on the sender and receiver of a message knowing and using the same secret key.
 - Symmetric Cryptography - sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message.
 - Because all keys in a secret-key cryptosystem must remain secret, the challenge has been providing secure key management.
- **Public-key cryptography** was introduced in 1976 by Whitfield Diffie and Martin Hellman in order to solve the secure key management problem.
 - Each person gets a pair of keys: one called the public key and the other called the private key.
 - Each person's public key is published while the private key is kept secret.
 - With this system, anyone can send a confidential message by using public information (public key of the recipient), but the message can only be decrypted using the private key of the intended recipient.
 - Can be used not only for privacy (encryption), but for authentication (digital signatures).

Encryption

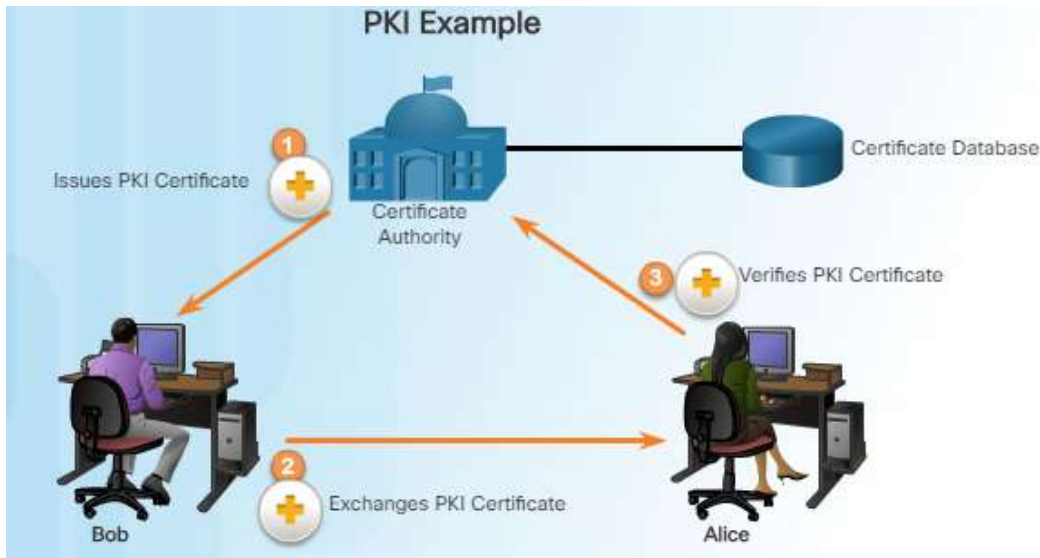
Authorities and the PKI Trust System



- Public Key Infrastructure (PKI) with its Certificate Authority (CA) is needed to support large-scale distribution and identification of public encryption keys. PKI is used to prove the identity of the IoT device.
 - PKI Certificate – Certificates contain an entity's or individual's public key.
 - Certificate Store – Resides on a local computer and stores issued certificates and private keys.
 - PKI Certificate Authority – Trusted third party that issues certificates. It signs these certificates using its private key.
 - Certificate Database – Stores all the certificates by the CA.

Encryption

Authorities and the PKI Trust System (Cont.)



- In the example, Bob has received his digital certificate from the CA. This certificate is used whenever Bob communicates with other parties.
- Bob communicates with Alice.
- When Alice receives Bob's digital certificate, she communicates with the trusted CA to validate Bob's identity.
- It is challenging using a PKI Authority with IoT devices. Certificate management with the large number of IoT devices is time-consuming and may be impossible to manage as more devices are added.

Chapter Summary

Summary

- IoT device hardware components
 - OWASP has compiled a list of vulnerabilities that should be addressed for each attack surface within the IoT system.
 - Where communication is available it is unlikely that encryption is implemented due to the limited processing power of constrained devices, particularly the Class 0 devices.
 - In IoT devices, the CPU, memory, and physical ports have the potential to be compromised by threat actors.
- IoT device software components.
 - Embedded systems may use an embedded operating system or be programmed directly using the machine code for the CPU.
 - Interpreted code is easy to modify because it is generally stored in a text format.
 - Even compiled code could be altered by an attacker using a debugger and replacing machine code instructions with malicious code.
 - If an attacker can gain access to the debug/boot mode it would be possible for them to make other changes to the system or even install a backdoor.

Chapter Summary

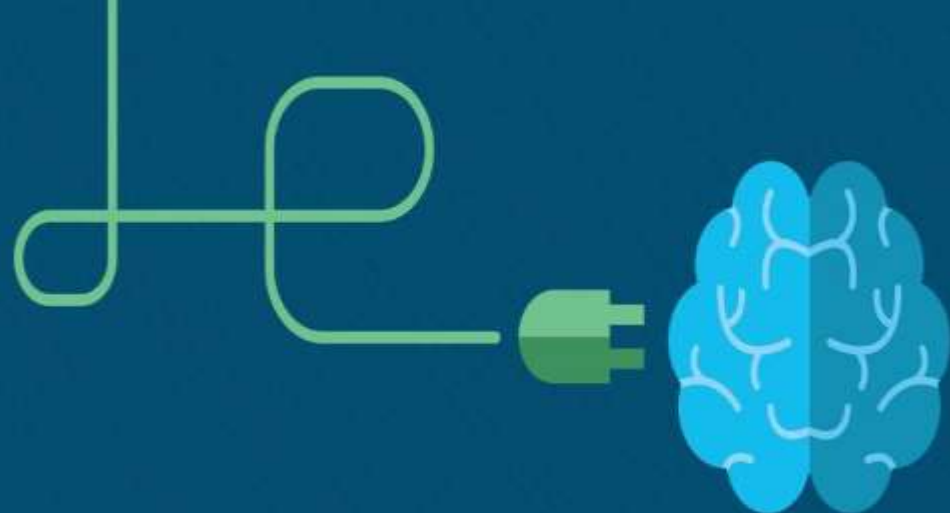
Summary (Cont.)

- Hardware Security
 - Constrained devices are placed in remote locations where physical security is difficult to implement.
 - Standard surveillance and security protocols should be implemented as a first layer of defense.
 - While most systems require some type of authentication when gaining access to the UART pins there are some devices that provide a command prompt with availability to numerous commands when connected.
- Firmware Vulnerabilities
 - IoT device firmware can contain security vulnerabilities that are discovered after their release.
 - In some cases, patches do not exist for security vulnerabilities for devices. In other cases, the device may not even be updatable or patchable.
 - Firmware should be updated as soon as new releases come out because it is likely these updates are addressing security vulnerabilities.
 - If the attacker can obtain physical access to a device, can connect to the IoT device using either the JTAG or UART serial interfaces to gain unauthorized access and hack into the device.

Chapter Summary

Summary (Cont.)

- Network access control concepts
 - A security analyst should understand the different basic access control models to have a better understanding of how attackers can break the access controls.
 - The OAuth 2.0 Authorization Framework can be used for access control of IoT devices to make them more secure by having an authorization server handle the authorization of resources.
 - IoT device identity management should handle IoT device access to other information from other resources in addition to handling access to that device's resources.
- Encryption
 - Encryption for IoT data is critical because some of the information being transmitted could be sensitive.
 - Most IoT devices do not have the processing power or resources necessary for the more robust encryption algorithms.
 - Implementing public key cryptography into IoT devices is the recommended method to ensure IoT device security.
 - PKI is used to prove the identity of the IoT device.



Chapter 4: IoT Communication Layer Attack Surface

IoT Security 1.0 v2.0



Chapter 4 - Sections & Objectives

- 4.1 Determine vulnerabilities of the IoT communication layer.
 - Explain the functions of the IoT network communication layer.
 - Determine vulnerabilities in IoT wireless network protocols.
- 4.2 Determine vulnerabilities in TCP/IP that impact IoT systems.
 - Determine vulnerabilities in IP that impact IoT systems.
 - Determine vulnerabilities in TCP and UDP that impact IoT systems.
- 4.3 Propose measures to mitigate threats at the IoT network layer.
 - Implement access control in IoT networks.

4.1 Functions of the IoT Communication Layer

OWASP Communication Layer Vulnerabilities

- Communication layer of the IoT is responsible for the transportation of data between devices, facilities, and applications.
 - Often occurs in the cloud.
 - Network security needs to be considered for all elements of the IoT system's attack surface.
 - Data in motion can be intercepted, damaged, or altered.
 - Because the purpose of much of the IoT is data collection, attacks on the systems that carry data can bring down an entire IoT system.

OWASP Communication Layer Vulnerabilities	
Attack Surface	Vulnerability
Device Network Services	Information disclosure Injection Denial of Service Unencrypted Services Poorly implemented encryption Test/Development Services Vulnerable UDP Services DoS Replay attack Lack of payload verification Lack of message integrity check
Network Traffic	LAN traffic LAN to Internet traffic Short range Non-standard protocols Wireless (Wi-Fi, Z-wave, XBee, Zigbee, Bluetooth, LoRA) Packet manipulation (protocol fuzzing)

Functions of the IoT Communication Layer

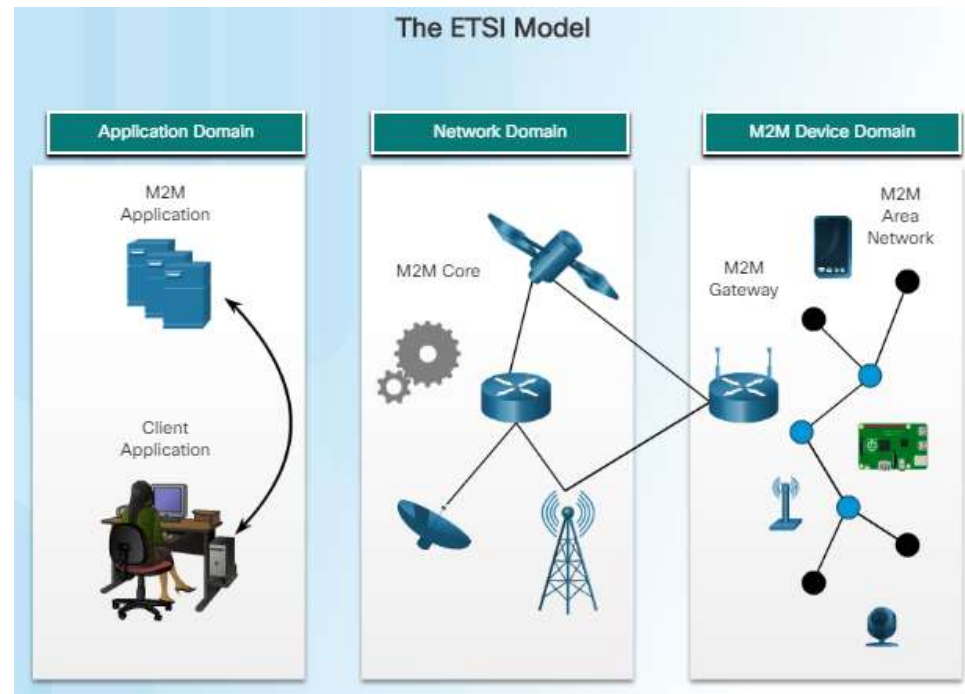
Communication Channels

- The figure shows some of the various types of wireless networks that exist in IoT.
- Each network may use different protocols to communicate between IoT devices, between IoT devices and gateways, and between IoT devices and the internet or the cloud.

Wireless Network	Use Case
WBAN: Wireless Body Area Network	A network of wireless sensor devices that are either worn or implanted into the body. May use various wireless protocols to communicate with a gateway to post data to cloud applications.
WPAN: Wireless Personal Area Network	Frequently employs Bluetooth to connect audio devices, personal fitness trackers, and smart watches to a cell phone that serves as a gateway.
WHAN: Wireless Home Area Network	Uses Bluetooth or other wireless protocols to connect appliances, alarm system components, and actuators to gateways and the Internet.
WFAN: Wireless Field (or Factory) Area Network	Ruggedized network components connect sensors and actuators at dispersed locations in challenging manufacturing environments.
WNAN: Wireless Neighborhood Area Network	A power grid network that exists in a limited geographic area and is frequently served by a field area router that may be located outdoors.

Functions of the IoT Communication Layer Communication Channels (Cont.)

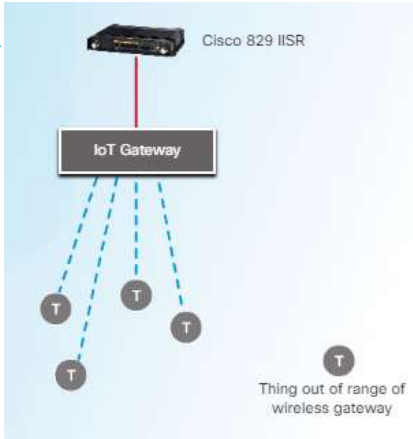
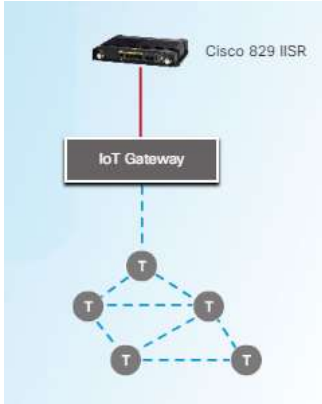
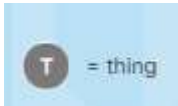
- Communication channels exist between low power devices and a gateway device.
 - Gateway translates wireless sensor network traffic to IP protocol traffic that can travel on data networks.
- Due to power constraints, nodes may only use very short-range radios.
 - Protocols are used that allow sensor data to travel from node to node until the data reaches the gateway.
- Communication channels, and the protocols that enable them, make up the IoT data communication attack surface.



Functions of the IoT Communication Layer

IoT Communication Scenarios

- IoT wireless protocols operate in several different topologies.
 - Mesh topology - smart objects forward data to other smart objects in order to reach a gateway (which may be out of range).
 - Enables sensor nodes and smart objects to be deployed over a larger area than would be possible if each node were required to communicate directly with a gateway.
 - Hub-and-spoke (star) topology - things must be deployed within range of the IoT gateway.
 - Limits the coverage area of the network and requires the purchase, installation, and configuration of additional gateway devices and links to the WAN or internet.
 - Gateway converts the traffic to Wi-Fi or Ethernet and encapsulates the data in IP packets for transmission



Functions of the IoT Communication Layer

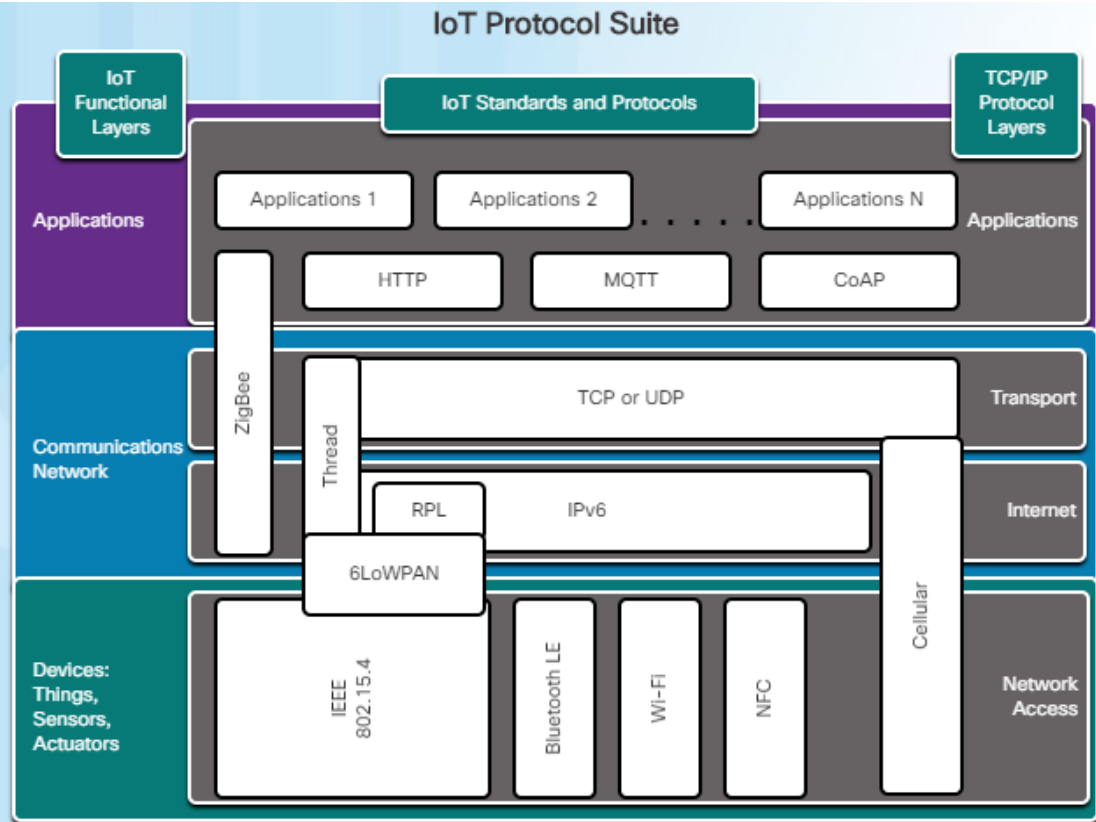
IoT Communication Scenarios (Cont.)

- This figure shows a scenario in which things can communicate directly with the cloud or data center. In this topology, each thing will have its own unique IPv6 address.
 - Things have their own IPv6 protocol stacks and messaging protocols.
 - Allows the sensor data to be sent through the IP network without requiring translation into IP by an IoT gateway.
 - Most things are not using Wi-Fi due to power and processing constraints so the gateway will convert the traffic to the appropriate Layer 2 encapsulation and the Layer 3 encapsulation will most likely be unaltered.



Wireless Protocols

Wireless Protocol Overview



- IoT Protocol Suite includes the following wireless protocols:
 - IEEE 802.15.4 - Standard for low-rate wireless personal area networks that is meant to be used by low-cost, low-speed devices. (ZigBee, Thread, and 6LoWPAN)
 - Bluetooth Low Energy (BLE) – Personal area network (WPAN) that uses the 2.4 GHz radio frequency.
 - Wi-Fi - Collection of IEEE 802.11 standards for WLANs that operate in the 2.4 GHz and 5 GHz frequencies.
 - Near Field Communication (NFC) – Protocols for device-to-device communications within 4 cm or 1.6 inches.
 - Cellular – Includes all the cellular technologies (3GPP, 4G, LTE, and 5G).

Wireless Protocols

Bluetooth and Wi-Fi

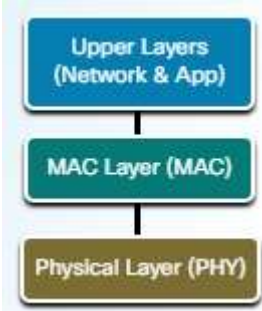
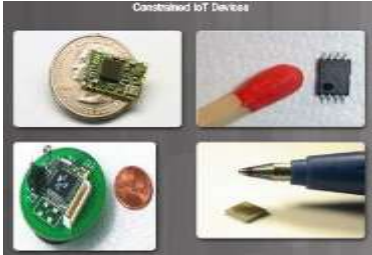


- Most common wireless protocols and found in connected homes, home automation and security applications.
 - Many vendors create devices that are easy to set up to help increase sales but sacrifices security.
 - When a new IoT device, such as a smart plug, is first installed, how does the consumer connect to it?
 - If designed to be controlled over the internet, it must connect to the home Wi-Fi network, even though it does not know the SSID.
 - Some of these devices will actually set up hot spots within the house that can be identified by the vendor's cell phone app or PC software.
 - Enables the buyer to connect to the device without having to configure it for the home network.
 - These minimally secured Wi-Fi hotspots are in operation but forgotten by the homeowner.
 - Combined with well-known login credentials, vulnerable to control by unauthorized users.
 - Poorly secured devices may provide a path into the rest of the network for threat actors and malware.

Wireless Protocols

IEEE 802.15.4 Overview

- The computational operating power constraints of many IoT things required new wireless protocols to be developed to enable things to communicate on networks.
- IEEE 802.15.4 protocol
 - Originally developed for use in personal area networks.
 - Popular in a wide range of applications.
 - 802.15.4 consists of media access layer (MAC) and physical (PHY) specifications.
 - Uses layered architecture, which allows developers to create upper-layer protocols on the same foundation.
 - ZigBee, Thread, and 6LoWPAN all run on top of 802.15.4



- Upper layers not specified. Various implementations exist.
- Medium Access Control Sub-Layer (MAC)
 - Responsible for reliable Layer 2 communication between two devices
 - Data framing and validation of RX frames
 - Device addressing
 - Channel access management
 - Device association/disassociation
- Physical Layer (PHY)
 - Provides bit stream air transmission
 - Activation/deactivation of radio transceiver
 - Carrier sensing (CSMA/CA)
 - Received signal strength indication (RSSI)
 - Link Quality Indicator (LQI)
 - Data coding and modulation, error correction

IEEE 802.15.4 Device Roles

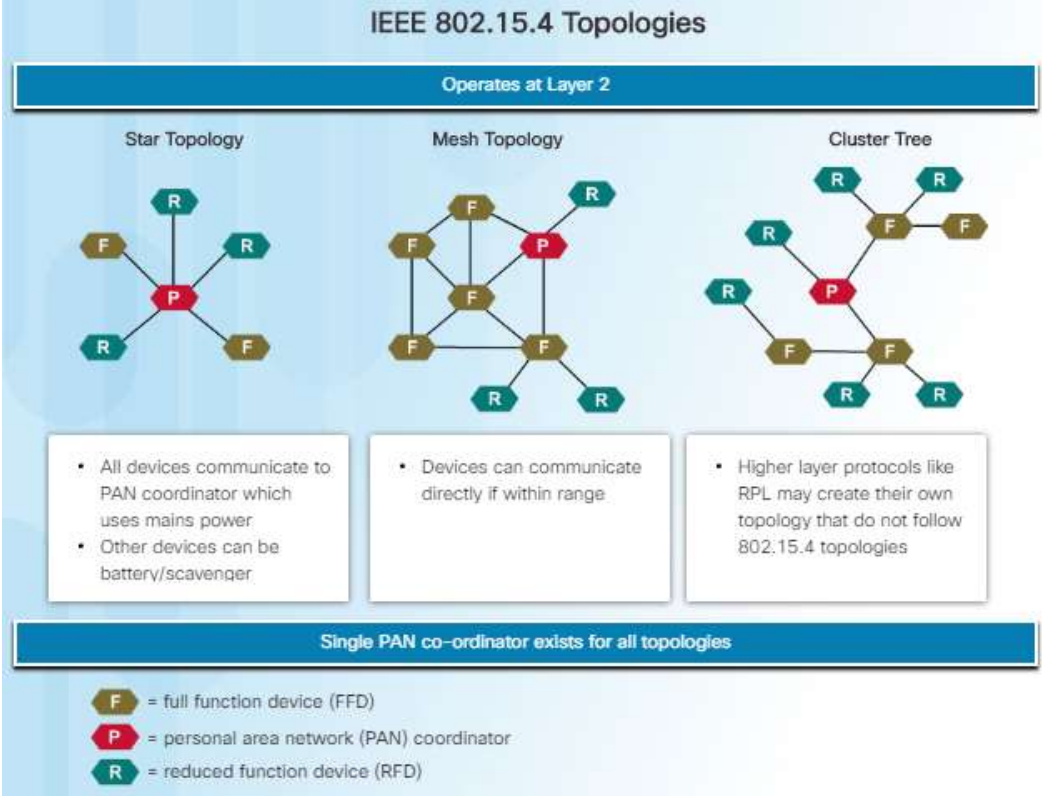
IEEE 802.15.4 Node Types

- **Full Function Device (FFD)**
 - Can operate as a PAN coordinator (allocates local addresses, gateway to other PANs)
 - Can communicate with any other device (FFD or RFD)
 - Ability to relay messages (PAN coordinator)
 - One PAN coordinator per topology
- **Reduced Function Device (RFD)**
 - Very simple device, modest resource requirements
 - Can only communicate with FFD
 - Intended for extremely simple applications

F = full function device (FFD)
P = personal area network (PAN) coordinator
R = reduced function device (RFD)

Wireless Protocols

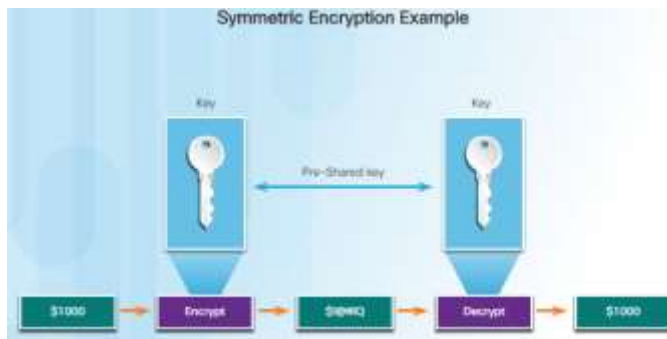
IEEE 802.15.4 Topologies



- **Star Topology** - Communication is established between devices and a single central controller, called the PAN coordinator. Each star network chooses a unique PAN identifier which allows each star network to operate independently.
- **Mesh Topology** - Also one PAN coordinator. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking would benefit from such a topology. RFDs must be able to connect to an FFD or PAN.
- **Cluster Tree Topology** - Special case of a mesh network in which most devices are FFDs. An RFD may connect to a cluster-tree network as a leaf node at the end of a branch.

Wireless Protocols

IEEE 802.15.4 Security



- Because 802.15.4 operates at the OSI physical and data link layers security of the frames is important.
- Four basic security services performed at the data link layer:
 - Access control – Prevents unauthorized devices from joining the network.
 - Message integrity – Protects against alteration of data while it is in transit by using an encrypted cryptographic key (message authentication code).
 - Message confidentiality – Prevents threat actors from reading the transmitted data. Message data payloads are encrypted to protect the confidentiality of the message.
 - Replay protection - Legitimate messages can be captured and sent out on the network at a later time. Because the messages are authenticated, they may be accepted by the relevant hosts. If these messages are replayed frequently, network performance can be degraded to the extent that legitimate data cannot reach the gateway.
- 802.15.4 uses symmetric key cyphers for encryption. Symmetric keys are less secure then asymmetric, or public key, cryptography.

Wireless Protocols

IEEE 802.15.4 Security (Cont.)

IEEE 802.15.4 Security Suites

	Confidentiality: Frame Encryption	Authenticity: Frame Integrity
Null	✗	✗
AES-CBC-MAC-32	✗	✓
AES-CBC-MAC-64	✗	✓
AES-CBC-MAC-128	✗	✓
AES-CTR	✓	✗
AES-CCM-32	✓	✓
AES-CCM-64	✓	✓
AES-CCM-128	✓	✓

AES-CBC-MAC = Uses message authenticity code at end of data payload
AES-CTR = Uses 128-bit AES to encrypt payload
AES-CCM = Uses both AES-CBC-CCM and AES-CTR

- 802.15.4 protocol provides security functionality in the form of security suites that can be specified by the overlaying application layers.
- Each security suite offers different encryption and authentication schemes with different key lengths.

Wireless Protocols

Mesh Protocols that use 802.15.4

- 6LowPan – Provides IPv6 services to low power devices in PANs. Can be added to other protocol stacks such as Zigbee and Thread.
- Zigbee – Simple and inexpensive group of upper-level wireless communication protocols that implement small lower-power PANs. Widely implemented in home automation, medical device data collection, and various other applications. Adds an optional security layer, a network layer including routing, and an application layer to the IEEE 802.15.4 PHY and MAC layers.
- Thread – Wireless mesh protocol built specifically for IoT home networks that uses IPv6 for addressing and 6LoWPAN as the foundation.
- WirelessHART – This is an international wireless specification for the Industrial Internet of Things (IIoT). It is built on an 802.15.4 mesh network.
- ISA 100.11a – U.S. standard for communication on the IoT.



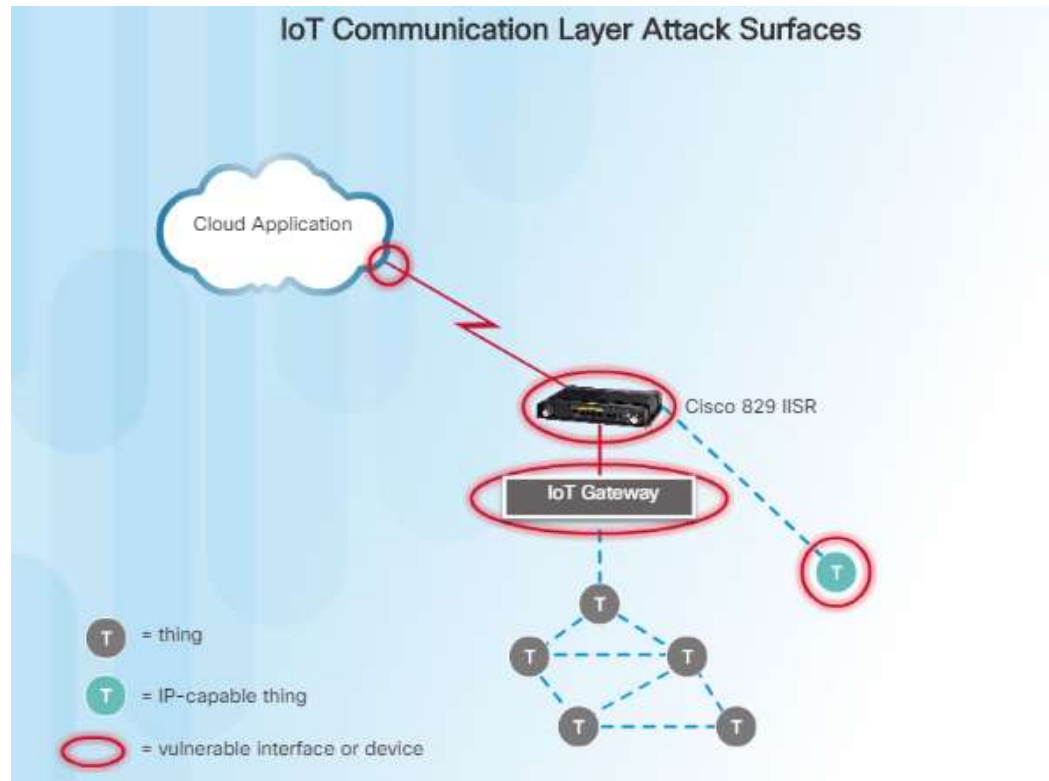
Other Wireless Options

- Other wireless protocols have been developed that support low power wide area networks (LPWAN).
- LoRa is one of a number LPWAN technologies and has become popular due to its low cost and wide implementation.
- LoRaWAN - The Things Network is an international organization which enables development of IoT proof-of-concept systems using the LoRa radio physical layer and LoRaWAN data link, and network layer elements of the protocol stack. The Things Network encourages individuals to create and maintain their own LoRaWAN gateways.
- Cellular - Cellular data standards, known as 3GPP, are implemented to extend IoT networks, but with devices that have a fixed power supply. Current cellular data services are not well suited to IoT applications due to power constraints. The fifth generation (5G) cellular specifications include LTE Advanced for Machine-Type Communication (LTE MTC). This technology includes features that greatly improve power consumption while providing simplified device capability for small periodic data transmission. Narrowband IoT (NB-IoT) is a low-power low-bandwidth protocol especially for indoor applications. It uses a portion of a wireless LTE carrier's frequency spectrum.

4.2 IP Vulnerabilities

IoT Communication Layer Vulnerabilities

- IP security is a serious concern since the main purpose of IoT is to transmit, store and analyze data over IP networks.
- IoT attack surface that will have IP vulnerabilities, are shown in the figure:
 - Sensor network
 - IoT gateway
 - Enterprise IT network (Cisco 829 router in the figure)
 - Uplink to the internet



Common IP Vulnerabilities

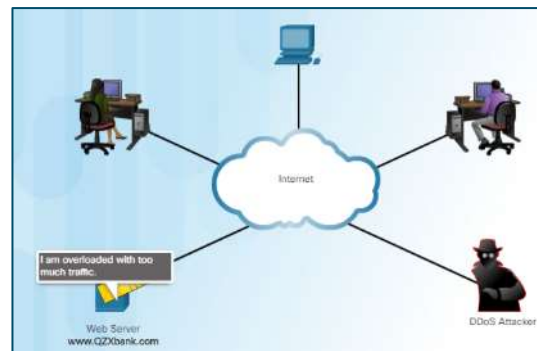
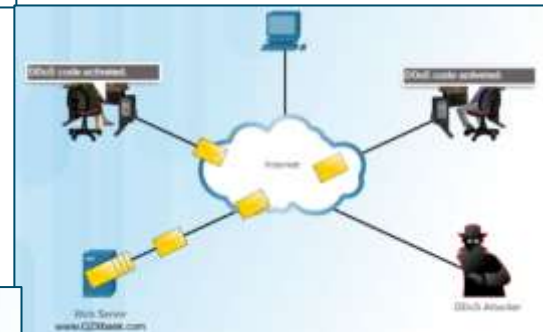
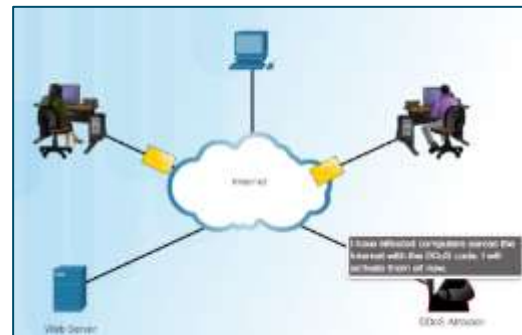
These are some of the more common IP-related attacks:

- **DoS attacks** - Threat actors attempt to prevent legitimate users from accessing information or services.
- **DDoS attacks** – This attack is similar to a DoS attack, but features a simultaneous, coordinated attack from multiple source machines.
- **ICMP attacks** - Threat actors use Internet Control Message Protocol (ICMP) echo packets (pings) to discover subnets and hosts on a protected network, to generate DoS flood attacks, and to alter host routing tables.
- **Address spoofing attacks** - The threat actor puts the source IP address in a packet to masquerade as a different source, tricking the destination into believing the packet came from a legitimate source.
- **Man-in-the-middle attack (MITM)** - Threat actors position themselves between a source and destination to transparently monitor, capture, and control the communication. They could simply eavesdrop by inspecting captured packets or alter packets and forward them to their original destination.
- **Session hijacking** - Threat actors gain access to the physical network, and then use an MITM attack to sniff a valid token for access to a web server.

IP Vulnerabilities

DoS Attacks

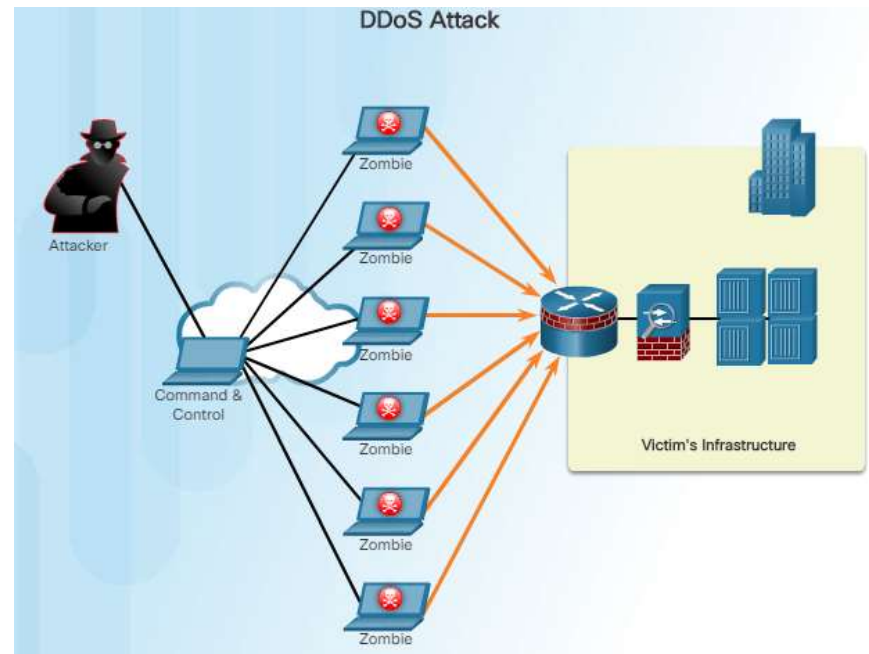
- The goal is to prevent legitimate users from gaining access to services.
- There are two major sources of DoS attacks:
 - **Maliciously Formatted Packets** – Threat actors craft a maliciously formatted packet and forward it to a susceptible host, causing it to crash or become extremely slow.
 - **Overwhelming Quantity of Traffic** – Threat actors overwhelm a target network, host, or application, causing it to crash or become extremely slow.
- DDoS attack is larger because it originates from multiple sources.



DoS Attacks (Cont.)

- This scenario is similar to that used by the Mirai Botnet attack. This attack could proceed as follows:

1. The threat actor (botmaster) builds or purchases the use of a botnet of zombie hosts.
2. Zombie computers continue to scan and infect more targets to create more zombies.
3. When ready, the botmaster uses the handler systems to make the botnet of zombies carry out the DDoS attack on the chosen target.

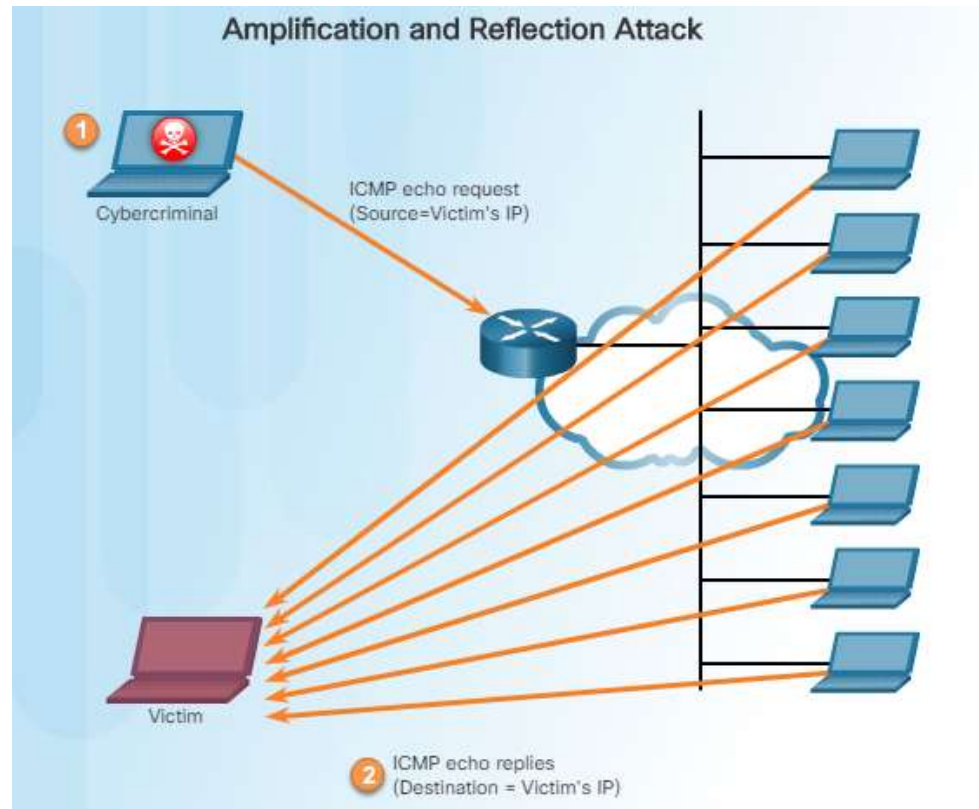


Amplification and Reflection Attacks

- The example in the figure illustrates how an amplification and reflection technique called a Smurf attack is used to overwhelm a target host:

1. Amplification - The threat actor forwards ICMP echo request messages that contain the source IP address of the victim to a large number of hosts.

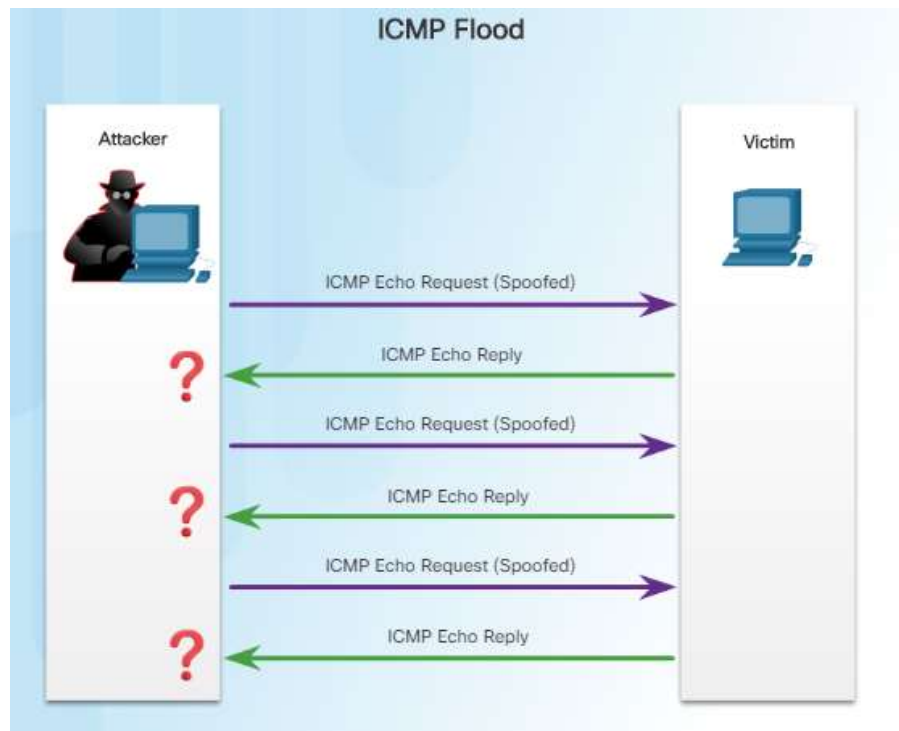
2. Reflection - The hosts all reply to the spoofed IP address of the victim to overwhelm it.



IP Vulnerabilities

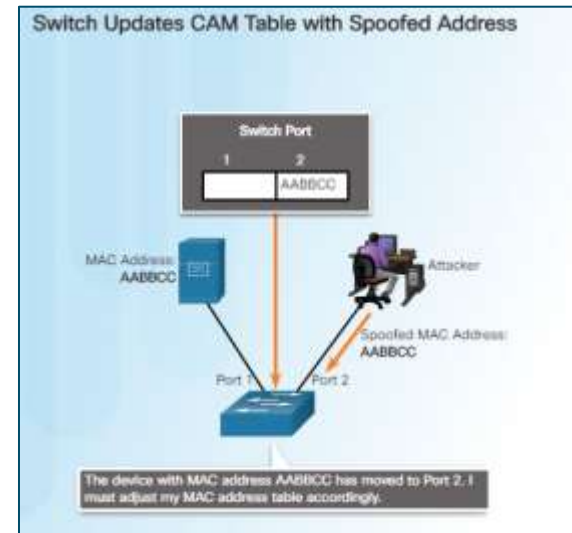
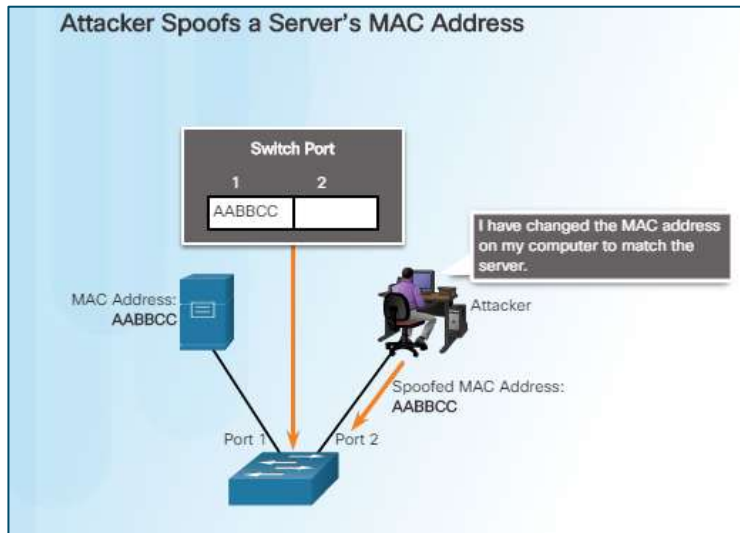
ICMP Attacks

- Internet Control Message Protocol (ICMP)
 - Developed to carry diagnostic messages and to report error conditions when routes, hosts, and ports are unavailable.
 - Ping command is a user-generated ICMP message that is called an echo request which is used to verify connectivity to a destination.
- Threat actors use ICMP for reconnaissance and scanning attacks.
- Threat actors often use ICMP to create DoS attacks.



Address Spoofing Attacks

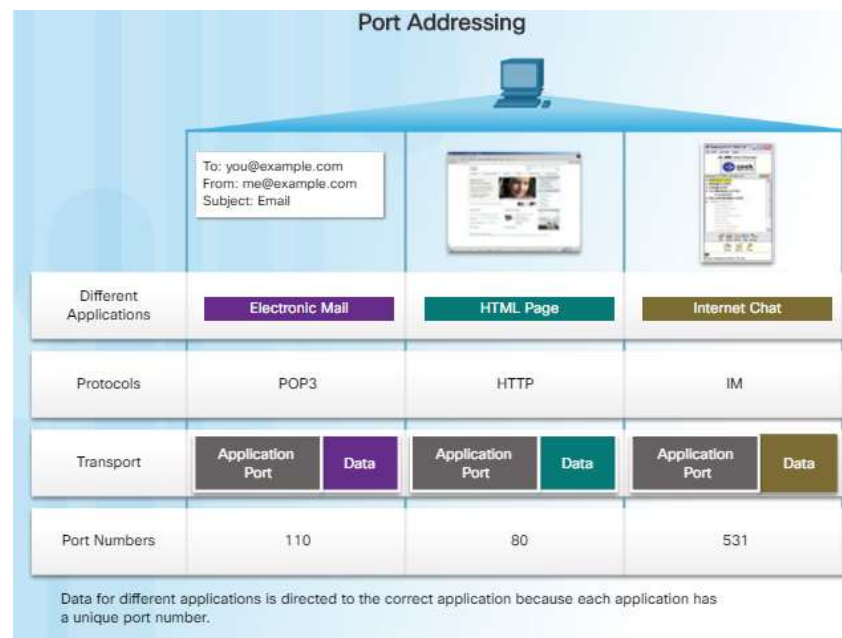
- IP address spoofing attacks occur when a threat actor creates packets with false source IP address information.
- Media Access Control (MAC) address spoofing attacks are used when threat actors have access to the internal network.



TCP and UDP Vulnerabilities

TCP Vulnerabilities

- TCP provides the following services:
 - **Reliable delivery** - TCP incorporates acknowledgments to guarantee deliver. If a timely acknowledgment is not received, the sender retransmits the data.
 - **Flow control** - TCP implements flow control to address delays. Rather than acknowledge one segment at a time, multiple segments can be acknowledged with a single acknowledgment segment.
 - **Stateful communication** - Before data can be transferred using TCP, a three-way handshake opens the TCP connection. If both sides agree to the TCP connection, data can be sent and received.



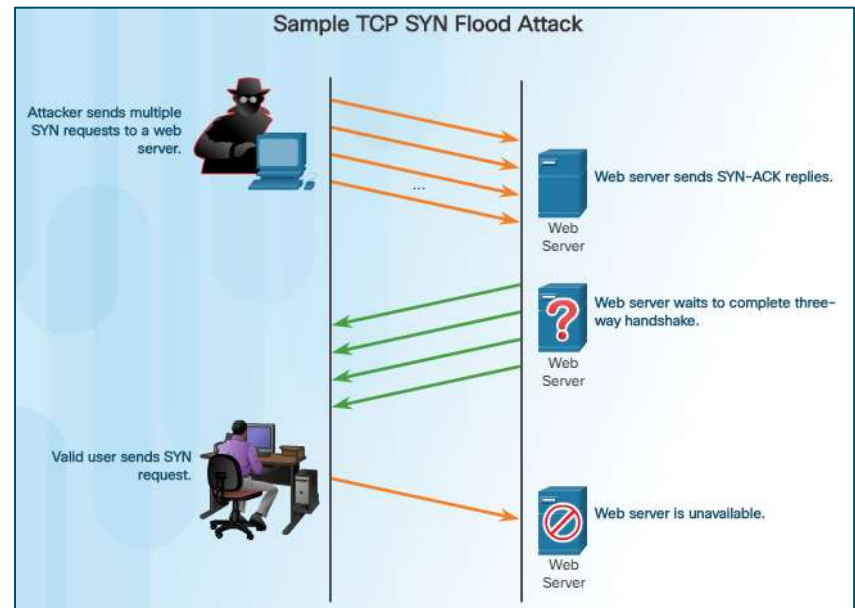
TCP Vulnerabilities (Cont.)

- Protocols at the Transport Layer of the OSI model use port addressing to enable multiple conversations to be tracked and connected with the correct applications.
 - Well-known port numbers identify commonly used applications.
 - An application such as Telnet can be assigned any port number within the range of open ports. Because Telnet is not secure, it should not be left running on IoT devices.
 - It is important that any smart object be evaluated regarding which communication protocols are enabled on it by default and which listening ports are open.
- TCP protocol is vulnerable to port scanning.
 - Threat actors conduct port scans of target devices to discover which services they offer.
 - Port scanners can supply very detailed information about the services that are running on a networked device. These services can be vulnerable to exploitation by threat actors.

TCP and UDP Vulnerabilities

TCP SYN Flood Attack

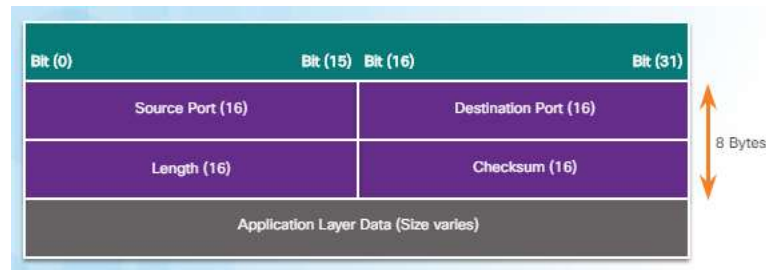
- TCP SYN Flood attack exploits the TCP three-way handshake.
- A TCP connection can be torn down when it receives an RST bit. This is an abrupt way to tear down the TCP connection and inform the receiving host to immediately stop using the TCP connection. A threat actor could launch a TCP reset attack and send a spoofed packet containing a TCP RST to one or both endpoints.
- TCP session hijacking is another TCP vulnerability. Although difficult to conduct, it enables a threat actor to overtake an already-authenticated host as it communicates with the target.



TCP and UDP Vulnerabilities

UDP Vulnerabilities

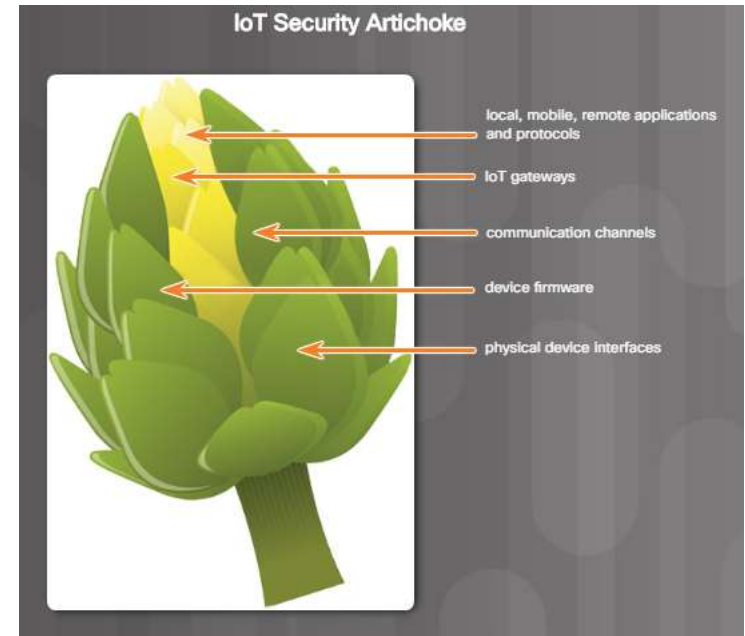
- UDP is a connectionless protocol. Used by DNS, TFTP, NFS, SNMP, media streaming, VoIP, and more. Lower overhead than TCP because it does not offer the retransmission, sequencing, and flow control mechanisms that provide reliability.
- UDP is not protected by any encryption. Possible to add encryption to but it is not available by default.
- Changing the data in the traffic will alter the 16-bit UDP checksum, which is mandatory when carried over IPv6.
- The threat actor can create a new checksum based on the new data payload and record it in the header as a new checksum. The destination device will find that the checksum matches the data without knowing the data has been altered.
- More common to see a UDP attack where all of the networked resources are consumed - UDP flood attack.



4.3 IoT Communication Security

Security for IoT Communication Protocols

- IoT devices must be secure physically and protected from damage, destruction, and tampering.
- Physical device firmware and interfaces must not be vulnerable.
- First stage in creating a threat model of IoT system security is to identify the component technologies and their features, like removing petals from the security artichoke.
- IoT communication must also be secure.
- Devices that join IoT sensor and actuator networks must use robust wireless protocols.
- Cryptographic approaches that protect credentials, encrypt data payloads, and create secure channels for protocol conversations are also an important part of IoT security.



IoT Communication Security

Isolation of IT and OT Traffic

- It is important to reduce the size of the overall attack surface by establishing smaller zones of trust through the use of firewalls and other security technologies.
- This helps to ensure that a successful attack on one part of the network does not permit access to all parts of the network.
- Data from multiple areas of the network may need to pass through these zones of trust. Therefore, traffic isolation techniques, such as using VLANs, is important.



IoT Communication Security

Case Study: No Isolation of IT and OT Traffic

- A major retailer discovered a security breach in which personal information, including credit card numbers, was stolen from over 50 million customers.
 - The breach occurred due to weak security at a contracting heating, ventilation, and cooling (HVAC) company.
 - Security vulnerabilities in the contractor network enabled threat actors to infect the retailer point-of-sale networks with malware.
- This case illustrates two important points:
 - Device, network, and application security are only as strong as the weakest link, which in this case was the security of the contractor who had access to the network.
 - The contractor should not have had access to the same network that carried confidential data. In this case, the contractor network became a part of the retailer's attack surface.



- At the TCP/IP protocol level of communication, only secure application layer protocols should be used.
- Transport security, in the form of TLS or Datagram Transport Layer Security (DTLS) should be implemented to authenticate and protect IoT data.

Threat Model for IoT Communication Technologies

I3: Insecure Network Services	I4: Lack of Transport Encryption
<ul style="list-style-type: none">• Ensure all devices operate with a minimal number of network ports active.• Ensure all devices do not make network ports and/or services available to the Internet.• Review all required network services for vulnerabilities such as buffer overflows or denial of service.	<ul style="list-style-type: none">• Ensure all communication between system components is encrypted, as well as encrypting traffic between the system or device and the Internet.• Use recommended and accepted encryption practices and avoid proprietary protocols.• Ensure SSL/TLS implementations are up to date and properly configured.• Consider making a firewall option available for the product.

- OWASP recommends guidelines to mitigate vulnerabilities for the communication channels in IoT systems as shown in figure.
- A threat model of these systems requires a thorough understanding of data flows on the network and the locations of trust boundaries relative to the flows.
- Trust boundaries are areas in the communication system where data could come from an untrusted source.
- Creating a threat model includes using the STRIDE model to identify the threats and the DREAD model to score the risk of each threat.

IoT Communication Security

IoT Communications Checklist



- Verify 802.15.4 security:
 - AES-CTR mode is not used.
 - Both encryption and authentication are used.
 - Acknowledgments are not relied upon in this channel.
 - Integrity of key management.
 - ACLs are maintained in low-power mode.
 - The latest version of the protocol is in use.
- Verify TCP/IP Communication security:
 - All data flows are thoroughly and accurately documented.
 - Data flow is adequately protected at trust boundaries.
 - Firewalls or other systems segregate traffic in different trust zones.
 - Transport encryption is utilized.

Chapter Summary

Summary

- Functions of the IoT communication layer:
 - Responsible for the transportation of data between devices, facilities and applications, often in the cloud.
 - Many IoT sensor nodes are constrained in terms of resources, power, and processing.
 - Therefore communication channels will exist between low power devices and a gateway device.
- Wireless protocols:
 - The IoT Protocol Suite includes IEEE 802.15.4, BLE, Wi-Fi, NFC, and cellular.
 - Bluetooth and Wi-Fi are easy to set up, but this often sacrifices security.
 - ZigBee, Thread, WirelessHART, ISA 100.11a, and 6LoWPAN all run on top of 802.15.4 devices which can operate as an FFD, PAN coordinator, or RFD.
 - The 802.15.4 protocol provides security functionality in the form of security suites that can be specified by the overlaying application layers
 - LoRa is one of a number of LPWAN technologies and has become popular due to its low cost and wide implementation.
 - Current cellular data services are not well suited to IoT applications due to power constraints.

Chapter Summary

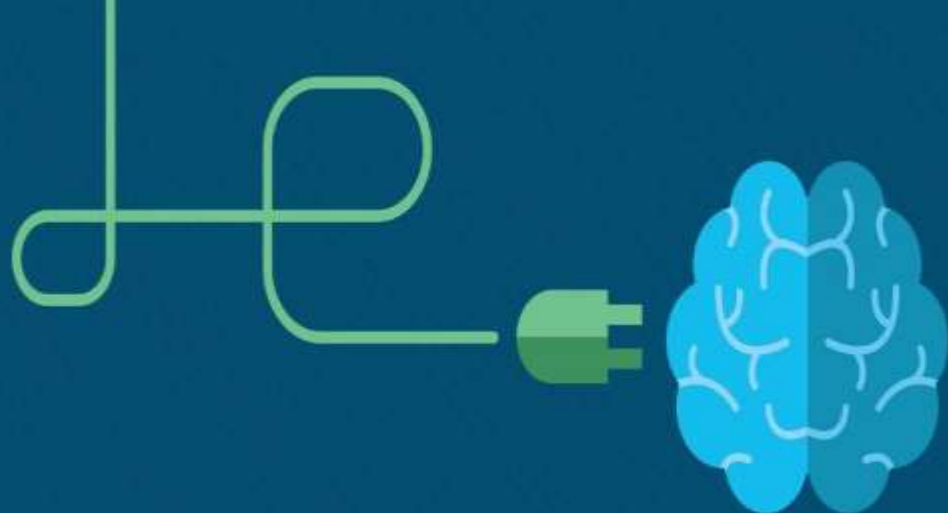
Summary (Cont.)

- IP vulnerabilities:
 - IP vulnerabilities include the sensor network, the IoT gateway, the enterprise IT network, and the uplink to the internet.
 - Attacks targeting IP including DoS, DDoS, ICMP, address spoofing, MITM, and session hijacking.
 - Threat actors often use amplification and reflection techniques to create DoS attacks.
 - Threat actors use ICMP for reconnaissance and scanning attacks.
 - Spoofing attacks can be conducted as either blind or non-blind.
- TCP and UDP vulnerabilities:
 - The TCP protocol is vulnerable to port scanning.
 - TCP attacks include the TCP SYN flood, TCP reset, and TCP session hijacking.
 - It is possible to add encryption to UDP, but it is not available by default.

Chapter Summary

Summary (Cont.)

- IoT communication security:
 - Cryptographic approaches that protect credentials, encrypt data payloads, and create secure channels for protocol conversations.
 - Reduce the size of the overall attack surface by establishing smaller zones of trust through the use of firewalls and other security technologies.
 - At the TCP/IP protocol level of communication, only secure application layer protocols should be used.
 - Creating a threat model includes using the STRIDE model to identify the threats and the DREAD model to score the risk of each threat.



Chapter 5: IoT Application Layer Attack Surface

IoT Security 1.0 v2.0



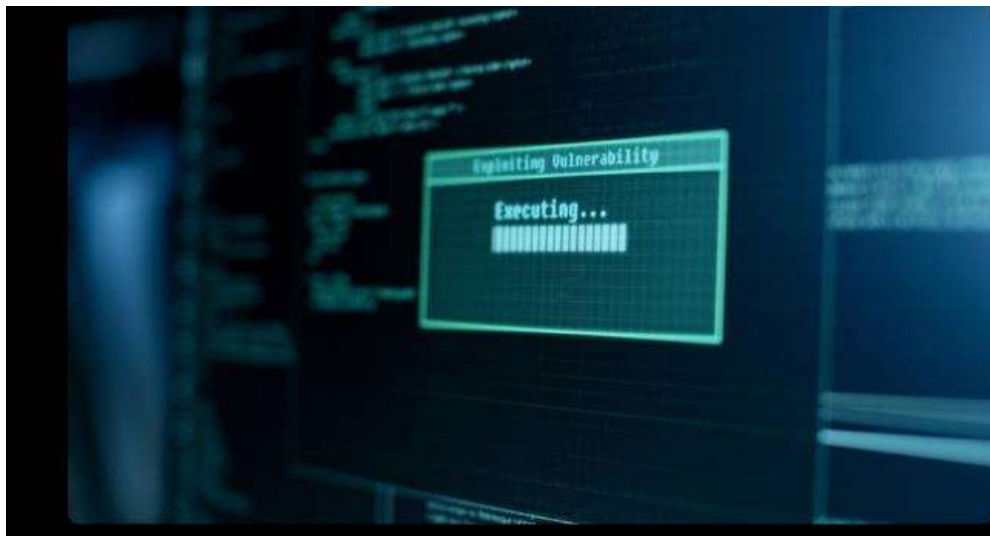
Chapter 5 - Sections & Objectives

- 5.1 Perform vulnerability assessment activities of IoT applications and protocols.
 - Perform vulnerability assessment activities of the IoT local applications.
 - Perform vulnerability assessment of IoT remote applications.
 - Perform vulnerability assessment of IoT application messaging protocols.

- 5.2 Recommend measures to mitigate threats to IoT applications.
 - Recommend measures to mitigate threats to IoT messaging protocols.

5.1 IoT Local Application Vulnerabilities

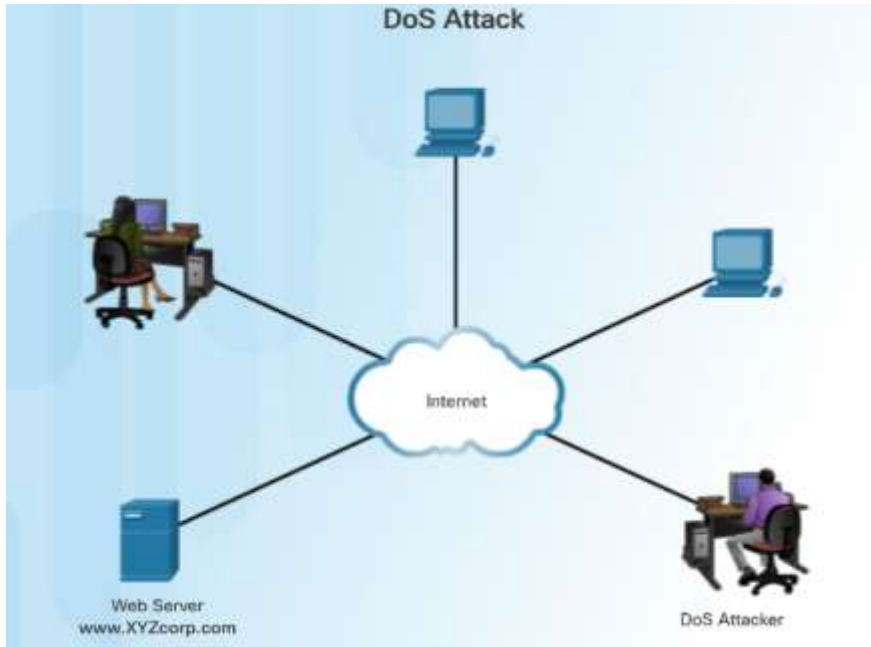
OWASP Application Vulnerabilities



- The most widely exposed IoT vulnerabilities listed by the Open Web Applications Security Project (OWASP):
 - **Username enumeration** – The threat actor is able to find valid usernames by interacting with the authentication mechanism of the application.
 - **Weak passwords** – The threat actor uses default passwords which have not been changed.
 - **Account lockout** – The threat actor attempts to authenticate which causes the account to be locked.
 - **Lack of multi-factor authentication**
 - **Insecure 3rd party components** – As vulnerabilities are discovered, patches must be installed.

IoT Local Application Vulnerabilities

Local Applications



- Some of the most popular local exploits:
 - **Firmware Replacement**
 - **Cloning**
 - **Denial of Service (DoS)**
 - **Extraction of Security Parameters**
- Some of the most popular remote exploits:
 - **Man-In-the-Middle Attack (MITM)** – The threat actor gets between devices in the system and intercepts data.
 - **Eavesdropping Attack** – When devices are being installed, the threat actor can intercept data such as security keys that are used by constrained devices.
 - **SQL Injection (SQLi)** – The threat actor uses a flaw in the Structured Query Language (SQL) application that allows access to modify data or gain administrative privileges.
 - **Routing Attack** – A threat actor could place a rogue routing device on the network.

IoT Local Application Vulnerabilities

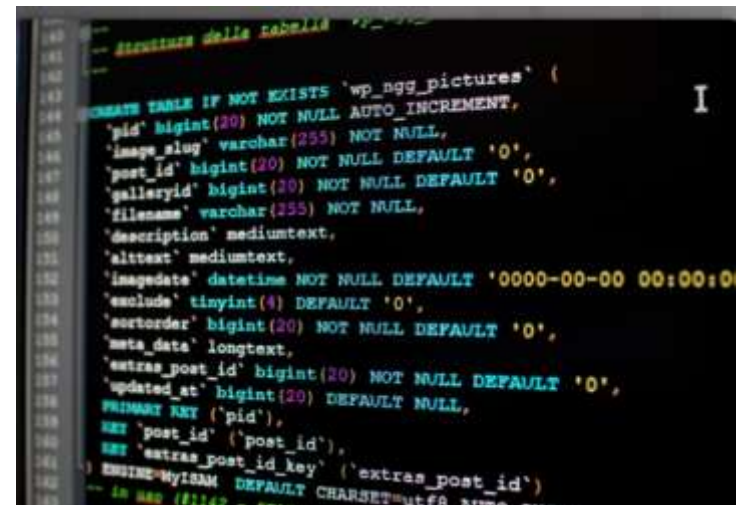
Mobile Applications

- Some of the most widely exposed mobile device vulnerabilities:
 - **Insecure communication** – The communication technology and channel must be secured.
 - **Insecure data storage** – Many applications have access to data storage areas of mobile devices, even though they may not need it.
 - **Insecure authentication** – Sessions must be managed properly to ensure secure authentication.
 - **Improper platform usage** – If security controls built into mobile apps are misused, access to the device and other apps can be compromised.
 - **Insufficient cryptography.**



OWASP Web and Cloud Application Vulnerabilities

- Some of the most common web and cloud application vulnerabilities:
 - **Injection** – An injection is where the threat actor sends a command, along with other data, to the interpreter. Often performed on SQL, NoSQL, Lightweight Directory Access Protocol (LDAP), and the OS.
 - **XML external entities (XXE)** - Using these external entities can disclose files and file shares, perform port scanning and code execution, or even launch a DoS attack.
 - **Sensitive data exposure** - Application Programming Interfaces (APIs) and web applications do not always protect data correctly.
 - **Broken access control** – If access is not configured properly, the threat actor might be able to access unauthorized data.
 - **Broken authentication** – Session management and authentication can be incorrectly implemented.



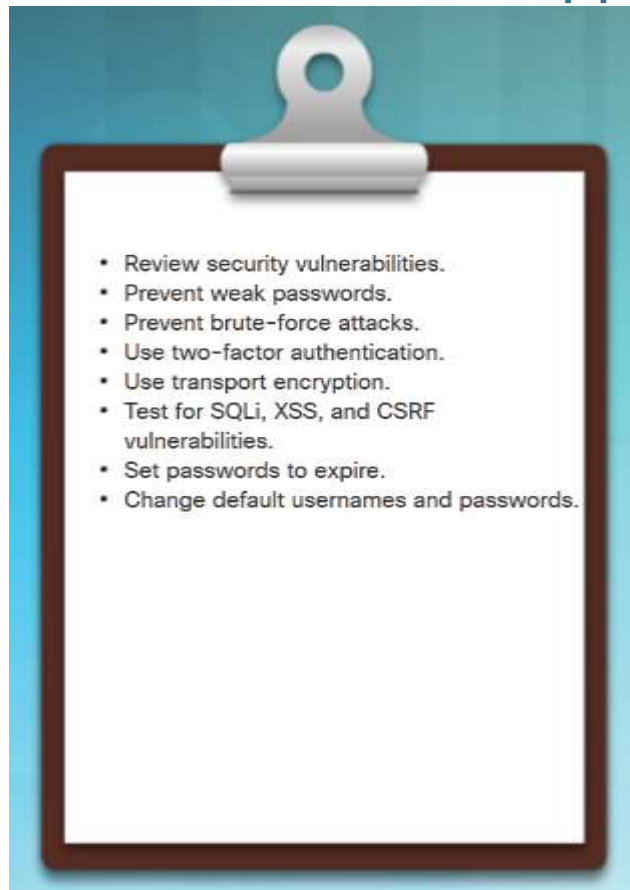
```
141 -- Struttura della tabella
142
143 CREATE TABLE IF NOT EXISTS `wp_blog_pictures` (
144   `pid` bigint(20) NOT NULL AUTO_INCREMENT,
145   `image_slug` varchar(255) NOT NULL,
146   `post_id` bigint(20) NOT NULL DEFAULT '0',
147   `galleryid` bigint(20) NOT NULL DEFAULT '0',
148   `filename` varchar(255) NOT NULL,
149   `description` mediumtext,
150   `alttext` mediumtext,
151   `inagestate` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
152   `exclude` tinyint(4) DEFAULT '0',
153   `sortorder` bigint(20) NOT NULL DEFAULT '0',
154   `meta_data` longtext,
155   `extras_post_id` bigint(20) NOT NULL DEFAULT '0',
156   `updated_at` bigint(20) DEFAULT NULL,
157   PRIMARY KEY (`pid`),
158   KEY `post_id` (`post_id`),
159   KEY `extras_post_id_key` (`extras_post_id`)
160 ) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Device Management and Data Applications



- IoT data can be stored at the edge of the network, or in a central location.
- Some processing of this data takes place at the mist or fog layers.
- Other data processing takes place in the cloud.
 - Cloud computing applications perform complex computations on enormous amounts of sensor data that can cause specific actions to be carried out by actuators, humans can be alerted, or a database updated.
 - These computations can be used to gain insight through cognitive and predictive analytics.
 - Also can be used in real time to adjust factory conditions, traffic control, or prevent an emergency.

Guidelines for Secure Web and Cloud Applications



IoT Web and Cloud Applications Vulnerabilities

Password Vulnerabilities

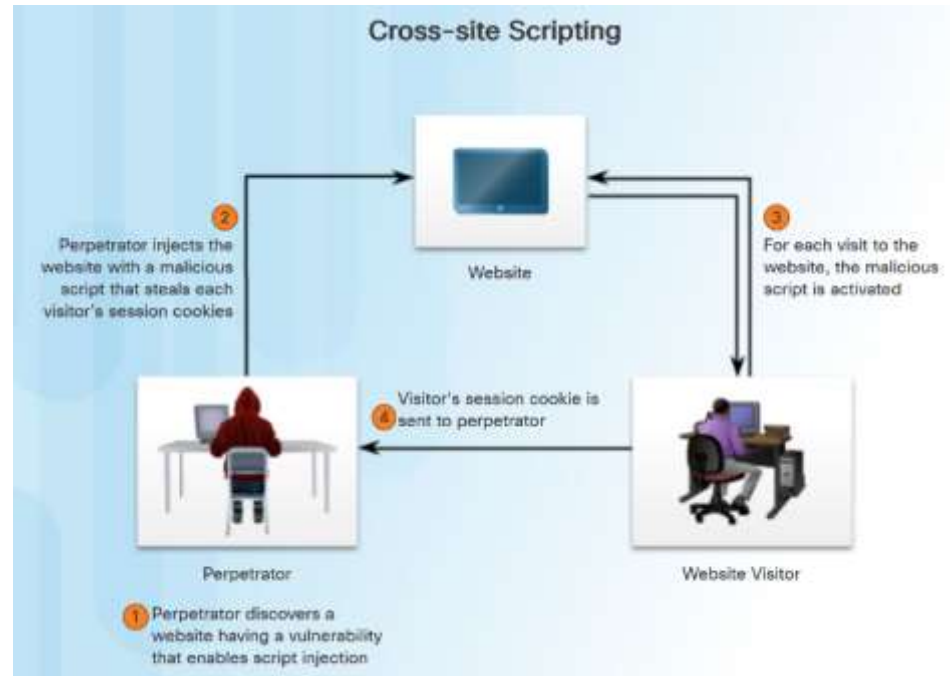


- Biggest problem with the conventional username and password combination is the user.
- Passwords should be easy to remember and difficult to crack.
- Most passwords are easy to guess, seldom changed, reused multiple times, and written down where they can be easily discovered.
- Many users do not even change the default password.
- Threat actors use this information to compromise and take over systems.

IoT Web and Cloud Applications Vulnerabilities

Web Frontend Vulnerabilities

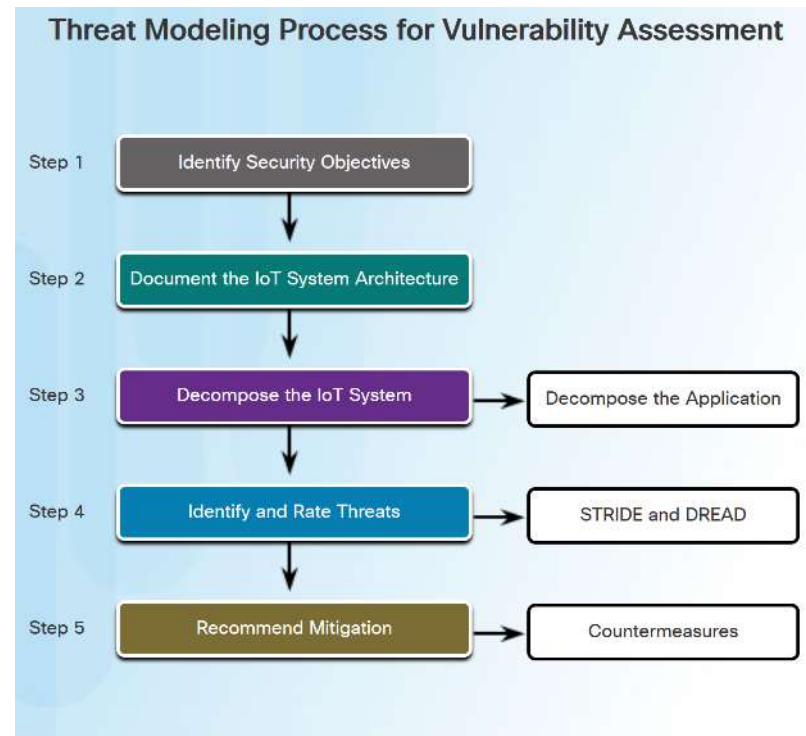
- Web frontend vulnerabilities apply to the apps, APIs, and services.
- The three most common web frontend vulnerabilities:
 - **Cross-Site Scripting** - In a XSS attack, the threat actor injects code, most often JavaScript, into the output of a web application
 - **SQL Injection** - SQLi is where the threat actor injects code into fields that will be used to query the SQL database. When the vulnerability is exploited, a threat actor can control an application's database.
 - **Broken Authentication** - A threat actor can hijack a session to assume the identity of a user when session tokens are left unexpired.



IoT Web and Cloud Applications Vulnerabilities

Threat Modeling at the Application Layer

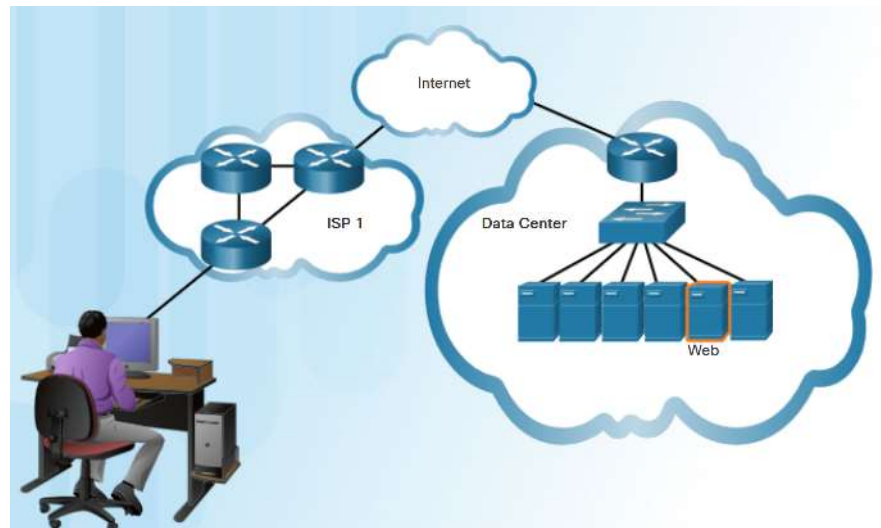
- Threat modeling can help to make sure applications are as secure as they can be.
- For the Application Layer of the IoT Protocol Suite, Threat Modeling focuses on Steps 3, 4 and 5.
 - Step 3: **Decompose the Application** - designed to gain a basic understanding of the application and its interaction with external entities.
 - Step 4: **Identify and Rate Threats** - used to categorize threats. Categorizations such as STRIDE and DREAD used in this course help vulnerability assessors identify and rate threats from the threat actor's perspective.
 - Step 5: **Recommend Mitigation** - a threat may be mitigated with a countermeasure. In some cases, the risk may be acceptable because the business impact is lower than the cost of the countermeasure.



IoT Application Layer Protocols

Role of Messaging Protocols

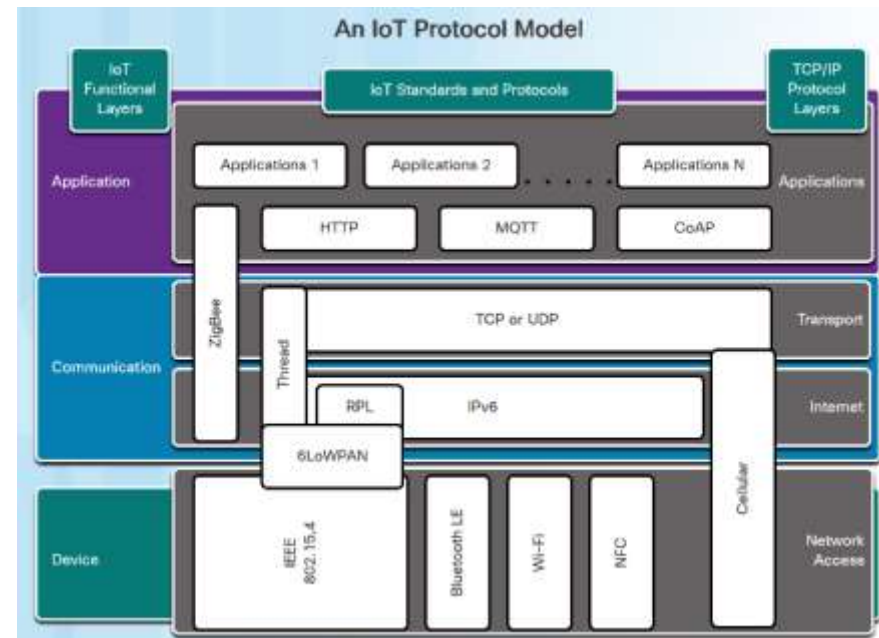
- IoT devices use messaging protocols to agree on how they will communicate.
- HTTP is inefficient for IoT devices for the following reasons:
 - IoT devices are resource-constrained and may not have the ability to install HTTP services.
 - HTTP is an inefficient messaging protocol that uses more resources.
 - Does not include a reliable publish and subscribe model.
 - Has no mechanism for preserving or retaining messages.
 - Has no method for informing users that the device is no longer available.



IoT Application Layer Protocols

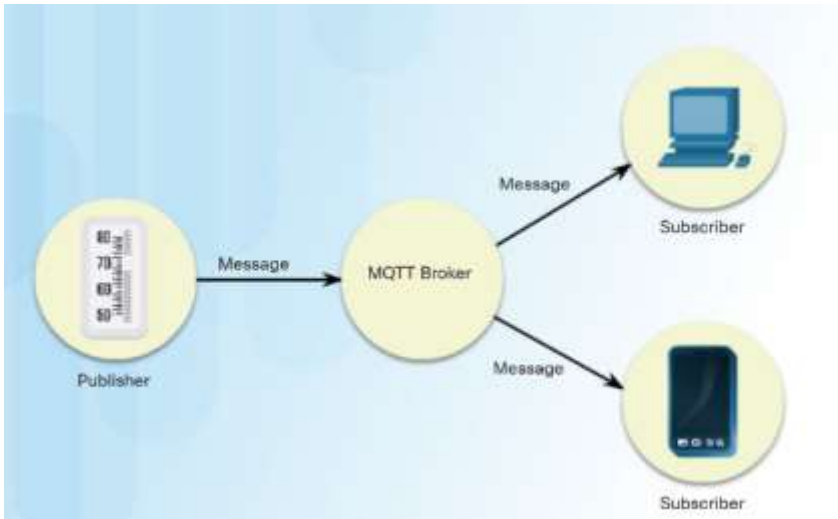
IoT Messaging Protocols

- Common IoT application layer protocols in use today:
 - **MQTT** – Message Queuing Telemetry Transport uses TCP and requires a message broker.
 - **CoAP** – Constrained Application Protocol is a document transfer protocol that uses UDP.
 - **XMPP** – Extensible Messaging and Presence Protocol uses TCP and was originally designed for instant messaging.
- Important characteristics for IoT protocols:
 - Power consumption
 - Speed
 - Latency
 - Security



IoT Application Layer Protocols

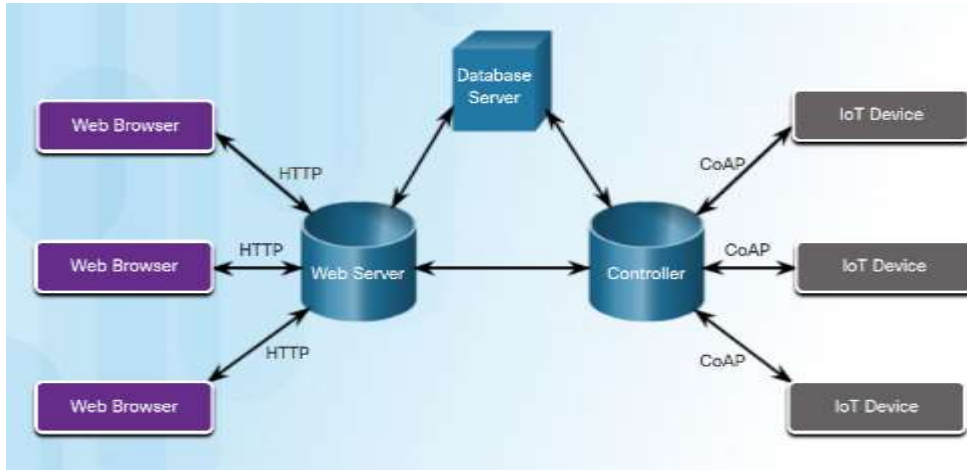
MQTT



- MQTT was developed by IBM specifically for the IoT, where lightweight machine to machine (M2M) communications are used.
 - MQTT uses a type client-server model called publish-subscribe.
 - A client either publishes a topic or subscribes to a topic. A topic is any specific type of message, like humidity, temperature, or light.
 - MQTT is designed to collect data from many devices and deliver it to the IT infrastructure.
 - Uses a hierarchical system for organizing topics.
- The publish-subscribe model works well in the IoT for two important reasons:
 - Clients that are not connected will not prevent the entire system from working.
 - Keeps traffic to a minimum reducing the amount of power used by IoT devices during communication.

IoT Application Layer Protocols

CoAP

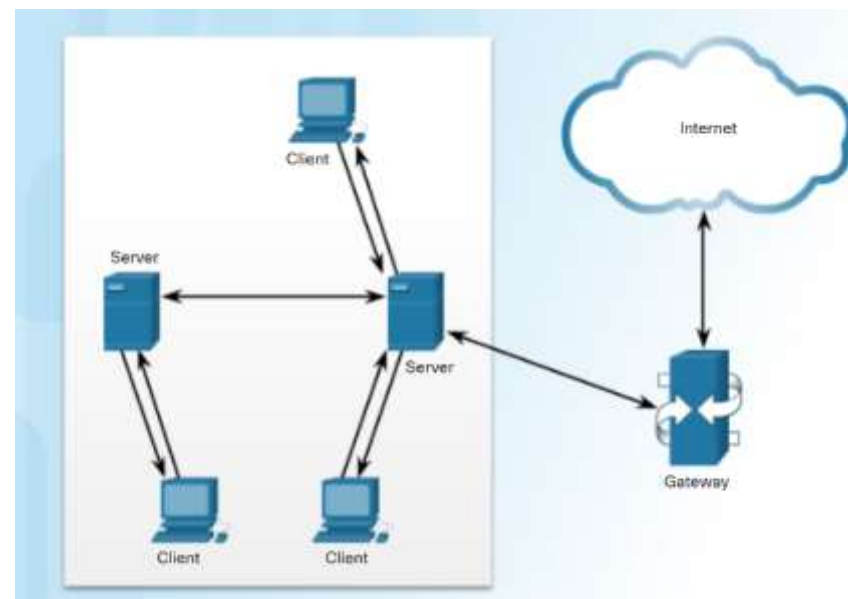


- CoAP is a lightweight protocol designed to be used for M2M communication.
 - Uses UDP.
 - Sensors and other nodes can connect with and publish to each other.
 - CoAP uses a client-server model where clients request from servers and servers respond to clients.
 - Supports the four HTTP methods: GET, POST, PUT, and DELETE.
 - Also has the ability to observe a resource. This allows the client to inform servers of state changes as they happen, which is crucial for many IoT devices.
- CoAP is commonly used when the state of an IoT device changes and that changed state needs to be reported.

IoT Application Layer Protocols

Other Application Messaging Protocols

- XMPP was originally created as an instant messaging protocol for the Jabber application.
 - It uses XML and runs over TCP
 - Used to connect home IoT devices to a web server so that they can be monitored from a smartphone or other device.
- DDS is used for M2M connections, specifically for devices that directly use device data.
 - Used in medical imaging, automotive testing, financial trading, air traffic control, and complicated high data use IoT networks.
- AMQP is used when reliability is the most important factor.



IoT Application Layer Protocols

A Note About UPnP

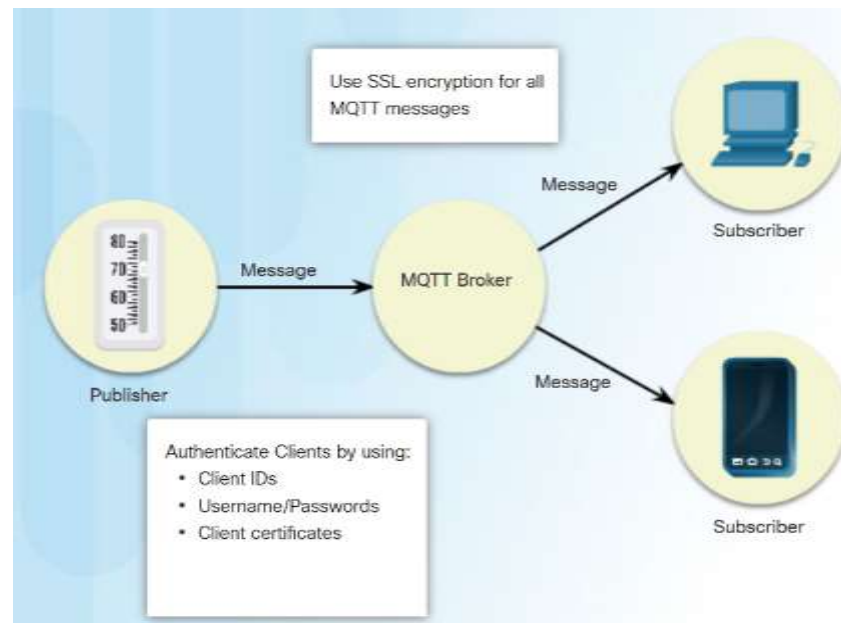
- **Universal Plug and Play (UPnP)** is a set of protocols that allows devices to detect the presence of other UPnP-enabled devices on the network with no intervention from the user.
 - Designed to be used on residential networks.
 - The multicast nature of UPnP consumes too many resources on networks that contain many UPnP-enabled devices, such as enterprise networks.
- UPnP has major security flaws.
 - Security flaws might give the threat actor remote control of devices such as cameras, lights, etc.
 - Home routers supporting UPnP can be fooled by malware to redirect DNS traffic to a rogue server using a single UPnP request.



5.2 Mitigate Security Issues in Messaging Protocols

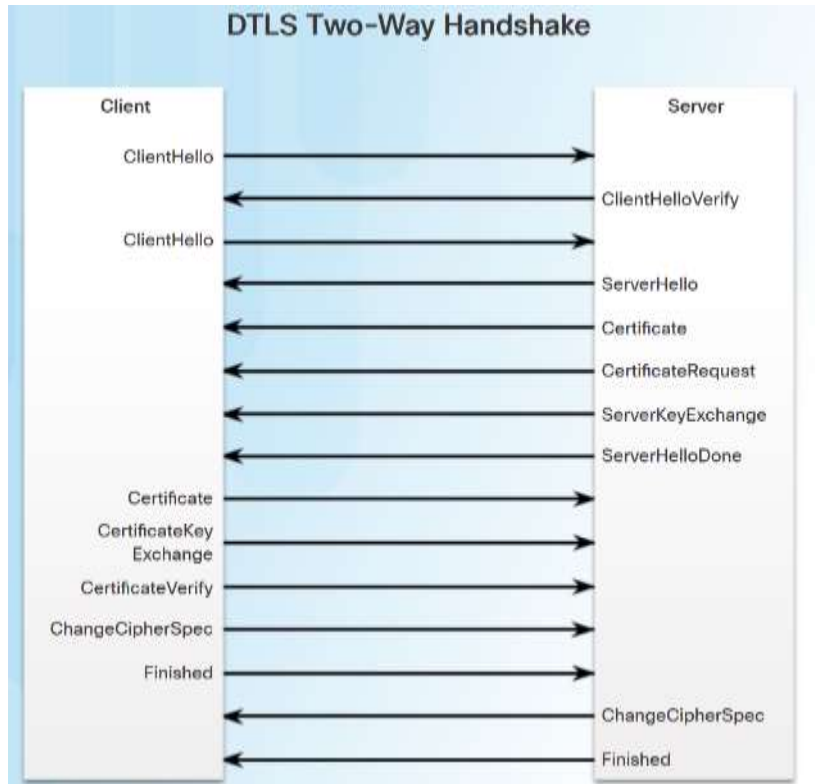
Securing MQTT

- Three ways a client can authenticate:
 - **Client ID** – Every client must be assigned a unique client ID. The client ID links topics to clients when it subscribes to a topic.
 - **Username and passwords** –username and password combination is sent in plaintext. Transport encryption must be used to secure the credentials.
 - **Client certificates** – x509 certificates can be deployed when a high level of security is necessary.
- Two ways to secure the messages:
 - **SSL encryption** – Same technology used to secure HTTP.
 - **Payload encryption** – Performed at the application layer, provides end to end encryption, protecting the message even from the broker.



Mitigate Security Issues in Messaging Protocols

Securing CoAP



- CoAP uses Datagram Transport Layer Security (DTLS). Sometimes referred to as secured CoAP (CoAPs).
 - A very simple implementation is available for embedded devices called **tinydtls**.
- DTLS can protect data with keys and algorithms, provide key exchange, and perform authentication.

Mitigate Security Issues in Messaging Protocols

Disable UPnP

UPnP

Apply ▶ Cancel Refresh

Turn UPnP On

Advertisement Period (in minutes) 30

Advertisement Time to Live (in hops) 4

UPnP Portmap Table

Active	Protocol	Int. Port	Ext. Port	IP Address
--------	----------	-----------	-----------	------------

- UPnP assumes that all local devices are friendly and trustworthy.
- Best practice is to turn off UPnP on all devices when possible.
 - Some manual configuration may be necessary to ensure devices and communications continue to work properly.
- Most home routers have the ability to turn UPnP on and off.

Mitigate Security Issues in Messaging Protocols

Password Issues

- Always change the default username and password on a new device.
- Strengthen and protect passwords using the following guidelines:
 - **Pass phrases** - Use a pass phrase instead of a password.
 - **Hardened passwords** - Use a minimum of 8 characters and allowing at least 64 characters to support passphrases.
 - **Password managers** - Use a password manager to avoid writing down passwords.
 - **Multi-factor authentication** - Using more than one form of authentication such as sending a text message to your phone with a code that must be entered.
- U. S. National Institute for Standards and Technology (NIST) recently published improved password requirements.



Mitigate Security Issues in Messaging Protocols

Harden Administrative Interfaces



- **SQLi** - To prevent SQL injection, the data must be kept separate from the commands and queries that are used by using a safe API.
- **eXternal Entity injection (XXE)** - the safest way to prevent it is to disable XML external entity and DTD processing in the application.
- **XSS** attacks are very dangerous because they can give the threat actor the ability to do whatever the user can. Three ways to prevent:
 - **Escaping** – This censors the data that the web page receives.
 - **Validating Input** – Whitelisting can be used to allow only known good characters into the application.
 - **Sanitizing** – Remove potentially harmful markup from any input.

Chapter Summary

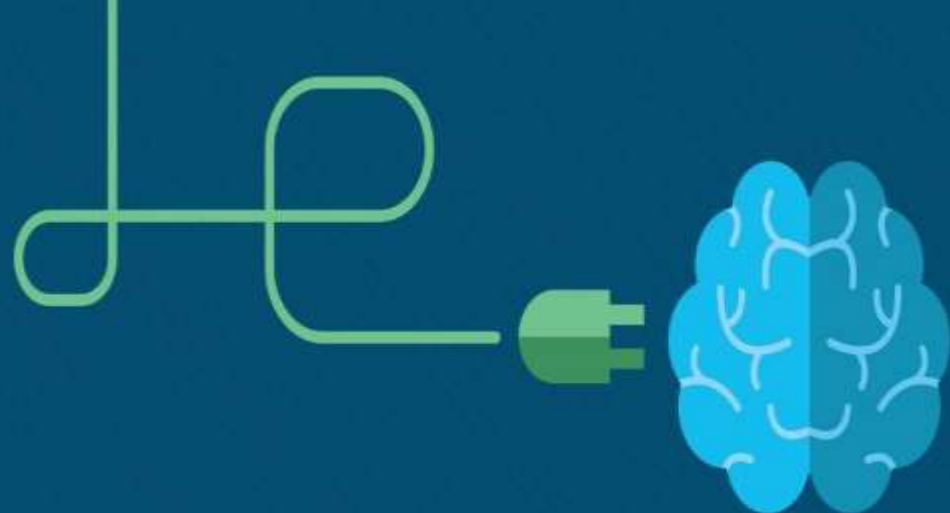
Summary

- IoT local application vulnerability is any weakness that a threat actor could use to compromise the security of that application.
- iOS and Android are relatively secure but they can still be compromised, especially through the applications that we install on them and use on the web.
- In order to secure exposed cloud applications, security applications such as static analysis, web application scanners, and runtime protection must be used.
- Web and cloud applications and cloud-based web interfaces must follow strict guidelines to keep them secure.
- Threat modeling can help to make sure applications are as secure as they can be.
- A message protocol defines the functions and rules for transmitting messages between devices.
- MQTT is used when there is a need to monitor many small devices and have their data available in the cloud to trigger M2M decisions.
- CoAP uses a client-server model where clients request from servers and servers respond to clients.

Chapter Summary

Summary (Cont.)

- Other message protocols used in the IoT include XMPP, DDS, AMQP, and UPnP.
- The security implementation for MQTT must be based on the capabilities of the broker and the clients.
- To secure CoAP use DTLS.
- If a device on your network becomes infected with malware, it could use UPnP to access information.
- Strengthen and protect passwords using strict guidelines.
- To harden administrative interfaces, use a mix of code review, static testing, and dynamic testing.



Chapter 6: Vulnerability and Risk Assessment in an IoT System

IoT Security 1.0 v2.0



Chapter 6 - Sections & Objectives

- 6.1 Explain how vulnerabilities are assessed in IoT Systems.
 - Explain how security vulnerabilities are assessed.
 - Explain how tools and services are used to assess vulnerabilities in IoT systems.
- 6.2 Evaluate security in an IoT system risk using assessment.
 - Explain approaches to security risk assessment.
 - Evaluate risks in an IoT system using assessment tools.
 - Use the STRDE and DREAD models as a part of a risk assessment process.
 - Explain approaches to risk management.
- 6.3 Explain innovations in IoT Security
 - Explain the role of blockchain in IoT systems.
 - Explain how blockchain works.

6.1 Vulnerability Assessment

Be a Bounty Hunter



- **Bug Bounty Hunters** -Talented ethical hackers hired by crowdsourced security services to test their clients' networks.
 - Company has access to a wide range of creative hacking talent.
- **HackerOne** is one of the first companies to provide these services.
 - Also tests core internet technologies such as Open SSL, various servers such as Nginx and Apache, and languages such as PHP, Python, and Perl.
 - Enlists nearly 100,000 hackers to discover vulnerabilities and have paid millions of dollars in internet bug bounties.

Vulnerability Assessment

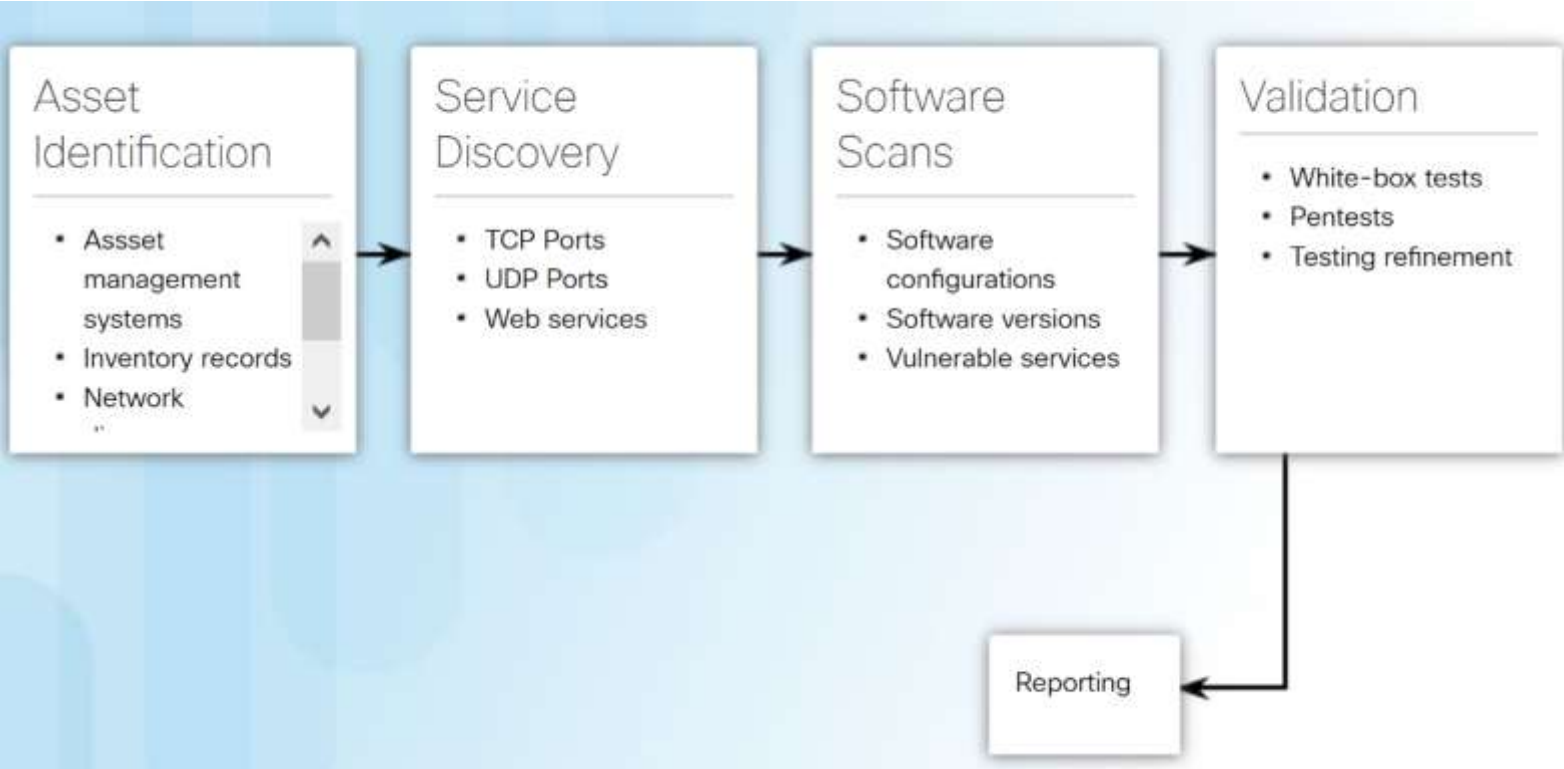
Vulnerability Assessment



- **Vulnerability assessment** identifies vulnerabilities that are likely to be exploited by threat actors.
- Vulnerability assessments can be routinely and regularly conducted, or may be targeted at specific components of an IoT system.
- Vulnerability assessments are frequently performed using off-the-shelf-tools such as those found in Kali Linux.

Vulnerability Assessment

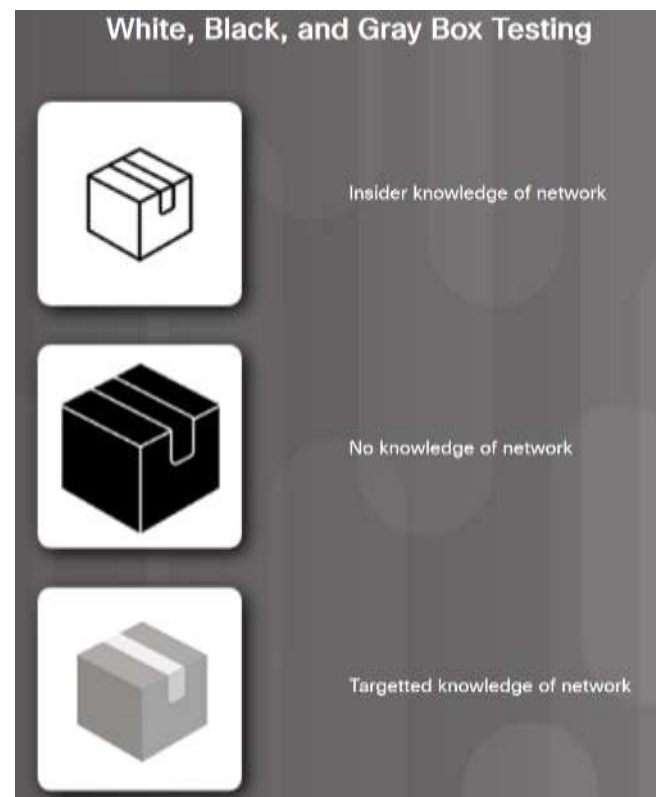
The Process of Vulnerability Assessment



Vulnerability Testing Types and Tools

Types of Vulnerability Assessment

- Vulnerability assessment can be classified into three types:
 - **White box** - Assessors have knowledge of the network systems and frequently operate from within the organization. They often focus on specific aspects of the system.
 - **Black box** - This assessment is the closest to an actual attack. The assessors, who are usually working for a third party, have no knowledge of the network architecture.
 - **Gray box** - Tester has partial knowledge of the network systems they are testing including access to the documentation of internal network architecture. Goal is to verify the vulnerabilities, determine the ease of exploiting them, and to determine the potential impacts of exploits.



Vulnerability Testing Types and Tools

Penetration Testing

- **Penetration testing** (pen testing) consists of actual focused attacks that uncover the potential impacts associated with known vulnerabilities.
 - Skilled ethical hackers take on the role of threat actors and launch actual attacks that are meant to replicate what malicious hackers might do.
 - Pen testing is an assessment tool used in black box testing - the hackers operate with no knowledge of the internal workings of the target system.
 - Pen testing is also used to confirm that vulnerabilities identified in other vulnerability assessments do exist—Gray box testing.
 - Pen tests are used to confirm that measures taken to eliminate a vulnerability have been effective.



Vulnerability Testing Types and Tools

Port Mapping Tools



- Port mapping tools are used for discovering open ports on end systems and network devices.
- Examples are Nmap, Netcat, or SolarWinds Port Scanner.
- Nmap's GUI form, called Zenmap can produce very detailed information about a single system or a range of systems on a network segment.
 - Can discover hosts on the network.
 - Can report the open ports.
 - Can identify the operating systems that are running on hosts.
 - Can reveal details about the services that are running on the open ports including the software versions in a process known as fingerprinting.

Password Vulnerability Tools

- Weak passwords on IoT application portals are a concern.
- Several common types of password attack methods that can be used to assess password security:
 - **Brute force** - This attack is a very time consuming, inefficient, automated means of trying every possible combination of letters, numbers, and symbols to challenge logins.
 - **Dictionary attack** - This attack uses lists of words that could be used as passwords.
 - **Password sniffing and cracking** - Protocol analyzers can be used to intercept authentication traffic that contains hashed passwords. Hashed passwords may also be discovered in the file systems of IoT devices. Tools such as John the Ripper and Aircrack-NG can be used to attempt to break the hash encryption

Vulnerability Testing Types and Tools

Password Vulnerability Tools (Cont.)



FIDO (Fast IDentity Online) Alliance shown in the figure has developed new authentication technologies and standards for the IoT.

- Defeat brute force attacks by allowing a limited number of authentication failures before an account is locked out.
- Username security is essential - account lock out can be a malicious denial of service in which attackers intentionally try to lockout legitimate users.
- Enhanced approaches to authentication in IoT systems must be considered when there is high risk of damage or physical harm to people, such as in Industrial Internet Control Systems (IICSS).
- United States National Institute of Standards and Technology (NIST) has guidelines for digital identities.

Vulnerability Testing Types and Tools

Web Application Vulnerability Tools

- In addition to the tools in the Kali suite that have been used in this course, some other prominent tools are:
 - **OWASP ZAP** - The Open Web Application Security Project (OWASP) is a primary reference for web application vulnerabilities. The OWASP ZED Attack Proxy (OWASP ZAP) is a free open-source vulnerability assessment tool used for black box pen testing.
 - **OpenVAS** - OpenVas framework combines a number of vulnerability scanning tools into a unified application that includes vulnerability data storage, scan scheduling, and reporting.
 - **Burp Suite** - Burp Suite is a comprehensive group of web application vulnerability testing tools that can identify the presence of the OWASP top 10 vulnerabilities. It includes a scanner, a configurable automated attack tool, and a web crawler that can map the file system of a web application.



Vulnerability Testing Types and Tools

Vulnerability Assessment Services

- Security as a service (SECaaS) companies provide a wide range of managed security services including vulnerability scanning.
 - Alienvault, Qualys and Mandiant offer these services.
 - Cisco offers network penetration assessment as part of its portfolio of security products and services.



Vulnerability Testing Types and Tools

Vulnerability Information Sources

- A number of threat intelligence sources work diligently to discover, investigate, and disseminate threat information as it is discovered.
- Cisco Talos Intelligence Group is one of the largest commercial threat intelligence teams in the world.
 - Talos defends Cisco customers against known and emerging threats.
- The NIST National Vulnerability Database (NVD) enhances Common Vulnerabilities and Exposures (CVEs) with additional analysis, a database, and a fine-grained search engine.
- Hardware and software vendors should inform customers and the public about vulnerabilities in their products and make patches available.



6.2 Risk Assessment Concepts and Approaches

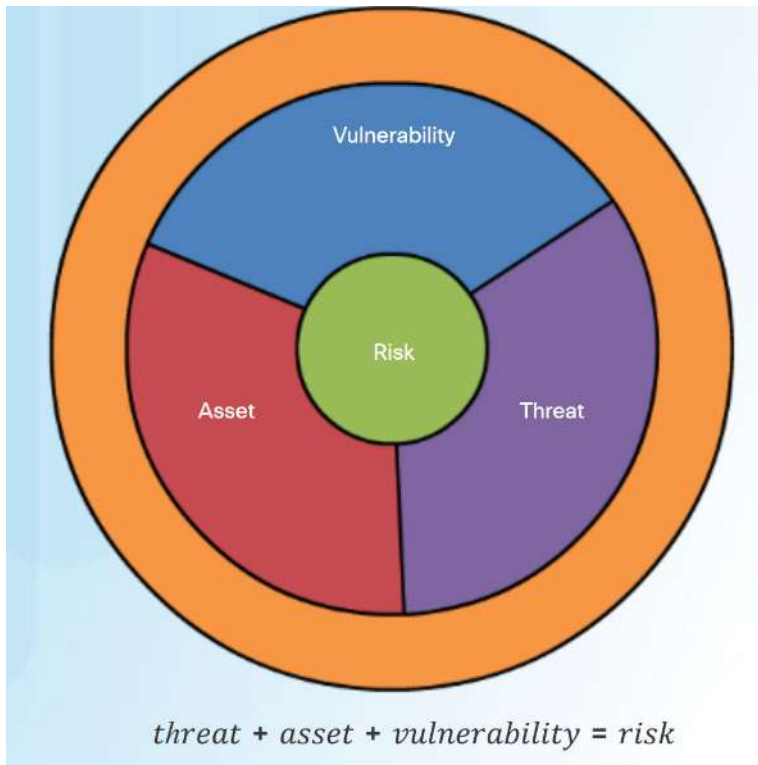
IoT Risk Assessment



- Most approaches to OT security are concerned with the safety of people and equipment.
- OT communication is frequently M2M with humans monitoring and controlling industrial, energy, environmental, or smart city systems, among others.
- There are a new set of security risks with the IoT because of the enormous IoT attack surface.
 - An example is compromising critical industrial infrastructure components that are now connected to the internet.

Risk Assessment Concepts and Approaches

Vulnerability and Risk

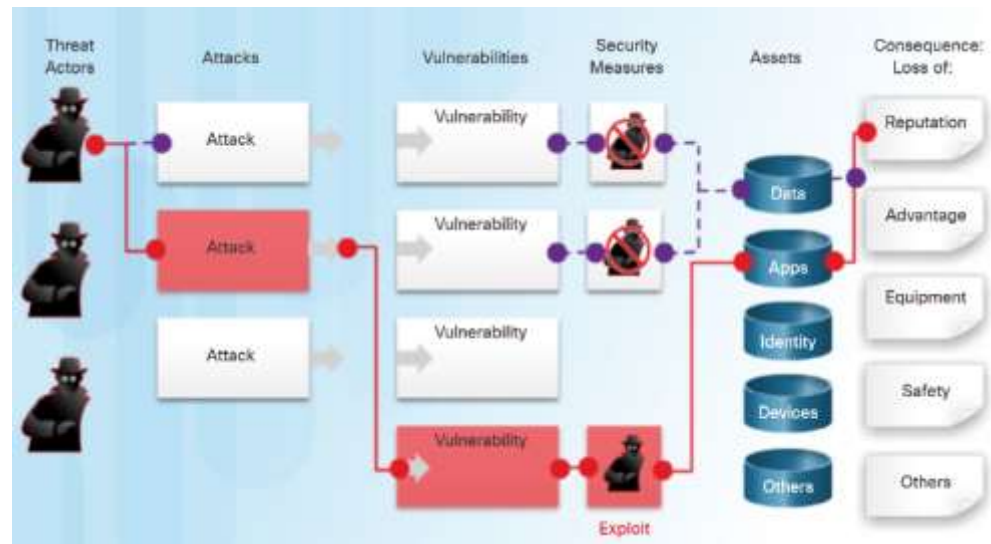


- Vulnerabilities are weaknesses in software and systems that can be exploited by threat actors in cyberattacks.
- Risks are those vulnerabilities assessed in the context of a specific organization.
 - A given vulnerability may have greater impact in one type of organization than another.
- The level of risk is dependent on the value of the asset, the vulnerability of that asset within the context of the software and systems on which they are used, and the likelihood that threats will be successfully executed against that asset.

Risk Assessment Concepts and Approaches

Thinking about Risk

- Determining risk first involves answering the following questions as part of a risk assessment:
 - Who are the threat actors who want to attack us?
 - What vulnerabilities can threat actors exploit?
 - How would the organization be affected by successful attacks?
 - What is the likelihood that different attacks will occur?
 - What can the organization do to address the risk?



Risk Assessment Concepts and Approaches

Common Vulnerability Scoring System

- The CVSS is a risk assessment designed to convey the common attributes and severity of vulnerabilities in computer hardware and software systems.
- CVSS 3.0, is a vendor-neutral, industry standard, open framework for weighting the risks of a vulnerability using a variety of metrics.
 - These weights combine to provide a score of the risk inherent in a vulnerability.
 - The numeric score can be used to determine the urgency of the vulnerability, and the priority for addressing it.
 - Does not include metrics around the issue of safety because it was designed for IT security. Future scoring systems should include additional metrics specific to IoT implementations.



Common Vulnerability Scoring System (CVSS-SIG)

- CVSS v3.0 Calculator
- CVSS v3.0 Specification Document
- CVSS v3.0 User Guide
- CVSS v3.0 Examples
- CVSS v3.0 Calculator Use & Design
- CVSS v2 Archive
- CVSS v1 Archive
- CVSS-SIG participants
- Scores and Calculators
- Identity & logo usage

Common Vulnerability Scoring System SIG

Mission

The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes.

CVSS is a published standard used by organizations worldwide, and the SIG's mission is to continue to improve it.

Goals/Deliverables

CVSS is currently at version 3.0. Links on the left lead to CVSS version 3.0's specification and related deliverables.

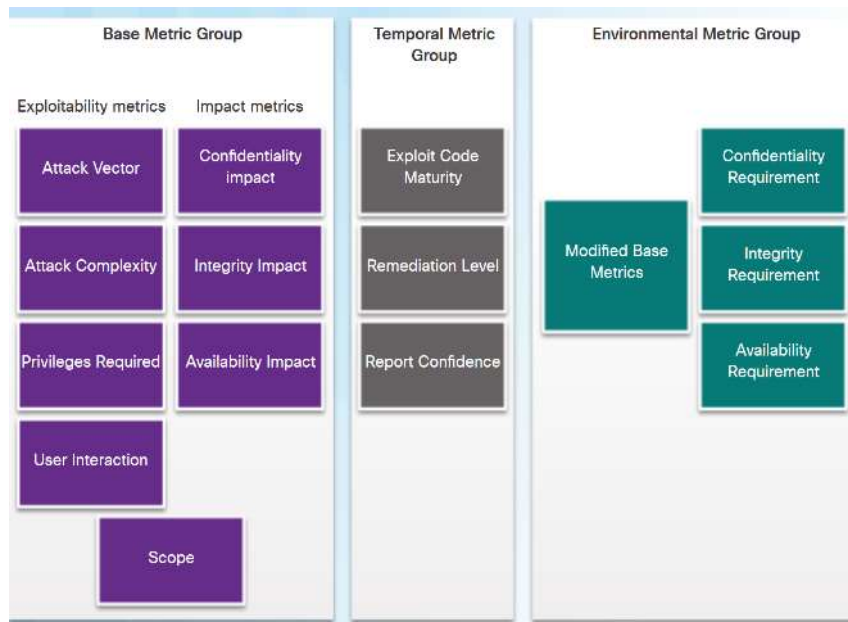
A self-paced on-line training course explains CVSS v3.0 and assumes no prior CVSS experience. It is based on FIRST's open training platform.

Current Initiatives

The CVSS Special Interest Group (SIG) is currently working on individual improvements that will form the basis of the next version of the CVSS standard. The SIG is composed of representatives from a broad range of industry sectors: from banking and finance to technology and academia. Organizations and individuals interested in joining

Risk Assessment Concepts and Approaches

The CVSS Metrics Groups



- The CVSS tool requires the assessor to select values in three metric groups for each vulnerability that has been identified.
- The figure shows the three metric groups and the individual metrics that make up each group.
 - **Base Metric Group** - represents the characteristics of a vulnerability that are constant over time and across contexts.
 - Exploitability - These are features of the exploit such as the vector, complexity, and user interaction required by the exploit.
 - Impact metrics - Impacts of the exploit are rooted in the CIA triad of confidentiality, integrity, and availability.
 - **Temporal Metric Group** - measures the characteristics of a vulnerability that may change over time, but not across user environments.
 - **Environmental Metric Group** - measures the aspects of a vulnerability that are rooted in a specific organization's environment.

Risk Assessment Concepts and Approaches

CVSS Base Metric Group



- Base Metric Group Exploitability metrics include:
 - **Attack vector** – reflects the proximity of the threat actor.
 - **Attack complexity** – expresses the number of components, software, hardware, or networks, that are beyond the attacker’s control and that must be present.
 - **Privileges required** – captures the level of access that is required.
 - **User interaction** - expresses the presence or absence of the requirement for user interaction.
 - **Scope** – expresses whether multiple authorities must be involved.
- Base Metric Group Impact metrics include:
 - **Confidentiality Impact** – measures the impact to confidentiality due to a successfully exploited vulnerability.
 - **Integrity Impact** – measures the impact to integrity due to a successfully exploited vulnerability.
 - **Availability Impact** – measures the impact to availability due to a successfully exploited vulnerability.

Risk Assessment Concepts and Approaches

The CVSS Process

- The CVSS Base Metrics Group is a way to assess security vulnerabilities found in software and hardware systems.
 - It describes the severity of a vulnerability based on the characteristics of a successful exploit of the vulnerability.
 - The other metric groups modify the base severity score by accounting for how the base severity rating is affected by time and environmental factors.
- The CVSS process uses a tool called the CVSS v3.0 Calculator
 - The calculator is similar to a questionnaire in which choices are made that describe the vulnerability for each metric group then a score is generated.

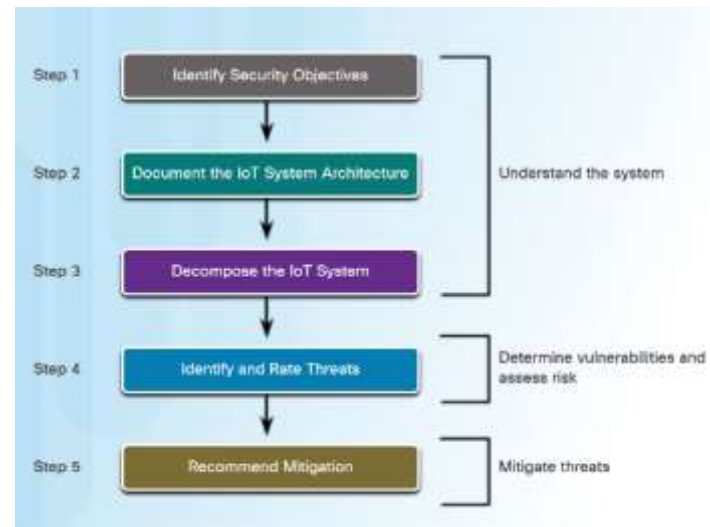
The screenshot shows the 'Common Vulnerability Scoring System Version 3.0 Calculator' web interface. The page features a navigation menu on the left with links to 'CVSS v3.0 Calculator', 'CVSS v3.0 Specification Document', 'CVSS v3.0 User Guide', 'CVSS v3.0 Examples', 'CVSS v3.0 Calculator Use & Design', 'CVSS v2 Archive', 'CVSS v1 Archive', 'CVSS MG Participants', 'Scores and Calculators', and 'Identify & Logo Usage'. The main content area is titled 'Common Vulnerability Scoring System Version 3.0 Calculator' and includes a brief introduction. Below this, there are several sections for selecting metric values, each with a 'Base Score' label and a 'Select values for all base metrics to generate score' button. The sections are: Attack Vector (AV) with options Network (N), Adjacent (A), Local (L), Physical (P); Attack Complexity (AC) with options Low (L), High (H); Privileges Required (PR) with options None (N), Low (L), High (H); User Interaction (UI) with options None (N), Required (R); Scope (S) with options Unrestricted (U), Changed (C); Confidentiality (C) with options None (N), Low (L), High (H); Integrity (I) with options None (N), Low (L), High (H); and Availability (A) with options None (N), Low (L), High (H).

- In addition to the numeric severity rating a vector string is also created that summarizes the choices made.
- The Temporal and Environmental metric values then modify the Base Metric results to provide an overall score.

Assessing Risk with Threat Modeling

Threat Modeling in Depth

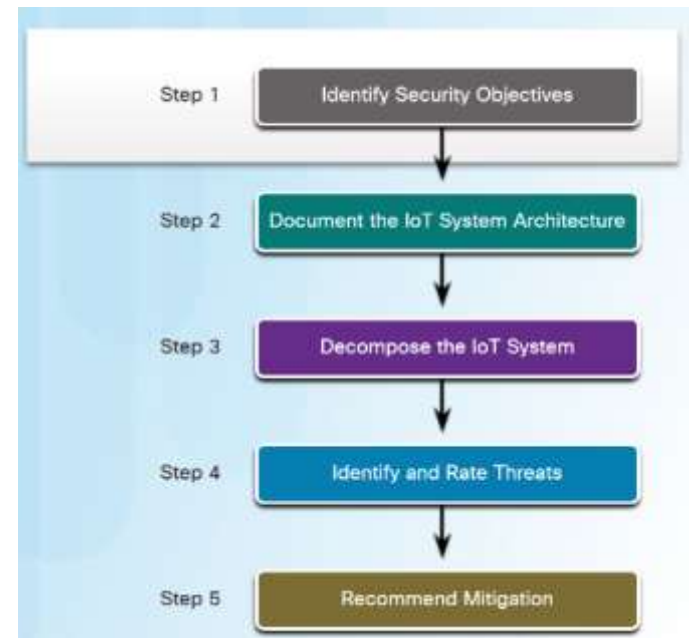
- Threat modeling is a proactive approach to assessing security of systems and software. Threat modeling is best applied throughout the development process.
- Three approaches to threat modeling:
 - Attack-centric – from the point of view of the attacker.
 - Defense-centric – analyzes the architecture system to identify threats to different elements. **Used in this class.**
 - Asset-centric – focuses on classifying assets and assigning value to them.
- The threat modeling process begins with understanding the system by answering the following questions:
 - What are we modeling? What are the potential threats?
 - What are the risks? What can be done to address the risks?



Assessing Risk with Threat Modeling

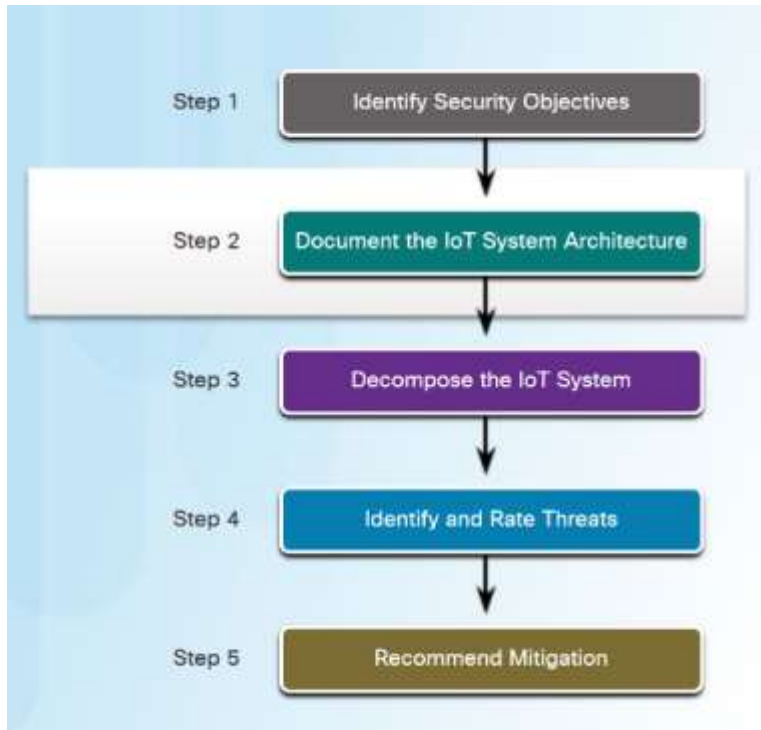
System Security Objectives

- First step is to determine what the security objectives are for the system, based on its purpose and operation.
- Important to understand what type of data is handled by the system and the consequences of data theft or destruction.
 - Will data loss result in financial losses? If so to what degree?
 - Will the reputation of the company be damaged? If so, what are the business impacts?
- Governments and other organizations are enacting regulations that govern how data is gathered, transmitted, and stored.
 - Violations of these regulations can result in serious financial and legal penalties.
- Critical infrastructure systems must always be available, disruptions can have serious impacts.



Assessing Risk with Threat Modeling

Map Data Flows











- After the security objectives are identified, the functions of the system architecture should be diagrammed.
- **Data flow diagrams (DFDs)** are extremely useful for visualizing an IoT system.
 - DFDs depict the pathways that data will take between different functional components of the system, including entry points into the system and the devices and people using those entry points.

Assessing Risk with Threat Modeling

Data Flow Diagrams Components

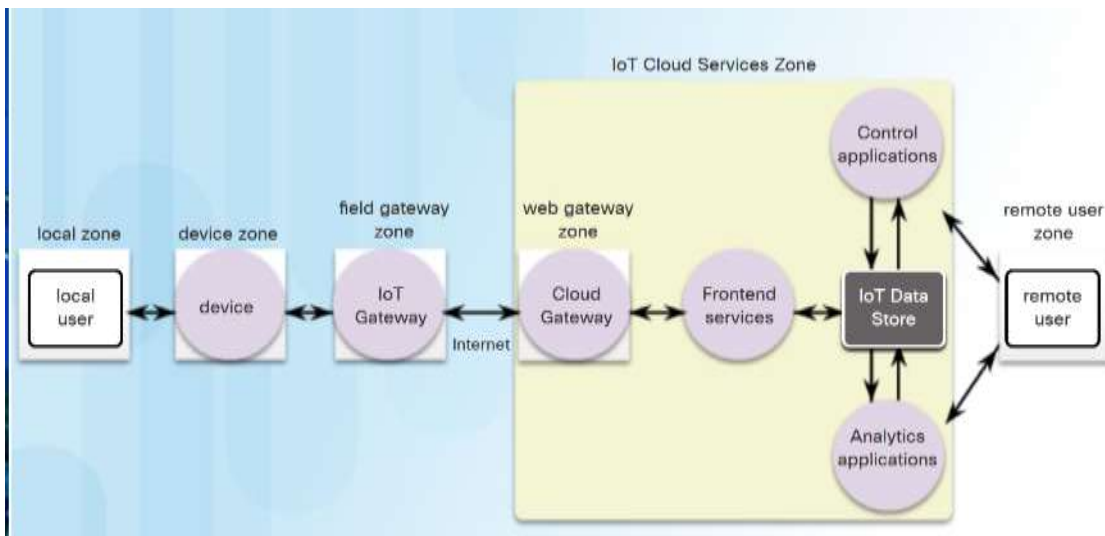
- Components included in an IoT system DFD:
 - IoT devices
 - IoT gateways – Enable sensor data to be sent across the IP network.
 - Local applications
 - Edge devices – Enable internal IP traffic to be sent between locations and the internet or cloud.
 - Data applications
 - Data storage
 - Control applications – Process data in order to make decisions that enact control.
 - Mobile applications
- DFDs use 4 symbols to represent these devices. This course uses Yourdon and Coad symbols.

Type	Description	Gane and Sarson	Yourdon and Coad
External Entity	Users, contractors, and partners outside of the control of the system that send or receive data		
Process	Data output from sensing, actuating, traffic forwarding, analysis, control systems		
Data Store	Data at rest in local, fog, cloud, or data center storage		
Data Flow	Single headed arrows indicated uni-directional data flow; Double headed arrows indicate bi-directional data flow		

- DFD entities should conform to basic rules:
 - Any process should have at least one input and one output.
 - Data stores should have flows for write and read access.
 - Data stored in a system must go through at least one process.

Assessing Risk with Threat Modeling

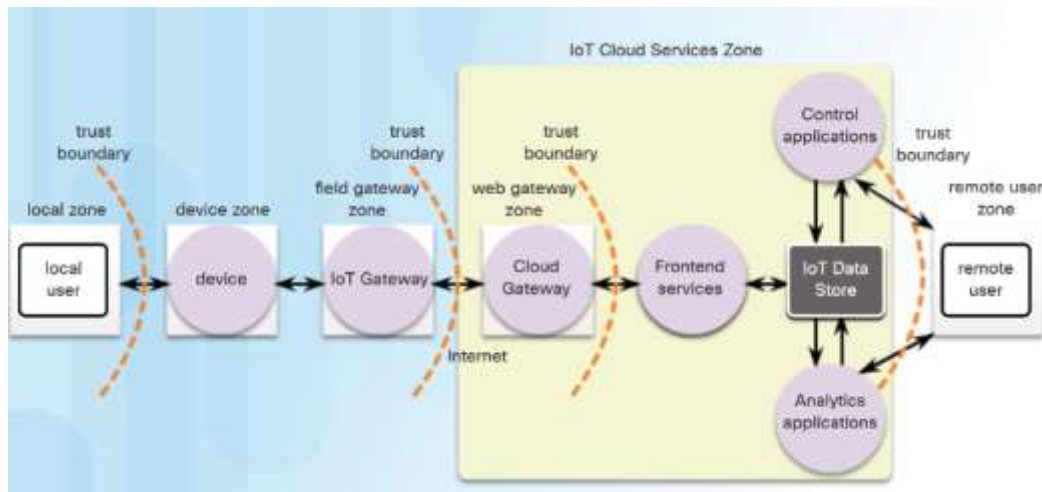
Zones of the System



- Zones can be defined as areas of the system that require different authorization and authentication.
- Zones also help to limit the exposure of different parts of the system to the vulnerabilities that are associated with each zone.
- Example zones might be the sensor area of the network, web applications, the IP gateway and network edge, etc.
- Zones can be nested when components are located within another organization.

Assessing Risk with Threat Modeling

Determine Trust Boundaries



- Trust boundaries delimit sections of the network where the level of trust between entities at either end of a flow is different.
 - For example, data flowing from an IoT Gateway to Cloud Gateway crosses a trust boundary.
 - The permissions for the IoT Gateway are different than those for the Cloud Gateway, which is exposed to the internet and accessed by many users.
 - Data traffic that crosses this boundary must be authorized and authenticated at the incoming device.

Threat Identification and Risk Prioritization

STRIDE

STRIDE Threat Classifications		
Threat Classification	Definition	Example Threats
Spoofing	Impersonating a legitimate user or device	<ul style="list-style-type: none"> Pretending to be a valid user or device Pretending to be another server Laptop impersonates IoT gateway to perform man in the middle data interception
Tampering	Modifying data, code, or device	<ul style="list-style-type: none"> Modifying sensor data Physical device hacking
Repudiation	Disabling ability to prove or disprove events	<ul style="list-style-type: none"> Corrupt or destroy log files Alter data record timestamps
Information Disclosure	Making privileged information available to unauthorized parties	<ul style="list-style-type: none"> Gathering sensitive information from log files Using SQL injection to steal personal data from web application
Denial of Service	Cause device to be unavailable to perform legitimate functions due to illegitimate traffic, data, or software	<ul style="list-style-type: none"> Crashing a web site Sending data absorbing CPU cycle, storage, or device power resources
Elevation of Privilege	Obtaining higher privileges than would normally be authorized	<ul style="list-style-type: none"> Allowing remote user to run commands, switch from a limited user to admin Using intercepted credentials to logon to data dashboard

- The STRIDE approach provides a set of categories that are very helpful for identifying potential threats in IoT systems.
- STRIDE is an acronym for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege.
- STRIDE provides a classification scheme for identifying threats for each element of the DFD.
- Understanding which vulnerabilities are relevant to which system elements will help save time in the threat modeling process.

Threat Identification and Risk Prioritization

Risk Assessment with DREAD

DREAD Categories
Damage potential
Reproducibility
Exploitability
Affected users
Discoverability

- Each threat identified by STRIDE must now be assessed for its degree of risk for the organization.
- The DREAD model results in a quantitative risk score.
- This score can be used with risk cost assessments to evaluate the desirability and feasibility of mitigating a threat.

Threat Identification and Risk Prioritization

The DREAD Threat Rating Model

- Small modifications may be required to apply these models to IoT systems since they were developed for software systems.
- Each vulnerability identified by STRIDE is rated according to the five DREAD categories. The rating values are:
 - 3 = high
 - 2 = medium
 - 1 = low
- Figure shows the meaning for the metric values for each category. These values are relative to the organization and system which is affected by the vulnerability

	Category	High (3)	Medium (2)	Low (1)
D	Damage potential	System down or under threat actor control; damage to people or facilities.	Loss of important data; some temporary system compromise or loss of availability.	Minor to medium loss of data or system impact.
R	Reproducibility	Every attempt will be successful.	Estimated to work half the time.	Difficult to reproduce, exploit requires special conditions.
E	Exploitability	Easily carried out by inexperienced threat actor.	Requires skilled attacker.	Requires very skilled attacker or attacking organization.
A	Affected users (or devices)	Enough devices to cause serious outages. All users who are up to standard.	Some devices that are not patched or in up to current standard.	Few users or devices under edge case configurations or roles.
D	Discoverability	Widely known in the attacker community. High value to attackers.	Little known and not widely present, some benefit to threat actors.	Little known and of little interest.

Threat Identification and Risk Prioritization

A DREAD Model Example

- A retail chain with 300 locations purchased network-connected (DVRs) for the security camera systems in each store. The DVRs are connected through the internet to regional offices that administer the DVRs and collect the videos. An embedded HTTP server in the DVR allows the device to be configured and controlled through a web page.
- The manufacturer of the DVR has published a notification regarding a recently discovered vulnerability. Threat actors can craft an HTTP cookie request and send it to the DVR embedded web server.

- For devices with this vulnerability, threat actors can gain control of the DVR, view live video, manipulate files, disable operations, etc.
- The figure shows that the rating for this vulnerability is High.
- Although the risk is High, the company will not update the firmware in all devices because the updating process is labor intensive and would prove costly.

Vulnerability	D	R	E	A	D	Total
Crafted cookie allows retrieval of user credentials across the internet	2	3	3	3	3	14

Risk rating	Result
High	12-15
Medium	8-11
Low	5-7

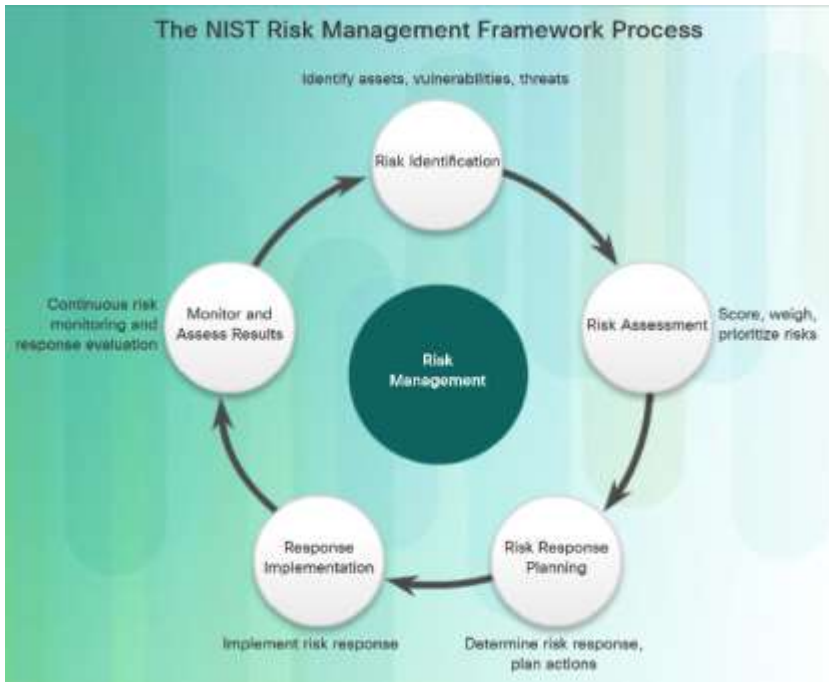
Threat Identification and Risk Prioritization

Recommending Mitigation

- Secure development and deployment of IoT systems is the most effect way to mitigate risk.
- Several guidelines for secure design:
 - Consider the attack surface area - IoT deployments will have a large attack surface at the device layer.
 - Consider devices that are built with secure default configurations and update mechanisms.
 - Consider the security policies of partners and third-party services.
 - Consider security in all things.
- General mitigation strategies:
 - Keep device firmware up to date.
 - Keep application software up to date.
 - Keep IT and IoT traffic separate on organization networks.
 - Insure physical security whenever possible.
 - Use secure messaging protocols and encryption.
 - Engage with the network security personnel within the organization.

Managing Risk in IoT Systems

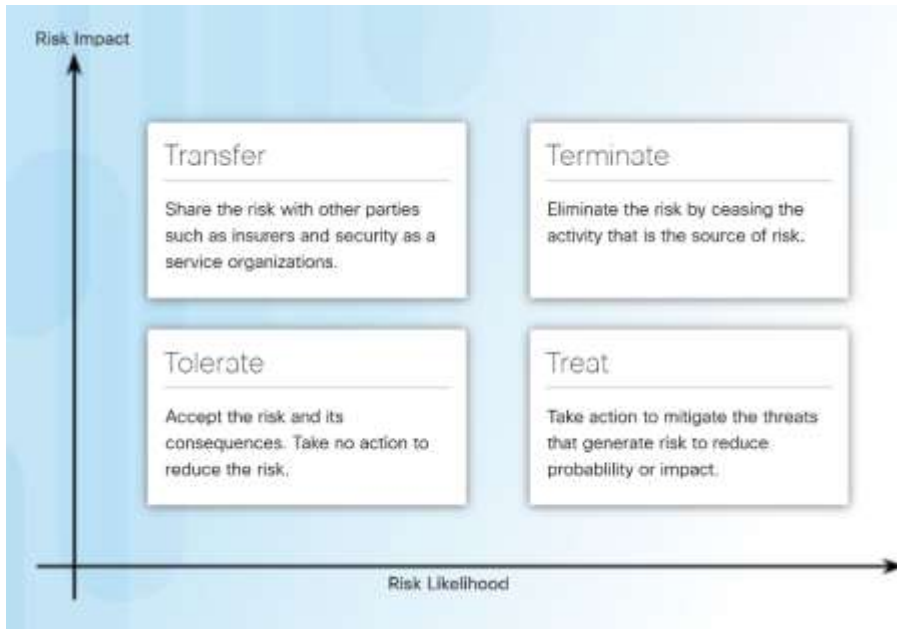
Risk Management Strategies



- The NIST Risk Management Framework (RMF) is a process that is cyclical and ongoing.
- Risk Identification and Risk Assessments parallel the threat modeling approach. RMF closes the circle by including the risk response, response evaluation, and response assessment activities.
- Identification and matching of threats with vulnerabilities is called threat-vulnerability (T-V) pairing.
- T-V pairs can be used as a baseline to indicate risk before security controls are implemented.
 - Baseline can be compared to ongoing risk assessments as a means of evaluating risk management effectiveness.
 - This determines the inherent risk profile of an organization.
- Risks may be scored or weighted as a way of prioritizing risk reduction strategies.

Managing Risk in IoT Systems

Risk Response



- The four "T's" of risk response:
 - **Risk avoidance (Terminate)** - Stop performing the activities that create risk.
 - **Risk reduction (Treat)** - Decrease the risk by taking measures to reduce vulnerability.
 - **Risk sharing (Transfer)** - Shift some of the risk to other parties.
 - **Risk retention (Tolerate)** - Accept the risk and its consequences.
- The figure shows that a guide to deciding which response to take involves weighing the potential impact of the risk against the probability that it will occur.

6.3 Introduction to Blockchain

The Promise of Blockchain

- Blockchain is primarily known as the technology behind Bitcoin.
- It is gaining a lot of attention from those interested in finding better ways to secure transactions, including information exchanged by IoT devices.
- Blockchain is a technology that solves the problem of trust.
 - is a distributed ledger whose continuous growing list of records, called blocks, are linked together and secured using cryptography.
 - Immutable - unable to be changed.



Introduction to Blockchain

IoT and Blockchain

- Both IoT and blockchain are disruptive technologies – a product or service that enters the market with a vastly different, even revolutionary approach.
- Cisco Systems is one of the leading members of the Trusted IoT Alliance, a consortium of 17 companies to help establish a standard protocol for a blockchain-based IoT security solution.



Introduction to Blockchain

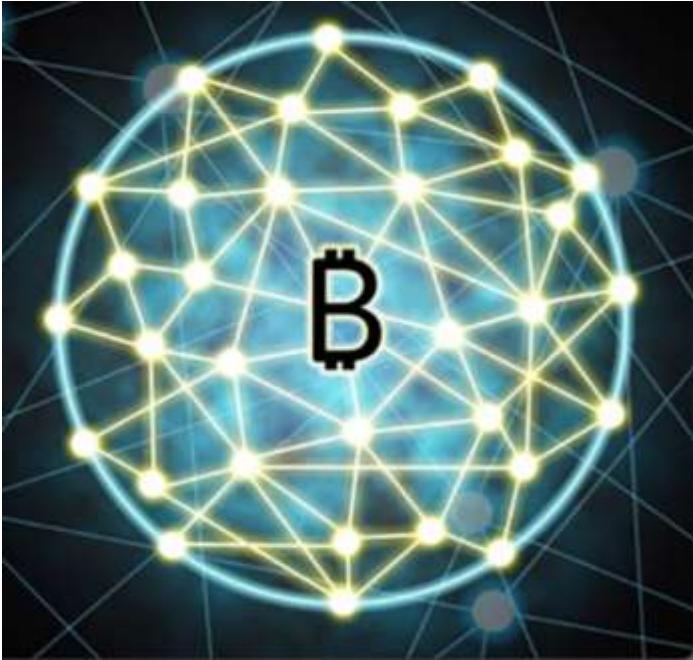
Current Trust Systems



- To best understand how blockchain technology works, we should first look at how trust works in our current monetary system.
 - When we purchase goods or services, both parties agree on the method of payment. Typically, traditional currency such as cash, debit card, credit card, or a check.
 - We rely on a third party, an intermediary, to guarantee the financial transaction between the buyer and the seller, for example, a bank, usually charging a fee.
 - These financial transactions are typically recorded in a single, centralized ledger which we trust is accurately maintained.
- The trust these intermediaries provide include:
 - Authenticating that the person making the transaction is who they say they are.
 - Ensuring that all transactions made to the ledger are accurate.
 - Not allowing any illegal transactions.

Introduction to Blockchain

Blockchain Trust System

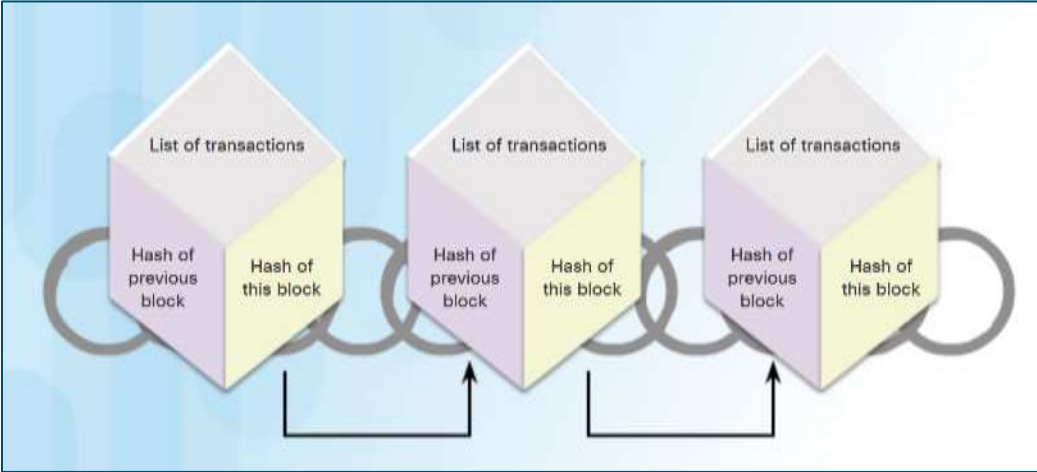


- Blockchain accomplishes trust in a very different manner.
- Cryptocurrencies such as Bitcoin do not use an intermediary to ensure the trust of the transaction.
- Instead, Bitcoin uses the blockchain itself to provide that trust between the buyer and the seller.
- This can be applied to any type of application that uses some type of transaction or ledger.

How Blockchain Works

Blockchain Features

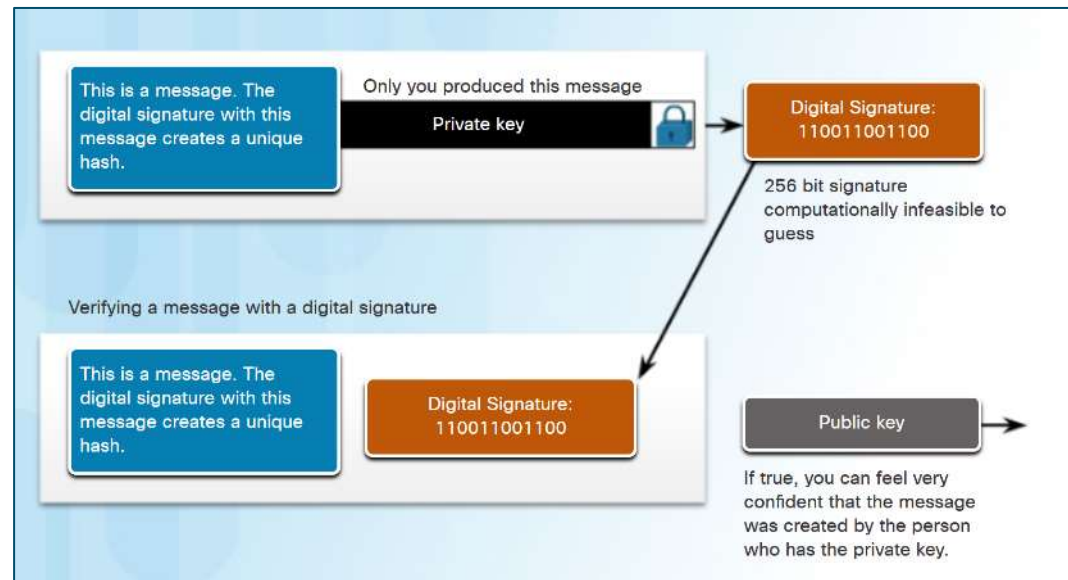
- Blockchain is a continuously growing list of transactions in the form of blocks. These blocks are linked and secured using cryptography.
- A blockchain uses the following:
 - Digital signatures
 - Decentralized ledger
 - An algorithm for reaching consensus
 - Each block includes the hash of the previous block, forming a chain of blocks known as a blockchain.



How Blockchain Works

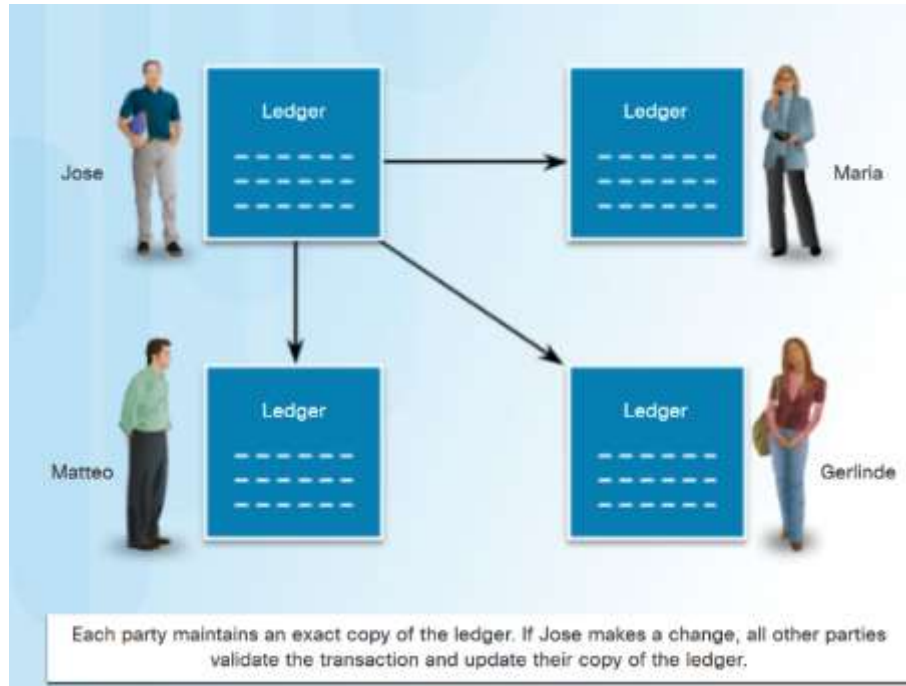
Digital Signature

- A digital signature is a mathematical scheme for demonstrating authenticating digital information.
- A digital signature cannot be copied because it is always different.
 - It uses the message or transaction to help derive the signature.
 - Changing the message even slightly makes the digital signature completely different.
- Digital signatures involve the message (or transaction), a private key, and a public key as shown in the figure.



How Blockchain Works

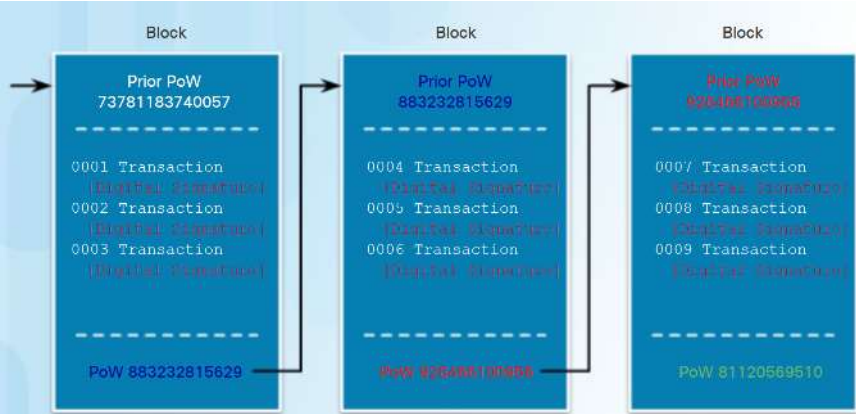
Decentralized Ledger



- Blockchain uses a decentralized ledger with all interested parties maintaining a copy.
- The trust is ensured by everyone receiving and believing any new transactions.
- Everyone must be using and working with the exact same ledger. This is done using a process known as Proof of Work.

How Blockchain Works

Reaching Consensus



- A block includes transactions along with their digital signatures.
- Validating transactions in a block uses a process known as Proof of Work (PoW).
- The PoW is an algorithm (hash) performed by computers that requires a large amount of computational work in relatively short amount of time. These computers are known as miners.
- A blockchain consists of blocks. Each block is a list of transactions, with a hash of the previous block and hash of this block including its PoW.
 - The hash is computed using the hash of the previous block (prior PoW), along with all the transactions in this block with their digital signatures.
 - This makes it computationally infeasible to modify a block or change the order of the blocks.

How Blockchain Works

Blockchain Applied to IoT Security

- Blockchain can be used to help solve many of the security and trust challenges for IoT:
 - Tracking sensor data measurements and preventing malicious data.
 - Providing IoT device identification, authentication, and secure data transfer.
 - Allow IoT sensors to exchange data directly with each other securely without an intermediary.
 - A distributed ledger eliminates a single source of failure within the IoT ecosystem.
 - IoT deployment is simplified and operation costs of IoT are reduced because there is no intermediary.
 - IoT devices are directly addressable with blockchain, providing an immutable history.



Chapter Summary

Summary

- **Vulnerability assessment:**
 - Vulnerability assessments can be routine and regularly conducted, often automated, or may be targeted at specific components of an IoT system.
 - The planning process includes defining the objectives and scope of the test, identifying the tools to be used, and determining who will perform the tests and a testing process.
- **Vulnerability testing types and tools:**
 - Vulnerability assessment can be classified into three different types; black box, white box, and gray box.
 - In penetration testing, skilled ethical hackers take on the role of threat actors and launch actual attacks that are meant to replicate what malicious hackers might do.
 - Open listening ports can provide access to the system by hackers.
- **Risk assessment concepts and approaches:**
 - The level of risk is dependent on the value of the asset, the vulnerability of that asset within the context of the software and systems on which they are used, and the likelihood that threats will be successfully executed against that asset.

Chapter Summary

Summary (Cont.)

- The numeric CVSS score can be used to determine the urgency of the vulnerability, and the priority of addressing it.
- The Base Metric Group Impact metrics increase with the degree or consequence of loss due to the impacted component.
- Assessing risk with threat modeling:
 - Three approaches to threat modelling: attack-centric, defense-centric, and asset-centric.
 - Threat modeling for risk assessment is a five step process.
 - Zones can be defined as areas of the system that require different authorization and authentication.
 - Trust boundaries delimit sections of the network where the level of trust between is different.
- Threat identification and risk prioritization:
 - The STRIDE approach provides a set of categories that are very helpful for identifying potential threats
 - The DREAD model should be applied to every threat that was uncovered during the threat identification process.
 - Each vulnerability identified by STRIDE is rated according the five DREAD categories.

Chapter Summary

Summary (Cont.)

- **Managing risk in IoT systems:**
 - A mandatory activity in risk assessment is the identification of threats and vulnerabilities and the matching of threats with vulnerabilities in what is often called threat-vulnerability (T-V) pairing.
 - Four potential ways to respond to risks that have been identified: terminate, treat, transfer, or tolerate.
- **Blockchain:**
 - A blockchain is a distributed ledger with a continuous growing list of records, called blocks, which are linked together and secured using cryptography.
 - Each block consists of several transactions, including its digital signature.
 - Blockchain uses a decentralized ledger with all interested parties maintaining a copy.
 - The Proof of Work is a hash performed by computers that requires a large amount of computational work in relatively short amount of time.

