CEH v12 Lesson 11 : Compromising IoT & OT platforms

Learning Outcomes

In this module, you will complete the following exercises:

- Exercise 1 — IoT Concepts
- Exercise 2 — IoT Device Exploitation and Prevention
- Exercise 3 — OT Concepts, Attacks, and Countermeasures

After completing this module, you will have further knowledge of:

- IoT Concepts
- IoT Technology Components
- IoT Architecture
- IoT Use Cases
- IoT Operating Systems
- IoT Communication Models IoT Security Issues
- OWASP Top 10 IoT Vulnerabilities
- OWASP IoT Attack Surfaces
- IoT Attacks
- IoT Hacking Methodology and Tools
- IoT Countermeasures Challenges of OT
- OT Vulnerabilities
- OT Attacks
- OT Hacking Methodology and Tools
- OT Countermeasures

Lab Duration

It will take approximately **30 minutes** to complete this lab.

Exercise 1 — IoT Concepts

The term IoT stands for the Internet of Things. IoT allows a device to be accessible on the Internet, such as a mobile device. IoT devices can communicate with each other and gather information that can be used for analysis.

For example, a smartwatch could track the number of hours of sleep and the number of steps a person takes. This data can then be analyzed by another IoT device, such as a mobile device, displaying the average number of hours of sleep and daily steps.

More recently, IoT devices have been used for different purposes. These include:

- Smart thermostats
- Medical devices
- Connected cars
- Activity trackers
- Smart refrigerators
- Parking sensors
- Medical sensors
- Smart security systems

Learning Outcomes

After completing this exercise, you will have further knowledge of:

- IoT Concepts
- IoT Technology Components
- IoT Architecture
- IoT Use Cases
- IoT Operating Systems

- IoT Communication Models

IoT Concepts

Each IoT device has an IP address that can be used for communication with the other IoT devices and remote monitoring and controlling. IoT devices use various sensors, such as:

- Temperature
- Humidity
- Acceleration
- Gyro
- Accelerometers

You can virtually turn computing device with the capability of autonomous computing into a smart device.

At present, IoT devices are being used across various industries, such as consumer electronics, travel, energy, healthcare, agriculture, facility automation, and security. For example, several housing complexes and office complexes use smart lighting. Lights in low traffic areas such as storerooms can be dimmed or turned off until traffic increases and then the lights will turn on or increase in brightness. This is smart facility automation.

IoT Technology Components

Different components in IoT technology work together to achieve a specific objective. These components are:

- **Sensors**: Devices that collect certain dataq such as temperature, heart rate etc. These devices will communicate this information to storage or analysis devices.
- **IoT Gateway**: The device that allows communication between sensors and the remote control, storage or analysis devices on the LAN / Cloud.
- **Storage**: A system that stores the data received from the IoT Gateway. It is typically a cloud location.

- **Remote Control**: An app on a mobile device or a web site that allows a user uses to interact with an IoT device. A user can achieve different tasks, such as viewing and retrieving historical data.

IoT Architecture

IoT architecture comprises five different layers where each layer has a distinct function to perform. The Edge Technology Layer is the topmost layer, followed by the Access Gateway, Internet, Middleware and Application Layers.

Edge Technology LayerAccess Gateway LayerInternet LayerMiddleware LayerApplication Layer

Figure 1.1: IoT Architecture, showing the Edge Technology, Access Gateway, Internet, Middleware and Application Layers

- **Edge Technology Layer**: Hardware components exist on this layer. Hardware can vary and can include things such as sensors and RFID tags. A device that has these hardware components is also included in this layer. Several tasks, such as data collection and device connectivity, are performed at this layer.
- **Access Gateway Layer**: Data handling takes place on this layer. Other tasks that are performed on this layer are message identification, message routing, and subscribing.
- **Internet Layer**: This layer is responsible for enabling communication between two devices and also enables device-to-gateway communication. It can be device-to-device communication or device-to-cloud.
- **Middleware Layer**: This layer is responsible for data-related tasks, such as aggregation and filtering. It is also responsible for access control.
- **Application Layer**: This layer is responsible for connecting with users to deliver services.

IoT Use Cases

IoT, as a technology, can be used in different places. It can be used in televisions and then even in fighter jets. Let’s look at some of the possible use cases of IoT across different domains:

BuildingsEnergyConsumerLife SciencesTransportation

- HVAC
- Transport
- Fire & Safety
- Security
- Access Control
- Lighting
- Turbines
- Windmills
- UPS
- Batteries
- Generators
- Meters
- Drills
- Digital Cameras
- Dishwashers
- Desktop computers
- Meters
- Lights
- Televisions
- Gaming consoles
- Alarms
- Surgical Equipment
- Pumps
- Monitors

- MRI Machines
- Vehicles
- Planes
- Tolls
- Signage
- Ships
- Lights

IndustrialRetailArmed ForcesIT Environment

- Fabrication Conveyors
- Valves
- Pumps
- Assembly
- Tanks
- POS Terminals
- Tags
- Vending Machines
- Signs
- Cash registers
- Fighter Jets
- Vehicles
- Ambulance
- Tanks
- Servers
- Storage

- Desktop Computers
- Routers
- Switches

IoT Operating Systems

An IoT operating system is installed on a device to function. It is like an embedded operating system within a device to work with system resources. It can be found in several IoT devices, such as digital cameras, elevators, ATMs, and cars.

Some of the key IoT operating systems are:

- Ubuntu Core
- RIOT
- Windows 10 IoT
- Amazon FreeRTOS
- Contiki
- Fuchsia
- ARM mbed OS
- Zephyr
- Nucleus RTOS
- Integrity RTOS
- NuttX RTOS

IoT Communication Models

IoT devices can communicate in different ways. It can be direct device-to-device communication or via a device connected with a cloud to store its data. Let's look at the different communication models used with IoT devices.

- **Device-to-Device**: in this communication model, two devices directly interact. These devices could be using different protocols such as ZigBee or Bluetooth. Examples can be wearable devices and CCTV cameras.

- **Device-to-Cloud**: IoT devices directly interact with the cloud to store data and receive commands. For example, if you need to inform the air conditioner to start at 5 PM, you could use a mobile app to send the commands through the cloud, not directly to the air conditioner.
- **Device-to-Gateway**: in this communication model, a mediator (such as a mobile app) plays the role of a gateway. Commands are passed to the IoT device through a mobile app.
- **Backend Data Sharing**: in this communication model, the IoT device stores data in the cloud, the backend. This model works in the data-sharing model, which involves third parties that use the data. Third parties may use the data for analysis, research, or other purposes.

Exercise 2 — IoT Device Exploitation and Prevention

IoT is a new technology that has surfaced and expanded greatly in recent years, and it is predicted that there are 35–40 billion IoT devices connected to the Internet as of the time of writing.

With such a large number of devices, the number of attacks has also risen, and it is crucial to use preventive measures for protection.

Despite their popularity and extensive use, the security of IoT devices remains a concern. Some of the key issues are:

- Hardware hacking via exposed ports
- Extracting information from flash memory
- Access to the root terminal using exposed ports
- Use of vulnerable Application Programming Interface (API)
- Insecure authentication and authorization
- Web interface vulnerabilities
- Hardcoded credentials
- Unnecessary and insecure services running
- Sniffing information from protocols, such as BLE and ZigBee

In this exercise, you will learn about IoT exploitation and prevention methods.

Learning Outcomes

After completing this exercise, you will know about:

- IoT Security Issues
- OWASP Top 10 IoT Vulnerabilities
- OWASP IoT Attack Surfaces
- IoT Attacks
- IoT Hacking Methodology and Tools
- IoT Countermeasures

IoT Security Issues

IoT devices, like any other computing device, have vulnerabilities that can be exploited. It could be default hard-coded credentials or even cleartext communication. These issues could make organizatoins vulenrablt to attacks.

Let's look at key IoT security issues at different levels within the IoT ecosystem.

- **Application**: default passwords, no security updates
- **Network**: no or improper communication data encryption, no security updates
- **Mobile**: no or improper communication data encryption, insecure APIs, data-at-rest encryption, weak authentication mechanism
- **Cloud**: weak authentication mechanism, no encryption for data-at-rest or data-in-transit

OWASP Top 10 IoT Vulnerabilities

Weak, Guessable, or Hardcoded PasswordsA password used with an IoT device is weak or easily guessable. Some IoT devices have hardcoded passwords used as backdoor access to the device's firmware or the installed software, which can allow an attacker unrestricted access if compromised.**Insecure Network Services**IoT devices can run unnecessary unsecure network services prone to an attack. If this happens, then the confidentiality, availability, and integrity of the information on the IoT device are at

stake. The danger to devices increases when they are exposed to a vulnerable network.**Insecure Ecosystem Interfaces**

IoT devices can be configured to use insecure web or mobile interfaces, which are generally weak or has no encryption being used. It could also be that there is a lack of authentication and authorization.

An insecure application programming interface (API) can also be an issue, leading to data compromise when the IoT device connects to the Internet and communicates with other devices.

Lack of Secure Update Mechanisms

Some common issues with secure update mechanisms include:

- There are no (or minimal) updates to IoT devices.
- Updates are not securely delivered to an IoT device. A man-in-the-middle attack can be conducted, and updates can be modified. This can include sending updates to a device without encrypting them.
- There are no or minimal security change notifications.

Use of Insecure or Outdated Components When outdated software or operating systems are being used, its vulnerabilities are known and often exploited.**Insufficient Privacy Protection**IoT devices can store a user's personal information. For example, a smartwatch can store the user's health information. The information must be securely stored with appropriate permissions. If the information is not encrypted, it can be hacked.**Insecure Data Transfer and Storage**

Another issue about the information stored in IoT devices is the lack of encryption applicable for the data in storage (at rest), during processing, or while in transit.

At all times, data must be secure and encrypted.

Lack of Device Management IoT devices can also lack security while they are deployed to users. The issue is also applicable for asset management, monitoring, and update management.**Insecure Default Settings**Like mobile devices, IoT devices also come with the factory default settings. Most users do not know that the factory default settings can be compromised. Therefore, IoT devices are used with their default settings,

allowing hackers to gain access. For example, each manufacturer uses a specific default password, which, if not changed, can be accessed by any hacker that finds the device on the Internet.

OWASP IoT Attack Surfaces

IoT devices are considered vulnerable due to a large attack surface that expands to communicate from the device to the network.

Some attack surfaces include:

IoT EcosystemMemoryPhysical InterfacesWeb interfacesFirmware

- System-wide failure
- Enrollment security
- Authentication
- Session management
- Handling of encryption keys
- Third-party credentials
- Cleartext user credentials (username and password)
- Privilege escalation
- Debug port
- Device ID exposure
- Firmware extraction
- User and admin command-line interface
- Weak passwords
- Account lockout
- Default credentials
- Web application vulnerabilities
- Hardcoded credentials

- Default user credentials
- Data exposure
- Encryption key management
- Backdoor user accounts

Network ServicesAdministrative InterfacesLocal Data StorageCloud Web InterfacesThird-party Backend APIs

- Denial-of-Service (DoS) attack
- Injection attacks
- Buffer overflow
- Unencrypted services
- User and admin command-line interface
- Web application vulnerabilities
- Scripting attacks
- Username enumeration
- Weak passwords
- Account lockout
- Known credentials
- Unencrypted data
- No integrity checks
- The static key for encryption and decryption of data
- Lack of proper credential management
- Known credentials
- Weak user credentials
- Cleartext data transmission

- Device information leakage
- Location information leakages

Update MethodsMobile ApplicationsVendor Backend APIsNetwork TrafficAuthentication and AuthorizationHardware

- Unencrypted updates transmission
- Unsigned updates
- No update verification
- Writable update location
- Unencrypted data transmission
- Unencrypted data storage
- Use of default credentials
- Weak user credentials
- Username enumeration
- Cleartext data transmission
- Implicit trust with cloud or mobile app
- Weak user credentials and authentication
- Weak access controls
- Injection attacks
- Protocol fuzzing
- Non-standard type
- Weak protocol usage
- Device-to-device
- Device-to-cloud
- Device-to-mobilen

- Mobile app-to-cloud
- Web application-to-cloud
- Physical damage
- Physical tampering

IoT Attacks

IoT devices are prone to several types of attacks. Some of the key attacks are:

- **DoS/DDoS**: is performed by hacking IoT devices and using them as bots.
- **Rolling Code**: is performed by sniffing traffic intended from an IoT device to a vehicle. An attacker extracts code from the traffic and steals the vehicle.
- **BlueBorne**: is an attack in which an infected device connects to the Bluetooth-enabled devices and infects them.
- **Jamming**: in this attack, an attacker jams the signal between two IoT devices, which prevents their communication.
- **Backdoor**: in this attack, an attacker converts an IoT device as a backdoor to get into a connected network.
- **SQL Injection**: an attacker exploits SQL vulnerabilities in a mobile app. After exploiting these vulnerabilities, an attacker can also control connected IoT devices.
- **Fault Injection**: an attacker introduces a fault behavior to an IoT device to exploit faults. An attacker can control the IoT device's security when these faults are exploited.
- **SDR-based**: an attacker captures the information floating on the IoT device and sends spam messages to IoT devices.
- **Network Pivoting**: an attacker exploits an IoT device and then, using this device, connects to a server. After connecting, the attacker pivots to the other servers and devices on the network.
- **Telnet Access**: an IoT device may have an open Telnet port. An attacker could exploit this port and then access stored data.

- **Sybil Attack**: an attacker creates forged identities in a peer-to-peer network to simulate heavy congestion.
- **Vulnerability Exploitation**: an attacker exploits open vulnerabilities on an IoT device.
- **Man-in-the-Middle (MITM**): an attacker intercepts traffic between two IoT devices.
- **Replaced Device**: an attacker replaces a legitimate IoT device with a malicious device.
- **Replay**: an attacker intercepts messages using MITM and then plays the intercepted messages to legitimate devices, which eventually becomes a DoS attack.

IoT Hacking Methodology and Tools

IoT hacking methodology is similar to any web attacking methodology, and an attacker would perform the same steps as they would perform in a web attack. For example, the attack would start with the Information Gathering phase, and proceed with Vulnerability Scanning, Launching Attacks, Gaining and Maintaining Access phases.

InformationGatheringVulnerabilityScanningLaunchAttacksGainAccessMaintainAccess

Figure 2.1: Diagram showing IoT Hacking Methodology, including Information Gathering, Vulnerability Scanning, Launching Attacks and Gaining / Maintaining Access phases.

- **Information Gathering**: An attacker looks for information that can exploit IoT devices. This information can include IP address, protocols, open ports, and the device's location.
- **Tools**: Shodan, Censys, Multiping, IoTSeeker, and Thingful
- **Vulnerability Scanning**: After the attacker has gathered the required information, the attacker looks for the vulnerabilities in the target IoT devices.
- **Tools**: Nmap, beSTORM, RIoT Vulnerability Scanner, and Foren6
- **Launch Attacks**: Based on which vulnerabilities are found, an attacker can launch different types of attacks. For example a MITM or rolling-code attack.
- **Tools**: RFCrack, DDoS, and Attify

- **Gain Access**: After initiating an attack, an attacker can access the targeted IoT device and gain remote access. If authentication is enabled, then an attacker can brute-force the password. An attacker would look for the open ports, and if Telnet is found to be open, it would be exploited.
- **Tool**: Telnet
- **Maintain Access**: After gaining access, an attacker needs to maintain access. To do this, they can modify the firmware of the targeted IoT device. After the firmware is modified, an attacker loads the firmware back onto the target IoT device.
- **Tools**: Firmware Mod Kit

IoT Countermeasures

Just as with a typical system or mobile device, you also need to protect IoT devices. There are several methods to do this, such as:

- Update firmware as and when required
- Block unnecessary open ports
- Disable Telnet, UPnP, and other vulnerable services
- Disable guest and demo accounts
- Use encrypted communication and use PKI if possible
- Use a strong password
- Use two-factor authentication
- Use drive encryption
- Configure user account lockout
- Perform periodic device assessment
- Use the secure password recovery
- Configure two-factor authentication
- Use secure coding practices to develop IoT applications
- Implement Intrusion Prevention System (IPS) and Intrusion Detection System (IDS)

- Ensure physical security of IoT devices
- Isolate IoT devices on a separate network

Exercise 3 — OT Concepts, Attacks, and Countermeasures

Operational Technology (OT) comprises hardware and software used to manage and monitor industrial operations. OT includes various components in an industrial environment, such as switches, systems, lights, CCTV cameras, cooling and heating systems. For example, a system runs a specific application to monitor a specific set of machines.

In this exercise, you will learn about OT attacks, hacking methodology, hacking tools, and countermeasures.

Learning Outcomes

After completing this exercise, you will know about:

- Challenges of OT
- OT Vulnerabilities
- OT Attacks
- OT Hacking Methodology and Tools
- OT Countermeasures

Challenges of OT

There can be various challenges of OT. For example, an OT device or system may not be updated, or the update may not be available. Some key challenges are as follows:

- **Lack of Visibility**: Organizations do not pay attention to security requirements necessary to safeguard their OT infrastructure. Without the proper security in place, it is difficult to have clear visibility and determine the risks beforehand. For example, if an organization cannot determine the risks or perform a risk assessment, the organization cannot be ready to handle risks proactively.
- **Plain-text Password**: OT networks that include various components may be using simple and plain-text passwords. Several organizations do not upgrade industrial machines managed by legacy applications. These applications may or may not use

complex passwords. Historically, most applications used plain-text passwords. If an organization uses a legacy application, it may be using a plain-text password.

- **Network Complexity**: An OT network consists of many devices, making it complex. It is not easy to define security requirements for a complex network. For example, a sensor may require a different level of security than a management application system.
- **Legacy Technology**: OT technology is expansive and, therefore, not regularly updated. For example, a machine may have a sensor that is a few years old. Therefore, the communication protocol that it uses may also be outdated.
- **Outdated or no Antivirus**: OT infrastructure is typically outdated and, therefore, may be running an outdated or no antivirus. It makes the entire OT infrastructure vulnerable.
- **Lack of Skills**: Organizations need skilled staff to manage an OT environment. However, since this is a specialized environment, not the typical IT environment, it is not easy to find a skilled workforce.
- **New Security Threats**: New security threats emerge daily. The OT infrastructure may not be regularly updated and, therefore, is more vulnerable to new security threats.
- **Insecure Connectivity**: OT networks may be using insecure wireless connectivity. For example, if the wireless network uses a weak wireless protocol, it is prone to a MITM attack.
- **Rogue Devices**: An attacker may install a rogue device within an OT network, which can further be used for initiating an attack.
- **Proprietary Software**: Several applications and hardware are proprietary in an OT environment. This causes the vendor lock-in situation. These devices are not easy to upgrade or update.
- **Communication Protocol Vulnerabilities**: Some of the key communication protocols, such as Modbus and Profinet, are used for monitoring and maintaining various components within an OT network. However, these protocols are vulnerable. For example, they lack authentication.

- **Remote Access**: OT devices and systems are accessed using different protocols, such as VNC, SSH, and RDP. The attacker can exploit the systems on the OT network after gaining remote access.

OT Vulnerabilities

OT networks are now interconnected with IT networks. Due to this, the attack surface increases and OT networks become vulnerable. Even though OT networks should be isolated, they often are not.

For example, an attack on an IT network can eventually provide access to the OT network through pivoting. Similarly, vulnerabilities in OT networks can also increase the attack surface. Let's look at some of the common vulnerabilities in OT networks.

- Due to the maintenance and management requirements, OT networks are mostly connected to the Internet. This is required for the third-party vendors to perform regular maintenance. However, OT networks are not protected with sufficient security controls. Because of the direct connectivity to the Internet, OT systems are prone to malware and Denial-of-Service (DoS) attacks.
- Passwords can be brute-forced to gain access to OT systems. It could also be possible that the default credentials are used for accessing OT systems. It is not difficult for the attacker to guess those weak passwords.
- Several organizations use jump boxes to allow access to OT networks. If there are vulnerabilities in jump boxes, an attacker can exploit them and access OT networks.
- It is not easy for organizations to keep up with the latest technology or update OT networks regularly. Such a limitation favors an attacker to discover an unpatched vulnerability and exploit it.
- Even though the main and OT networks are segregated, they need to have a firewall in between to allow access to either network. If a firewall is not properly configured or excess permissions are allowed, it is easy for an experienced attacker to access the OT network.
- It can be possible that an OT management console is installed on a system on a main network. If an attacker can penetrate the main network, then the management console can also control the OT network.

- An organization may have a flat network that contains OT and production systems on a single network. This allows an attacker to hack into one network and access both systems.
- If OT systems are not using updated wireless networks, an attacker can sniff the traffic.

OT Attacks

OT networks can be prone to several attacks. Some attacks can be due to existing vulnerabilities, and some can be due to misconfigurations. Let's look at some possible attack types:

- **Zero-day Vulnerabilities**: an attacker may discover a zero-day vulnerability and exploit it to gain administrative access to an OT network.
- **Unpatched Vulnerabilities**: is quite common for OT networks to remain unpatched — either patches or updates were never applied or, in some cases, even released by vendors. Attackers look out for such unpatched vulnerabilities. By exploiting these vulnerabilities, the attackers may disrupt the OT network functionality or plant malware into the network.
- **Data Theft or Leakage**: an attacker can steal data or send it to the command-and-control center after exploiting an OT network. Attackers can look for the configuration or confidential files.
- **Legacy Protocols Exploitation**: many OT networks still rely on legacy protocols such as Modbus, which attackers can exploit. After exploitation, they may gain access to administrative controls of the OT networks.
- **Remote Attacks**: most OT networks work with minimal or weak authentication, allowing an attacker to break it and gain access to the OT network.
- **DoS Attacks**: because OT systems are often directly connected to the internet, they may be prone to DoS attacks.
- **Human-Machine Interface (HMI)-based Attacks**: most HMI applications are not secure enough and are not protected with Defense-in-Depth, making them vulnerable to several attacks, such as memory corruption, code injection, buffer overflow, and authentication attacks.

- **Radio Frequency (RF) Attacks**: OT networks use RF to connect with various components. By default, RF technology has inherent vulnerabilities, making it easier for attackers to hack into a network and gain control. An attacker can conduct attacks such as replay, command injection, and malicious programming by injecting malware into the firmware.
- **Spear Phishing**: attackers can use social engineering to perform spear-phishing attacks. Once malicious attachments are downloaded and clicked, attackers enter the OT network.

OT Hacking Methodology and Tools

OT hacking methodology is the same as the IoT hacking methodology. It follows the same steps:

- **Information Gathering**: An attacker looks for information that can exploit OT networks. Information can include connected devices, open ports, and the device's location. For example, an attacker can use the Nmap tool to find the HMI systems.
- **Tools**: Shodan, Censys, CRITIFENCE, SCADAPASS, SCADA Shutdown Tool, RedPoint, s7scan, Kamerka-GUI, and Nmap
- **Vulnerability Scanning**: After an attacker has gathered the required information, they look for vulnerabilities in target devices connected to the OT network. Attackers can target protocols, applications, and systems to find vulnerabilities.
- **Tools**: Nessus, Skybox Vulnerability Control, NetworkMiner, GRASSMARLIN, CyberX, SmartRF Packet Sniffer, and Wireshark
- **Launch Attacks**: Based on vulnerabilities found, an attacker can launch different types of attacks. For example, depending on the type of vulnerability, they can be a MITM or a rolling-code attack. Other than using software tools, as mentioned below, the attacker can use hardware tools, such as Signal Analyzer, Oscilloscope, Multimeter, and Digital Microscope.
- **Tools**: GDB, OpenOCD, BinWalk, Fritzing, Radare2, OllyDbg, IDA Pro, Metasploit, ICS Exploitation Framework, PLCInject, Moki Linux, and modbus-cli
- **Gain Access**: After initiating an attack, an attacker can gain remote access of a targeted IoT device. If authentication is enabled, then an attacker can brute-force the

password. They look for the open ports, and if Telnet is found to be open, it is exploited.

- **Tool**: DNP3
- **Maintain Access**: After gaining access, an attacker needs to maintain access. To do this, they can modify the firmware of OT systems. After the firmware is modified, they load it back on to the OT network.

OT Countermeasures

Due to threats and vulnerabilities present in an OT network, several countermeasures can be implemented. Key countermeasures include:

- There should be defense-in-depth to protect an OT network.
- Since OT systems are publicly accessible, they should implement multi-factor authentication. Even if passwords are breached, the second factor can still protect a user's account. However, an organization should also enforce strong and complex password policies to prevent users from using simple passwords.
- Ingress and egress traffic is filtered using a firewall.
- Remote connectivity should be enabled through VPN or SSH.
- Device and system hardening practices should be implemented.
- OT networks should be segregated with limited access to individuals. No one should be allowed access unless necessary.
- A gateway should be implemented between IT and OT networks to filter traffic.
- Default user credentials should be changed at implementation from devices and applications.
- A regular risk assessment should be performed, and risk mitigation should be carefully implemented, keeping return-on-investment (ROI) in mind.
- Patching policies should be implemented. After patches are installed, there should be an audit.
- There should be regular vulnerability and penetration testing. Their results should be carefully reviewed and remediated.

- Information in storage and transmission should be protected using encryption.
- Strong wireless encryption protocols should be used.