**Lesson 12**

Implementing Host Security Solutions

Lesson Objectives

In this lesson, you will:

* Implement secure firmware.
* Implement endpoint security.
* Explain embedded system security implications.

**Topic 12A**

**Implement Secure Firmware**

The security of the hardware underpinning our network and computing devices is often overlooked. In part, this is because it is difficult for most companies to make their own investigations in this area. They have to rely on the market and security agencies to identify bad actors in supply chains. Nevertheless, it is important that you understand the issues involved in secure systems design so that you can evaluate product offerings and make recommendations for purchasing and device configuration.

**HARDWARE ROOT OF TRUST**

A **hardware Root of Trust (RoT)** or trust anchor is a secure subsystem that is able to provide attestation. Attestation means that a statement made by the system can be trusted by the receiver. For example, when a computer joins a network, it might submit a report to the network access control (NAC) server declaring, "My operating system files have not been replaced with malicious versions." The hardware root of trust is used to scan the boot metrics and OS files to verify their signatures, then it signs the report. The NAC server can trust the signature and therefore the report contests if it can trust that the signing entity's private key is secure.

The RoT is usually established by a type of cryptoprocessor called a **trusted platform module (TPM)**. TPM is a specification for hardware-based storage of encryption keys, hashed passwords, and other user and platform identification information. The TPM is implemented either as part of the chipset or as an embedded function of the CPU.

Each TPM is hard-coded with a unique, unchangeable asymmetric private key called the endorsement key. This endorsement key is used to create various other types of subkeys used in key storage, signature, and encryption operations. The TPM also supports the concept of an owner, usually identified by a password (though this is not mandatory). Anyone with administrative control over the setup program can take ownership of the TPM, which destroys and then regenerates its subkeys. A TPM can be managed in Windows via the tpm.msc console or through group policy. On an enterprise network, provisioning keys to the TPM might be centrally managed via the Key Management Interoperability Protocol (KMIP).

* The problem with establishing a hardware root of trust is that devices are used in environments where anyone can get complete control over them. There cannot be complete assurance that the firmware underpinning the hardware root of trust is inviolable, but attacks against trusted modules are sufficiently difficult so as to provide effective security in most cases.

**BOOT INTEGRITY**

Most PCs implement the **unified extensible firmware interface (UEFI)**. UEFI provides code that allows the host to boot to an OS. UEFI can enforce a number of boot integrity checks.

**Secure Boot**

**Secure boot** is designed to prevent a computer from being hijacked by a malicious OS. UEFI is configured with digital certificates from valid OS vendors. The system firmware checks the operating system boot loader and kernel using the stored certificate to ensure that it has been digitally signed by the OS vendor. This prevents a boot loader or kernel that has been changed by malware (or an OS installed without authorization) from being used. Secure boot is supported on Windows ([docs.microsoft.com/en-us/windows/security/information-protection/secure-the-windows-10-boot-process](https://docs.microsoft.com/en-us/windows/security/information-protection/secure-the-windows-10-boot-process)) and many Linux platforms ([wiki.ubuntu.com/UEFI/SecureBoot](https://wiki.ubuntu.com/UEFI/SecureBoot)). Secure boot requires UEFI, but does not require a TPM.

**Measured Boot**

A trusted or **measured boot** process uses platform configuration registers (PCRs) in the TPM at each stage in the boot process to check whether hashes of key system state data (boot firmware, boot loader, OS kernel, and critical drivers) have changed. This does not usually prevent boot, but it will record the presence of unsigned kernel-level code.

**Boot Attestation**

**Boot attestation** is the capability to transmit a boot log report signed by the TPM via a trusted process to a remote server, such as a network access control server. The boot log can be analyzed for signs of compromise, such as the presence of unsigned drivers. The host can be prevented from accessing the network if it does not meet the required health policy or if no attestation report is received.

**DISK ENCRYPTION**

**Full disk encryption (FDE)** means that the entire contents of the drive (or volume), including system files and folders, are encrypted. OS ACL-based security measures are quite simple to circumvent if an adversary can attach the drive to a different host OS. Drive encryption allays this security concern by making the contents of the drive accessible only in combination with the correct encryption key. Disk encryption can be applied to both hard disk drives (HDDs) and solid state drives (SSDs).

FDE requires the secure storage of the key used to encrypt the drive contents. Normally, this is stored in a TPM. The TPM chip has a secure storage area that a disk encryption program, such as Windows BitLocker, can write its keys to. It is also possible to use a removable USB drive (if USB is a boot device option). As part of the setup process, you create a recovery password or key. This can be used if the disk is moved to another computer or the TPM is damaged.

One of the drawbacks of FDE is that, because the OS performs the cryptographic operations, performance is reduced. This issue is mitigated by **self-encrypting drives (SED)**, where the cryptographic operations are performed by the drive controller. The SED uses a symmetric data/media encryption key (DEK/MEK) for bulk encryption and stores the DEK securely by encrypting it with an asymmetric key pair called either the authentication key (AK) or **key encryption key (KEK)**. Use of the AK is authenticated by the user password. This means that the user password can be changed without having to decrypt and re-encrypt the drive. Early types of SEDs used proprietary mechanisms, but many vendors now develop to the **Opal** Storage Specification ([nvmexpress.org/wp-content/uploads/TCGandNVMe\_Joint\_White\_Paper-TCG\_Storage\_Opal\_and\_NVMe\_FINAL.pdf](https://wmx-api-production.s3.amazonaws.com/courses/5731/supplementary/TCGandNVMe_Joint_White_Paper-TCG_Storage_Opal_and_NVMe_FINAL.pdf)), developed by the Trusted Computing Group (TCG).

**USB AND FLASH DRIVE SECURITY**

As revealed by researcher Karsten Nohl in his BadUSB paper ([srlabs.de/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf](https://wmx-api-production.s3.amazonaws.com/courses/5731/supplementary/SRLabs-BadUSB-BlackHat-v1.pdf)), exploiting the firmware of external storage devices, such as USB flash drives (and potentially any other type of firmware), presents adversaries with an incredible toolkit. The firmware can be reprogrammed to make the device look like another device class, such as a keyboard. In this case it could then be used to inject a series of keystrokes upon an attachment or work as a keylogger. The device could also be programmed to act like a network device and corrupt name resolution, redirecting the user to malicious websites.

Another example is the O.MG cable ([theverge.com/2019/8/15/20807854/apple-mac-lightning-cable-hack-mike-grover-mg-omg-cables-defcon-cybersecurity](https://www.theverge.com/2019/8/15/20807854/apple-mac-lightning-cable-hack-mike-grover-mg-omg-cables-defcon-cybersecurity)), which packs enough processing capability into an ordinary-looking USB-Lightning cable to run an access point and keylogger.

A modified device may have visual clues that distinguish it from a mass manufactured thumb drive or cable, but these may be difficult to spot. You should warn users of the risks and repeat the advice to never attach devices of unknown provenance to their computers and smartphones. If you suspect a device as an attack vector, observe a sandboxed lab system (sometimes referred to as a sheep dip) closely when attaching the device. Look for command prompt windows or processes such as the command interpreter starting and changes to the registry or other system files.

*Not all attacks have to be so esoteric. USB sticks infected with ordinary malware are still incredibly prolific infection vectors. Hosts should always be configured to prevent autorun when USB devices are attached. USB ports can be blocked altogether using most types of Host Intrusion Detection Systems (HIDS).*

**THIRD-PARTY RISK MANAGEMENT**

A root of trust is only trustworthy if the vendor has implemented it properly. Hardware and firmware vulnerabilities and exploits demonstrate the necessity of third-party risk management. A supply chain is the end-to-end process of supplying, manufacturing, distributing, and finally releasing goods and services to a customer. For example, for a Trusted Platform Module (TPM) to be trustworthy, the supply chain of chip manufacturers, firmware authors, OEM resellers, and administrative staff responsible for provisioning the computing device to the end user must all be trustworthy. Anyone with the time and resources to modify the computer's firmware could (in theory) create some sort of backdoor access. The same is true for any kind of computer or network hardware, right down to USB cables.

Establishing a trusted supply chain for computer equipment essentially means denying malicious actors the time or resources to modify the assets being supplied.

*For most businesses, use of reputable OEMs will represent the best practical effort at securing the supply chain. Government, military/security services, and large enterprises will exercise greater scrutiny. Particular care should be taken if use is made of second-hand machines.*

When assessing suppliers for risk, it is helpful to distinguish two types of relationship:

* Vendor—this means a supplier of commodity goods and services, possibly with some level of customization and direct support.
* Business partner—this implies a closer relationship where two companies share quite closely aligned goals and marketing opportunities.

For example, Microsoft is a major software vendor, but it is not feasible for it to establish direct relationships with all its potential customers. To expand its markets, it develops partner relationships with original equipment manufacturers (OEMs) and solution providers. Microsoft operates a program of certification and training for its partners, which improves product support and security awareness.

**END OF LIFE SYSTEMS**

When a manufacturer discontinues sales of a product, it enters an **end of life (EOL)** phase in which support and availability of spares and updates become more limited. An **end of service life (EOSL)** system is one that is no longer supported by its developer or vendor. EOSL products no longer receive security updates and so represent a critical vulnerability if any remain in active use.

For example, in Microsoft's support life cycle policy, Windows versions are generally given five years of mainstream support and five years of extended support (during which only security updates are shipped). You can check the support status for a particular version of Windows at [support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet](https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet).

Most OS and application vendors have similar policies. Care also needs to be taken with open source software. If the software is well-maintained, the development group will identify versions that have Long Term Support (LTS). Other builds and version branches might not receive updates.

It is also possible for both open source and commercial projects to be abandoned; if a company continues to rely on such abandonware, it will have to assume development responsibility for it. There are many instances of applications and devices (peripheral devices especially) that remain on sale with serious known vulnerabilities in firmware or drivers and no prospect of vendor support for a fix. The problem is also noticeable in consumer-grade networking appliances and in the Internet of Things. When provisioning a supplier for applications and devices, it is vital to establish that they have effective security management life cycles for their products.

**ORGANIZATIONAL SECURITY AGREEMENTS**

It is important to remember that although one can outsource virtually any service or activity to a third party, one cannot outsource legal accountability for these services or actions. You are ultimately responsible for the services and actions that these third parties take. If they have any access to your data or systems, any security breach in their organization (for example, unauthorized data sharing) is effectively a breach in yours. Issues of security risk awareness, shared duties, and contractual responsibilities can be set out in a formal legal agreement. The following types of agreements are common:

**Memorandum of understanding (MOU)**—A preliminary or exploratory agreement to express an intent to work together. MOUs are usually intended to be relatively informal and not to act as binding contracts. MOUs almost always have clauses stating that the parties shall respect confidentiality, however.

**Business partnership agreement (BPA)**—While there are many ways of establishing business partnerships, the most common model in IT is the partner agreements that large IT companies (such as Microsoft and Cisco) set up with resellers and solution providers.

Nondisclosure agreement (NDA)—Legal basis for protecting information assets. NDAs are used between companies and employees, between companies and contractors, and between two companies. If the employee or contractor breaks this agreement and does share such information, they may face legal consequences. NDAs are useful because they deter employees and contractors from violating the trust that an employer places in them.

**Service level agreement (SLA)**—A contractual agreement setting out the detailed terms under which a service is provided.

**Measurement systems analysis (MSA)**—quality management processes, such as Six Sigma, make use of quantified analysis methods to determine the effectiveness of a system. This can be applied to cybersecurity procedures, such as vulnerability and threat detection and response. A measurement systems analysis (MSA) is a means of evaluating the data collection and statistical methods used by a quality management process to ensure they are robust. This might be an onboarding requirement when partnering with enterprise companies or government agencies.

A legal agreement is all very well, but it is still up to you to make sure that your suppliers, vendors, and contractors can live up to it. If they can't, you may successfully sue them, but if they go out of business, you are still accountable for their actions or failures to act.

**Topic 12B**

**Implement Endpoint Security**

Host hardware integrity is not of much use if the OS and applications software running on it is weakly configured. As a security professional, you will often assist with drafting configuration baselines, ensuring hosts comply with those baselines, and implementing endpoint protection security agents.

**HARDENING**

The process of putting an operating system or application in a secure configuration is called **hardening**. When hardening a system, it is important to keep in mind its intended use, because hardening a system can also restrict the system's access and capabilities. The need for hardening must be balanced against the access requirements and usability in a particular situation.

For an OS functioning in a given role, there will usually be a fairly standard series of steps to follow to apply a secure configuration to allow the OS and applications software to execute that role. Many of the requirements can be applied automatically via a configuration baseline template. The essential principle is of least functionality; that a system should run only the protocols and services required by legitimate users and no more. This reduces the potential attack surface.

Interfaces provide a connection to the network. Some machines may have more than one interface. For example, there may be wired and wireless interfaces or a modem interface. Some machines may come with a management network interface card. If any of these interfaces are not required, they should be explicitly disabled rather than simply left unused.

Services provide a library of functions for different types of applications. Some services support local features of the OS and installed applications. Other services support remote connections from clients to server applications. Unused services should be disabled.

Application service ports allow client software to connect to applications over a network. These should either be disabled or blocked at a firewall if remote access is not required. Be aware that a server might be configured with a nonstandard port. For example, an HTTP server might be configured to use 8080 rather than 80. Conversely, malware may try to send nonstandard data over an open port. An intrusion detection system should detect if network data does not correspond to the expected protocol format.

Persistent storage holds user data generated by applications, plus cached credentials. Disk encryption is essential to data security. Self encrypting drives can be used so that all data-at-rest is always stored securely.

It is also important to establish a maintenance cycle for each device and keep up to date with new security threats and responses for the particular software products that you are running.

**BASELINE CONFIGURATION AND REGISTRY SETTINGS**

You will have separate configuration baselines for desktop clients, file and print servers, Domain Name System (DNS) servers, application servers, directory services servers, and other types of systems. In Windows, configuration settings are stored in the registry. On a Windows domain network, each domain-joined computer will receive policy settings from one or more group policy objects (GPOs). These policy settings are applied to the registry each time a computer boots. Where hosts are centrally managed and running only authorized apps and services, there should be relatively little reason for security-relevant registry values to change. Rights to modify the registry should only be issued to user and service accounts on a least privilege basis. A host-based intrusion detection system can be configured to alert suspicious registry events.

Baseline deviation reporting means testing the actual configuration of hosts to ensure that their configuration settings match the baseline template. On Windows networks, the Microsoft Baseline Security Analyzer (MBSA) tool was popularly used to validate the security configuration. MBSA and other Microsoft reporting tools have now been replaced by the Security Compliance Toolkit ([docs.microsoft.com/en-us/windows/security/threat-protection/security-compliance-toolkit-10](https://docs.microsoft.com/en-us/windows/security/threat-protection/security-compliance-toolkit-10)).

**PATCH MANAGEMENT**

No operating system, software application, or firmware implementation is wholly free from vulnerabilities. As soon as a vulnerability is identified, vendors will try to correct it. At the same time, attackers will try to exploit it. Automated vulnerability scanners can be effective at discovering missing **patches** for the operating system, plus a wide range of third-party software apps and devices/firmware. Scanning is only useful if effective procedures are in-place to apply the missing patches, however.

On residential and small networks, hosts will be configured to auto-update, meaning that they check for and install patches automatically. The major OS and applications software products are well-supported in terms of vendor-supplied fixes for security issues. In Windows, this process is handled by Windows Update, while in Linux it can be configured via yum-cron or apt unattended-upgrades, depending on the package manager used by the distribution. Enterprise networks need to be cautious about this sort of automated deployment, however, as a patch that is incompatible with an application or workflow can cause availability issues. In rare cases, such as the infamous SolarWinds hack (<https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack?t=1631031433646>), update repositories can be infected with malware that can then be spread via automated updates. There can also be performance and management issues when multiple applications run update clients on the same host. For example, as well as the OS updater, there is likely to be a security software update, browser updater, Java updater, OEM driver updater, and so on. These issues can be mitigated by deploying an enterprise **patch management** suite. Some suites, such as Microsoft’s System Center Configuration Manager (SCCM)/Endpoint Manager ([docs.microsoft.com/en-us/mem/configmgr](https://docs.microsoft.com/en-us/mem/configmgr/)), are vendor-specific while others are designed to support third-party applications and multiple OSes.

It can also be difficult to schedule patch operations, especially if applying the patch is an availability risk to a critical system. If vulnerability assessments are continually highlighting issues with missing patches, patch management procedures should be upgraded. If the problem affects certain hosts only, it could be an indicator of compromise that should be investigated more closely.

Patch management can also be difficult for legacy systems, proprietary systems, and systems from vendors without robust security management plans, such as some types of Internet of Things devices. These systems will need compensating controls, or some other form of risk mitigation if patches are not readily available.

**ENDPOINT PROTECTION**

Another crucial step in hardening is to configure endpoint protection for automatic detection and prevention of malware threats. There have been many iterations of host-based/endpoint protection suites and agents. It is important to consider the contrasting functions performed, as individual software tools or protection suites often combine multiple functionality.

**Antivirus (A-V)/Anti-Malware**

The first generation of antivirus (A-V) software is characterized by signature-based detection and prevention of known viruses. An "A-V" product will now perform generalized malware detection, meaning not just viruses and worms, but also Trojans, spyware, PUPs, cryptojackers, and so on. While A-V software remains important, signature-based detection is widely recognized as being insufficient for the prevention of data breaches.

**Host-Based Intrusion Detection/Prevention (HIDS/HIPS)**

Host-based intrusion detection systems (HIDS) provide threat detection via log and file system monitoring. HIDS come in many different forms with different capabilities, some of them preventative (HIPS). File system integrity monitoring uses signatures to detect whether a managed file image—such as an OS system file, driver, or application executable—has changed. Products may also monitor ports and network interfaces, and process data and logs generated by specific applications, such as HTTP or FTP.

**Endpoint Protection Platform (EPP)**

Endpoint protection usually depends on an agent running on the local host. If multiple security products install multiple agents (say one for A-V, one for HIDS, another for host-based firewall, and so on), they can impact system performance and cause conflicts, creating numerous technical support incidents and security incident false positives. An endpoint protection platform (EPP) is a single agent performing multiple security tasks, including malware/intrusion detection and prevention, but also other security features, such as a host firewall, web content filtering/secure search and browsing, and file/message encryption.

**Data Loss Prevention (DLP)**

Many EPPs include a data loss prevention (DLP) agent. This is configured with policies to identify privileged files and strings that should be kept private or confidential, such as credit card numbers. The agent enforces the policy to prevent data from being copied or attached to a message without authorization.

**Endpoint Protection Deployment**

While specific products vary widely in terms of features and implementation detail, some generic tasks to implement endpoint protection include:

1. Configure the management system to push the agent software and any updates to all desktops. This will require configuring permissions and firewall settings.
2. Assign hosts to appropriate groups for policy assignment. For example, client endpoints have very different security requirements to servers. While it may be appropriate to use a preventative mechanism to immediately isolate a client when a threat is detected, automatically doing this for a critical server could cascade to loss of functionality across the network.
3. Test the different host group configuration settings to ensure that the expected range of threats is detected.
4. Use a monitoring dashboard to verify status across all network hosts. Apart from detection events, if the agent is disabled or missing, there should be an alert.

**NEXT-GENERATION ENDPOINT PROTECTION**

Where EPP provides mostly signature-based detection and prevention, next-generation endpoint protection with automated response is focused on logging of endpoint observables and indicators combined with behavioral- and anomaly-based analysis.

### Endpoint Detection and Response (EDR)

An **endpoint detection and response (EDR)** product's aim is not to prevent initial execution, but to provide real-time and historical visibility into the compromise, contain the malware within a single host, and facilitate remediation of the host to its original state. The term EDR was coined by Gartner security researcher Anton Chuvakin, and Gartner produces annual "Magic Quadrant" reports for both EPP ([gartner.com/en/documents/3848470](https://www.gartner.com/en/documents/3848470)) and EDR functionality within security suites ([gartner.com/en/documents/3894086/market-guide-for-endpoint-detection-and-response-solutio](https://www.gartner.com/en/documents/3894086/market-guide-for-endpoint-detection-and-response-solutio)).

Where earlier endpoint protection suites report to an on-premises management server, next-generation endpoint agents are more likely to be managed from a cloud portal and use artificial intelligence (AI) and machine learning to perform user and entity behavior analysis. These analysis resources would be part of the security service provider's offering.

### Next-Generation Firewall Integration

An analytics-driven next-gen antivirus product is likely to combine with the perimeter and zonal security offered by next-gen firewalls. For example, detecting a threat on an endpoint could automate a firewall policy to block the covert channel at the perimeter, isolate the endpoint, and mitigate risks of the malware using lateral movement between hosts. This type of functionality is set out in more detail in Sophos's white paper on synchronized security ([sophos.com/en-us/lp/synchronized-security.aspx](https://www.sophos.com/en-us/lp/synchronized-security.aspx)).

**ANTIVIRUS RESPONSE**

An on-access antivirus scanner or intrusion prevention system works by identifying when processes or scripts are executed and intercepting (or hooking) the call to scan the code first. If the code matches a signature of known malware or exhibits malware-like behavior that matches a heuristic profile, the scanner will prevent execution and attempt to take the configured action on the host file (clean, quarantine, erase, and so on). An alert will be displayed to the user and the action will be logged (and also may generate an administrative alert). The malware will normally be tagged using a vendor proprietary string and possibly by a CME (Common Malware Enumeration) identifier. These identifiers can be used to research the symptoms of and methods used by the malware. This may help to confirm the system is fully remediated and to identify whether other systems have been infected. It is also important to trace the source of the infection and ensure that it is blocked to prevent repeat attacks and outbreaks.

### Advanced Malware Tools

Malware is often able to evade detection by automated scanners. Analysis of SIEM and intrusion detection logs might reveal suspicious network connections, or a user may observe unexplained activity or behavior on a host. When you identify symptoms such as these, but the AV scanner or EPP agent does not report an infection, you will need to analyze the host for malware using advanced tools.

There is a plethora of advanced analysis and detection utilities, but the starting point for most technicians is Sysinternals ([docs.microsoft.com/en-us/sysinternals](https://docs.microsoft.com/en-us/sysinternals/)).

### Sandboxing

Sandboxing is a technique that isolates an untrusted host or app in a segregated environment to conduct tests. Sandbox environments intentionally limit interfaces with the host environment. The analysis of files sent to a sandbox can include determining whether the file is malicious, how it might have affected certain systems if run outside of the sandbox, and what dependencies it might have with external files and hosts. Sandboxes offer more than traditional anti-malware solutions because you can apply a variety of different environments to the sandbox instead of just relying on how the malware might exist in your current configuration.

**Topic 12C**

**Explain Embedded System Security Implications**

As well as the obvious computing hosts within your networks, you must also account for the security of embedded systems. Embedded computing functionality can be found in consumer electronics devices and in specialist monitoring and control systems, so it is important that you know how to identify and secure these devices.

**EMBEDDED SYSTEMS**

An **embedded system** is a complete computer system that is designed to perform a specific, dedicated function. These systems can be as contained as a microcontroller in an intravenous drip-rate meter or as large and complex as the network of control devices managing a water treatment plant. Embedded systems can be characterized as static environments. A PC is a dynamic environment. The user can add or remove programs and data files, install new hardware components, and upgrade the operating system. A static environment does not allow or require such frequent changes.

In terms of security this can be ideal, because unchanging environments are typically easier to protect and defend. Static computing environments pose their own risks, however. A static environment is often a black box to security administrators. Unlike an OS environment such as Windows, there may be little support for identifying and correcting security issues.

### Cost, Power, and Compute Constraints

Embedded systems are usually constrained in terms of processor capability (cores and speed), system memory, and persistent storage. Cost is an important factor. As devices may be used in large numbers and are designed for fairly predictable processing workloads, there is no obvious reason to over-provision compute resources and the price per unit can be driven as low as possible.

The other factor determining compute resources is power. Many embedded devices are battery-powered, and may need to run for years without having to replace the cells. This means that processing must be kept to the minimum possible level.

### Crypto, Authentication, and Implied Trust Constraints

The lack of computer resources means that embedded systems are not well-matched to the cryptographic identification and authentication technologies that are widely used on computer networks. As embedded systems become more accessible via those networks, however, they need to use cryptoprocessors to ensure confidentiality, integrity, and availability. This is prompting the development of ciphers that do not require such large processing resources.

On PC hardware, a root of trust is established at the hardware level by a TPM. Without this explicit trust anchor, a network has to use an implied trust model. Implied trust means that every device that has been added to the network is trusted, on the assumption that it was added and continues to be operated by a legitimate administrator. Until there is widespread adoption of embedded TPM, embedded networks have to rely on the perimeter security model.

### Network and Range Constraints

Minimizing compute functions also has an impact on choices for network connectivity. The Wi-Fi and 4G/5G standards developed for use with computer and smartphone networking use power-hungry antennas to maximize data rates and range, plus processing to encrypt the communications. Networks for embedded systems emphasize power-efficient transfer of small amounts of data with a high degree of reliability and low latency.

**LOGIC CONTROLLERS FOR EMBEDDED SYSTEMS**

Embedded systems are normally based on firmware running on a **programmable logic controller (PLC)**. These PLCs are built from different hardware and OS components than some desktop PCs.

### System on Chip (SoC)

Desktop computer system architecture uses a generalized CPU plus various other processors and controllers and system memory, linked via the motherboard. **System on chip (SoC)** is a design where all these processors, controllers, and devices are provided on a single processor die (or chip). This type of packaging saves space and is usually power efficient, and so is very commonly used with embedded systems.

**Raspberry Pi** ([raspberrypi.org](https://www.raspberrypi.org/)) and **Arduino** ([arduino.cc](https://www.arduino.cc/)) are examples of SoC boards, initially devised as educational tools, but now widely used for industrial applications, and hacking.

### Field Programmable Gate Array (FPGA)

A microcontroller is a processing unit that can perform sequential operations from a dedicated instruction set. The instruction set is determined by the vendor at the time of manufacture. Software running on the microcontroller has to be converted to these instructions (assembly language). As many embedded systems perform relatively simple but repetitive operations, it can be more efficient to design the hardware controller to perform only the instructions needed. One example of this is the application-specific integrated circuits (ASICs) used in Ethernet switches. ASICs are expensive to design, however, and work only for a single application, such as Ethernet switching.

A **field programmable gate array (FPGA)** is a type of controller that solves this problem. The structure of the controller is not fully set at the time of manufacture. The end customer can configure the programming logic of the device to run a specific application.

### Real-Time Operating Systems (RTOS)

Many embedded systems operate devices that perform acutely time-sensitive tasks, such as drip meters or flow valves. The kernels or operating systems that run these devices must be much more stable and reliable than the OS that runs a desktop computer or server. Embedded systems typically cannot tolerate reboots or crashes and must have response times that are predictable to within microsecond tolerances. Consequently, these systems often use differently engineered platforms called **real-time operating systems (RTOS)**. An RTOS should be designed to have as small an attack surface as possible. An RTOS is still susceptible to CVEs and exploits, however.

**EMBEDDED SYSTEMS COMMUNICATIONS CONSIDERATIONS**

Historically, embedded systems used proprietary vendor communications technologies. As technologies improve and closer integration with IT networks becomes more important, greater use of standardized communication technologies is becoming more prevalent.

**Operational Technology (OT) Networks**

A cabled network for industrial applications is referred to as an operational technology (OT) network. These typically use either serial data protocols or industrial Ethernet. Industrial Ethernet is optimized for real-time, deterministic transfers. Such networks might use vendor-developed data link and networking protocols, as well as specialist application protocols.

**Cellular Networks**

A cellular network enables long-distance communication over the same system that supports mobile and smartphones. This is also called **baseband radio**, after the baseband processor that performs the function of a cellular modem. There are several baseband radio technologies:

* Narrowband-IoT (NB-IoT)—this refers to a low-power version of the Long Term Evolution (LTE) or 4G cellular standard. The signal occupies less bandwidth than regular cellular. This means that data rates are limited (20-100 kbps), but most sensors need to send small packets with low latency, rather than making large data transfers. Narrowband also has greater penetrating power, making it more suitable for use in inaccessible locations, such as tunnels or deep within buildings, where ordinary cellular connectivity would be impossible.
* LTE Machine Type Communication (LTE-M)—this is another low-power system, but supports higher bandwidth (up to about 1 Mbps).

While not yet completely standardized, both NB-IoT and LTE-M are designed to be compatible with 5G networks. This means they do not interfere with 5G signaling and can use tower relays developed for 5G. They may support higher data rates, though latency and reliability tend to be more important considerations.

Any LTE-based cellular radio uses a **subscriber identity module (SIM)** card as an identifier. The SIM is issued by a cellular provider, with roaming to allow use of other suppliers' tower relays. As a removable card is not really a suitable form factor for embedded, an eSIM incorporates the same function as a chip on the system board or SoC design.

Encryption of frames between the endpoint and the cell tower and within the backhaul to Internet routers is the responsibility of the network operator. Over-the-air encryption is performed by encryption schemes devised by the cellular standards body 3GPP. Backhaul security is usually enforced using IPSec. The embedded system can use application layer encryption for additional security.

**Z-Wave and Zigbee**

Z-Wave and Zigbee are wireless communications protocols used primarily for home automation. Both create a mesh network topology, using low-energy radio waves to communicate from one appliance to another. In **Z-Wave**, devices can be configured to work as repeaters to extend the network but there is a limit of four "hops" between a controller device and an endpoint. Z-Wave uses ~900 Mhz frequencies.

**Zigbee** has similar uses to Z-Wave and is an open source competitor technology to it. The Zigbee Alliance operates certification programs for its various technologies and standards. Zigbee uses the 2.4 GHz frequency band. This higher frequency allows more data bandwidth at the expense of range compared to Z-Wave and the greater risk of interference from other 2.4 GHz radio communications. Zigbee supports more overall devices within a single network and there is no hop limit for communication between devices.

Both Z-Wave and Zigbee have communications encryption. The main threats are from re-pairing attacks and from rogue devices. A re-pairing attack allows a threat actor to discover the network key by forcing a device off the network, causing it to try to re-connect ([checkpoint.com/press/2020/the-dark-side-of-smart-lighting-check-point-research-shows-how-business-and-home-networks-can-be-hacked-from-a-lightbulb](https://www.checkpoint.com/press/2020/the-dark-side-of-smart-lighting-check-point-research-shows-how-business-and-home-networks-can-be-hacked-from-a-lightbulb/)). If the user connects a rogue device to the network, the system depends on application-level security to prevent the device from compromising higher value targets, such as a smart hub, alarm, or door entry mechanism.

**INDUSTRIAL CONTROL SYSTEMS**

Industrial systems have different priorities to IT systems. Often, hazardous electromechanical components are involved, so safety is the overriding priority. Industrial processes also prioritize availability and integrity over confidentiality—reversing the CIA triad as the AIC triad.

**Workflow and Process Automation Systems**

**Industrial control systems (ICSs)** provide mechanisms for workflow and process automation. These systems control machinery used in critical infrastructure, like power suppliers, water suppliers, health services, telecommunications, and national security services. An ICS that manages process automation within a single site is usually referred to as a distributed control system (DCS).

An ICS comprises plant devices and equipment with embedded PLCs. The PLCs are linked either by an OT fieldbus serial network or by industrial Ethernet to actuators that operate valves, motors, circuit breakers, and other mechanical components, plus sensors that monitor some local state, such as temperature. Output and configuration of a PLC is performed by one or more **human-machine interfaces (HMIs)**. An HMI might be a local control panel or software running on a computing host. PLCs are connected within a control loop, and the whole process automation system can be governed by a control server. Another important concept is the **data historian**, which is a database of all the information generated by the control loop.

**Supervisory Control and Data Acquisition (SCADA)**

A **supervisory control and data acquisition (SCADA)** system takes the place of a control server in large-scale, multiple-site ICSs. SCADA typically run as software on ordinary computers, gathering data from and managing plant devices and equipment with embedded PLCs, referred to as field devices. SCADA typically use WAN communications, such as cellular or satellite, to link the SCADA server to field devices.

**ICS/SCADA Applications**

These types of systems are used within many sectors of industry:

* Energy refers to power generation and distribution. More widely, utilities includes water/sewage and transportation networks.
* Industrial can refer specifically to the process of mining and refining raw materials, involving hazardous high heat and pressure furnaces, presses, centrifuges, pumps, and so on.
* Fabrication and manufacturing refer to creating components and assembling them into products. Embedded systems are used to control automated production systems, such as forges, mills, and assembly lines. These systems must work to extremely high precisions.
* Logistics refers to moving things from where they were made or assembled to where they need to be, either within a factory or for distribution to customers. Embedded technology is used in control of automated transport and lift systems plus sensors for component tracking.
* Facilities refers to site and building management systems, typically operating automated heating, ventilation, and air conditioning (HVAC), lighting, and security systems.

ICS/SCADA was historically built without regard to IT security, though there is now high awareness of the necessity of enforcing security controls to protect them, especially when they operate in a networked environment.

**INTERNET OF THINGS**

The term **Internet of Things (IoT)** is used to describe a global network of appliances and personal devices that have been equipped with sensors, software, and network connectivity. This compute functionality allows these objects to communicate and pass data between themselves and other traditional systems like computer servers. This is often referred to as Machine to Machine (M2M) communication. Each “thing” is identified with some form of unique serial number or code embedded within its own operating or control system and is able to inter-operate within the existing Internet infrastructure either directly or via an intermediary. An IoT network will generally use the following types of components:

* Hub/control system—IoT devices usually require a communications hub to facilitate Z-Wave or Zigbee networking. There must also be a control system, as most IoT devices are headless, meaning they have no user control interface. This could be a smart hub, with voice control, or a smartphone/PC app.
* Smart devices—IoT endpoints implement the function, such as a smart lightbulb or a video entryphone that you can operate remotely. These devices implement compute, storage, and network functions that are all potentially vulnerable to exploits. Most smart devices use a Linux or Android kernel. Because they're effectively running mini-computers, smart devices are vulnerable to some of the standard attacks associated with web applications and network functions. Integrated peripherals such as cameras or microphones could be compromised to facilitate surveillance.
* Wearables—some IoT devices are designed as personal accessories, such as smart watches, bracelets and pendant fitness monitors, and eyeglasses. Current competing technologies are based on FitBit, Android Wear OS, Samsung's Tizen OS, and Apple iOS, each with their own separate app ecosystems.
* Sensors—IoT devices need to measure all kinds of things, including temperature, light levels, humidity, pressure, proximity, motion, gas/chemicals/smoke, heart/breathing rates, and so on. These are implemented as thermocouples/thermistors, infrared detectors, inductive, photoelectric, and capacitative cells, accelerometers, gyroscopes, and more.

Home automation products often use vendor-specific software and networking protocols. As with embedded devices, security features can be poorly documented, and patch management/security response processes of vendors can be inadequate. When they are designed for residential use, IoT devices can suffer from weak defaults. They may be configured to "work" with a minimum of configuration effort. There may be recommended steps to secure the device that the customer never takes.

**SPECIALIZED SYSTEMS FOR FACILITY AUTOMATION**

A specialized system refers to the use of embedded systems and/or IoT devices for a specific purpose or application.

**Building Automation System (BAS)**

A **building automation system (BAS)** for offices and data centers ("smart buildings") can include physical access control systems, but also heating, ventilation, and air conditioning (HVAC), fire control, power and lighting, and elevators and escalators. These subsystems are implemented by PLCs and various types of sensors that measure temperature, air pressure, humidity, room occupancy, and so on. Some typical vulnerabilities that affect these systems include:

* Process and memory vulnerabilities, such as buffer overflow, in the PLCs. These may arise from processing maliciously crafted packets in the automation management protocol. Building automation uses dedicated network protocols, such as BACnet or Dynet.
* Use of plaintext credentials or cryptographic keys within application code.
* Code injection via the graphical web application interfaces used to configure and monitor systems. This can be used to perform JavaScript-based attacks, such as clickjacking and cross-site scripting (XSS).

It is possible that control of these systems could be used to perform some sort of DoS or ransom demand (consider disrupting HVAC controls within a data center, for instance). However, as with the Target data breach, the aim is likely to access the corporate data network from the automation and monitoring system, which may be accessible via a supplier company ([krebsonsecurity.com/tag/fazio-mechanical](https://krebsonsecurity.com/tag/fazio-mechanical/)).

**Smart Meters**

A **smart meter** provides continually updating reports of electricity, gas, or water usage to the supplier, reducing the need for manual inspections. Most meters use cellular data for communication back to the supplier, and an IoT protocol, such as ZigBee, for integration with smart appliances.

**Surveillance Systems**

A physical access control system (PACS) is a network of monitored locks, intruder alarms, and **video surveillance**. A PACS can either be implemented as part of a building automation system or a separate system in its own right. Gaining physical access to premises, or even just access to video monitoring systems, gives an adversary many opportunities to develop additional attacks. As with building automation, a PACS is likely to be installed and maintained by an external supplier. This can lead to it being omitted from risk and vulnerability assessments, as highlighted by the US Government Accountability Office's 2014 report into PACS at federal offices ([gao.gov/assets/670/667512.pdf](https://wmx-api-production.s3.amazonaws.com/courses/5731/supplementary/667512.pdf)).

Physical security systems use networked camera systems (CCTV) for surveillance. Unfortunately, some makes of camera systems have been found to have numerous serious vulnerabilities that allow attackers either to prevent intrusions from being recorded or to hijack the cameras to perform their own surveillance. These issues tend to affect cheap consumer-grade systems rather than enterprise models, but in both cases it is necessary to evaluate the supplier to demonstrate that their security monitoring and remediation support services are effective.

**SPECIALIZED SYSTEMS IN IT**

There are also specialized systems installed within office networks, such as printer and Voice over IP (VoIP) equipment. These systems must not be overlooked by security monitoring procedures.

### Multifunction Printers (MFPs)

Most modern print devices, scanners, and fax machines have hard drives and sophisticated firmware, allowing their use without attachment to a computer and over a network. Often these print/scan/fax functions are performed by single devices, referred to as **multifunction printers (MFPs)**. Unless they have been securely deleted, images and documents are frequently recoverable from all of these machines. Some of the more feature-rich, networked printers and MFPs can also be used as a pivot point to attack the rest of the network. These machines also have their own firmware that must be kept patched and updated.

### Voice over IP (VoIP)

Types of embedded systems are used to implement both Voice over IP (VoIP) endpoints and media gateways. Endpoints can be individual handsets or conferencing units. A media gateway might use a separate firmware/OS to implement integration with telephone and cellular networks.

**SPECIALIZED SYSTEMS FOR VEHICLES AND DRONES**

Automobiles and unmanned aerial vehicles (UAV), or drones, contain sophisticated electronics to control engine and power systems, braking and landing, and suspension/stability. Modern vehicles are increasingly likely to have navigation and entertainment systems, plus driver-assist or even driverless features, where the vehicle's automated systems can take control of steering and braking. The locking, alarm, and engine immobilizer mechanisms are also likely to be part of the same system. Each of these subsystems is implemented as an electronic control unit (ECU), connected via one or more controller area network (CAN) serial communications buses. The principal external interface is an Onboard Diagnostics (OBD-II) module. The OBD-II also acts as a gateway for multiple CAN buses.

The **CAN bus** operates in a somewhat similar manner to shared Ethernet and was designed with just as little security. ECUs transmit messages as broadcast so they are received by all other ECUs on the same bus. There is no concept of source addressing or message authentication. An attacker able to attach a malicious device to the OBD-II port is able to perform DoS attacks against the CAN bus, threatening the safety of the vehicle. There are also remote means of accessing the CAN bus, such as via the cellular features of the automobile's navigation and entertainment system ([wired.com/2015/07/hackers-remotely-kill-jeep-highway](https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/)). Some vehicles also implement on-board Wi-Fi, further broadening the attack surface.

**SPECIALIZED SYSTEMS FOR MEDICAL DEVICES**

Medical devices represent an array of systems potentially vulnerable to a wide range of attacks. It is important to recognize that use of these devices is not confined to hospitals and clinics but includes portable devices such as cardiac monitors/defibrillators and insulin pumps. As well as unsecure communication protocols, many of the control systems for these devices run on unsupported versions of operating systems (such as Windows XP) because the costs of updating the software to work with newer OS versions is high and disruptive to patient services. Some of the goals of attacks on medical devices and services are as follows:

* Use compromised devices to pivot to networks storing medical data with the aim of stealing protected health information (PHI).
* Hold medical units ransom by threatening to disrupt services.
* Kill or injure patients (or threaten to do so) by tampering with dosage levels or device settings.

**SECURITY FOR EMBEDDED SYSTEMS**

Embedded systems must not be overlooked when designing the security system. The following methods can be used to mitigate risk in such environments.

**Network Segmentation**

Network segmentation is one of the core principles of network security. Network access for static environments should only be required for applying firmware updates and management controls from the host software to the devices and for reporting status and diagnostic information from the devices back to the host software. This control network should be separated from the corporate network using firewalls and VLANs.

With environments such as SCADA, the management software may require legacy versions of operating systems, making the hosts particularly difficult to secure. Isolating these hosts from others through network segmentation and using endpoint security (preventing the attachment of USB devices) can help to ensure they do not become infected with malware or exposed to network exploits.

**Wrappers**

One way of increasing the security of data in transit for embedded systems is through the use of wrappers, such as IPSec. The only thing visible to an attacker or anyone sniffing the wire is the IPSec header, which describes only the tunnel endpoints. This is useful for protecting traffic between trusted networks when the traffic has to go through an untrusted network to go between them, or between trusted nodes on the same network.

**Firmware Code Control and Inability to Patch**

Embedded systems demonstrate one of the reasons that supply chain risks must be carefully managed. Programming logic implemented in FPGA and firmware code must not contain backdoors. Firmware patching is just as vital as keeping host OS software up to date, but for many embedded systems, it is far more of a challenge:

* Many embedded systems and IoT devices use low-cost firmware chips and the vendor never produces updates to fix security problems or only produces updates for a relatively short product cycle (while the device could remain in operational use for much longer).
* Many embedded systems require manual updates, which are perceived as too time-consuming for a security department with other priorities to perform.
* Availability is a key attribute for most embedded deployments. Patching without service interruption may not be possible, and opportunities for downtime servicing extremely limited.