“Is There an End in Sight?”

(Endpoint Security)

Tuesday, November 5th

1. Learning Outcomes

1. **Understand Endpoint Protection Technologies**: Identify and describe key endpoint protection technologies, including anti-malware software, endpoint detection and response (EDR), and data loss prevention (DLP), and discuss their roles in securing diverse endpoints against threats.
2. **Analyze Host-based Security Measures**: Explain the purpose and functionality of host-based firewalls, intrusion detection systems (HIDS), and intrusion prevention systems (HIPS) in protecting individual endpoints, including their differences and how they contribute to endpoint security.
3. **Identify Endpoint Hardening Practices**: Demonstrate knowledge of endpoint hardening techniques, including software updates, patch management, and port management, to minimize vulnerabilities and secure endpoints effectively against unauthorized access and malware.

2. Endpoint

1. An **endpoint** is:
2. **Unified Endpoint Management (UEM)** helps manage and secure endpoints from a single console, regardless of the device type or operating system. UEM allows updates, security policy applications, and remote data removal for lost or compromised devices.
3. **Endpoint Protection** is:
4. **Endpoint Detection and Response (EDR)** specifically monitors:
   1. **Whitelisting** is a security measure that restricts system access to only pre-approved applications, files, or users. In endpoint security, whitelisting allows only trusted software or processes to run on devices (endpoints) like computers, smartphones, and servers, effectively blocking any other software or processes from executing.
5. Each endpoint type, from mobile devices to smart IoT gadgets, requires a tailored security approach. For example, IoT devices, like smart meters, need frequent updates and checks due to their limited security features.
6. **Malware Protection** defends against:
7. **Next-generation firewalls (NGFWs)** offer:
   1. **Anti-virus and Antimalware –** stop viruses, worms, trojans; spyware, ransomware, fileless malware.
   2. **NGFW** can be called:
      1. Application layer gateway
      2. Stateful multilayer inspection
      3. Deep packet inspection
      4. Can prevent access to URLs, identify attacks and malware, allow or disallow application features.

3. Endpoint protection

1. **Data Loss Prevention (DLP)** uses
2. DLP helps organizations comply with privacy laws and regulations, such as HIPAA, PCI-DSS, and GDPR, which safeguard personal, financial, and medical data. DLP can prevent an employee from emailing a customer's social security number or deleting sensitive financial files.
3. **Host-Based Systems** provide additional security on specific devices (endpoints).
   1. A **host-based firewall**:
   2. A **host-based intrusion detection system (HIDS):**
   3. A **host-based intrusion prevention system (HIPS)** can:
   4. **Host-based firewalls:**
   5. Software-based firewall can run on every endpoint.
   6. Allow or disallow incoming or outgoing application traffic
   7. Identify and bloc unknown processes – stop malware before it can start
   8. Can be managed centrally
   9. **Host-based intrusion detection system (HIDS)**
4. Uses log files to identify intrusions
5. Can reconfigure firewalls to block
   1. Host-based intrusion prevention system (HIPS)
      1. Recognize and block known attacks
      2. Secure OS and application configurations
      3. Often built into endpoint protection software
      4. Uses signatures, heuristics, and behavioral to identify attacks
      5. Prevent buffer overflows, registry updates, or writing files

5. Endpoint hardening: Disk, registry, ports, and services

1. **Data Encryption** involves converting readable data into a secure code to protect it from unauthorized access. Two primary methods of encryption are **Full Disk Encryption (FDE)** and **Self-Encrypting Drives (SEDs)**.
2. **FDE**:
3. **SEDs:**

1. **Opal** standards, set by the Trusted Computing Group, provide guidelines to further secure these self-encrypting drives and enforce user authentication at startup to protect against unauthorized access.
   1. **Registry** in Windows is a critical database where system configurations and user settings are stored. It contains information that controls device functionality, user profiles, and application behavior. Securing the registry is essential to avoid unauthorized changes that could compromise security or system performance. To protect this, administrators restrict registry editing permissions, often through **Group Policy settings** that disable registry access for regular users.
   2. **Open Ports and Services** are access points on a device used by applications and services to communicate over a network. Ports are assigned specific numbers, where each one corresponds to a type of network service. For example, **port 80** is commonly used for web browsing. Ports can be “open” (accepting data), “closed” (rejecting data), or “filtered” by a firewall to control access. Leaving unnecessary ports open can increase exposure to attacks, so managing ports is critical for reducing the device’s attack surface.

|  |  |
| --- | --- |
| **Port type** | **Port number range** |
| **Well-known** | **0 to 1,023** |
| **Registered** | **1,024 to 49,151** |
| **Dynamic** | **49,152 to 65,535** |

1. **TCP (Transmission Control Protocol) Ports:**
2. **Connection-Oriented**: TCP is a connection-oriented protocol, meaning it establishes a reliable connection between two devices before transmitting data. This process includes a "three-way handshake" where the devices acknowledge each other's readiness to communicate, ensuring data integrity and reliability.
3. **Reliable Data Transfer**: TCP guarantees data delivery in the correct order and checks for errors. If any data packet is lost or corrupted during transmission, TCP detects it and resends the packet. This reliability makes TCP suitable for applications where accuracy is crucial, such as web browsing (HTTP/HTTPS), email (SMTP, IMAP), and file transfers (FTP).
4. **Slower but Secure**: The added reliability comes at the cost of speed since TCP has to manage data checking and resending, leading to higher latency. It also requires more bandwidth to handle error-checking and acknowledgement packets.
5. **Examples of TCP Ports**: Common TCP ports include 80 (HTTP), 443 (HTTPS), 25 (SMTP), and 21 (FTP).
6. **UDP (User Datagram Protocol) Ports:**
7. **Connectionless**: UDP is a connectionless protocol, which means it does not establish a connection before sending data. It sends data packets (datagrams) without waiting for an acknowledgment from the receiving device, making the communication faster but less reliable.
8. **Faster but Less Reliable**: UDP’s lack of error-checking and resending of lost packets means it can transmit data more quickly than TCP, with lower latency. However, it does not guarantee that data will arrive in order or without errors, making it best for applications where speed is more critical than accuracy.
9. **Used for Real-Time Applications**: UDP is ideal for real-time applications like video streaming, gaming, and Voice over IP (VoIP), where a delay in data transmission is more problematic than occasional data loss.
10. **Examples of UDP Ports**: Common UDP ports include 53 (DNS), 67/68 (DHCP), and 123 (NTP).
11. **Key Differences:**

In summary, **TCP ports** are better for applications requiring reliable and ordered data delivery, while **UDP ports** are more suitable for speed-dependent applications where occasional data loss is acceptable. This trade-off between speed and reliability is what differentiates TCP and UDP in networking.

|  |  |  |
| --- | --- | --- |
| **Feature** | **TCP** | **UDP** |
| **Protocol Type** | Connection-oriented | Connectionless |
| **Reliability** | High (error-checking and retransmission) | Low (no error-checking) |
| **Speed** | Slower due to connection setup and checks | Faster, with no connection setup |
| **Use Cases** | Web browsing, file transfers, email | Real-time video, gaming, DNS queries |
| **Overhead** | Higher (additional data for tracking) | Lower (minimal data tracking) |
| **Examples** | HTTP (80), HTTPS (443), FTP (21) | DNS (53), VoIP, DHCP (67/68) |

1. **Common Application Ports (Spend some time learning these ports; just review the next sections)**
2. **Port 20 & 21 (FTP)** – File Transfer Protocol for transferring files between devices. Port 20 is used for data transfer, and port 21 for control commands.
3. **Port 22 (SSH)** – Secure Shell for secure logins, file transfers, and port forwarding, commonly used for remote access.
4. **Port 23 (Telnet)** – Used for unencrypted text communication. Telnet is largely deprecated in favor of SSH due to security concerns.
5. **Port 25 (SMTP)** – Simple Mail Transfer Protocol for sending emails.
6. **Port 53 (DNS)** – Domain Name System, which translates domain names to IP addresses.
7. **Port 80 (HTTP)** – Hypertext Transfer Protocol for web traffic. HTTP traffic is unencrypted.
8. **Port 110 (POP3)** – Post Office Protocol, used by email clients to retrieve emails from servers.
9. **Port 143 (IMAP)** – Internet Message Access Protocol, another protocol for retrieving emails that supports more features than POP3.
10. **Port 443 (HTTPS)** – Secure version of HTTP, encrypting web traffic with SSL/TLS to protect data.
11. **Commonly Used Network Services**
12. **Port 88 (Kerberos)** – Authentication protocol used by many organizations to secure access and logins.
13. **Port 389 (LDAP)** – Lightweight Directory Access Protocol, used for directory services (like Active Directory).
14. **Port 500 (IKE)** – Internet Key Exchange, used for setting up VPN tunnels in IPsec.
15. **Port 636 (LDAPS)** – Secure LDAP, the encrypted version of LDAP.
16. **Port 989/990 (FTPS)** – FTP Secure, providing encrypted FTP data transfer.
17. **Port 1194 (OpenVPN)** – OpenVPN tunneling protocol, widely used for secure VPNs.
18. **Port 1812 & 1813 (RADIUS)** – Remote Authentication Dial-In User Service, used for user authentication and accounting.
19. **Port 2049 (NFS)** – Network File System, used for file sharing across Unix/Linux systems.
20. **Ports for Threat Monitoring and Incident Response**
21. **Port 53 (DNS)** – DNS is a target for DNS poisoning and DNS amplification attacks.
22. **Port 123 (NTP)** – Often targeted in NTP amplification attacks.
23. **Port 514 (Syslog)** – Important for forwarding system logs, helping monitor network events.
24. **Port 3389 (RDP)** – High-risk port often targeted in brute-force and ransomware attacks.
25. **Port 445 (SMB)** – Often exploited in ransomware attacks and network infiltration.
26. **Default Passwords and Unnecessary Software** are potential security risks if left unchanged or unused on devices. Default passwords are often easy to find or guess, and leaving them unchanged makes it easier for attackers to gain access. Updating these with strong, unique passwords is a simple but essential security step. Similarly, unused software can become a security risk over time if not maintained. Regularly reviewing and removing unnecessary programs minimizes possible vulnerabilities and optimizes system performance, keeping the environment secure and efficient.

6. Endpoint hardening: Software updates, patch management, and operating system vulnerabilities

1. **Software Updates** are essential for keeping programs secure, fixing errors, adding features, and improving performance. There are three main types:
2. **Patches**:
3. **Hotfixes**:
4. **Service Packs**:
   1. **Patch Management** is the process
      1. **Auto-update** is an option:
      2. Patch management applies not only to operating systems but also to **third-party software** (software not included with the OS, like antivirus programs) that requires regular updates.
   2. **Operating System Vulnerabilities** are flaws in an OS that attackers can exploit, like buffer overflows, where sending too much data can crash the system or run malicious code. Vulnerabilities in the OS, which manages all hardware and software, can impact the entire system, so regular updates, firewalls, and access controls are essential defenses.
   3. **Malicious Software Updates**:

7. Boot Integrity

1. The **Unified Extensible Firmware Interface (UEFI)** is a system that controls how a computer starts up, ensuring it does so securely by preventing malware from loading before the operating system. UEFI replaces the older BIOS system, offering faster startup times, enhanced security, and better functionality like remote troubleshooting and mouse support for setup screens.
   1. UEFI is a modern replacement for the traditional BIOS (Basic Input/Output System) that has been used in computers for decades. It's a specification that defines a software interface between an operating system and platform firmware.
   2. Faster boot times: UEFI can initialize multiple hardware devices simultaneously, reducing boot time.
   3. Network capabilities: UEFI can access networks, which allows for remote troubleshooting and diagnostics before the OS loads.
   4. CPU-independent architecture: UEFI is not tied to any specific type of processor architecture.
2. **Key Features of UEFI:**
   1. **Secure Booting**: UEFI supports **secure boot** which only allows:
      1. Digital Signatures: Each piece of software is signed with a digital signature. These signatures are checked against a database of allowed signers. Bootloader must be signed with a trusted certificate.
      2. Key Management: UEFI maintains several key databases:
         1. PK (Platform Key) is:
         2. KEK (Key Exchange Key) is:
         3. db (Signature Database): Contains allowed signatures
         4. dbx (Forbidden Signature Database): Contains forbidden signatures
      3. Revocation is:
   2. **Measured Boot**: This option:
      1. These measurements are stored in the Trusted Platform Module (TPM).
         1. Specification for cryptographic functions in application in operating systems.
         2. Persistent memory – comes with unique keys burned in during production
         3. Password protected – built with anti-brute force technology.
      2. Key aspects of Measured Boot:
         1. TPM Usage: The TPM is:
         2. Chain of Trust is:
         3. Remote Attestation:
         4. Hard to avoid because it is hardware.
         5. UEFI stores a hash of the firmware, boot drivers, and everything else loaded during the secure boot and trusted boot process, stored in the TPM.
   3. Together, secure and measured boot add layers of protection against unauthorized software.
3. **Root of Trust (RoT)**: This is:
4. **Hardware Vulnerabilities**: Hardware and firmware can become vulnerable over time, especially if they’re outdated or no longer receive updates. Old hardware lacks modern security features and may be easier to exploit. Regular updates, replacing outdated devices, and careful management of older systems help minimize these risks.
   1. Some notable examples include:
      1. Spectre and Meltdown: These vulnerabilities affected nearly all modern processors and allowed malicious programs to steal data being processed on the computer.
      2. Rowhammer: A vulnerability in DRAM chips that allows an attacker to change the contents of memory by repeatedly accessing nearby memory locations.
      3. Intel Management Engine vulnerabilities: Flaws in Intel's ME, a separate processor and operating system embedded in Intel chipsets, could allow attackers to take control of the system below the operating system level.
   2. Mitigation strategies:
   3. Firmware updates: Regularly applying firmware updates can help address some hardware vulnerabilities.
   4. Hardware replacement: In some cases, the only solution is to replace vulnerable hardware components or entire systems.
   5. Virtualization and isolation: Using virtualization technologies can help isolate potentially vulnerable hardware components.

8. Example Test Questions:

16) Gary wants to deploy a tool that will allow him to identify and effectively respond to ransomware that might target systems that his company owns. He knows that he is likely to need to identify threats based on behavior rather than just using signatures, and he wants to have a dashboard-style view of his data. What tool should Gary select to meet this need?

a. IPS

b. NAC

c. DLP

d. EDR

19) Susan has been tasked with hardening the systems in her environment and wants to ensure that data cannot be recovered from systems if they are stolen or their disk drives are stolen and accessed. What is her best option to ensure data security in these situations?

a. Deploy folder-level encryption

b. Deploy full-disk encryption

c. Deploy file-level encryption

d. Degauss all the drives

23) Tracy wants to protect desktop and laptop systems in her organization from network attacks. She wants to deploy a tool that can actively stop attacks based on signatures, heuristics, and anomalies. What type of tool should she deploy?

a. A firewall

b. Antimalware

c. HIDS

d. HIPS

29) Mary wants to harden workstations she is responsible for against malware attacks. Which of the following is not a common solution to this?

a. Installing EDR

b. Limiting administrative access

c. Installing antivirus

d. Using disk encrypt

36) What is the primary concern for security professionals about legacy hardware?

a. Its likelihood of failure

b. Lack of patches and updates

c. Lack of vendor support

d. Inability to support modern protocols