

Core review

Thursday, August 15, 2024 12:21 PM

Step 1: Review Documentation

1.1 Gather Inventory Documentation

- **Action:** Obtain the latest inventory lists for all information assets, including:
 - Workstations, laptops, servers, security devices, network devices, and software applications.
- **Verification:** Confirm the inventory includes:
 - Device types, operating systems, manufacturer, versions, the number of instances, and the date of the last update.

Positive Findings

- "The credit union has developed a comprehensive, regularly updated inventory of all information assets, ensuring appropriate security controls can be applied in accordance with Appendix A, Section III(B)."
- "The asset inventory is reviewed and updated quarterly to reflect changes, reducing the risk of unauthorized access or misuse."

Negative Findings

- **Incomplete or Inaccurate Inventory:**
 - Missing assets such as workstations, laptops, or security devices.
 - Outdated inventory records that do not reflect recent changes.
 - Lack of detailed information on asset types, operating systems, and versions.
 - **Reference:** Appendix A, Section III(B).

1.2 Review Policies and Procedures

- **Action:** Evaluate inventory management policies for completeness.
 - Ensure policies include guidelines for adding, updating, and removing assets, and define roles and responsibilities.

Positive Findings

- "The credit union has a comprehensive Information Asset Management Policy covering acquisition, tracking, updating, and decommissioning of assets, reviewed annually to ensure alignment with evolving technologies and threats."

Negative Findings

- **Inadequate Policies and Procedures:**
 - No formal inventory management policies in place.
 - Unclear roles and responsibilities for managing the inventory.
 - **Reference:** Appendix A, Section II(A).

Step 2: Validate CORE Statements

2.1 Stmt 3.1 & 3.2: Workstations, Laptops, and Servers

- **Action:** Cross-check inventory lists with actual physical and virtual assets using tools such as Active Directory or SCCM.

Positive Findings

- "Access control mechanisms are in place for all information assets, including multi-factor authentication and role-based access controls, meeting Appendix A, Section III(C)."
- "Sensitive member data stored and transmitted through assets is protected by strong encryption standards, such as AES-256, in compliance with Appendix A, Section III(A)."

Negative Findings

- Missing or outdated inventory entries for workstations, laptops, and servers.
- **Reference:** Appendix A, Section III(C).

2.2 Stmt 3.3: Security Devices

- **Action:** Verify the inventory of security devices such as firewalls, IDS/IPS systems, and VPN gateways.

Positive Findings

- "The credit union conducts regular security audits of all information assets, ensuring compliance with policies and safeguarding against unauthorized access, in line with Appendix A, Section III(C)."
- "Quarterly vulnerability scans and annual penetration testing identify potential weaknesses and verify that controls function as intended."

Negative Findings

- Unaccounted-for security devices.
- Outdated firmware or inadequate monitoring on security devices.
- **Reference:** Appendix A, Section III(A).

2.3 Stmt 3.4: Network Devices

- **Action:** Validate the inventory of network devices such as routers and switches.

Positive Findings

- "Lifecycle management tracks the End-of-Life (EOL) and End-of-Support (EOS) dates for all hardware and software, ensuring timely upgrades or replacements."
- "Secure decommissioning procedures, including secure data wiping and physical destruction, prevent unauthorized access to residual information."

Negative Findings

- Missing inventory entries for network devices.
- Devices running outdated firmware or lacking adequate monitoring.
- **Reference:** Appendix A, Section III(B).

2.4 Stmt 3.5: Software Applications

- **Action:** Cross-check software inventory against installed applications.

Positive Findings

- "The credit union uses a software allowlist to prevent unauthorized applications from running, ensuring only approved software is operational."

Negative Findings

- Missing or unauthorized software installations.
- Inaccurate tracking of software versions.
- **Reference:** Appendix A, Section III(B).

Core + Review

Tuesday, December 10, 2024 12:19 PM

3.1 Stmt 3.6: Accurate, Detailed Inventory

- **Action:** Spot-check entries for accuracy in documenting assets processing member data.

Positive Findings

- "Comprehensive inventory ensures all systems involved in processing member information are accurately recorded."

Negative Findings

- Missing detailed records of assets storing or processing sensitive member data.
- **Reference:** Appendix A, Section III(B).

3.2 Stmt 3.7: EOL/EOS Tracking

- **Action:** Confirm tracking and replacement of systems nearing EOL/EOS.

Positive Findings

- "Lifecycle management ensures outdated systems are proactively replaced to minimize risks, meeting Appendix A, Section III(A)."

Negative Findings

- Systems past EOL/EOS are still in use.
- No mechanisms for tracking EOL/EOS dates.
- **Reference:** Appendix A, Section III(A).

3.3 Stmt 3.8: Detection of Unauthorized Assets

- **Action:** Review NAC systems that detect unauthorized assets.

Positive Findings

- "Unauthorized devices are promptly detected and remediated, ensuring the integrity of network security."

Negative Findings

- Unauthorized devices connected to the network.
- No NAC system in place.
- **Reference:** Appendix A, Section III(C).

1. Incomplete or Inaccurate Inventory

- **Finding:** Missing assets such as workstations, laptops, servers, or security devices are not accounted for in the inventory.
 - **Reference: Appendix A, Section III(B)** – Requires maintaining an accurate and complete inventory of information systems to identify and safeguard member information.
- **Finding:** Outdated inventory records that do not reflect the most recent asset changes.
 - **Reference: Appendix A, Section III(B)** – The inventory must be updated regularly to account for new or retired assets to ensure the security of member information.
- **Finding:** Lack of detailed information on asset types, operating systems, and versions.
 - **Reference: Appendix A, Section III(B)** – Requires an accurate and detailed asset inventory to ensure security controls are appropriately implemented for each asset.

2. Inadequate Inventory Management Policies and Procedures

- **Finding:** No formal inventory management policies or procedures in place.
 - **Reference: Appendix A, Section II(A)** – Policies and procedures should be appropriate for the credit union's size and complexity, ensuring effective management and safeguarding of member information through asset tracking.
- **Finding:** Unclear roles and responsibilities for managing the inventory.
 - **Reference: Appendix A, Section II(A)** – The ISP must clearly define roles and responsibilities for those managing information assets and ensure accountability.

3. Gaps in Security Devices and Network Devices Inventory

- **Finding:** Unaccounted-for security devices, such as firewalls or VPN gateways, are missing from the inventory.
 - **Reference: Appendix A, Section III(A)** – Security measures and devices must be listed in the inventory to ensure they are effectively used to protect member information from unauthorized access.
- **Finding:** Network devices like routers and switches are running outdated firmware or lack adequate monitoring.
 - **Reference: Appendix A, Section III(B)** – Network devices must be maintained and monitored to ensure the confidentiality, integrity, and availability of member information.

4. Inaccurate Software Inventory

- **Finding:** Missing or unauthorized software installed on devices is not tracked in the inventory.
 - **Reference: Appendix A, Section III(B)** – Requires maintaining an inventory of all software applications to ensure that unauthorized or outdated software does not compromise the security of member information.
- **Finding:** Software versions are not tracked accurately.
 - **Reference: Appendix A, Section III(B)** – Accurate tracking of software versions is required to ensure systems are updated and secure from vulnerabilities.

5. End-of-Life (EOL) and End-of-Support (EOS) Tracking Issues

- **Finding:** Systems that have reached EOL/EOS are still in use, posing security risks.
 - **Reference: Appendix A, Section III(A)** – The use of systems past their EOL/EOS introduces significant risks and must be addressed proactively through replacement or upgrades.
- **Finding:** No tracking of EOL/EOS dates for hardware and software.
 - **Reference: Appendix A, Section III(A)** – Institutions must have mechanisms to track when systems reach their EOL/EOS and must plan for replacements to ensure ongoing protection of member information.

6. Unaddressed Unauthorized or Unmanaged Devices

- **Finding:** Unauthorized devices connected to the network, exposing the credit union to risk.
 - **Reference: Appendix A, Section III(C)** – Institutions must implement controls to detect and respond to unauthorized devices accessing the network, which could lead to unauthorized access to member information.
- **Finding:** No network access control (NAC) system in place to detect unauthorized devices.
 - **Reference: Appendix A, Section III(C)** – Institutions must detect unauthorized network access, including rogue devices that could compromise member information.

7. Insufficient Use of Asset Discovery Tools

- **Finding:** No active or passive discovery tools in place to maintain an accurate inventory.
 - **Reference: Appendix A, Section III(B)** – Active monitoring and discovery tools must be used to maintain an accurate and up-to-date inventory of information assets.
- **Finding:** Discovery tools are not effectively used to update the inventory.
 - **Reference: Appendix A, Section III(B)** – Institutions must ensure that discovery tools are operational and consistently update the asset inventory to protect member data.

8. Lack of Software Allowlist and Controls

- **Finding:** No software allowlist is implemented, allowing unauthorized software installations.
 - **Reference: Appendix A, Section III(B)** – Institutions must maintain a list of authorized software to prevent the use of unauthorized applications that could compromise member information.
- **Finding:** Software allowlist is not enforced, allowing unauthorized applications to run.
 - **Reference: Appendix A, Section III(B)** – The software allowlist must be enforced to protect sensitive systems and data.

9. Weaknesses in Software Version Control

- **Finding:** Lack of formal processes for controlling and updating software versions.
 - **Reference: Appendix A, Section III(A)** – Institutions must have controls in place to ensure software is regularly updated to address vulnerabilities and protect member information.
- **Finding:** Inconsistent software updates, leaving systems vulnerable to known security risks.
 - **Reference: Appendix A, Section III(A)** – Outdated software introduces significant vulnerabilities, which must be mitigated by consistent software version control processes.

10. Insufficient Asset Lifecycle Management

- **Finding:** Failure to retire obsolete or underutilized assets, leaving them susceptible to unauthorized access or misuse.
 - **Reference: Appendix A, Section III(B)** – Institutions must properly manage the entire lifecycle of information assets, including decommissioning or retiring assets to ensure they no longer pose risks to member information.
- **Finding:** Inadequate documentation or procedures for removing assets from the inventory.
 - **Reference: Appendix A, Section III(B)** – Institutions must document the removal of assets from the inventory to ensure clarity and avoid confusion about which assets remain active.

11. Lack of Documentation and Reporting

- **Finding:** Discrepancies in the inventory are not documented or tracked, leading to persistent inaccuracies.
 - **Reference: Appendix A, Section III(B)** – Discrepancies in the inventory must be documented and addressed to ensure accurate tracking of all information assets.
- **Finding:** No actionable recommendations are provided after identifying gaps or inaccuracies in the inventory.
 - **Reference: Appendix A, Section III(A)** – Institutions must document and implement corrective actions for identified inventory gaps to protect member information effectively.
- **Finding:** Inadequate reporting on the status of the inventory to the Board or senior management.
 - **Reference: Appendix A, Section I(C)(2)** – The Board must receive regular reports on the information security program, including the status of the asset inventory, to ensure appropriate oversight.

Remediation

Thursday, September 19, 2024 10:26 AM

Here are the remediation steps for each of the potential findings related to the **Inventory of Information Assets**, aligned with **Appendix A to Part 748, Title 12**:

1. Incomplete or Inaccurate Inventory

- **Finding:** Missing assets or outdated inventory records.
 - **Remediation:**
 1. **Conduct a full inventory audit** to identify all physical and virtual assets, ensuring that no devices (e.g., workstations, laptops, servers) are omitted.
 2. **Update inventory management processes** to ensure real-time updates whenever assets are added, modified, or decommissioned.
 3. **Implement regular reviews** to reconcile discrepancies between inventory lists and actual assets.
 4. **Utilize asset discovery tools** to automatically track devices and update inventory lists.
- **Finding:** Lack of details on asset attributes.
 - **Remediation:**
 5. **Standardize asset tracking fields** to include detailed information such as device type, OS, version, and firmware.
 6. **Ensure consistent data entry** by enforcing inventory data standards across all departments.
 7. **Use automated tools** that populate detailed asset information to avoid manual data entry errors.

2. Inadequate Inventory Management Policies and Procedures

- **Finding:** No formal inventory management policies or unclear roles.
 - **Remediation:**
 1. **Develop and formalize inventory management policies** outlining procedures for adding, updating, and removing assets from the inventory.
 2. **Define clear roles and responsibilities** for personnel involved in managing and maintaining the asset inventory.
 3. **Ensure policies include timelines** for reviewing and updating the inventory.
 4. **Conduct training** for staff to ensure adherence to inventory management procedures.

3. Gaps in Security Devices and Network Devices Inventory

- **Finding:** Unaccounted-for or outdated security devices.
 - **Remediation:**
 1. **Inventory all security devices** such as firewalls, IDS/IPS systems, and VPN gateways, and ensure they are tracked.
 2. **Regularly review and update firmware** on all network and security devices to keep them secure.
 3. **Implement automated monitoring** to track the operational status of security devices and alert administrators to missing or outdated devices.
- **Finding:** Inconsistent tracking of device configurations.
 - **Remediation:**
 4. **Ensure security device configurations** (e.g., firewall rules, VPN settings) are documented and stored securely.
 5. **Review configurations regularly** to confirm that they are up to date and aligned with security policies.
 6. **Use configuration management tools** to monitor changes and ensure consistency across network devices.

4. Inaccurate Software Inventory

- **Finding:** Missing or unauthorized software installations.
 - **Remediation:**
 1. **Conduct an audit of all installed software** across the organization to identify any unauthorized applications.
 2. **Develop a software allowlist** and enforce policies to ensure only authorized software can be installed.
 3. **Use automated tools to monitor software installations** and flag any unauthorized software for review.
- **Finding:** Software versions not tracked accurately.
 - **Remediation:**
 4. **Implement software version control** tools to track and document software versions installed on all devices.
 5. **Automate software version tracking** to ensure that outdated or unsupported versions are identified for upgrade or replacement.
 6. **Enforce policies that require immediate updates** to software identified as vulnerable or out of date.

5. End-of-Life (EOL) and End-of-Support (EOS) Tracking Issues

- **Finding:** Systems past EOL/EOS are still in use.
 - **Remediation:**
 1. **Identify all EOL/EOS assets** in the inventory and develop a replacement or upgrade plan for each.
 2. **Prioritize critical systems** for immediate upgrade or replacement, ensuring that no unsupported systems handle sensitive member information.
 3. **Schedule regular reviews** to identify any upcoming EOL/EOS dates for systems and plan proactive upgrades.
- **Finding:** No tracking of EOL/EOS dates.
 - **Remediation:**
 4. **Integrate EOL/EOS tracking** into the inventory management system to ensure that all hardware and software nearing EOL/EOS are flagged.
 5. **Create a lifecycle management process** for timely upgrades, including budget allocation and implementation plans for replacing EOL systems.

6. Unaddressed Unauthorized or Unmanaged Devices

- **Finding:** Unauthorized devices are connected to the network.
 - **Remediation:**
 1. **Implement Network Access Control (NAC)** systems to automatically detect and block unauthorized devices from connecting to the network.
 2. **Conduct regular network scans** to identify any rogue or unmanaged devices and take immediate remediation action.
 3. **Create policies that restrict unauthorized devices** from accessing sensitive network resources and ensure compliance.
- **Finding:** No NAC system in place.
 - **Remediation:**
 4. **Deploy a NAC solution** to monitor all devices attempting to connect to the network and enforce access restrictions.
 5. **Set up real-time alerts** for unauthorized devices accessing the network to allow quick remediation.
 6. **Regularly review NAC logs** to ensure that only approved devices have network access.

7. Insufficient Use of Asset Discovery Tools

- **Finding:** No active or passive discovery tools in place.
 - **Remediation:**
 1. **Deploy active and passive discovery tools** (e.g., network scanners) to continuously track information assets in real-time.
 2. **Integrate discovery tools** with the inventory management system to automatically update the inventory as new assets are added or changed.
 3. **Schedule regular asset discovery scans** to ensure no devices are missed in the inventory.
- **Finding:** Discovery tools are not effectively used.
 - **Remediation:**
 4. **Ensure that discovery tools are properly configured** and scheduled for regular scans to identify and update all assets on the network.
 5. **Train IT staff** on how to use discovery tools effectively and interpret the data collected to maintain an accurate asset inventory.
 6. **Establish review cycles** for verifying the completeness of discovery tool results against the actual network environment.

8. Lack of Software Allowlist and Controls

- **Finding:** No software allowlist implemented.
 - **Remediation:**
 1. **Develop and enforce a software allowlist policy** to ensure only authorized software is permitted for installation.
 2. **Implement software control tools** that prevent unauthorized software installations.

3. **Regularly review and update the allowlist** to ensure that new, authorized software is included and outdated or unauthorized software is removed.
- **Finding:** Failure to enforce the allowlist.
 - **Remediation:**
 4. **Implement monitoring tools** that enforce allowlist rules and automatically block unauthorized applications.
 5. **Conduct audits of software usage** across the organization to ensure compliance with the allowlist.
 6. **Train employees** on the importance of using only authorized software and the potential risks of using unauthorized applications.

9. Weaknesses in Software Version Control

- **Finding:** Lack of formal processes for controlling and updating software versions.
 - **Remediation:**
 1. **Develop formal version control procedures** that require regular software updates and document all changes to software versions.
 2. **Implement software update tools** to automatically apply patches and updates to software across all systems.
 3. **Monitor version control adherence** through audits to ensure no systems are running outdated or unsupported versions.
- **Finding:** Inconsistent software updates.
 - **Remediation:**
 4. **Enforce consistent update policies** that require software to be updated as soon as new versions or patches are released.
 5. **Automate software updates** through management tools to ensure timely patching and version control.
 6. **Establish a testing environment** for validating updates before deploying them across the organization to reduce the risk of compatibility issues.

10. Insufficient Asset Lifecycle Management

- **Finding:** Failure to retire obsolete or underutilized assets.
 - **Remediation:**
 1. **Develop an asset retirement policy** that ensures obsolete or underutilized assets are decommissioned securely and removed from the inventory.
 2. **Create an asset lifecycle management plan** that tracks the acquisition, usage, and retirement of assets to avoid keeping unnecessary or vulnerable devices.
 3. **Ensure secure data destruction** procedures are followed when retiring assets, especially those containing sensitive member information.
- **Finding:** Inadequate documentation of asset removal.
 - **Remediation:**
 4. **Ensure proper documentation** is completed whenever assets are removed or retired, including records of secure disposal or reallocation.
 5. **Integrate asset retirement processes** into inventory management systems to automatically update the inventory when assets are decommissioned.

11. Lack of Documentation and Reporting

- **Finding:** Discrepancies in the inventory are not documented.
 - **Remediation:**
 1. **Create a discrepancy tracking process** to document and address any inventory inaccuracies identified during audits or reviews.
 2. **Ensure real-time updates** to the inventory when discrepancies are identified and corrected.
- **Finding:** No actionable recommendations are provided after identifying gaps.
 - **Remediation:**
 3. **Develop a process for providing actionable recommendations** to address gaps found in the inventory.
 4. **Assign accountability** for implementing recommendations and set deadlines for completion.
- **Finding:** Inadequate reporting to the Board or senior management.
 - **Remediation:**
 5. **Establish regular reporting cycles** to update senior management and the Board on the status of the inventory and any identified risks or issues.
 6. **Create dashboards or summary reports** for presenting key inventory findings, including trends, gaps, and corrective actions.

By implementing these **remediation steps**, credit unions can address the potential findings in their **Inventory of Information Assets** and ensure compliance with **Appendix A to Part 748, Title 12**. These actions will help protect member information, improve security, and meet regulatory expectations.

To ensure compliance with **Appendix A to Part 748, Title 12 for Information Assets**, credit unions must follow a structured process to develop, maintain, and monitor an accurate and secure inventory of all information assets. These assets include workstations, laptops, servers, network devices, security devices, and software applications, all of which must be safeguarded to protect member information. Here are the **compliance steps for Information Assets** according to Appendix A to Part 748, Title 12:

1. Establish and Maintain an Accurate Inventory of Information Assets

Step 1.1: Create an Information Asset Inventory

- **Action:** Develop a comprehensive inventory that includes all information assets, such as:
 - Workstations, laptops, and servers.
 - Network devices (e.g., routers, switches).
 - Security devices (e.g., firewalls, IDS/IPS, VPNs).
 - Software applications (including licenses, versions, and instances).
 - Virtual and cloud assets (if applicable).
- **Reference:** **Appendix A, Section III(B)** – Ensure an accurate and comprehensive inventory of systems and assets to identify and safeguard member information from unauthorized access or misuse.

Step 1.2: Ensure Detailed Documentation

- **Action:** For each asset, document key attributes such as:
 - Device type, manufacturer, and model.
 - Operating system version and patch level.
 - Hardware specifications.
 - Physical location or virtual assignment.
 - Last update or maintenance date.
- **Reference:** **Appendix A, Section III(B)** – Information assets must be documented with sufficient detail to ensure appropriate security controls can be applied.

Step 1.3: Regularly Update the Asset Inventory

- **Action:** Establish a process for regularly updating the inventory as new assets are added, modified, or decommissioned.
- **Frequency:** Set a schedule for periodic reviews, such as quarterly or semi-annually.
- **Reference:** **Appendix A, Section III(B)** – The inventory must be updated to reflect the addition of new systems and the decommissioning of obsolete assets to maintain an accurate record.

2. Implement and Enforce Information Security Controls

Step 2.1: Implement Access Control Mechanisms

- **Action:** Ensure that all information assets are protected by appropriate access control mechanisms, limiting access to authorized users only.
- **Control Types:** Use methods such as multi-factor authentication, role-based access controls, and encryption to protect sensitive data.
- **Reference:** **Appendix A, Section III(C)** – Implement controls to prevent unauthorized access to member information.

Step 2.2: Apply Encryption to Sensitive Information

- **Action:** Encrypt sensitive data at rest and in transit across all relevant information assets, including databases, storage devices, and communications.
- **Reference:** **Appendix A, Section III(A)** – Implement encryption measures to safeguard sensitive member information from unauthorized disclosure.

Step 2.3: Use Secure Configuration Standards

- **Action:** Ensure that all devices and software are configured securely, in line with industry best practices (e.g., disabling unnecessary services, applying security patches).
- **Reference:** **Appendix A, Section III(A)** – Devices must be securely configured to minimize vulnerabilities that could lead to data breaches.

3. Monitor and Test the Effectiveness of Controls

Step 3.1: Conduct Regular Security Audits

- **Action:** Schedule and perform regular security audits to ensure that all assets are properly inventoried, secure, and compliant with established policies.
 - This includes reviewing network devices, security devices, and software applications for vulnerabilities.
- **Reference:** **Appendix A, Section III(C)** – Testing and auditing are necessary to validate the effectiveness of controls designed to protect member information.

Step 3.2: Perform Vulnerability Scanning and Penetration Testing

- **Action:** Regularly scan all systems for vulnerabilities and conduct penetration testing on critical assets, especially those that store or process member information.
- **Frequency:** Conduct vulnerability scans at least quarterly and penetration tests annually.
- **Reference:** **Appendix A, Section III(C)** – Regular vulnerability assessments are critical to identifying and remediating weaknesses in the security of information assets.

Step 3.3: Monitor Access to Information Assets

- **Action:** Implement continuous monitoring of access to critical systems and assets, using logging and alerting tools to detect and respond to unauthorized access attempts.
- **Reference:** **Appendix A, Section III(C)** – Continuous monitoring helps detect and respond to unauthorized access to sensitive member information.

4. Manage the Lifecycle of Information Assets

Step 4.1: Track End-of-Life (EOL) and End-of-Support (EOS) Dates

- **Action:** Maintain a schedule to track when hardware or software assets reach their EOL or EOS dates, ensuring timely upgrades or replacements.
- **Reference:** **Appendix A, Section III(A)** – Systems and software that are no longer supported or updated introduce significant security risks and must be addressed proactively.

Step 4.2: Securely Decommission Obsolete Assets

- **Action:** When retiring or decommissioning hardware or software, ensure that all sensitive data is securely wiped or destroyed.
- **Process:** Follow secure data destruction procedures, such as physical destruction of hard drives or using data erasure software.
- **Reference:** **Appendix A, Section III(A)** – Ensure that obsolete assets are disposed of securely to prevent unauthorized access to residual member information.

5. Ensure Proper Vendor and Third-Party Management

Step 5.1: Inventory Service Providers with Access to Information Assets

- **Action:** Maintain a list of third-party service providers that have access to critical systems or handle member information.
- **Reference:** **Appendix A, Section III(C)** – Service providers with access to member information must be included in the information asset inventory and their security practices must be evaluated.

Step 5.2: Enforce Security Requirements in Contracts

- **Action:** Ensure that contracts with third-party providers include explicit security requirements for protecting member information, including regular audits and compliance reviews.
- **Reference:** **Appendix A, Section III(A)** – Contracts must include provisions that ensure third-party service providers implement adequate security measures to safeguard member information.

6. Develop and Maintain Policies for Information Asset Management

Step 6.1: Develop a Comprehensive Information Asset Management Policy

- **Action:** Create a written policy detailing how the credit union manages its information assets, including the acquisition, tracking, updating, and decommissioning of assets.
- **Reference:** **Appendix A, Section II(A)** – Policies and procedures must be appropriate for the size and complexity of the credit union's operations and reflect its risk management strategy.

Step 6.2: Review and Update Policies Regularly

- **Action:** Review the Information Asset Management Policy at least annually to ensure it remains current with evolving technologies and threats.
- **Reference:** **Appendix A, Section II(A)** – Regular reviews ensure policies are effective in safeguarding member information and aligned with the credit union's overall risk management framework.

7. Report to the Board of Directors

Step 7.1: Prepare an Annual Report on the Information Security Program

- **Action:** Include details about the status of the inventory, key risks, testing results, and any security incidents related to information assets.
 - Include updates on third-party vendor relationships, access control measures, and ongoing monitoring efforts.
- **Reference:** **Appendix A, Section I(C)(2)** – The Board must receive an annual report on the Information Security Program, which includes information on the security of assets that protect member information.

8. Regularly Train Employees on Information Asset Security

Step 8.1: Provide Ongoing Security Awareness Training

- **Action:** Train employees on the importance of protecting information assets, including identifying unauthorized devices, secure use of software, and proper asset management practices.
- **Reference:** **Appendix A, Section III(C)** – Training is essential to ensure that employees understand and adhere to security practices that protect member information.

By following these **compliance steps**, credit unions can maintain an accurate, secure, and up-to-date inventory of information assets, ensuring compliance with **Appendix A to Part 748, Title 12**. These actions will help safeguard member information, minimize risks, and provide effective governance and oversight of information systems.

Tools

Thursday, August 15, 2024 12:25 PM

To effectively validate the inventory of information assets and ensure compliance with both CORE and CORE+ statements, a combination of automated tools, manual checks, and specialized software is required. Below is a comprehensive list of tools that can be used for this purpose:

1. Inventory Management and Asset Tracking Tools

1.1 ServiceNow

- **Use:** Centralized platform for IT asset management. Tracks hardware, software, and network devices. Includes features for managing asset lifecycle, including EOL/EOS tracking.
- **Validation:** Provides comprehensive inventory reports and supports spot-checking of assets.

1.2 SolarWinds Network Performance Monitor (NPM)

- **Use:** Monitors network devices, providing detailed inventory of switches, routers, and other network hardware.
- **Validation:** Offers real-time visibility into network devices and their configurations.

1.3 Lansweeper

- **Use:** Agentless network discovery tool that scans and inventories hardware, software, and network assets.
- **Validation:** Automates asset discovery and provides detailed reports on installed software and hardware configurations.

1.4 ManageEngine AssetExplorer

- **Use:** Web-based asset management software that helps in tracking and managing hardware and software assets.
- **Validation:** Useful for maintaining an up-to-date inventory and tracking software licenses and versions.

2. Network Discovery and Security Tools

2.1 Nmap

- **Use:** Open-source network scanner that discovers devices connected to the network, including servers, workstations, and network devices.
- **Validation:** Validates the presence and status of networked assets through active discovery.

2.2 Tenable Nessus

- **Use:** Vulnerability scanner that identifies networked devices and their security status.
- **Validation:** Detects unauthorized devices and ensures they are properly inventoried.

2.3 Cisco Discovery Protocol (CDP)

- **Use:** Cisco's network discovery protocol that helps in identifying Cisco devices on the network.
- **Validation:** Assists in identifying network devices and ensuring they are correctly inventoried.

2.4 Microsoft System Center Configuration Manager (SCCM)

- **Use:** Comprehensive tool for managing large groups of Windows-based systems, including software distribution, security, and inventory management.
- **Validation:** Automates the discovery and inventory of connected devices and software.

3. Software Inventory and Management Tools

3.1 Flexera Software Asset Management

- **Use:** Helps organizations manage software licenses, optimize usage, and ensure compliance.
- **Validation:** Provides detailed inventory of software applications, including version control and instance tracking.

3.2 Ivanti IT Asset Management

- **Use:** Automates asset tracking and software inventory management, including license compliance.
- **Validation:** Ensures all software applications and versions are accurately documented and controlled.

3.3 Qualys Asset Inventory

- **Use:** Provides a continuous, real-time inventory of all IT assets, including software applications and hardware.
- **Validation:** Ensures that all assets are accounted for, including unauthorized and unknown assets.

3.4 Symantec Control Compliance Suite

- **Use:** Provides detailed software and hardware inventory management with compliance checks.
- **Validation:** Automates software and hardware inventory validation against regulatory and internal standards.

4. End-of-Life and End-of-Support Tracking Tools

4.1 Spiceworks Inventory

- **Use:** Free IT asset management software that tracks hardware and software EOL/EOS.
- **Validation:** Keeps track of asset lifecycles and provides alerts for EOL/EOS dates.

4.2 Lansweeper (EOL/EOS Features)

- **Use:** Includes EOL/EOS tracking capabilities, providing alerts when assets reach their end of support.
- **Validation:** Helps ensure that inventory reflects the current support status of assets.

5. Unauthorized Asset Detection Tools

5.1 CrowdStrike Falcon

- **Use:** Endpoint detection and response (EDR) tool that identifies and responds to unauthorized devices and software.
- **Validation:** Ensures that only authorized assets are connected to the network.

5.2 McAfee ePolicy Orchestrator

- **Use:** Centralized security management software that detects unauthorized devices and software within the network.
- **Validation:** Helps enforce policies to prevent unauthorized assets from accessing the network.

5.3 Tripwire Enterprise

- **Use:** Provides continuous monitoring of IT environments for unauthorized changes or configurations.
- **Validation:** Detects and alerts on unauthorized assets or software installations.

6. Active and Passive Discovery Tools

6.1 NetFlow Analyzer

- **Use:** Monitors network traffic and provides insights into active network devices and communications.
- **Validation:** Assists in the discovery of all active devices on the network.

6.2 Cisco Stealthwatch

- **Use:** Network security analytics tool that uses NetFlow data for passive discovery of network devices.
- **Validation:** Provides passive monitoring of assets connected to the network, ensuring they are accurately inventoried.

6.3 Rumble Network Discovery

- **Use:** Active network discovery tool that scans the network to identify all connected devices.
- **Validation:** Ensures all network-connected assets are detected and logged into the inventory.

6.4 ARPwatch

- **Use:** Monitors Ethernet traffic on a network and keeps a database of Ethernet/IP address pairings.
- **Validation:** Provides passive discovery of devices connected to the network, ensuring they are tracked.

7. Software Allowlisting and Version Control Tools

7.1 AppLocker

- **Use:** Microsoft tool that helps control which applications can run on a network.
- **Validation:** Ensures that only authorized software is allowed to run, and unauthorized software is blocked.

7.2 Carbon Black

- **Use:** Endpoint security tool that includes allowlisting features to control software execution.
- **Validation:** Enforces software allowlists across the network and prevents unauthorized software from running.

7.3 Puppet or Chef

- **Use:** Configuration management tools that help ensure consistent software versions across environments.
- **Validation:** Automates version control, ensuring that the software across the network is consistent and up-to-date.

8. Technical Controls for Software Libraries

8.1 Microsoft Group Policy

- **Use:** Manages the configuration of Windows operating systems, including controlling which software libraries are allowed to run.
- **Validation:** Ensures that only authorized .dll, .ocx, .so, etc., files are loaded into system processes.

8.2 Symantec Endpoint Protection

- **Use:** Endpoint security tool that includes controls for software libraries and other critical files.
- **Validation:** Prevents unauthorized software libraries from loading and enforces compliance with the allowlist.

9. Manual Tools and Methods

9.1 Manual Spot-Checks

- **Use:** Physical inspection or remote verification of selected assets from the inventory.
- **Validation:** Ensures the accuracy of the inventory by confirming the presence and status of assets.

9.2 Excel/Google Sheets

- **Use:** Simple tools for manual tracking, validation, and cross-referencing of inventory data.
- **Validation:** Useful for smaller environments or as a secondary verification method.

9.3 Interviews and Observations

- **Use:** Conduct interviews with IT staff and observe inventory management processes in real-time.
- **Validation:** Provides qualitative insights and confirms that documented processes are being followed.

10. Reporting and Compliance Tools

10.1 Splunk

- **Use:** Data analytics platform that can aggregate and analyze inventory data for reporting purposes.
- **Validation:** Generates compliance reports and dashboards for inventory validation.

10.2 PowerBI

- **Use:** Business analytics tool that can visualize and report on inventory data.
- **Validation:** Creates detailed reports and visual representations of the inventory, aiding in validation.

10.3 Tableau

- **Use:** Data visualization software that can help in the presentation and analysis of inventory data.
- **Validation:** Provides insights through visual dashboards, helping to identify gaps in the inventory.

Resources

Thursday, August 15, 2024 12:25 PM

[CIS](#)

CM-8: System Component Inventory

<https://csf.tools/reference/nist-sp-800-53/r5/cm/cm-8/>

<https://center-for-threat-informed-defense.github.io/mappings-explorer/external/nist/attack-14.1/domain-enterprise/nist-rev5/CM-08/>

https://mitre-attack.github.io/attack-navigator/#layerURL=https%3A%2F%2Fraw.githubusercontent.com%2Fcenter-for-threat-informed-defense%2Fattack-control-framework-mappings%2Fmain%2Fframeworks%2Fattack_12_1%2Fnist800-53_r4%2Fplayers%2Fby_family%2FConfiguration_Management%2FCM-8.json

Get list of installed software on a local machine

```
Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*" |  
Select-Object DisplayName, DisplayVersion, Publisher, InstallDate |  
Sort-Object DisplayName
```

Get list of installed software on a remote machine

```
Invoke-Command -ComputerName "RemoteMachineName" -ScriptBlock {  
    Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*" |  
    Select-Object DisplayName, DisplayVersion, Publisher, InstallDate |  
    Sort-Object DisplayName  
}
```

Export the list to a CSV file

```
Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*" |  
Select-Object DisplayName, DisplayVersion, Publisher, InstallDate |  
Sort-Object DisplayName |  
Export-Csv -Path "C:\SoftwareInventory.csv" -NoTypeInformation
```

Palo Alto Firewall Allowlist

Thursday, September 5, 2024 1:59 PM

Configuring allowlisting in Palo Alto Networks Next-Generation Firewalls (NGFW) is typically done using the firewall's Application Control feature, which allows administrators to control and restrict which applications can run on the network. Palo Alto uses App-ID technology to identify applications and then allows you to set policies to permit or block them.

Here's a step-by-step guide to configuring allowlisting on a Palo Alto firewall:

1. Log in to the Palo Alto Firewall

- Access the Web Interface by entering the IP address of the firewall in your web browser.
- Log in using an account with administrative privileges.

2. Create a Security Policy for Allowlisting

Palo Alto uses Security Policies to define which applications or traffic are allowed or denied. To create an allowlist, you will set up a policy that explicitly permits approved applications while blocking all others.

Steps to Create the Policy:

1. Go to Policies > Security.
2. Click on "Add" to create a new security policy rule.

Policy Configuration:

- Name: Name your rule (e.g., "Allowlisted Apps").
- Source: Define the Source Zone (e.g., the internal or trusted network).
- Destination: Define the Destination Zone (e.g., external or untrusted network, or a specific zone depending on your use case).
- Application: Use the App-ID feature to select which applications are allowed. You can:
 - Click on the "Application" tab in the rule editor.
 - Click "Add" and search for and select the applications you want to allow. Palo Alto has a prebuilt database of application signatures.
- Action: Set the action to Allow.

3. Commit the changes to apply the new rule.

3. Block All Other Applications

After creating the allowlist, you need to block all other traffic (i.e., any application that isn't explicitly allowed).

- Create a Deny-All Policy:
 1. Go back to Policies > Security.
 2. Click "Add" to create another rule and position it below your allowlist rule.
 3. Name this rule (e.g., "Block All Others").
 4. In the Application tab, set it to any.
 5. In the Action tab, set the action to Deny.
- This will ensure that only the applications listed in your allowlist policy are allowed, and all others will be blocked.

4. Enable URL Filtering (Optional)

You can enhance the allowlist by using URL Filtering to control web traffic. You can allow access only to specific URLs or categories of websites and block everything else.

Steps to Configure URL Filtering:

1. Go to Objects > Security Profiles > URL Filtering.
2. Create a new profile and specify categories or URLs that are allowed.
3. Assign this profile to the Security Policy you created for allowlisting.

5. Configure Logging and Monitoring

Monitoring and logging help track which applications are being allowed or blocked:

- Enable logging for the allowlist and deny-all policies:
 - Under each policy, go to the Actions tab.
 - Enable Log at Session Start and Log at Session End.
- Go to Monitor > Traffic Logs to review logs of allowed and denied traffic.

6. Commit and Test

- Once all configurations are completed, click on the Commit button in the top-right corner of the interface to apply changes.
- Test the allowlist by attempting to access allowed applications and verifying that non-allowlisted applications are blocked.

7. Ongoing Maintenance

- Regularly update the allowlist to accommodate new applications or remove old ones as your environment evolves.
- Monitor traffic logs to ensure that applications are being correctly allowed or blocked.

Best Practices:

- Regularly Update Application Signatures: Palo Alto regularly updates its App-ID database, so ensure that you are keeping it up-to-date to recognize new applications or threats.
- Test Policies Before Full Deployment: Apply the allowlist policy to a small group or environment first to ensure it doesn't block critical applications.
- Backup Configuration: Always create a backup of your firewall configuration before making significant changes.

By following these steps, you can successfully configure an allowlist on your Palo Alto firewall to restrict network access to only approved applications while blocking unauthorized or malicious traffic.

Aruba ClearPass Allowlisting

Thursday, September 5, 2024 2:00 PM

Aruba ClearPass allows you to define policies that control what devices, users, or applications are allowed to connect to the network. Below are several use cases where ClearPass can enforce allowlisting:

1. Device-Based Allowlisting

ClearPass can allowlist specific devices (e.g., laptops, smartphones, printers, etc.) that are permitted to access the network. This is commonly done by identifying devices based on:

- MAC address.
- Certificate-based authentication.
- Device fingerprinting (e.g., OS version, hardware type).
- Role-based access control (RBAC).

How to Configure Device Allowlisting:

- Create a Device Group: You can create a group of approved devices and allow only these devices to connect to the network.
- Configure Policy Manager: Use ClearPass Policy Manager to define a policy that permits devices that match the criteria (e.g., MAC address or device profile).
- Block All Other Devices: Set a default policy that denies network access for any device not in the allowlist group.

2. User-Based Allowlisting

ClearPass can allowlist users based on identity, allowing only authenticated and approved users to access network resources. This is typically done using:

- 802.1X Authentication (with credentials like username/password or certificates).
- LDAP/Active Directory Integration to verify if the user is authorized.
- Role Assignment: Based on user attributes, ClearPass can assign roles and permissions.

How to Configure User Allowlisting:

- Integrate with Identity Sources: Connect ClearPass to an identity source like Active Directory, LDAP, or a RADIUS server.
- Create Access Control Policies: Define policies that allow only users in certain groups (e.g., corporate employees, specific departments) to access network resources.
- Default Deny: Set a default deny policy for users not in the allowlist (e.g., guest users or unknown identities).

3. Application-Based Allowlisting

Aruba ClearPass can be used to enforce application control by allowing only approved applications or traffic patterns to flow through the network. While Aruba ClearPass itself doesn't inspect application content directly (like a firewall), it integrates with Aruba's Policy Enforcement Firewall (PEF) to enable application-level control.

How to Configure Application Allowlisting:

- Integrate with Aruba Network Devices: ClearPass works with Aruba switches, wireless access points, and controllers to enforce application control policies.
- Define Roles and Policies: Use ClearPass to assign specific user or device roles that are allowed to use certain applications (e.g., VoIP, web traffic, etc.).
- Allowlist Certain Applications: In conjunction with Aruba's AppRF feature (Application Recognition Firewall), you can allowlist specific applications based on deep packet inspection, ensuring that only approved traffic (e.g., Office 365, Salesforce) is permitted.

4. Endpoint Compliance and Posture-Based Allowlisting

ClearPass includes posture assessment capabilities, allowing you to allowlist endpoints that meet specific security requirements, such as:

- Antivirus/anti-malware installed.
- Device encryption enabled.
- Operating system patches applied.

How to Configure Compliance-Based Allowlisting:

- Set Posture Policies: Define compliance policies for devices (e.g., up-to-date antivirus, latest OS patches).
- Allow Network Access Only for Compliant Devices: Create rules that allow only devices meeting these posture checks to connect to the network. Devices that don't comply can be quarantined or denied access.

5. Guest Access Control (Allowlisting Guests)

For guest access, ClearPass can provide temporary network access to pre-approved devices or users, while blocking all other guests from accessing corporate networks.

How to Configure Guest Allowlisting:

- Guest User Role: Create specific guest roles and assign limited access policies based on a predefined allowlist of approved guest users.
- MAC Authentication for Guests: After the guest is onboarded and approved, ClearPass can allowlist their MAC address for future connections.
- Time-Based Allowlist: You can create temporary allowlists that expire after a set time period for guests or contractors.

Steps to Configure Allowlisting in Aruba ClearPass

1. Set Up Policy Manager:
 - Log into ClearPass Policy Manager.
 - Go to Configuration > Enforcement > Policies to create or modify policies for access control.
 - Use conditions like MAC addresses, user roles, or device posture to define what is allowed.
2. Create Device or User Groups:
 - Go to Identity > Endpoints or Users and create groups for devices or users that should be on the allowlist.
 - Associate these groups with the policies you created.
3. Integrate with Network Devices:
 - Integrate ClearPass with Aruba switches, access points, and controllers to enforce policies.
 - In wired environments, ensure that ClearPass is communicating with the switches via 802.1X or MAC authentication for allowlisting.
4. Test the Configuration:
 - Test the allowlist by attempting to connect with both approved and unapproved devices or users.
 - Monitor logs and reports to ensure the allowlist is functioning as expected.
5. Set Default Policies to Deny All Others:
 - Configure default deny policies in ClearPass to block all devices, users, or traffic that is not explicitly allowed.

By using these features, Aruba ClearPass can be a powerful tool for enforcing allowlisting in your network, controlling access based on device type, user identity, application usage, or compliance posture.

Configuring an application blocklist is an essential security measure to prevent unauthorized, harmful, or non-business-related applications from running on your systems. The best way to configure application blocklists depends on your environment, available tools, and desired level of control. Below are best practices and steps for configuring application blocklists using various tools and strategies.

Best Practices for Application Blocklisting

1. Understand the Environment:
 - Inventory of Applications: Identify all legitimate applications in your environment and create a baseline of what should be allowed or blocked.
 - Risk Assessment: Assess the security risks associated with certain applications. Block applications known to be risky (e.g., torrent clients, P2P software, remote access tools like TeamViewer, etc.).
2. Use Policy-Based Management:
 - Create policies that are centrally managed and automatically enforced across endpoints. Avoid manually managing blocklists on individual machines.
3. Layered Approach:
 - Combine application blocklisting with other security measures, such as firewalls, antivirus, and endpoint protection solutions, for a more comprehensive defense.
4. Regular Updates:
 - Continuously update your blocklist to account for new threats, software vulnerabilities, or unauthorized software.
5. Role-Based Policies:
 - Set up blocklist policies based on user roles. For example, block certain applications for standard users but allow them for IT administrators if necessary.

Tools for Configuring Application Blocklists

Different tools and platforms can be used to configure application blocklists depending on your environment.

1. Microsoft Defender Application Control (WDAC)

Best for: Organizations using Windows environments.

Steps:

- Create Application Control Policies: In the Windows Defender Security Center, go to Application Control.
- Block Specific Applications: Use Publisher, File Hash, or Path Rules to block specific applications.
- Set Enforcement Mode: Set the policy enforcement to block mode to prevent the execution of disallowed applications.
- Deploy via Group Policy: Use Group Policy or Microsoft Endpoint Manager to deploy the policy across multiple devices.
- Test and Deploy: Always test the blocklist in a non-production environment first before full deployment.

Advantages:

- Deeply integrated into Windows, providing native security.
- Allows granular control of applications based on file hashes, publishers, and paths.

2. Third-Party Endpoint Protection Solutions (e.g., Carbon Black, McAfee, Symantec)

Best for: Enterprises looking for comprehensive endpoint protection.

Steps:

- Access Application Control Settings: Log into the management console (e.g., Carbon Black, Symantec Endpoint Protection, or McAfee ePO).
- Create Blocklist Policies:
 - Define the applications to block using file signatures, file paths, or file hashes.
 - These platforms often include intelligence on potentially harmful applications, so you can block based on threat intelligence as well.
- Apply the Policy: Apply the blocklist policy to all endpoints, groups, or specific users depending on your environment.
- Monitor and Update: Use the dashboard to monitor blocked applications and update the blocklist regularly.

Advantages:

- Provides centralized control and advanced reporting capabilities.
- Blocklists are often updated with real-time threat intelligence, making them effective against new threats.

3. Firewalls with Application Control (e.g., Palo Alto Networks, Cisco Firepower, Fortinet)

Best for: Network-level control for blocking applications across the entire organization.

Steps:

- Log into the Firewall Management Console.
- Create Application-Based Rules:
 - Define rules in the firewall to block applications by category, such as gaming, social media, file sharing, or specific application signatures (e.g., blocking BitTorrent or Facebook).
 - Most modern firewalls use App-ID or similar technologies to identify and block applications at Layer 7 (application layer).
- Deploy to Zones or Groups: Apply these policies across specific network zones, user groups, or the entire network.
- Monitor Traffic: Monitor traffic logs to detect any attempts to bypass blocked applications.

Advantages:

- Allows for granular control over applications at the network level, preventing unauthorized traffic.
- Can enforce application blocklists without requiring changes on the endpoint itself.

4. Group Policy (for Windows environments)

Best for: Small to medium-sized Windows-only environments.

Steps:

- Open Group Policy Management: On your domain controller, open Group Policy Management.
- Navigate to Application Control Policies:
 - Go to User Configuration > Administrative Templates > System > Don't run specified Windows applications.
- Specify Blocked Applications:
 - Add the executable names (e.g., chrome.exe, spotify.exe) of the applications you want to block.
 - Be specific to avoid blocking legitimate software.
- Apply Policy to Organizational Units (OUs): Link the Group Policy to the desired OUs to enforce it across users and devices.
- Monitor and Test: Use logs to verify that blocked applications are not being executed.

Advantages:

- Native to Windows environments, making it a low-cost solution for small to medium-sized businesses.
- Simple to configure and manage for Windows-only environments.

5. Application Blockers for macOS and Linux (e.g., Jamf for macOS, AppArmor for Linux)

- Jamf (for macOS):
 - Best for: macOS environments in enterprises.
 - How to Configure: Use Jamf Pro to create a configuration profile that blocks specific applications by bundle ID or file path.
 - Deploy via MDM: Deploy the configuration profile to managed devices through mobile device management (MDM).
- AppArmor (for Linux):
 - Best for: Securing Linux systems.
 - How to Configure: Use AppArmor profiles to enforce application control. You can create profiles that block specific applications from running or restrict their permissions.
 - Advantages: Provides granular control over what applications can run and their capabilities on Linux systems.

6. Mobile Device Management (MDM) for Mobile Platforms

Best for: Organizations that need to block applications on mobile devices.

Steps:

- Choose an MDM Solution: Use MDM solutions like Microsoft Intune, Jamf, VMware Workspace ONE, or Cisco Meraki.
- Create Application Blocklist: In the MDM console, define a list of applications (by app name or bundle ID) that are restricted from being installed or executed.
- Deploy to Devices: Push the blocklist policy to all enrolled devices or specific groups of users.
- Monitor Compliance: Use the MDM platform to monitor whether devices are compliant with the application blocklist.

Advantages:

- MDM solutions give you centralized control over mobile devices and can prevent users from installing or running disallowed applications.
- Suitable for BYOD or corporate-owned mobile environments.

Key Considerations

- Ease of Management: Choose a tool that provides centralized and automated policy enforcement to minimize administrative overhead.
- Granularity of Control: Ensure that the solution allows you to define blocklist policies based on multiple criteria (file hashes, paths, users, roles).
- Visibility and Reporting: Your chosen tool should provide detailed logs and reports to help you monitor blocked applications and quickly adjust policies if needed.
- Compliance and Security Needs: Consider regulatory compliance requirements (e.g., GDPR, HIPAA) that might necessitate blocking specific categories of software (e.g., data-sharing applications).

By following these strategies and using the right tools for your environment, you can create and enforce an effective application blocklist that helps mitigate security risks and enforces productivity.

Allowlist

Tuesday, September 17, 2024 2:41 PM

The best solution for application allowlisting depends on the specific needs of your organization, such as the environment you're managing (Windows, Linux, macOS), network complexity, and the scale of deployment. Below are some of the top solutions for application allowlisting, each suited to different scenarios:

1. Microsoft Defender Application Control (WDAC)

Best for: Organizations using Windows environments (Enterprise, Education, and Server).

Key Features:

- Built-in to Windows: No additional software is needed for basic allowlisting.
- Policy-based control: You can define policies to allow only specific applications, scripts, and drivers.
- Integration with Microsoft Defender: Provides robust security by integrating with other Microsoft security tools.
- Flexible rules: Allows control based on publisher, file hash, or path rules.
- Block malicious code: Zero-day attacks and unapproved applications are blocked.

Why it's a great choice:

- WDAC is deeply integrated into Windows, making it easy to manage through Group Policy or Intune.
- It's an enterprise-grade solution for businesses heavily reliant on the Windows ecosystem.
- Free for Windows users, providing robust security without additional costs.

2. Carbon Black (VMware)

Best for: Enterprises needing advanced threat detection and endpoint protection.

Key Features:

- Dynamic application control: Carbon Black's application control continuously enforces policies, ensuring that only approved applications run.
- Real-time monitoring: Tracks application behavior and blocks unauthorized software in real time.
- Comprehensive security: Carbon Black integrates allowlisting with broader endpoint security, protecting against malware, ransomware, and exploits.
- Cloud or on-premise management: Offers flexibility for managing allowlists in cloud or traditional IT infrastructures.

Why it's a great choice:

- Highly effective for large-scale deployments in enterprises with complex security needs.
- Real-time, behavioral-based security adds an extra layer of defense beyond simple allowlisting.

3. Symantec Endpoint Protection

Best for: Enterprises looking for comprehensive endpoint security with allowlisting capabilities.

Key Features:

- Application control: Includes allowlisting as part of a broader endpoint protection strategy.
- Behavioral analysis: Combines allowlisting with heuristic and signature-based detection.
- Centralized management: Policies can be managed centrally for large organizations with numerous endpoints.
- Integrated security: Works well with other Symantec tools to provide comprehensive protection.

Why it's a great choice:

- Combines allowlisting with traditional antivirus and intrusion prevention features.
- Best suited for enterprises that need a unified solution for endpoint management and application control.

4. McAfee Application Control

Best for: Organizations needing dynamic allowlisting with minimal management overhead.

Key Features:

- Dynamic allowlisting: Automatically updates allowlists based on approved software to reduce administrative effort.
- Tamper protection: Prevents unauthorized users from changing or circumventing policies.
- Integration with ePolicy Orchestrator (ePO): Provides centralized policy management and reporting across large networks.
- Small footprint: Minimal impact on system resources.

Why it's a great choice:

- Ideal for organizations where IT teams want minimal manual intervention in managing allowlists.

- Supports both Windows and Linux, making it suitable for heterogeneous environments.

5. Ivanti Application Control (formerly AppSense)

Best for: Enterprises with complex, multi-device environments requiring granular control.

Key Features:

- Granular policies: Allows highly detailed control over which applications can run based on user, location, or device.
- Role-based access control: You can create policies that allow applications to run based on user roles or profiles.
- Least privilege enforcement: Helps enforce the principle of least privilege by dynamically elevating privileges for authorized applications only.
- Advanced reporting: Provides detailed insights into application usage, ensuring compliance and security.

Why it's a great choice:

- Best for organizations that require highly customizable allowlisting based on user roles and detailed policies.
- Integrates well into virtual desktop infrastructures (VDI) and remote work environments.

6. Faronics Anti-Executable

Best for: Small to mid-sized businesses and education environments.

Key Features:

- Simple allowlisting: Blocks all applications except those explicitly allowed.
- Application blocking: Prevents unauthorized executables and applications from running, ensuring a secure environment.
- Lightweight: Minimal system impact, making it a great choice for lower-resource environments.
- Remote management: Allows centralized management for easy deployment across multiple endpoints.

Why it's a great choice:

- Suitable for smaller organizations or educational institutions needing a straightforward, easy-to-manage allowlisting solution.
- Affordable and user-friendly, making it a good entry-level option for small businesses.

7. Tenable Application Control (formerly FlawCheck)

Best for: Containerized environments and DevOps-focused organizations.

Key Features:

- Container security: Focuses on allowlisting applications within containers, ensuring that only approved images and software are allowed to run.
- DevOps integration: Integrates seamlessly with CI/CD pipelines, ensuring security in DevOps workflows.
- Vulnerability management: Works in tandem with Tenable's vulnerability scanning tools to ensure only secure applications are deployed.

Why it's a great choice:

- Ideal for cloud-native applications and containerized environments.
- Provides real-time security for DevOps and microservices-based applications.

8. Cisco Secure Endpoint (formerly AMP for Endpoints)

Best for: Organizations seeking cloud-based, application-aware endpoint security with allowlisting capabilities.

Key Features:

- Cloud-managed application control: Policies can be managed and updated from a centralized cloud interface.
- Threat intelligence integration: Integrates with Cisco's Threat Intelligence to dynamically adjust allowlists and block known malicious applications.
- Continuous monitoring: Provides real-time monitoring and control over applications to detect any suspicious behavior.
- Endpoint protection: Integrates allowlisting with broader endpoint security features, including malware detection and sandboxing.

Why it's a great choice:

- Ideal for hybrid environments with both cloud and on-premise infrastructure.
- Provides comprehensive protection with built-in intelligence, reducing manual intervention.

Choosing the Best Solution

When selecting the best application allowlisting solution for your organization, consider the following factors:

- Platform Compatibility: Ensure the solution supports your operating systems (Windows, Linux, macOS, etc.).
- Scale: For large enterprises, tools like Carbon Black or Symantec offer centralized management and scalability. Smaller businesses might prefer simpler tools like

Faronics Anti-Executable.

- **Integration with Existing Security Tools:** Look for solutions that integrate well with your existing security stack (firewalls, antivirus, threat intelligence platforms).
- **Management Overhead:** Consider whether you need dynamic allowlisting (like McAfee Application Control) or if a static approach is sufficient.
- **Industry-Specific Needs:** If you operate in a containerized or cloud environment, consider solutions like Tenable Application Control.

By aligning these factors with your organization's specific requirements, you can choose the most effective application allowlisting solution for securing your environment.

1. Credential Harvesting and Escalation

- **Estimated Effectiveness: 80%**
 - Application controls could have blocked unauthorized use of tools like PowerShell, `certutil.exe`, and `WMIC` for credential harvesting and privilege escalation.
 - Effectiveness is slightly reduced because attackers might bypass basic controls through misconfigurations or signed, but malicious, scripts.

2. Data Exfiltration

- **Estimated Effectiveness: 90%**
 - Tools like `Rclone.exe` and unauthorized access to SharePoint files would have been blocked.
 - Most data exfiltration tools would fail if properly managed allowlists were in place.
 - A small reduction in effectiveness accounts for potential misconfigurations or attackers finding ways to use authorized tools for exfiltration.

3. Data Destruction

- **Estimated Effectiveness: 85%**
 - Application controls could restrict access to backup management software, making it difficult to delete snapshots or manipulate backup infrastructure.
 - Effectiveness depends on the segregation of privileges and enforcement of controls on critical systems.

4. Ransomware Deployment

- **Estimated Effectiveness: 95%**
 - Ransomware binaries and scripts are often unsigned or unapproved and would likely be blocked by strong application control policies.
 - High effectiveness is due to the limited ability of ransomware to function without executing malicious code.

5. Indicators of Compromise (IOCs)

- **Estimated Effectiveness: 90%**
 - Unauthorized applications like `ConnectionManager.exe` would have been logged and potentially flagged during execution attempts.
 - Combined with monitoring tools, application control can greatly enhance detection of suspicious activities.

Overall Estimated Effectiveness: 88%

Application controls are highly effective for preventing unauthorized software execution and limiting lateral movement during attacks.

However, residual risks include:

1. **Initial Exploits:** If attackers exploit legitimate, pre-approved applications or signed scripts, application controls alone may not stop them.
2. **Policy Gaps:** Misconfigured or outdated allowlists could reduce effectiveness.
3. **Bypass Techniques:** Sophisticated attackers may find ways to bypass application controls using trusted processes or vulnerabilities.

Maximizing Effectiveness

To achieve closer to 100% effectiveness:

- **Regular Policy Updates:** Continuously update allowlists and blocklists to reflect evolving threat landscapes.
- **Integration with Monitoring Tools:** Pair application control with SIEM or EDR tools for enhanced visibility.
- **User Education:** Train employees to avoid behaviors that could inadvertently bypass application controls (e.g., executing macros or untrusted files).

Conclusion

While application controls are not foolproof, their potential to block nearly 90% of malicious actions in this scenario underscores their value as a critical layer in a defense-in-depth strategy.

Application Control

Tuesday, September 17, 2024 2:42 PM

1. Approach to Software Control

- **Application Whitelisting:**

Whitelisting focuses on **explicitly allowing only approved applications** to run on a system. This means that even if a user manages to download or attempt to install a program, it won't execute unless it's been pre-approved and added to the whitelist.

- **Blocking Software Downloads:**

Blocking downloads is a more limited approach that **restricts users from downloading software** from the internet or other sources. It prevents unauthorized software from being obtained, but if software is already present or installed through other means (e.g., removable media, file transfer, or network shares), it may still run unless further controls are in place.

2. Scope of Control

- **Application Whitelisting:**

It controls **all execution** of applications, regardless of how they were obtained (downloaded, copied, or pre-installed). Only authorized software is allowed to execute, providing **complete control** over what can run on the system.

- **Blocking Software Downloads:**

This method typically applies only to internet-based downloads and does not prevent users from running software that's already installed or transferred via other methods. It offers **limited control** since users might find ways to circumvent the restriction or use non-internet sources.

3. Security Impact

- **Application Whitelisting:**

Offers a **proactive, all-encompassing** security measure by preventing the execution of unauthorized software, even if it's already present on the system. It protects against malware, zero-day exploits, and insider threats by ensuring only trusted applications are run.

- **Blocking Software Downloads:**

While it can reduce the risk of users downloading harmful software (malware, adware, etc.), it's more **reactive and partial** since it doesn't account for other attack vectors, like installing software via USB or running pre-installed vulnerable applications. It can be **easier to bypass**, especially by knowledgeable users.

4. Granularity of Control

- **Application Whitelisting:**

Whitelisting operates at a **fine-grained level** by specifying which applications are allowed to execute on the system, typically down to the file or executable level. This allows administrators to tailor security controls to specific needs, such as allowing only necessary versions of software or specifying rules for software behavior.

- **Blocking Software Downloads:**

This approach offers **coarse control** by generally preventing users from downloading executable files or installers from the web. It's less customizable and doesn't differentiate between types of software that might be useful or necessary for the user's job and those that are potentially harmful.

5. User Experience and Flexibility

- **Application Whitelisting:**

This method can be more **restrictive**, as users can only run pre-approved applications, which could impact flexibility. However, it provides more targeted security and can be adapted based on roles, permissions, or specific requirements, allowing for different whitelists for different departments or tasks.

- **Blocking Software Downloads:**

It's generally less intrusive since users may still be able to run or install pre-approved software in other ways. However, it can create frustration if users need legitimate software that is blocked from downloading and doesn't prevent them from running applications they already have access to.

6. Management and Maintenance

- **Application Whitelisting:**

Requires ongoing management to **maintain the whitelist**—adding approved applications, updating versions, and removing obsolete or insecure software. It can be more resource-intensive to implement and maintain but provides a high level of security control.

- **Blocking Software Downloads:**

Is usually easier to set up and maintain as it often involves **simple policy configurations** (e.g., browser or system restrictions). However, it's less effective for comprehensive security control and doesn't need constant updates like a whitelist.

7. Protection Against Insider Threats

- **Application Whitelisting:**

Protects against both **malicious insiders and accidental actions** by ensuring that no unauthorized software can run, even if a user with higher privileges or knowledge tries to bypass restrictions.

- **Blocking Software Downloads:**

While it can help mitigate some insider threats, it is **less robust** since users with appropriate permissions may still install or run unauthorized software through other methods.

Conclusion

- **Application whitelisting** provides **comprehensive, proactive control** over what software can run on a system, offering more robust protection but requiring more effort to maintain.

- **Blocking software downloads** is a more **reactive, limited approach** that prevents users from obtaining software from certain sources but doesn't control what can be executed or installed through other channels.

Ransomware

Tuesday, September 17, 2024 2:43 PM

How Application Whitelisting Prevents Ransomware:

1. **Restricts Unauthorized Executables:**
 - o Ransomware typically operates by executing malicious software to encrypt files. With application whitelisting in place, only pre-approved (whitelisted) applications can execute, effectively **blocking any unauthorized ransomware from running**.
 - o This prevents both traditional ransomware, which comes as an executable, and more advanced types that use scripts or macros to start the encryption process.
2. **Prevents Zero-Day Exploits:**
 - o **Zero-day ransomware** variants exploit unpatched vulnerabilities in legitimate software. Application whitelisting restricts which applications and processes can run, even if they attempt to exploit a vulnerability. Since **only approved apps can run**, even if ransomware tries to piggyback on an authorized program, it is likely to be blocked.
3. **Stops Drive-By Downloads and Malicious Attachments:**
 - o Many ransomware attacks begin when a user downloads malicious files (e.g., through email attachments or compromised websites). Application whitelisting prevents these malicious executables from running, stopping ransomware at its entry point.
4. **Blocks Macro-Based Ransomware:**
 - o Ransomware can be delivered via malicious macros in Office documents. Whitelisting prevents unauthorized macros or scripts from executing unless explicitly allowed. This drastically reduces the risk of ransomware spread through **document-based attacks**.
5. **Defense Against Fileless Malware:**
 - o Some ransomware operates using **fileless methods**, where malicious code is executed directly in memory, without writing files to disk. Whitelisting can help block these attempts by controlling which scripts, PowerShell commands, or interpreters are allowed to run, reducing the attack surface.

Why Application Whitelisting Is Not a Complete Solution:

1. **Whitelist Mismanagement:**
 - o If the whitelist is not carefully managed, attackers may find ways to exploit **whitelisted applications** or processes to run malicious code. For instance, if a vulnerable, trusted application (like Microsoft Word or PowerShell) is allowed, ransomware may use it to launch the attack.
2. **Social Engineering and Insider Threats:**
 - o Application whitelisting doesn't prevent users from **falling victim to phishing attacks** or social engineering techniques. If a user unknowingly approves a malicious application or grants unnecessary privileges, ransomware can bypass the whitelist.
3. **Exploiting Authorized Applications:**
 - o Some ransomware may hijack legitimate applications that are on the whitelist or use approved software to run scripts that encrypt files (e.g., through PowerShell or Windows Script Host). Whitelisting cannot stop this unless these scripting environments are also restricted.
4. **File Encryption and Data Theft:**
 - o While whitelisting may prevent the ransomware from executing, it doesn't stop **data encryption or exfiltration** by other means, like through **legitimate remote access software** or poorly configured access controls. Attackers may still be able to steal data using whitelisted applications before the ransomware is activated.

Best Practices to Enhance Ransomware Protection with Whitelisting:

1. **Combine with Other Security Measures:**
 - o Application whitelisting is more effective when combined with other layers of defense:
 - **Endpoint Detection and Response (EDR):** Detects suspicious activity.
 - **Behavioral Analysis:** Monitors unusual patterns in file activity or encryption.
 - **Antivirus/Anti-Malware:** To detect known ransomware variants.
2. **Whitelist Maintenance:**
 - o Regularly **review and update the whitelist** to ensure only legitimate, up-to-date applications are included. Remove obsolete or potentially vulnerable applications that may become a target for ransomware.
3. **Limit Script Execution:**
 - o Whitelist **specific scripts and script interpreters** (like PowerShell) very cautiously. Disable or restrict PowerShell, macros, and other commonly exploited utilities unless absolutely necessary.
4. **User Education and Training:**
 - o Train users to recognize phishing attempts, social engineering, and other common attack vectors that could lead to ransomware bypassing whitelisting protections.
5. **Restrict Administrator Privileges:**
 - o Limit the use of administrative privileges to reduce the chances of ransomware using elevated permissions to execute or spread.

Software Inventory

Tuesday, September 17, 2024 2:44 PM

1. Create a Software Inventory

- **Automated Discovery Tools:** Use software inventory tools to automatically gather information about all installed software across the network. These tools can track software versions, installation dates, and update statuses. Popular tools include:
 - **Microsoft System Center Configuration Manager (SCCM)** for Windows environments.
 - **ManageEngine Desktop Central** for endpoint management.
 - **SolarWinds Network Performance Monitor** for tracking software on servers and devices.
 - **Open-source solutions** like OCS Inventory NG or GLPI.
- **Manual Inventory:** If automation tools are not available, manually generate a list of installed software by:
 - Running a command to list installed software on each device (e.g., `wmic product get name,version` for Windows, `dpkg --get-configurefiles` for Linux, or using package managers on macOS).
 - Exporting this data into a spreadsheet or centralized database.

2. Identify Outdated Software

- **Compare Software Versions with Vendor Databases:**
 - Use your inventory tool's built-in capabilities or vendor documentation to compare installed software versions with the latest available versions. Some tools automatically flag outdated software.
 - Many commercial tools can help with this process:
 - **Patch My PC** or **Ivanti Patch Management** for checking patch status.
 - **WSUS (Windows Server Update Services)** for patching and checking Microsoft applications.
 - **Vulnerability scanners** (like Nessus, Qualys, or OpenVAS) that detect out-of-date software with known vulnerabilities.
- **Look for End-of-Life (EOL) Software:**
 - Outdated software may no longer receive security updates if it has reached its end-of-life. Refer to vendor websites or security advisories for EOL notifications, and update or replace these applications.
 - Some tools provide EOL reporting capabilities (e.g., **Flexera** or **Lansweeper**).

3. Remove Unneeded Software

- **Assess Usage:**
 - Use a **Software Usage Monitoring Tool:** Solutions like **Lansweeper**, **ManageEngine**, or **SCCM** can track how frequently each application is used. Flag applications that have not been used in the last 90 days (or a time frame appropriate for your organization).
 - **Manually Analyze Usage:** Review the software's last accessed date on individual machines or check user logs to see if certain applications have been idle for long periods.
- **Categorize Software:**
 - **Critical Applications:** Necessary for daily business operations, these should be regularly updated and kept in use.
 - **Productivity Applications:** Useful but not essential software. Consider if alternatives exist or whether these tools are necessary for specific users or departments.
 - **Legacy or Outdated Applications:** Identify software no longer compatible with modern systems or security standards.
 - **Unapproved Software:** Flag any software that has been installed without IT or security team approval. These could be potential security risks.
- **Assess Licensing:** Identify software that is no longer being used but still consumes licenses. Removing these can help avoid unnecessary licensing costs.

4. Automate the Removal Process

- **Create Removal Policies:** Define policies that automatically flag and uninstall software if it hasn't been used for a specific time period or if it's identified as out-of-date and not critical.
 - For Windows, you can use **Group Policy**, **PowerShell scripts**, or **SCCM** to remove software centrally.
 - For Linux, use package managers (e.g., `apt`, `yum`) with scripts to uninstall unused packages.
 - **Endpoint management tools** like **Jamf** for macOS can automate software removal across devices.
- **Batch Uninstallation:** Use scripts or management tools to uninstall unused or outdated software in bulk. For example, on Windows, PowerShell scripts or WMIC commands can be used:
`Get-WmiObject -Query "SELECT * FROM Win32_Product WHERE Name = 'OutdatedSoftware'" | ForEach-Object { $_.Uninstall() }`

On Linux, use commands like:

```
sudo apt-get remove --purge package_name
```

5. Review Vulnerabilities and Risks

- **Vulnerability Scanners:** Use vulnerability scanning tools to identify any outdated software with known vulnerabilities. These tools provide reports highlighting which applications pose security risks due to outdated versions (e.g., **Nessus**, **Qualys**, **Rapid7 Nexpose**).
- **Security Patching Tools:** Implement patch management solutions (like **Ivanti**, **WSUS**, or **SolarWinds Patch Manager**) to automate the process of updating software with known vulnerabilities.

6. Review Vendor and End-of-Life Lists

- Regularly review lists of software reaching **end-of-life** or **end-of-support** from vendors like Microsoft, Adobe, or Oracle. Ensure such software is either removed or replaced in your environment.
- Cross-reference these lists with your inventory to spot potential issues.

7. Plan for Application Updates and Removals

- **Update Outdated Software:** For essential software that is outdated but still needed, schedule updates. Many enterprise tools like **WSUS** or **SCCM** can handle this automatically.
- **Remove Unneeded Applications:** After identifying unnecessary or outdated software, create a plan for removal, ensuring to:
 - **Back up data** if any critical information is tied to the application.
 - Inform users and departments of any upcoming changes.
 - Test the removal process in a controlled environment before deploying it across the network.

8. Regular Maintenance and Review

- **Schedule Regular Audits:** Set up periodic reviews (e.g., quarterly) to reassess the software inventory, ensure critical applications are updated, and remove unneeded ones.
- **Integrate with IT Asset Management (ITAM):** Use ITAM practices to maintain a continuously updated inventory of software and track its lifecycle.

Tools for Automating Software Inventory Management:

1. **Microsoft SCCM:** Provides comprehensive software inventory, usage monitoring, and patch management for large environments.
2. **Lansweeper:** Offers software discovery, inventory, and end-of-life tracking features.
3. **Flexera:** Helps manage software licenses, lifecycle, and vulnerabilities.
4. **ManageEngine Desktop Central:** Centralized management of software installation, usage tracking, and automatic removal of unneeded apps.
5. **SolarWinds Patch Manager:** Handles automatic updates and vulnerability remediation.

Hardware Inventory

Tuesday, September 17, 2024 2:44 PM

1. Define Your Inventory Scope

- **Identify Asset Types:** Decide what hardware assets you need to track (e.g., workstations, laptops, servers, network devices, printers, mobile devices, etc.).
- **Location and Ownership:** Determine if you're accounting for assets across multiple locations (e.g., different offices, remote employees) and whether you need to include both company-owned and leased/rented assets.
- **Relevant Asset Information:** Define the information you need to capture about each asset, such as:
 - Asset Tag Number
 - Manufacturer
 - Model
 - Serial Number
 - IP Address
 - MAC Address
 - Processor, RAM, Storage
 - Purchase Date
 - Warranty Expiry Date
 - Location (Physical/Network)
 - Assigned User or Department

2. Use Automated Hardware Discovery Tools

Automated discovery tools can quickly scan the network to detect and list hardware assets along with detailed information about each device. Some popular tools include:

a. Network Discovery Tools

- **Lansweeper:** Can scan your network for all connected devices, pulling information like serial numbers, manufacturers, and IP addresses, and it supports a wide range of device types.
- **Microsoft System Center Configuration Manager (SCCM):** Allows for robust asset discovery and inventory management of Windows environments.
- **Spiceworks Inventory:** Provides network device scanning, including details like device specs and warranty information.
- **SolarWinds Network Performance Monitor:** Tracks network-connected devices and gathers data on network devices (routers, switches) and endpoints.

b. Agent-Based Tools

- **ManageEngine Asset Explorer:** Uses software agents installed on each device to continuously track the hardware inventory of computers, mobile devices, etc.
- **Ivanti IT Asset Management:** Allows tracking of IT assets, providing real-time data on hardware.
- **JAMF** (for macOS environments): Manages and tracks Apple hardware across your network.

c. Cloud-Based Solutions

- **AWS Systems Manager or Microsoft Intune:** These tools are useful for environments that include cloud infrastructure and devices managed remotely.
- **Google Admin Console** (for Chromebooks): Tracks hardware assigned to users in Google Workspace environments.

3. Ensure Asset Tagging for Physical Devices

- **Physical Asset Tags:** For tracking physical hardware like desktops, laptops, printers, and servers, ensure each device has a unique asset tag (either a barcode or QR code). These tags can be scanned using a **barcode scanner** or a **QR code reader** for fast identification and input into your inventory system.
- **Assign Tags During Acquisition:** When new hardware is purchased, assign a tag immediately and record its details (e.g., manufacturer, model, purchase date) in the inventory system.

4. Perform Manual Audits (If Necessary)

- **Physical Audit:** Walk through your office locations or data centers to verify that the hardware listed in your inventory matches what is physically present. Use the asset tags or serial numbers to match physical devices with the recorded inventory.
- **User Survey:** For assets assigned to remote workers or field employees, conduct surveys to confirm the presence and use of devices. Ask users to validate device information like model, serial number, and tag.

5. Integrate with Existing Systems

- **Active Directory (AD) Integration:** Many inventory tools integrate with Active Directory to automatically pull information about domain-joined devices. This ensures all systems connected to the corporate domain are included in the inventory.
- **DHCP and DNS Logs:** Use DHCP and DNS logs to track devices by their IP addresses, MAC addresses, and hostname.
- **IT Service Management (ITSM) Tools:** Tools like **ServiceNow** or **BMC Remedy** allow tracking of hardware assets in combination with service desk functions, helping to keep track of hardware lifecycle (purchasing, maintenance, decommissioning).

6. Regularly Update the Inventory

- **Scheduled Scans:** Use the automated tools mentioned above to schedule regular scans (daily, weekly, or monthly) to ensure the inventory is up to date with newly added or removed assets.
- **Real-Time Alerts:** Set up alerts to detect when a new device connects to the network or when an asset is decommissioned to avoid discrepancies between physical assets and your inventory.

7. Categorize and Organize the Data

- After scanning, categorize the devices in the inventory by:
 - **Device Type** (e.g., workstation, server, mobile device).
 - **Department or User:** Which department or individual owns or uses the device.
 - **Location:** Physical location or data center where the hardware resides.
 - **Lifecycle Status:** Mark devices as "in use," "in repair," "retired," or "decommissioned."
- Store the data in a **centralized database** or asset management tool that allows for easy access and reporting.

8. Review Hardware Lifecycles

- **Track Warranties:** Use the inventory tool to track the **warranty expiry dates** of all hardware assets. This helps you identify when a piece of equipment needs replacement or when extended warranties may be required.
- **End-of-Life (EOL) Planning:** Cross-reference your inventory with vendor-provided **end-of-life (EOL)** lists to ensure you're aware of upcoming EOL dates for servers, workstations, and network devices. Devices reaching EOL should be flagged for decommissioning or replacement.

9. Data Validation

- **Cross-Check Against Purchase Records:** Periodically cross-check the hardware inventory against purchase records or leasing contracts to ensure all assets are accounted for.
- **Reconcile with Other Systems:** Reconcile the inventory with **financial records** (e.g., fixed assets register) to ensure the asset values and lifespans align with accounting and procurement records.

10. Reporting and Documentation

- **Automate Reporting:** Use your inventory tool to generate reports on the hardware assets, showing metrics such as:
 - Total number of assets per location.
 - Assets nearing end-of-life.
 - Warranty expiration dates.
 - Under-utilized or unassigned devices.
 - Financial depreciation of assets.
- **Maintain Documentation:** Store all asset-related documentation (e.g., purchase orders, warranty information, device configurations) alongside the inventory for easy reference during audits.

Key Tools for Hardware Inventory Management

1. **Lansweeper:** Provides detailed scans of all connected devices and offers easy reporting for IT teams.
2. **Microsoft SCCM:** Tracks both software and hardware, offering detailed reports on asset lifecycle and configurations.
3. **ManageEngine AssetExplorer:** Tracks hardware assets throughout their lifecycle, helping with procurement, usage, and decommissioning.
4. **Ivanti IT Asset Management:** Delivers real-time visibility into hardware, software, and device usage across the organization.
5. **JAMF:** Focuses on Apple devices, providing hardware tracking and lifecycle management for macOS environments.
6. **ServiceNow:** Integrates hardware asset management with service management for comprehensive tracking and reporting.

Comments

Thursday, September 26, 2024 12:54 PM

1. Establish and Maintain an Accurate Inventory of Information Assets

- **Finding:** "The credit union has developed a comprehensive, regularly updated inventory of all information assets, including workstations, laptops, servers, network devices, security devices, and software applications. This inventory is documented with detailed information, such as device type, manufacturer, and operating system, ensuring appropriate security controls can be applied in accordance with Appendix A, Section III(B)."
- **Finding:** "The asset inventory is reviewed and updated quarterly to reflect new additions, modifications, or decommissioned assets. This process ensures the inventory remains accurate, reducing the risk of unauthorized access or misuse."

2. Implement and Enforce Information Security Controls

- **Finding:** "Access control mechanisms are in place for all information assets, including multi-factor authentication and role-based access controls. These controls restrict access to sensitive data and ensure that only authorized personnel can access member information, meeting the requirements of Appendix A, Section III(C)."
- **Finding:** "Sensitive member data stored and transmitted through relevant information assets is protected by strong encryption standards, such as AES-256. Encryption measures are applied consistently to protect against unauthorized disclosure, in line with Appendix A, Section III(A)."

3. Monitor and Test the Effectiveness of Controls

- **Finding:** "The credit union conducts regular security audits of all information assets, including network devices, security devices, and software applications. These audits ensure compliance with established security policies and confirm that assets are safeguarded against unauthorized access, fulfilling Appendix A, Section III(C) requirements."
- **Finding:** "Vulnerability scans are performed on a quarterly basis, and annual penetration testing is conducted on critical assets. These assessments help identify potential weaknesses and verify that the credit union's controls are functioning as intended, as required by Appendix A, Section III(C)."

4. Manage the Lifecycle of Information Assets

- **Finding:** "A lifecycle management program is in place to track the End-of-Life (EOL) and End-of-Support (EOS) dates for all hardware and software. This ensures that outdated or unsupported systems are replaced or upgraded in a timely manner, minimizing security risks, in compliance with Appendix A, Section III(A)."
- **Finding:** "The credit union follows secure decommissioning procedures for obsolete hardware and software, including secure data wiping and physical destruction of hard drives. These practices prevent unauthorized access to residual member information, in accordance with Appendix A, Section III(A)."

5. Ensure Proper Vendor and Third-Party Management

- **Finding:** "All third-party service providers with access to critical systems or member information are inventoried and their security practices are regularly reviewed. Due diligence ensures that their security controls align with the credit union's security requirements, as required by Appendix A, Section III(C)."
- **Finding:** "Contracts with third-party vendors explicitly include security requirements for protecting member information. Regular audits and compliance reviews of these providers confirm adherence to security standards, fulfilling the requirements of Appendix A, Section III(A)."

6. Develop and Maintain Policies for Information Asset Management

- **Finding:** "The credit union has a comprehensive Information Asset Management Policy that covers acquisition, tracking, updating, and decommissioning of assets. The policy is reviewed annually to ensure it remains current and aligned with evolving technologies and threats, as required by Appendix A, Section II(A)."
- **Finding:** "Regular policy reviews have ensured that the Information Asset Management Policy remains effective in managing and safeguarding member information assets. These reviews help the credit union respond proactively to changing risks and technologies."

7. Report to the Board of Directors

- **Finding:** "The annual report to the Board of Directors includes detailed updates on the information asset inventory, testing results, third-party vendor relationships, and key security incidents. The report also highlights any risks identified during audits and provides recommendations for mitigation, in compliance with Appendix A, Section I(C)(2)."
- **Finding:** "The Board receives regular updates on the Information Security Program's status, including the security of information assets. These updates ensure ongoing oversight and governance, strengthening the credit union's commitment to protecting member information."

8. Regularly Train Employees on Information Asset Security

- **Finding:** "The credit union provides ongoing security awareness training for all employees, including specific guidance on the secure management of information assets. This training helps employees recognize unauthorized devices, use software securely, and adhere to asset management practices, ensuring compliance with Appendix A, Section III(C)."
- **Finding:** "Training programs are reviewed and updated annually to reflect new security threats and evolving best practices. Employees are well-versed in protecting information assets, which enhances the overall security posture of the credit union."

1. End of Support and Patching Issues

- **Security Updates:** Microsoft ended mainstream support for Windows Server 2012 in 2018 and will end extended support in October 2023. After this, no further security patches will be provided, leaving the system vulnerable to new exploits.
- **Vulnerability Exploitation:** Without regular security updates, your DC is more susceptible to known vulnerabilities, which can be exploited by attackers to gain unauthorized access or execute malicious activities.

2. Outdated Security Features

- **Lack of Modern Security Controls:** Windows Server 2012 lacks some of the advanced security features present in newer versions of Windows Server, such as improved encryption standards, more robust auditing capabilities, and advanced threat protection.
- **Weak Encryption:** Older versions may still use weaker encryption algorithms (e.g., RC4, SHA-1) by default, which are vulnerable to modern cryptographic attacks.

3. Compatibility Issues

- **Application and Tool Compatibility:** Newer security and management tools may not be fully compatible with Windows Server 2012, limiting your ability to implement advanced security measures, such as next-generation antivirus, endpoint detection and response (EDR), or Security Information and Event Management (SIEM) systems.
- **Interoperability with Newer Systems:** As organizations upgrade other systems or introduce new technologies, compatibility issues may arise, leading to potential security gaps or operational inefficiencies.

4. Increased Attack Surface

- **Exploitable Legacy Protocols:** Older domain controllers may still support legacy protocols like NTLMv1, SMBv1, or older versions of Kerberos, which are known to have security weaknesses.
- **Higher Risk of Lateral Movement:** If an attacker compromises the outdated DC, they could more easily use it as a platform to move laterally within the network, gaining access to other systems or escalating privileges.

5. Limited Incident Response and Forensics

- **Insufficient Logging:** Windows Server 2012 has less advanced logging capabilities compared to newer versions, potentially making it harder to detect and investigate security incidents.
- **Forensics Challenges:** The limitations in built-in tools and logging can also complicate forensic analysis, making it more difficult to understand the full scope of a security breach.

6. Operational Risks

- **Resource Constraints:** As the hardware and software ecosystem evolves, running a legacy system like Windows Server 2012 may lead to performance bottlenecks or resource constraints, especially under the load of modern applications and services.
- **Complexity in Management:** Maintaining an outdated server requires more effort in terms of management, troubleshooting, and ensuring compatibility with newer systems, which can strain IT resources.

7. Regulatory and Compliance Issues

- **Non-Compliance with Standards:** Using unsupported software can lead to non-compliance with various industry regulations and standards, such as GDPR, HIPAA, or PCI-DSS, which often require systems to be regularly updated and patched.
- **Increased Audit Scrutiny:** Organizations using outdated systems may face increased scrutiny during audits, potentially leading to fines, penalties, or the need for costly remediation efforts.

8. Risk of Legacy Applications

- **Unsupported Legacy Applications:** Applications designed for Windows Server 2012 may be outdated and unsupported, introducing additional security vulnerabilities that can be exploited by attackers.
- **Difficulty in Migration:** Migrating from Windows Server 2012 to a newer platform might be challenging due to dependencies on legacy applications, leading to delays in addressing the security risks.

1. End of Support and Patching Issues

- **Risk:** Vulnerabilities due to lack of security updates.
- **Mitigation:**
 - M1030 - Patch Management: Regularly update all systems. While Windows Server 2012 is nearing the end of its support, consider upgrading to a supported version of Windows Server to continue receiving security updates.
 - M1042 - Disable or Remove Feature or Program: Remove unnecessary services and software from the Windows Server 2012 DC to reduce the attack surface, especially those that are no longer receiving updates.

2. Outdated Security Features

- **Risk:** Lack of modern security controls.
- **Mitigation:**
 - M1043 - Credential Access Protection: Implement multi-factor authentication (MFA) and stronger password policies to mitigate risks associated with weaker security features on Windows Server 2012.
 - M1053 - Data Backup: Regularly back up critical data, including DC configurations, to ensure quick recovery in case of a security incident.

3. Compatibility Issues

- **Risk:** Incompatibility with modern security tools and systems.
- **Mitigation:**
 - M1016 - Vulnerability Scanning: Regularly scan the DC and associated systems for vulnerabilities and misconfigurations, using tools that are compatible with older systems.
 - M1049 - Antivirus/Antimalware: Deploy antivirus and antimalware tools that are compatible with Windows Server 2012 to maintain a baseline level of protection.

4. Increased Attack Surface

- Risk: Exploitable legacy protocols and services.
- Mitigation:
 - M1041 - Encrypt Sensitive Information: Ensure that any communications involving the DC, especially those using legacy protocols, are encrypted to prevent data interception.
 - M1037 - Filter Network Traffic: Disable legacy protocols (e.g., SMBv1, NTLMv1) that are no longer necessary and use network filtering to block outdated protocols from being used.

5. Limited Incident Response and Forensics

- Risk: Insufficient logging and monitoring capabilities.
- Mitigation:
 - M1051 - Network Intrusion Prevention: Implement enhanced logging and monitoring using third-party tools compatible with Windows Server 2012. Forward logs to a centralized SIEM for better analysis.
 - M1022 - Restrict File and Directory Permissions: Limit access to critical logs and configuration files to ensure integrity and availability during forensic investigations.

6. Operational Risks

- Risk: Performance bottlenecks and management complexity.
- Mitigation:
 - M1057 - Audit: Conduct regular audits of system performance and resource utilization to identify and address potential bottlenecks before they impact operations.
 - **M1018 - Application

Finding: Non-Compliant Software Management Practices

Summary

An assessment of the software inventory and management practices revealed multiple compliance and security issues, including unauthorized, outdated, and unnecessary applications across the organization's systems. These deficiencies pose potential risks to the confidentiality, integrity, and availability of sensitive information, in violation of **12 CFR Part 748, Appendix A, Guidelines for Safeguarding Member Information**.

Details of the Findings

1. Software Inventory Management
- The organization lacks a comprehensive, centralized software inventory to track all installed applications, contrary to the guidelines under **12 CFR Part 748, Appendix A, III(C)(1)**, which requires maintaining an up-to-date inventory of information system components.

Multiple instances of duplicate software (e.g., redundant versions of 4K Video Downloader and Adobe tools) indicate ineffective inventory control.
2. Presence of Outdated and End-of-Support Software
- Identified applications such as **Microsoft Silverlight, MSXML 4.x, and Internet Explorer 11** are no longer supported by vendors, exposing the organization to potential security vulnerabilities. This violates the requirement to manage and remediate vulnerabilities as outlined in **12 CFR Part 748, Appendix A, III(C)(2)**.
3. Unnecessary or Unused Applications (Bloatware)
- Numerous non-business-critical applications, including **3D Builder, Farm Heroes Saga**, and promotional software (e.g., "Holiday Glow" and "Fields of Flowers"), were identified. These applications contribute to resource inefficiencies and increase the attack surface, contrary to the principles of minimizing risk in **12 CFR Part 748, Appendix A, II(C)**.
4. Insufficient Software Update and Patch Management
- Several applications, including outdated Adobe tools and OpenSSL versions, have not been updated to the latest secure versions. This violates **12 CFR Part 748, Appendix A, III(C)(3)**, which mandates maintaining and applying security patches in a timely manner.
5. Unauthorized and Prohibited Software
- Instances of unauthorized software were found that are not aligned with business requirements or security policies. The absence of proper controls for identifying and removing prohibited software is non-compliant with **12 CFR Part 748, Appendix A, III(C)(1)**.
6. Licensing Compliance Issues
- Licensing records for some commercial software are incomplete or unverified. This lack of oversight potentially breaches **12 CFR Part 748, Appendix A, III(A)(2)**, which requires appropriate oversight for third-party software and tools.

Regulatory Impact

The identified issues demonstrate a lack of adherence to the required safeguards under **12 CFR Part 748**, which mandates credit unions to implement appropriate measures to protect member information against risks associated with unauthorized access or use.

Recommendations

1. Develop and Maintain a Comprehensive Software Inventory
- Implement a centralized software asset management system to track all installed applications and their licensing status.
2. Remove Outdated and Unsupported Software
- Conduct a thorough review and immediately uninstall all software that is no longer supported by vendors.
3. Streamline Applications
- Remove redundant, unused, and non-essential applications to minimize the organization's attack surface and optimize system resources.
4. Enhance Software Update and Patch Management
- Establish a process to ensure all applications are regularly updated and patched in accordance with the organization's risk management policies.
5. Enforce Prohibited Software Policies
- Regularly review installed software and implement monitoring tools to identify and remove applications that violate organizational policies.
6. Ensure Licensing Compliance
- Audit all commercial software to confirm valid licensing and document proof of compliance.
7. Establish Continuous Monitoring

Allowlist for Authorized Software
The credit union **isolation and enforcement** of Authorized Software. An Allowlist for Authorized Software is a crucial line of defense for protecting enterprises given today's threat landscape, and it has an inherent advantage over traditional antivirus solutions. Specifically, an Allowlist for Authorized Software moves away from an application trust model where all applications are assumed trustworthy to one where applications must earn trust in order to run. An Allowlist for Authorized Software can help mitigate these types of security threats by restricting the applications allowed by users to run and the code that runs in the System Core (kernel). Allowlist for Authorized Software policies can also block unsigned scripts and MSIs, and restrict Windows PowerShell to run in Constrained Language Mode. The credit union should evaluate various solutions and conduct a risk analysis for the deployment of Allowlist for Authorized Software.

Asset Management: Incomplete Asset Management Practices

The current fragmented asset management approach creates **security risks, compliance gaps, and operational inefficiencies**. By implementing a structured process aligned with the criteria (e.g., inventory creation, classification, risk assessment, and policy enforcement), the organization can:

1. Identify and manage assets effectively.
2. Enhance security visibility and monitoring.
3. Mitigate risks associated with untracked and unsupported systems.
4. Ensure compliance with 12 CFR Part 748 and industry best practices.

Key Issues Identified

1. Limited Use of Discovery Features:
- Tools like **Defender** and **Qualys** were not fully utilized to discover and track all assets. This led to unmonitored and untracked devices and software.
2. Inconsistent Asset Classification:
- Lack of classification resulted in unclear prioritization of risks.
3. Lack of Continuous Monitoring:
- No automated alerts for unauthorized changes or installations, reducing visibility into potential threats.
4. Minimal Integration into Centralized Reporting:
- Asset data was not effectively integrated into reporting systems for analysis, delaying decision-making.
5. Absence of Policy Enforcement:
- Policies for enforcing **approved software usage** and blocking **unauthorized applications** were lacking.

Consequences of Fragmented Asset Management

1. Security Risks
- Untracked Assets:** Increase exposure to adversarial techniques such as **MITRE ATT&CK Initial Access (TA0001)**:
Example: **Exploitation of Public-Facing Applications (T1190)**, where unmonitored systems with unpatched software are exploited.

Unmanaged Endpoints: Create opportunities for malware delivery through techniques like **Spearphishing Attachment (T1566.001)**.

Persistence Risks: Unmonitored systems are vulnerable to attackers installing backdoors via **Web Shell (T1505.003)**.
2. Compliance Gaps
- 12 CFR Part 748:** Inadequate asset tracking hinders compliance with maintaining accurate inventories and timely patch management.

Example: Techniques such as **Valid Accounts (T1078)**, where adversaries exploit unmanaged credentials and systems.
3. Operational Inefficiencies
- Delayed responses to incidents like **ransomware attacks (MITRE Impact TA0040)**:
Example: Incomplete inventories delay isolating affected systems, exacerbating data loss and downtime.
4. Business Risk
- Privilege Escalation Risks (TA0004):** Unpatched software allows exploitation of vulnerabilities (e.g., **Exploitation for Privilege Escalation T1068**).

Increased Visibility Gaps: Lack of centralized reporting increases the persistence of threats and missed opportunities to remediate vulnerabilities.

Steps to Address Gaps

1. Discovery and Inventory
- Fully leverage **Microsoft Defender's Device Inventory** and **Qualys Global IT Asset Inventory** to:
Identify **all hardware and software assets**, including cloud and hybrid environments.
Use automated discovery tools to maintain **completeness**.
2. Classification and Prioritization
- Apply **tagging** and **categorization** features to prioritize assets based on:
Criticality, compliance requirements, and risk levels.
3. Risk Assessment and Monitoring
- Use **Threat & Vulnerability Management (TVM)** in Microsoft Defender and **Qualys Vulnerability Management** to:
Continuously assess and prioritize risks.
Enable real-time alerts for unauthorized changes or unapproved software installations.
4. Policy Enforcement
- Enforce software usage policies using tools like:
Microsoft Endpoint Manager (Intune) to block unapproved software.
Qualys Policy Compliance to ensure adherence to standards.
5. Integration and Reporting
- Integrate asset inventory data into centralized reporting platforms, such as:
Power BI for actionable insights.
ServiceNow for incident management.
- Generate compliance and vulnerability reports to support audits and executive decision-making.
6. Regular Reviews
- Schedule periodic reviews of:
Asset inventories to identify unsupported software and EOL hardware.
Asset management policies to align with **regulatory changes and emerging threats**.

Notes

Wednesday, September 4, 2024 4:29 AM

1. Privilege Escalation

- Mitigation:
 - M1042 - Disable or Remove Feature or Program: Restrict the ability to request certificates to authorized users only. Carefully configure certificate templates with minimal privileges.
 - M1026 - Privileged Account Management: Implement strict control over who can issue certificates and what privileges those certificates grant. Regularly review and audit certificate issuance processes.

2. Misissued Certificates

- Mitigation:
 - M1016 - Vulnerability Scanning: Regularly scan and audit issued certificates to ensure they are correctly authorized and aligned with security policies.
 - M1036 - Account Use Policies: Enforce policies that limit which users or systems can request specific types of certificates, reducing the risk of unauthorized certificate issuance.

3. Certificate Authority (CA) Compromise

- Mitigation:
 - M1043 - Credential Access Protection: Secure the CA infrastructure, including physical and logical access controls. Implement multi-factor authentication (MFA) for administrative access to the CA.
 - M1050 - Exploit Protection: Regularly patch and update the CA server and related systems to protect against vulnerabilities that could lead to CA compromise.

4. Lack of Proper Certificate Revocation

- Mitigation:
 - M1017 - User Training: Train administrators on the importance of maintaining and publishing certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP) responses.
 - M1031 - Network Segmentation: Ensure CRLs and OCSP services are accessible across the network and regularly updated to reflect the current status of issued certificates.

5. Weak Encryption or Hashing Algorithms

- Mitigation:
 - M1041 - Encrypt Sensitive Information: Use strong, modern encryption and hashing algorithms (e.g., SHA-256) for all certificates and ensure that older, weaker algorithms like SHA-1 are phased out.
 - M1053 - Data Backup: Regularly back up the CA and related configuration data to quickly recover in case of a cryptographic compromise.

6. Insufficient Logging and Monitoring

- Mitigation:
 - M1051 - Network Intrusion Prevention: Implement robust logging and monitoring of all AD CS activities, including certificate issuance, CRL updates, and CA configuration changes. Use a Security Information and Event Management (SIEM) system to detect and alert on suspicious activities.
 - M1018 - Application Developer Guidance: Ensure that all applications that interact with AD CS are configured to log relevant events and are integrated with the organization's monitoring systems.

7. Complexity and Misconfiguration

- Mitigation:
 - M1037 - Filter Network Traffic: Apply strict access controls and limit the scope of AD CS to reduce complexity. Regularly review configurations and templates for adherence to best practices.
 - M1022 - Restrict File and Directory Permissions: Limit who can modify AD CS configuration files and settings to prevent unauthorized changes that could introduce vulnerabilities.

8. Lack of Regular Updates

- Mitigation:
 - M1030 - Patch Management: Regularly update the Windows Server and AD CS to the latest supported version to ensure that known vulnerabilities are patched. Consider upgrading to a newer version of Windows Server if possible.
 - M1029 - Remote Data Storage: Use centralized patch management tools to ensure all systems running AD CS are kept up to date.

9. Trust Issues Across Domains or Forests

- Mitigation:
 - M1014 - Application Isolation and Sandboxing: Isolate and limit the trust relationships between domains and forests. Implement strict policies for cross-domain trust and regularly audit trust relationships.
 - M1044 - Network Segmentation: Segment networks to limit the impact of potential certificate misuse across different