Software Asset Management (SWAM) Illustrative Process

Introduction

The *Software Assessment Management* (SWAM) capability provides an organization visibility into the software installed and operating on their network(s) so they can appropriately manage authorized software and remove unauthorized software. This capability is dependent on the existence of a set of device roles defined for the D/A and an authorized hardware inventory as developed for the Hardware Asset Management Capability.

Device roles and software profiles are used to allow authorized variations of installed software products to exist on different devices without being labeled as defects. *Management of Assets Device Role Definition: An Illustrative Process* identifies how a D/A can initially create and then further refine a set of device roles for the organization. Every time a new device role is created, it must be associated with exactly one software profile. If an appropriate software profile does not exist, then one should be created. Because of this relationship between device roles and software profiles, the development or refinement of one often results in the development or refinement in the other. This document will not reiterate activities already described in *Management of Assets Device Role Definition: An Illustrative Process*. This document will identify when activities for SWAM would either lead to, or result from, the device role definition and refinement process.

Proper management of software assets begins with lists of authorized (whitelist) and prohibited (blacklist) software products and executables. Authorized products are allowed or permitted, but not required. Some lists, like organizationally prohibited software products may be defined globally, while others like authorized or mandatory software products are more likely to be defined per device role. The D/A actively monitors the network for software installed or executing on devices, by hash-type signatures and other residual information in the system such as registry entries, filenames, or running processes. This is compared against the software profiles associated with that device (based on its role assignments). If a piece of software is not authorized for use, the D/A will mitigate the risk by either removing it or properly authorizing the software for use. If the software is explicitly blacklisted, then the D/A will mitigate the risk by removing the software.

Special software –anti-malware or antivirus – is installed on devices to detect (and if configured to do so, protect against) known malicious and other unauthorized software (as defined by the D/A) from being downloaded, existing on the device, or executing. SWAM also ensures that lists or signatures used by this special software are kept up to date in accordance with the D/A's policy. These lists are referred to as "known-bad blacklists" for the purposes of this document.

Additionally, any discovered software that is not explicitly listed in the D/A's whitelist or blacklist will be initially placed into a graylist. Software included in the D/A's graylist will either be treated-as-authorized and be permitted to remain installed on the device until authorization is determined, or treated-as-unauthorized and uninstalled until authorization is determined. Items on the graylist must be moved to

either a whitelist or a blacklist within some time frame, ensuring that the D/A continues to improve upon this capability.

Prioritization of SWAM Implementations

All of the CDM capabilities can, and most likely will, be implemented in the D/A environment based on some prioritization strategy. It is expected that D/As will deploy the SWAM capability in a phased approach using a D/A specific prioritization model or criteria listing that *might* include:

- Ease of Definition
  - It might be easier to define software profiles for groups of devices based on organization, location, device type, or technology.
- Mission Criticality
  - Groups of devices might be more important because of their importance to the mission and higher potential impact if compromised.
- Threat
  - Groups of devices might be more important because of the threats that are targeting them.

As such, the D/A should determine a prioritization strategy and apply the Quick Win actions to the initial set of in-scope devices and their roles. The D/A should continue to perform Quick Win actions for devices (and associated device roles) as they are brought in-scope for CDM based on that prioritization strategy. They should then advance the capability for those in-scope devices by performing the Short Term actions, and then later the Long-Term actions (adding and refining devices and device roles each subsequent phase until the total population is captured at the appropriate level of refinement). The D/A should create a strategy and implementation plan for how to balance bringing the next round of devices in-scope (and associated device roles and software profiles) with advancing the capability for the previously defined in-scope devices.

Data and Processes Required for the SWAM Capability

Quick win activities

Quick wins are intended to provide opportunities for organizations to collect and prepare information to establish the foundation of the *Software Asset Management* capability.

The following are considered Quick Win activities:

Desired State

Verify that Device Roles have been defined.

Ensure that an initial set of device roles have been established for the D/A as described in *Management of Assets Device Role Definition: An Illustrative Process*. Additionally, verify that the following activities have been completed*:*

- Each device in the D/A has been assigned to at least one Device Role
- Mutually exclusive Device Roles have been identified per D/A policy

Develop an organizational software whitelist.

Work with D/A operations personnel (or similar organization responsible for the management and administration of D/A software) to identify the overall population of software within the D/A. The D/A can use some method of the following techniques to build the organization white list:

- Configuration/Change Control Board (CCB) – The D/A's CCB likely maintains a listing of approved software based upon requests over some period of time. This listing should be used as one of the primary conduits to populate the organization whitelist.
- Software Depot – The D/A Software Depot may contain information pertaining to deployed instances of software throughout the D/A that also includes license and version information pertaining to the purchased and deployed software.
- Software Patch Management System – D/A personnel can inspect the software patch management log to determine the most up to date, deployed versions of software in the D/A
- Active/Passive discovery – The D/A may employ software asset management sensors to extract implemented software information on deployed hosts.

Each entry in the software listing should contain the following attributes:

- Data necessary to accurately identify the software product and compare to the actual state data collected
    - Vendor
    - Product
    - Version/Release/Patch level
    - Software identification tag (SWID): Software ID tags provide authoritative identifying and associated information for installed software or other licensable item
    - CPE: Common Platform Enumeration (CPE) is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name
- Authoritative listing of executable files associated with product – associated executable files that execute source code for a specific piece of software that enable successful usage of the software.
- Software Manager - individual responsible for the mitigation of defects and maintenance of the software.
- Expiration policy – timeframe before the software must be reauthorized to validate its usage in the D/A environment
- Software authorization status - Current state of a software product authorization. States can be 'pending', 'authorized, 'suspended', 'expired', or 'revoked'
- Date initially authorized – date the software was initially authorized
- Date last authorized – date the software was last authorized
- Date revoked – date the software authorization was revoked

Identify management responsibility.

It is important to identify management responsibility for software management functions for each authorized software product. Work with D/A operations personnel (or similar organization responsible

for the management and administration of D/A software) and the individuals assigned software management responsibilities to identify the additional software management roles and responsibilities. Due to the large number of possible defects related to the management of D/A software (and the instances of the software in a D/A environment), the D/A may choose to break out management responsibilities beyond just one software manager for a piece of software. For example, the D/A may choose to designate additional managers to handle the deployment of software patches and the management of software licenses.

Establish prohibited software (organization) blacklist.

Work with members of the Operations group (and D/A equivalent that manages software) to develop a listing of software products that are prohibited across the organization. This listing will include applications, operating systems, etc. that aren't necessarily "known-bad" software, but software deemed to put the D/A (and their subsequent resources and assets) at too great of risk to allow implementation. An example might include operating systems that are no longer supported by the vendor or peer to peer software.

Note: Given the functionality needs of legacy applications, the D/A may not be able to update and patch software vulnerabilities. The D/A should seek to identify all these instances and minimize their instantiations through assignments to only devices (via profiles/Device Roles) that require this software in order to provide key mission or business functionality in the environment.

Identify known-bad blacklists

Work with members of the information security group (or D/A equivalent) to identify the different lists of known-bad software (blacklists). Often times this list is managed via the technology vendor of the software that the D/A has deployed (such as anti-virus or -malware software). The D/A should collect the following data elements pertaining to each known-bad blacklist:

- Known-bad blacklist expiration policy – the minimum required frequency at which the D/A should ensure the known-bad blacklist has been updated to include the latest executables (or equivalent) from an authoritative source. If the requirement is to use an external service providers "list" by connecting to their service every time a check is run, then this must be identified as such. If the real-time connection is to an internally updated image of external providers "list", then the update frequency needs to be specified for that server.
- Known-bad blacklist name or unique identifier
- Version of the known-bad blacklist
- Location of known-bad blacklist – if managed externally, the Uniform Resource Locator (URL) where the known-bad blacklist pulls the latest updates based on the expiration policy.

Note: Known bad blacklists are quite large, very dynamic, and often maintained by an antivirus or antimalware vendor. It is not expected that the D/A will know what software is on the list, but that they will know what blacklist is to be used and how frequently it is to be updated.

Establish the enterprise software "graylist."

A "Quick Wins" actual state step seeks to identify all of the software installed in the D/A. The D/A should have established an initial white list and blacklists (organizational and known-bad) as a part of the previous desired state "quick win" steps. Any piece of software found during the actual state collection of deployed software that does not map to a listing in the enterprise white list or the blacklists should be added to the graylist for further investigation. This investigation will determine whether the software should be added to the D/A whitelist or organizational blacklist. The D/A must decide the

default authorization setting for software on the graylist (e.g., should it be treated as authorized or unauthorized during the investigation process?). The D/A graylist attributes will include:

- Software "name" for a piece of software
- Date/time a particular piece of software was first discovered
- Date/time a particular piece of software was last seen
- Graylist grace period – the permitted timeframe of investigation before the D/A must move any piece of software to either the D/A whitelist or organizational blacklist.

Note: While identification and management of the graylist exists as a desired state activity, the list itself and associated attributes will be derived from actual state collection.

Identify the set of software profiles for the D/A.

Work with D/A operations personnel, CCB (or similar organization responsible for the management and administration of D/A software) to create software profiles that will be associated with specific device roles. Each software profile will be derived in accordance with the device role mission or business objective, including only software needed to accomplish the associated job functions. Each software profile should contain the following attributes:

- Allowed from whitelist – software that the D/A permits to be installed on the device. This listing is a derivative of the organizational whitelist.
- Mandatory whitelist – the list of software that must be installed on the device in order to maintain an intended functionality in the scope of the mission or organization. This listing is also a derivative of the organizational whitelist.
- Prohibited software - the listing of software from the organization white list that is prohibited from being installed on the device. The organization blacklist will automatically be associated with this list.
- Treat graylist as authorized status – Determination for how newly discovered and graylisted software will be treated for the profile. Options include:
    - Use organization default
    - Treat as unauthorized
- Known-bad blacklist used - the list of known-bad blacklists (derived from the overall population of possible known-bad blacklists in the organization) that should be deployed as a part of the profile

Note: Certain profiles may require known-bad blacklist expiration policies more or less stringent from the policy set forth by the D/A. As such, the D/A should identify those instances (rather, the profiles) where that requirement exists and the expiration policy that will replace it.

Associate a software profile to each device role.

There will be <u>one</u> software profile assigned to each device role. Multiple device roles can be assigned to a single software profile. Once assigned, the software profile establishes a key portion of the desired state specification for many Software Asset Management (SWAM) defect checks (e.g., unauthorized software on device, mandatory software not installed on device).

Actual State

Identify the software installed on each device within the D/A.

The D/A will use some automated solution that might seek to pull information from the "add/remove programs" for Windows devices or a similar script that seeks to use Linux/Unix resources (i.e., RPM/JAR)

to list installed programs on those types of deployed devices. The quick wins SWAM actual state collection solution must have the ability to create "names" for the software discovered in the production environment such that it can be compared to the desired state specifications.

Additionally, the D/A should have the ability to determine (and historically maintain) information pertaining to the length of time that a piece of unauthorized software remained on a device. At a minimum the collected information should include:

- Date/time it was first discovered
- Date/time it was last seen

Inspect the known-bad blacklist.

The D/A should collect information pertaining to the known-bad blacklist that includes:

- Current deployed version numbers and/or dates of the last update
- Status of mechanism deployment – will be Boolean (deployed or not deployed)

Depending on the type of SWAM solution deployed, this information may be on device itself (if using a host-based solution) or the server that maintains the actual known-bad blacklist (for network based solutions).

Short-term activities

Short-term steps should be established to improve upon the information initially collected and to build sustainability for further data collection. All information captured in the Quick Wins steps should be updated using an established process that automatically updates on an interval deemed feasible by the D/A. The goal is to attain more timely and accurate data to further mature the capability and to provide stakeholders better knowledge on discovered risk conditions.

The following are short-term activities:

Desired State

Continue identification of device roles and software profiles in the D/A environment.

Based on the deployment prioritization criteria that the agency employed in the quick win processes, the D/A should continue the identification of device roles and the software profiles associated with each device role. In this phase, the D/A will likely focus on refining the device roles further to capture more mission and organization specific roles, expanding upon the base level, device type/technology specific roles identified during the quick wins portion.

The D/A will continue to collect the attributes associated with each of the steps identified above for all the devices roles and software profiles not identified during the quick wins phase until all device types and profiles have been identified.

Modify desired state specifications based on systemic defects.

Establish a process for reviewing SWAM capability results and typical mitigations being employed by the organization in response to those defects. For example, devices with multiple device roles might contain conflicting pieces of software based on the profiles associated with the device roles. Software that is allowed for one device role assigned to a device could be listed as prohibited for another device role assigned to the same device. While the total population of software associated with the multiple software profiles might be needed to accomplish a specific mission or business objective in the

organization for that device, it will still yield defects according to the conflicting software profile policies. As such, the D/A will have to decide how to handle these situations. Often times, the correct decisions might ultimately result in a new device role being created that seeks to solve the conflicting software profile issue. Additionally, the D/A might seek to solve the problem through the association of an alternate profile to the device role. The D/A needs to decide what the appropriate policy or action should be based on factors such as risk tolerance and legal obligations of the D/A.

Lastly, work with D/A operations personnel, CCB, or similar organization responsible for the management and administration of D/A software to identify instances where the combination of two roles on a device result in a more restrictive set of approved software (more restrictive profile). If these multiple roles are allowed to coexist, but increase risk of impact if the device is compromised, then a different singular profile may be created and applied to the device. In these cases, the organization needs to specify the:

- Software profile(s) replaced – the name or unique identifier of the software profiles that no longer apply as currently defined
- Software profile(s) used – the name of the software profile(s) to use instead for that device

In general, it is more efficient and less work to define a new device role in such a circumstance.

Continue assignment of device roles to each device in the D/A.

Based on the identification of new device roles and the update of existing device roles to address systemic defects, assign the new device roles to devices noted in the HWAM inventory. This process should be established to recur as much as feasible in the D/A's environment. While a single device's device role assignments change infrequently, it is reasonable that a large D/A will have frequent (maybe even daily) changes to devices or their authorization status across the organization.

Develop a process for updating D/A software lists (black, white, and graylists).

As new software is discovered or new software requests are made, the D/A should develop and refine the process for adding the software to the appropriate list dependent on the balancing of legitimate business or mission functionality needs versus the inherent risk associated with the implementation and deployment of software.

Additionally, the D/A should develop a process to manage the software on the D/A graylist as established in the previous quick wins phase. The D/A should gradually move software from the graylist to the whitelist or blacklist as deemed appropriate. The D/A should reduce the "graylist grace period" that defines the maximum amount of time that a piece of software can remain on the graylist before it should be moved to the D/A whitelist or blacklist. This reduces the risk of allowing unmanaged software to remain in the production environment for long periods of time.

Actual State

Improve software inventory detection capability.

Work with software management team (or D/A equivalent) to improve the capability to detect software changes in the production environment in a more efficient fashion. Initially, the D/A might develop a script to parse the add/remove programs (for Windows), or query a resource in Linux/Unix (i.e., RPM database) to extract a list of installed software. This detection capability should be matured a step further at this juncture to extract a listing of all executables on each device in the D/A (as an example) in order to map them to a specific piece of software (and the associated CPE or SWID).

Increase timeliness of the capture of changes in the actual state software inventory.

The D/A should work to improve the timeliness of capturing changes in the population of deployed software from a previously established "pull" method to more of an automated "push." Specifically, a portion of the enterprise should be able to provide notifications of software inventory changes (as they occur) without the D/A having to query or extract that information.

Long-term activities

Long-term activities should be established to implement D/A-wide automated processes such that updates to device roles, profiles, and authorized software are immediately replicated from the authoritative source and visible to or consumable by CDM.

The following are long-term activities for this capability:

Desired State

Implement a centralized system/mechanism for changes in the authoritative sources.

Implement a centralized system/mechanism that provides real time visibility into the changes in the authoritative sources. All required desired state data elements will need to be updated in CDM in real-time. The software needed to support D/A mission and business functionality in addition to software needed to address the evolving threat vectors imply that D/A software inventories will remain more dynamic than static in nature. For the capability to score risk in an optimal fashion, changes to the existing population of authorized software, device roles, and software profile attributes must be seamlessly updated to ensure an accurate comparison to actual state collection.

Continue to augment the software graylist management process.

The D/A should continue to work with the Operations personnel and the software manage group to refine the graylist management process in order to increase the ability to seamlessly handle the discovery of new instances of software and appropriately classify each piece of software in a near real-time manner.

Actual State

Implement a centralized system/mechanism for changes in the installed software.

Implement a centralized system/mechanism that provides near real time visibility into changes in the installed software. All instances of deployed software will need to be updated in CDM as they are updated in the environment. For example, every time a new piece of software is installed in the D/A environment, this information could be made available to CDM instead of having to be periodically collected ("pulled"). This provides a greater level of assurance that the D/A has deployed only authorized software in the production environment and that specific devices only contain the necessary software to carry out their specific mission or business objective in the D/A. Additionally, prohibiting the known-bad and organizationally identified prohibited software from being installed on D/A devices decreases potential attack vectors and pivot points for adversaries to exploit.

Identify Risk Conditions

The following conditions are examples of comparisons that can be completed by the *SWAM* capability to identify defects.

- **Software authorization is expired.** Compare the date that the software was last authorized to the current date. Using the software expiration policy, determine whether or not the software's authorization has expired. This defect will be reported for every instance of that software installed at the D/A.
- **The D/A's device role policy is violated.** Review the list of mutually exclusive device roles as defined in D/A desired state policy to the listing of device role assignments on deployed D/A devices. If a device has two device roles assigned that are identified as mutually exclusive, this defect is reported.
- **Known-bad blacklist is not deployed for device.** Verify that the proper configuration settings are applied such that the defined known-bad blacklist is installed or deployed on the device.
- **Known-bad blacklist is out of date.** Compare the date that the known-bad blacklist was last updated to the current date. Using the known-bad expiration policy, determine whether or not the known-bad blacklist has expired. If the known-bad blacklist uses versions instead of dates to determine which file is most current, compare the most recent version with the one deployed. If the known-bad blacklist is expired or out of date but still deployed, this defect is reported.
- **Device has unauthorized, but not blacklisted or graylist-treat as unauthorized, software installed.** Compile the desired state policies for all the software profiles mapped to all the device roles for each device in the D/A. Inspect the software installed on each device in the D/A to determine if only mandatory and allowed software is installed on the device. If software is installed on the device that is NOT listed as prohibited, NOT currently on the graylist and listed as treat as unauthorized, and NOT listed as allowed or mandatory, then unauthorized software has been installed.
- **Installed software has not been reported within a set timeframe for a device (Non-Reporting for software).** Non-Reporting defects may exist if information updates for installed software on a device have not been reported within a defined period of time.

Attack Type Descriptions

Attackers continually scan networks looking for unauthorized server software running on ports of a device that can be reached from the attacker's location. This could include web servers, database servers, email servers, chat servers, blog servers, and other type of server software.

Attackers stage client-side attacks using directed spear-phishing techniques and other social engineering approaches as well as indirect approaches. A common indirect approach is to insert malware on legitimate web servers or direct a user using e-mail, social media, or other approaches to browse to a malicious website that contains embedded malware. Alternatively, an attacker can coerce a user to download malicious content that will run an exploit when the application runs. In either case, when a user clicks on the infected webpage or opens the infected link malicious content is executed that exploits vulnerabilities in the client browser or helper applications used to display images, play movies, edit documents, or perform other functions. Often these vulnerabilities are only known to the attacker (i.e. zero-day exploit) and cannot be patched.

Attackers use many tactics to transport viruses, worms, and other types of malware to devices. That malware will then attempt to execute and compromise the host device and then to spread to other devices. Irrespective of how the malware gets on a device, antivirus and other types of malware detection tools can detect the malware and sometimes prevent it from executing. Application whitelisting technologies can prevent malware or other unauthorized software from installing.

Appendix A - Definitions

Term	Definition
Authorized Hardware Inventory	List of authorized hardware assets for an organization or subnet.
Authorized Software Inventory	Managed software whitelists and blacklists for the organization and each device role.
Blacklist	List of prohibited software.
Common Platform Enumeration (CPE)	Common Platform Enumeration (CPE) is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name.[1]
Device	IP-addressable asset on the network or a non-addressable component (e.g. removable media) able to interact with the D/A's data and resources.
Device Role	An enterprise-wide label for a business or mission function that is associated with D/A-defined information technology assets. The device role is intended allow the refinement of authorization policies related to hardware and software in a manner that ensures that they are appropriately addressing mission needs/risks.
Executable file	For CDM, a specific file in persistent memory that can be loaded into active memory and executed by the CPU.
Expiration Policy	The policy that defines the requirements for when an asset's authorization expires. Examples include a piece of software is authorized for use in the enterprise for 2 years or device role authorizations are good for 3 years.
Graylist	List of software not authorized, but not explicitly prohibited.
Known Bad Blacklist	List of executables that are known to be malicious, fraudulent, or harmful.
Prohibited Software Products Blacklist	List of software products prohibited by policy. This enables a D/A to limit the not only software products in their environment, but also what versions of a software product that are authorized for use.
Scoring	The process of calculating the risk points for a defect. Identified defects will be "scored" based on the amount of perceived risk they create. The CDM program will be providing a scoring system that is generic across D/As. Each D/A may adapt this with additional D/A specific information to better prioritize defects for action.
Software	For CDM, software includes firmware, basic input/output systems (BIOS), operating systems, applications, services, and malware such as rootkits, trojans, viruses, and worms.
Software Asset Management (SWAM) capability	The CDM capability that ensures unauthorized and/or unmanaged software is 1) identified, 2) authorized, and 3) assigned for management, or 4) removed before it can be exploited compromising confidentiality, integrity, and availability.
Software identification tag (SWID)	Software ID tags provide authoritative identifying information for installed software or other licensable item.[2]

[1] http://nvd.nist.gov/cpe.cfm
[2] ISO/IEC 19770-2: Software identification tag

Term	**Definition**
Software Authorization Status	Current state of a software product authorization. States can be 'pending', 'authorized, 'suspended', 'expired', or 'revoked'.
Software product	The level of abstraction by which software is typically licensed, listed in registries during installation, and executed by users. Software products are roughly equivalent to the software identified by the NIST Common Product Enumeration (CPE) codes, and also by the ISO SWIDs.
Software Profile	A listing of authorized (permitted), mandatory, and prohibited software products and other software policy (e.g., update frequency of known bad blacklists) requirements. Profiles are used in combination with device roles to manage software on individual devices in a manner that scales.
Whitelist	List of authorized software for a D/A or device.