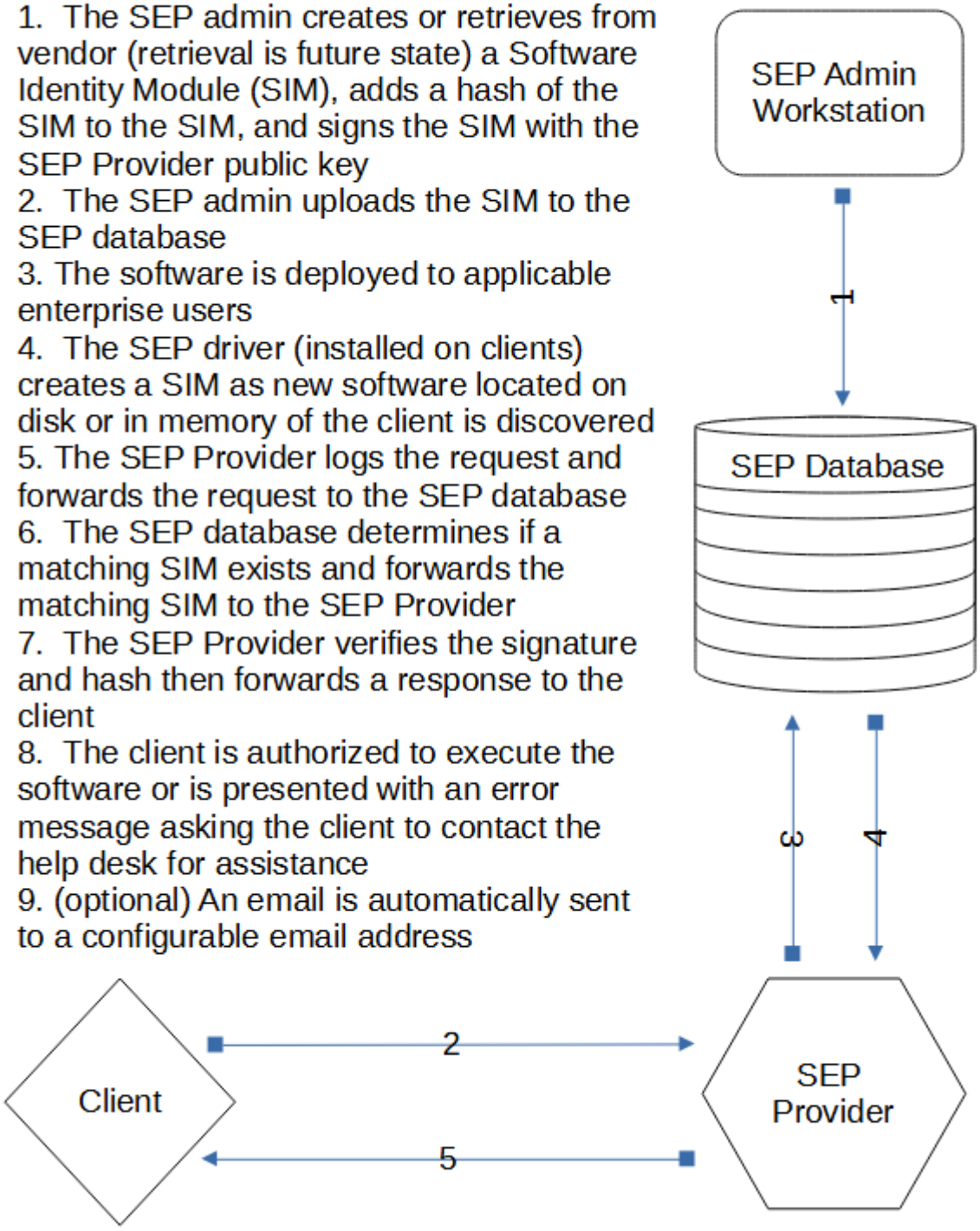
Software Eligibility Protocol

1. The SEP admin creates or retrieves from vendor (retrieval is future state) a Software Identity Module (SIM), adds a hash of the SIM to the SIM, and signs the SIM with the SEP Provider public key
2. The SEP admin uploads the SIM to the SEP database
3. The software is deployed to applicable enterprise users
4. The SEP driver (installed on clients) creates a SIM as new software located on disk or in memory of the client is discovered
5. The SEP Provider logs the request and forwards the request to the SEP database
6. The SEP database determines if a matching SIM exists and forwards the matching SIM to the SEP Provider
7. The SEP Provider verifies the signature and hash then forwards a response to the client
8. The client is authorized to execute the software or is presented with an error message asking the client to contact the help desk for assistance
9. (optional) An email is automatically sent to a configurable email address



Software Eligibility Protocol

Written by: Andrew Kosakowski

Table of Contents

Document Introduction

As of the time of publishing this document Software Eligibility Protocol (SEP) is a notional protocol that is being released for discussion and consideration for development. This protocol and related documentation is free for all uses, development, implementation, discussion, and/or dissemination as long as this document and this permission-set is attached to all derivative, non-derivative, and/or future works, discussions, development, solutions, implementations, dissemination, and/or all other uses. This document is a technical introduction into SEP. The overview is designed to provide a provide an introductory overview of SEP to permit a base layer of understanding that can be leveraged throughout the remainder of the document. Assumption are then provided to provide insights into what is considered to be within the responsibility of SEP and what is the responsibility of other solutions and implementations as well as other relevant assumption. The SEP provider is covered in detail as the first part of the technical portion of this SEP introduction as it is the anchor of the SEP protocol much like a DNS server is the anchor in the DNS protocol. Technical database, administrator, and client SEP integration follow to provide deeper insights. Once the technical details of SEP are fundamentally understood this document continues with ways to automate and scale portions of SEP. A document summary is provided to reiterate the key points that have been presented.

Overview of SEP

Software Eligibility Protocol (SEP) is a protocol designed to increase client system confidentiality, integrity, and availability by denying access to processor compute time by any software, script files, or plugins that has not been explicitly authorized to run on the endpoint. This has been accomplished by using the BEP (Better Encryption Protocol) architecture and adapting it to the specific needs of software authorization and integrity. When a patch or update is released, software is approved for installation, or a script file is needed an admin workstation creates a SIM file and saves it to the SEP database. After the SIM is saved to the SEP database endpoints are able to install or run the associated software after authorization is provided be the SEP Provider using a SIM verification process. Once authorization is verified by the SEP Provider a client-side database is updated and future authorizations are not required for the cryptographically identical SIM. The database is encrypted using the client's private key and only new and previously rejected software requires successful authorization. As part of the authorization process the SIM Provider creates a SIM Authorization log file that can be leveraged to determine is a single unauthorized software is spreading throughout the environment, if a single client is submitting a large number of failed authorization, or any other associated issues.

Assumptions

The assumption made when creating this document and SEP include:

- The SEP Database will not permit reads from any identities other than the SEP Provider
- The SEP Database will not permit any write actions from any identity other than the SEP administrator workstation
- SEP will not prevent DDOS, Brute Force, Certificate, or any other network or identity attacks
- Client workstations will protect identity specific private keys

SEP Provider

The SEP Provider is a security device that acts as the sole identity which holds the SEP Provider private key that is used to verify SIM signatures and authenticity. The SEP Provider acts as a proxy

between client identities and the SEP Database to remove the authentication of SIM verification away from the database or clients. In this way neither an intrusion of the SEP database or an intrusion of the or client will compromise the integrity of authorized SIM entries. This is accomplished by the SEP Provider creating a root certificate that is used to create client certificates and the SEP Provider public-private key pair. The root certificate key is available for a one-time export to a physically attacked storage device when it is created. This copy of the root certificate can be used for backup and recovery purposes as well as to scale the SEP Provider to meet demand.

Software Identity Module

The software identity module is the name used to identify a unique data-set that defines a particular software executable, script file, or module. The content of the three types of SIMs are described here.

Software SIM

The software SIM is used to uniquely identify software that is considered “installed” on a workstation and runs from file such as executables and browser plug-ins. The contents of this SIM are as follows:

- A Software SIM specific identification number
- The software name
- The Software version number
- The Software vendor
- The SHA512 hash of the file encrypted with the SEP Provider public key
- The MD5 hash of the file encrypted with the SEP Provider public key
- The date and time of SIM creation
- A SHA512 Hash of the concatenated above listed contents encrypted with the SEP Provider public key

Script File SIM

The script file SIM is used to uniquely identify files that have been saved with (well)known script file types. The contents of this SIM are as follows:

- A Script File SIM specific identification number
- The script name
- The script file version (if available)
- The script author (based on file metadata)
- The SHA512 hash of the file encrypted with the SEP Provider public key
- The MD5 hash of the file encrypted with the SEP Provider public key
- The date and time of SIM creation
- A SHA512 Hash of concatenating the identification number, the hashes, and the date of SIM creation encrypted with the SEP Provider public key

Module SIM

The module SIM is used to uniquely identify all task and auto-run software that is not mapped to a script file SIM or a software SIM. The scripts and “file-less” software that can be run from these locations present risks that do not always receive proper attention and care. The contents of these SIMs are as follows:

- A module SIM specific identification number
- The date and time the entry was created
- The location of the new entry
- The SHA512 hash of the contents encrypted with the SEP Provider public key
- The MD5 hash of the contents encrypted with the SEP Provider public key

- The date and time of SIM creation
- A SHA512 Hash of the concatenated above listed contents encrypted with the SEP Provider public key

SEP Private (signature) Key Management

The SEP Provider holds the private key for the asymmetric key-pair that is used to sign SIM entries. SEP Private keys are managed using the following usage and life-cycle.

1. The SEP Provider root certificate is used to create the public-private key pair and associated certificates.
2. The private key certificate is exported for back-up, recovery, and to scale the SEP Provider.
3. The private key is initialized for use in SIM authorization and authentication use.
4. After an enterprise specified time the root certificate is used to create a new private key.
5. The former private key and new public key are used to resign a sample set of authentic SIM entries in the SEP database.
6. The new private key verifies the correct application of the new signatures on the sample set of authentic SIM entries.
7. The former private key and new public key are used to resign the remainder of the authentic SIM entries in the SEP database.
8. The former public key is revoked and the private key is retired and the new key is initialized.
9. Six months after being retired the former key is destroyed.

Client Key Management

The SEP Provider provisions client certificates for each client identity that is authorized to request SIM verification from the SEP Provider. The client certificate management by the SEP Provider uses the following life-cycle:

1. The SEP Provider reads the SEP Provider specific group membership in the identity provider.
2. For each new certificate found in the identity provider the SEP Provider creates a new asymmetric key-pair and associated certificates.
3. The SEP Provider creates a one-time retrieval pass-code to retrieve the client private certificate.
4. The SEP Provider admin identity forwards the pass-code to the client identity (or retrieves the private certificate and saves it for backup and restoration purposes).
5. The client identity saves the private certificate to the client identities certificate store.
6. The client certificate is used to request SIM authentication and verification from the SEP Provider.
7. The client certificate is retired and destroyed when the client identity is removed from the identity provider’s SEP specific group(s), upon expiration, or when the SEP Provider admin identity manually revokes the certificate.

SIM Verification

When the SEP Provider receives a request for authentication and verification of a SIM entry the following steps are taken:

1. The SEP Provider authenticates the requester using the client identity public key.
2. The SEP Provider accepts the client SIM
3. The SEP Provider verifies the signature and hash on the client SIM with the client’s public key.
4. The client SIM type and hash is forwarded to the SEP database.
5. The SEP Provider accepts the SEP database SIM.
6. The SEP database SIM’s hash and signature is verified by the SEP Provider.

7. The SEP Provider compares the file size, software vendor, and version number as a complimentary verification mechanism.
8. The SEP Provider returns a verification and authorization response to the client:
    a) A positive response to authorize the use of the software; or
    b) A negative response with an error message to indicate:
        i. Hash not found in SEP database
        ii. SEP Provider signature not verified
        iii. Type 2 error indicating which check in step 7 above failed

SIM Verification Request Logging

Every client SIM verification request is logged by the SEP Provider to ensure anomalous activity can be identified and to ensure post-incident forensics and investigation activity is captured disparate to the management plane logging. The SIM verification request logging includes (but may not be limited to) the following data for each entry:

- Date and time of request
- Client request number
- Client SIM
- Date and time request was forwarded to the SEP database
- Response received from SEP database
- Date and time response was received from the SEP database
- disposition of hash, signature, and data checks
- response sent to client
- date and time response was sent to the client

SEP Database Integration

The SEP database is used to store SIM entries based on SIM type, receive verification and authorization requests, identify existing SIM entries for requests, and deliver the requests to the SEP Provider. The detailed flow is as follows:

Adding New SIMs to the SEP Database

1. The SEP database receives a SIM entry from a SEP administrator workstation
2. The SEP database verifies the legitimacy of the SIM by decrypting the SIM using the admin public key
3. The SIM type identifier is used to determine which table or storage space (software, script file, or module) to place the SIM entry
4. The SIM entry is saved/committed to the database

Responding to Verification and Authorization Requests

1. The SEP database receives a request from the SEP Provider
2. The request is verified by decrypting the SIM using the SEP Provider’s public key
3. The database uses the SIM identifier to locate which table/space to search for a matching SIM
4. The database uses the encrypted SHA512 entry for a first match
    a) If a match is found the flow proceeds to step 5
    b) If no match is found the database notifies the SEP Provider that no match was found
5. The uses the encrypted MD5 entry to verify a match
    a) If there is a match the flow proceeds to step 6

b) If there is not a match the database continues to search for a SHA512 match
6. The database encrypts the matching SIM with the SEP Provider's public key and transmits the SIM to the SEP Provider

SEP Administration Workstation(s) Integration

The SEP administration workstation(s) integration ensures only authorized identities can create SIMs on authorized endpoints. This is achieved using administrator workstation private key(s) and a software package. The administration workstation loads the script file, software, or module then uses the SIM creator to to select the SIM type and select the software based on location. Once the software is selected the software package creates the SIM, presents it to the admin for approval (if configured) then forwards the SIM to the SEP database.

SEP Client Integration

SEP clients have a software driver that sits at layer 0 and acts as a pre-processor proxy. When a software is placed into memory or saved to disk the following steps are taken:

1. The software driver performs a local SIM verification check by comparing the two encrypted hash values in the SIM against the encrypted hash values of the software being inspected.
2. The local SIM store is verified by decrypting the local SIM store signature (has) with the SEP public key
   a) If the decrypted hash matches the flow proceeds to step 4
   b) If the decrypted hash does not match the local store proceed to step 4 and start rebuilding the store by sending requests to the SEP Provider
3. If there is no corresponding entry in the local SIM store a SIM verification and authorization request is sent to the SEP Provider.
4. Once the response is received from the SEP Provider the client
   a) if the software is not verified and authorized
      1. Deletes the software associated with the SIM (best choice)
      2. Permits the software associated with the SIM to persist
      3. Permits the software to run (not recommended)
   b) If the software is authorized the software is permitted to persist in the authorized state and run
5. The authenticated and authorized SIM is added to the local SIM store and a new hash is created and signed with the client private key

Automation and Scaling

There are a number of places that automation can be introduced into SEP to include the following:

1. The creation of an administration workstation SIM as part of the patching and update process
2. The creation and deployment of client private certificates and drivers as part of the provisioning process
3. The back-up of certificates
4. Horizontal scaling of the SEP Provider using a secure certificate server and a SEP Provider provisioning script
5. Exports of SEP Provider SIM authentication and authorization logs

Caveats

The known caveats of SEP include the following:

- If client private certificates are compromised local stores can be modified without authorization
- If client SIM stores are deleted continuously without authorization a DDOS condition can exists
- If the administration workstation is not limited to least privilege unauthorized SIM entries can be submitted
- A backup or clone of the database must be performed if the SEP Provider private key is going to be rotated to ensure recover-ability if the database becomes corrupted
- A SEP admin workstation should not be a SEP client

Summary

Software Eligibility Protocol (SEP) is a protocol developed by the author to open a discussion about advancing software authorization to clients to increase confidentiality, integrity, and availability in a way that can be scaled easily and quickly. With SEP the ability to manage, control, authorize, and authenticate software, script files, and modules through explicit authorization can be achieved. SEP is a not a replacement for legacy software management solutions, rather it is a starting place to consider new techniques that can advance practices. SEP is intended to be viewed as an advancement to software management but should not (at this time) be considered representative of a complete and independent solution.