SOP Application Software Identification

**Introduction**

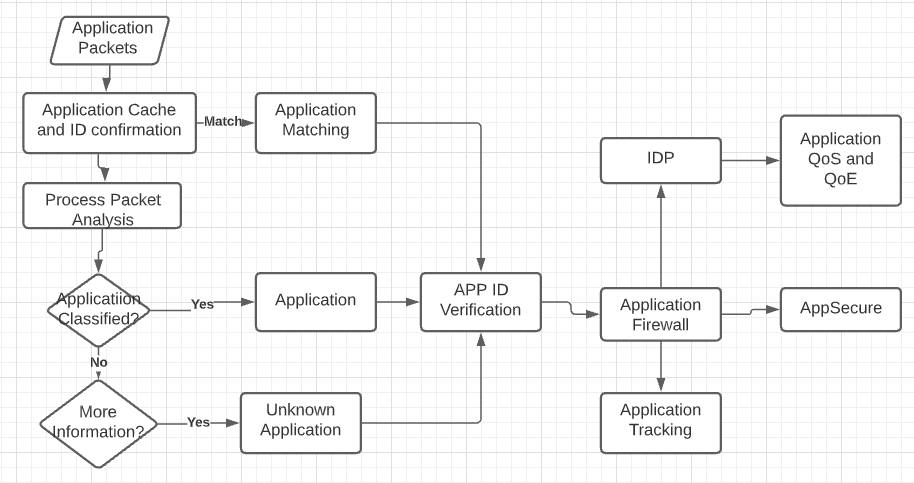
Standard Operating Procedure (SOP) is a step-by-step instruction to help perform a specific function. It adds more details on elements of a high-level process, workflow, and specific assignment that conforms with a company or organization's standards. In developing SOPs, one also needs to describe what happens for an outcome and provide steps and information detailing who, when, and where an event took place. All businesses require SOPs to help employees perform a routine job safely and consistently, complying with the industry's regulations.

**Software Identification SOP**

Application identification enables one to observe application behavioral characteristics and how they work. Different identification mechanisms such as software ID are used to detect the application in a network. Applications can be identified using their signatures, parsing information, and information from session management (“Application Identification.”, 3). Application signature mapping uses traffic on the network to identify an application. A set of data used in the application identification is shown in the table below.

|  |  |
| --- | --- |
| Property | Description |
| Product Name | Windows installer Package |
| Product Version | Windows Installer package version 6.0.5.7 |
| Software ID | Windows installer 98894\_895-WI3566 |
| Software Creator | Microsoft Corporation |
| Software Creator RegID | Unknown |
| Software Licensor | Freeverse Inc |
| Software Licensor RegID | Unknown |
| Tag Creator | Skype Communication |
| Tag Creator RegID | RehID2019.SDF.skypecommunication, Admin |
| Generate Software Tag files | Unknown |

The SOP for Application Software Identification is shown below.



The application packet, which contains application information as listed in the table, are passed to the application identification system, where the information is processed. The data in the application cache; majorly the application ID is used in monitoring the software whose ID matches the information in the identifier database.

The identifier will not operate with a limited dataset. In this situation, the dataset from the application cache will be held and the application will not be identified until additional data is provided. If the application is unknown, the identification process will search for more data such as application ID for further analysis.

The identification system matches the application and configures AppSecure for monitoring and controlling the application. The AppSecure also provides detection and prevention of traffic, prioritization of tasks performed by the application.

The application tracking toolkit tracks all applications identified passing through various devices and can be used in reporting any malicious activity or mal-functionality of the software. To enhance session management, the application identifier applies advanced policy-based routing. The routing protocol classifies application sessions and applies to route protocols to reroute application traffic.

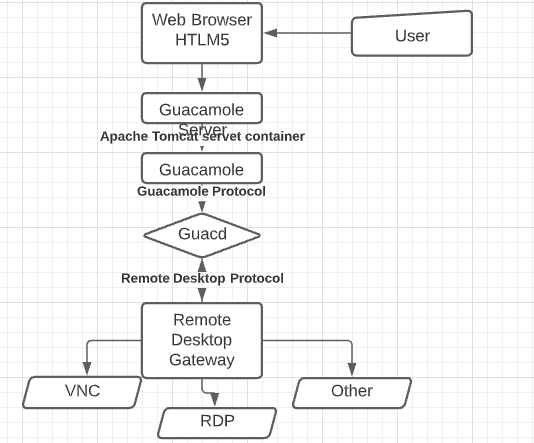
The Instruction Detection and Prevention toolkit is used to track and implement appropriate actions against any attack on the application. The application identification performs intrusion detection and prevention without decoders.

The application identification process also aims at protecting devices by preventing the infiltration of malicious applications using application firewalls. The application firewall is implemented using application-based protocols.

The application identification process also improves application quality of service and experience through application quality and quality of experience toolkits, respectively. The quality of the service toolkit improves application prioritization while the quality of experience monitors the application performance.

One can use predefined application signature identification tools on the verified Juniper Network Security website. The website will first update protocol bundles when a new application is added for identification.

**Guacamole Apache Protocol**



Users of the system are linked to the cloud services via the Guacamole server. They initially access the service using a web application. They are connected back to the server implemented by HTTP based on Guacamole communication rules. The web application interprets the Guacamole communication rules and forwards the information to the Guacd. Guard is a proxy that interprets the protocol content and creates a removed desktop for the user.

**Web Application**

The application does not implement any remote desktop protocol and does not contain any protocol that supports VNC or RDP or any other protocol defined in the Guacamole stack. It only understands the Guacamole protocol (Chapter&nbsp;1, 5). It receives the protocol and forwards it to the Guard. The web application converges the data from clients used to the Guard to determine the remote desktop allocated to users.

**The Guacamole Protocol**

The protocol is involved in writing a middle layer protocol between it and the remote desktop protocol. It is only involved in translating the content to the remote desktop. It only acts as a remote display and interaction protocol. The protocol operates on its own after the Guacd processed and loaded the client plugins. At this stage, it will have full communication regulation function between the web application and itself. It further maintains the connection until the operation is terminated.

**Guacd**

The Guard interprets the data from the web application and connects users to remote desktops based on the data provided from the Guacamole protocol. It processes a specific remote protocol to the point that Guacamole protocol can be loaded and the argument that must be passed to the protocol.

**VNC Client**

The client is purely JavaScript and translates the VNC to an XML version of its content. Additionally, its function has been limited to forwarding a single connection to a group of users. A remote desktop gateway is faster than presenting multiple remote desktop protocols, including VNC.

Works Cited

“Application Identification.” Application Identification | Application Security User Guide for Security Devices | Juniper Networks TechLibrary, www.juniper.net/documentation/us/en/software/junos/application-identification/topics/topic-map/security-application-identification-overview.html.

Chapter&nbsp;1.&nbsp;Implementation and Architecture, guacamole.apache.org/doc/gug/guacamole-architecture.html.