Toward Automated Threat Detection and Actuation

Takeshi Takahashi, Yu Tsuda, Koei Suzuki, Yaichiro Takagi, and Daisuke Inoue
*National Institute of Information and Communications Technology, Tokyo, Japan
E-mail: takeshi_takahashi@ieee.org

***Abstract*—Various cyber attacks are being generated via the Internet and are becoming a major problem for society. In particular, advanced persistent attacks, a type of cyber attack targeting a specific organization, are a huge threat. We have developed NIRVANA KAI, an integrated analysis platform that monitors, analyzes, and visualizes an organization's internal live network traffic in real time. It aggregates traffic flowing through live networks and alerts given out by security appliances installed inside the organization and visualizes them. This paper introduces an overview of NIRVANA KAI and discusses several interoperability issues that we encountered during its development.**

I. Overview of NIRVANA KAI

NIRVANA KAI helps system administrators to discover security incidents rapidly by analyzing and visualizing network traffic and alert information from security appliances installed within organizations. Figure 1 shows the visualization of network conditions using NIRVANA KAI. This shows the entirety of the IPv4 Internet space (/0 network), and each of the panels listing values from 0 to 255 corresponds to a Class A address block. The trigonal pyramid objects that jump between the panels show the sending and receipt of the packets in that space. The flower-shaped objects at the top of the address block panel visualize the alerts aggregated by the different types of security appliance. The individual parts that look like flower petals correspond with a security appliance, and this means that an irregularity has been detected within that address block.

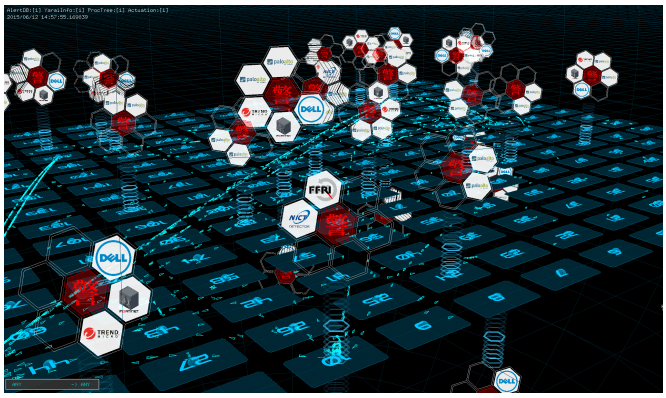

Fig. 1. Visualizing traffic and alerts in NIRVANA KAI

NIRVANA KAI allows us to drill down from the entirety of the Internet space to the Class A address (/8 network), the Class B address (/16 network), and the Class C address, to small address blocks with an alert source, finally arriving at the interior of one host (/32 network), as shown in Figure 2. It also has components that operate at the backend and execute actions such as the aggregation and analysis of information needed for visualization, i.e. network traffic aggregator, security alert aggregator, host information aggregator, event analysis platform, and actuator. Each of the components is implemented so that they operate independently.

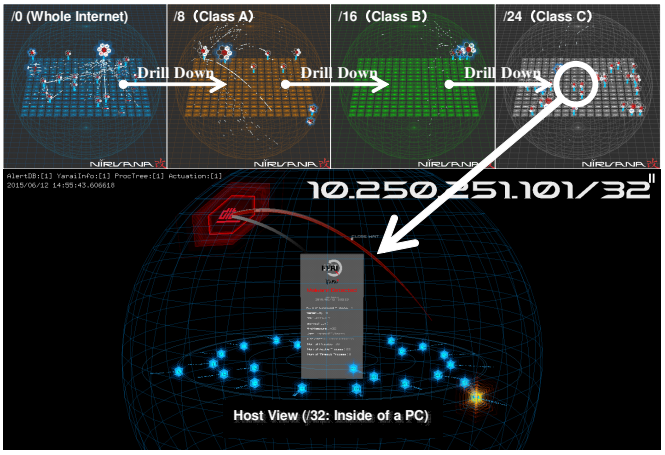

Fig. 2. Drill down from network view to host view

A. Security Alert Aggregator

NIRVANA KAI aggregates alert information, which includes detected attack information from devices introduced within an organization and detected malware information from endpoint-security software. The security alert aggregator in NIRVANA KAI aggregates syslog messages, which are commonly used when the log message of a device is shared.

A log message via syslog differs in terms of format for each device, so the received syslog message must be shaped so that NIRVANA KAI can handle it. First, all messages sent by syslog are received by the aggregator. The aggregator classifies log messages based on the IP address of the sender of the syslog transmission and then shapes its log message. During this shaping process, the aggregator extracts the sender/recipient IP address, port number, alert contents, severity, occurrence time, and so on using regular expressions. Next, the extracted data is stored in a database (alert database). The individual pieces of data accumulated in the alert database are displayed on the visualization user interface as individual petals of flower-shaped objects.

B. Host Information Aggregator

In addition to alert information, NIRVANA KAI also aggregates information from within hosts. To aggregate the different types of information from the hosts, an agent tool that cooperates with endpoint-security software needs to be introduced in advance. When malware is detected by the endpoint-security software, this agent tool notifies the host information aggregator of processes that have been judged as malware by the endpoint-security software together with a detection reason. Additionally, at fixed intervals, the agent tool sends basic information (e.g. OS type/version, user information, MAC address, etc.), running processes information and transmission generated by processes to the aggregator. Following this, malware detection information is sent by syslog to the security alert aggregator, and other information is stored in a database (host information database).

NIRVANA KAI constructs process trees based on process ID (PID) and parent PID from the process information accumulated in the host information database. The basic information and process trees of the host that are accumulated in the host information database are visualized in its Host View. Further, the establishment of the TCP session against an external network from malware can also be visualized.

C. Event Analysis Platform

As the scope of network/security devices and executing PCs in organizations grows larger, alert information is predicted to increase, and it is possible that important alert information will be overlooked. Accordingly, NIRVANA KAI analyzes the aggregated alert information and is equipped with an event analysis platform that extracts the important events that need to be dealt with by a security operator. The platform has a mechanism that allows for the easy implementation of an analysis engine that analyzes the information accumulated in the alert database, and the security operator is able to independently add in an analysis engine. We have defined a Domain-Specific Language to allow writing to an analysis engine.

The event information extracted from the analysis engine is stored in a database (event database). On the visualization user interface, event information is represented in the shape of an effect on the flower-shaped object that is the alert information.

D. Actuator

NIRVANA KAI can take various actuations (i.e. sending defense commands into several devices) against events obtained from analysis results. For example, when a large-scale infection of malware has been recognized inside a given network, a defense policy is preset to send a defense command to the firewall.

Figure 3 shows an overview of the actuator that fulfills the role of defense in NIRVANA KAI. Taking the event information obtained from the analysis engine described above, it applies the preset actuation policy to extract candidates for actuations. The policy can be written in the desired programming language, and the extracted candidates for measures and the corresponding defense commands are set and stored in a database. Network administrators select items to which actuations should be applied via a defense determinant user interface. Having done so, an actuation engine generates defense commands and send them to each of the network appliances that need to implement defensive measures. The actuation module can be added when appropriate, and the defense commands can be written in the desired programming language. The commands are communicated through interfaces such as NETCONF and REST API. Please note that when defensive measures are being implemented, this is visualized on NIRVANA KAI.

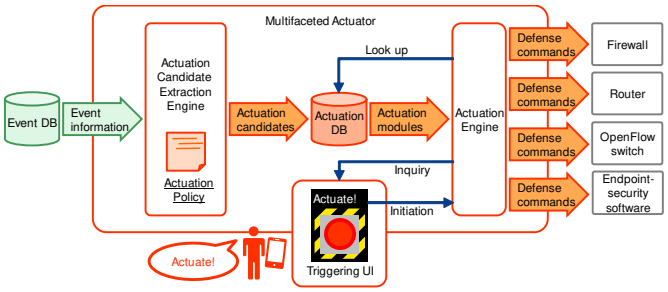

Fig. 3. Overview of the Actuation System

II. Boosting Security Operations Automation

To achieve more effective and efficient threat detection and actuation, the following issues need to be addressed.

Security alerts: Many security appliances provide security alerts in their own formats. Although most of them provide such alerts using syslog messages, their formats are not consistent. There are already common formats, such as Common Event Format (CEF) and Log Event Extended Format (LEEF), and they are already widely used by many appliances. However, there are still many appliances that do not support those formats and use their own formats. Meanwhile, the common formats are flexible and allow security appliances to describe data using their own styles. Moreover, there are a few security appliances that sometimes provides alerts that violate the data structures defined by the common formats.

Actuation instructions: Actuation instructions are communicated with network appliances through the interfaces provided by each network appliance, such as NETCONF and REST APIs. These REST APIs differ among vendors and appliances. Even if NETCONF is supported, its supported data models differ among vendors, so it still allows room for custom descriptions among appliances. There are even appliances without useful network APIs, so NIRVANA KAI needs to access them through SSH connections.

III. Summary

Several interoperability issues stand against the automation of security analysis and the other security operations. We believe that better interoperability expedites further developments of security operation automation techniques.