
OSSIM Fast Guide

February 8, 2004

Julio Casal
<jcasal@ossim.net>
<http://www.ossim.net>

WHAT IS OSSIM? In three phrases:

- VERIFICATION may be OSSIM's most valuable contribution at this time. Using its correlation engine, OSSIM screens out a large percentage of false positives.

- The second advantage is that of INTEGRATION, we have a series of security tools that enable us to perform a range of tasks from auditing, pattern matching and anomaly detection to forensic analysis in one single platform. We take responsibility for testing the stability of these programs and providing patches for them to work together.

- The third is RISK ASSESSMENT, OSSIM offers high level state indicators that allow us to guide inspection and measure the security situation of our network.

*** DISTRIBUTION**

- OSSIM integrates a number of powerful open source security tools in a single distribution. These include:

- Snort
- Nessus
- Ntop
- Snortcenter
- Acid
- Riskmeter
- Spade
- RRD
- Nmap, P0f, Arpwatch, etc..

- These tools are linked together in OSSIM's console giving the user a single, integrated navigation environment.

*** ARCHITECTURE**

- OSSIM is organized into 3 layers:

- . Sensors
- . Servers
- . Console

- The database is independent of these layers and could be considered to be a fourth layer.

Sensors

- Sensors integrate powerful software in order to provide three capabilities:
 - . IDS
 - . Anomaly Detection
 - . Real time Monitoring
- Sensors can also perform other functions including traffic consolidation on a segment and event normalization.
- Sensors communicate and receive orders from the server using a proprietary protocol.

Server

- OSSIM's server has the following capabilities:
 - . Correlation
 - . Prioritization
 - . Online inventory
 - . Risk assessment
 - . Normalization

Console

The console's interface is structured hierarchically with the following functions:

- . Control Panel
- . Risk and usage monitors
- . Forensic console
- . The Configuration Framework

Databases

- OSSIM uses an open interface that gives it the ability to communicate with any SQL database
- The distribution uses Postgress or Mysql as its open source relational database

Communication Protocol

- The protocol used for communication between server and sensors is proprietary and utilizes a TCP port.
- With this protocol, the user can activate, configure, request and receive data from the sensors.
- This protocol includes all of the functionality of IDMEF, it even adds an option for on-demand queries from the correlation engine to the monitors, sometime in the future it will be IDMEF compliant.

*** NORMALIZATION & AGGREGATION**

- OSSIM is currently able to read alerts from:
 - . Snort

- . Real Secure
- . Spade
- . Any data from NTOP
- . Firewall-1
- . Iptables
- . Apache
- . IIS
- . Cisco Routers
- . Unix Servers

- Normalization is performed with a configurable parser using XML files. Adding new agents can be accomplished in a matter of hours.

- Information from each detector is normally sent to the nearest sensor using native delivery capacity. OSSIM allows delivery using the following methods:

- . Snmp
- . Syslog
- . Rawsockets
- . SQL
- . OPSEC

- Aggregation between sensor and server is executed using OSSIM's proprietary protocol.

- If encrypted communications and authentication is required, it can be established using tunnels at the application level, typically ssh or ssl.

*** FUNCTIONALITY**IDS

- OSSIM includes snort, although it is capable of receiving and saving alerts from other IDSes

- Snort is configured and parametered for maximum performance, we also include a number of our own alerts, especially ATTACK-RESPONSE alerts, since they allow OSSIM to verify attacks, which is one of its main objectives

Anomaly Detection

- OSSIM includes three types of anomaly detection:

- Connections that are anomalous in origin or destination (ex: abnormal connection to an open port)
- Use data that is anomalous in relation to a threshold (ex: more than 100k throughput by host H)
- Anomalies in data with periodic tendencies learned using the holt-winters forecasting algorithm (as it learns the algorithm adjusts thresholds, for example high traffic during office hours and low traffic on nights and weekends)

- Anomalies can be correlated to identify malicious use or behavior (ex: a worm that sends 300% more traffic, makes 400% more connections, and a number of abnormal connections to machines and ports).

- Abnormal behavior can in turn be correlated with pattern alerts, thereby providing much superior detection and verification.

Correlation

- OSSIM has a powerful correlation engine that can:

1. Correlate an alert according to the version of the affected product and operating system. (If the attack affects an IIS-Windows machine, it is discarded if the target is Apache-Linux)
2. Correlate snort with nessus (if there is a possible buffer-overflow and nessus determines that we are vulnerable, the alert is prioritized)
3. Define logical directives for sequences of events that can correlate:
 - a. alerts
 - b. anomalies
 - c. states by queries to monitors

- To achieve the above functionalities OSSIM also employs the following processes:

- Online inventory Maintenance
- Maintenance of alert-version and alert-vulnerability relationship tables
- Real time monitor querying

- As a result correlation becomes a powerful tool for VERIFICATION, and identifies a high percentage of false positives generated by an IDS. Let's take a look at some examples of the correlation process:

- . Verification of an intrusion attack:
 - . Wait for attack responses
 - . Verify the existence of persistent sessions
 - . Verify connect-back
 - . Verify anomalous behavior of the target following an attack
- . Verification of a denial of service attack:
 - . Verify that service is down (using monitors)
- . Verification of web attacks:
 - . Verify negative response from web server
 - . Verify positive response
 - . Verify error response
- . Etc

(For a more in-depth look at this subject a paper specifically on correlation will soon be published.)

Prioritization

- Prioritization provides the system with the following information:

- What is important for security (assessment of assets)
- Which origins we should be worried about
- Which destinations we should be worried about

- The relationships among these data are laid out in a policy, similar to that of a firewall, in which we can configure, for example:

- If the attack originated from the Internet and goes to the internal network, it should be prioritized
- If the attack is carried out against a printer it should be deprioritized
- If the alert is a known configuration error for the network it should be deprioritized
- Etc.

- The policy allows us to define objects, groups of objects, ranges of directions, etc.

Inventory

- OSSIM can automatically and instantaneously inventory the following network information:

- Operating system
- Mac address
- netbios name, DNS
- Open services
- Products and versions of open services
- Various data on use (traffic/connections/time of day) found in the Usage Monitor's database

- This information is collected both passively and actively (listening and asking) using various specific programs.
- OSSIM can detect changes in any of the above parameters and send anomaly alerts if configured to do so.

Forensic Console

- OSSIM utilizes an extension of Acid for its Forensic Console, this console allows us to exploit the event database (EDB) collected through the process of normalization.
- Using Acid OSSIM allows us to store and exploit other events besides those of snort, as mentioned earlier (Firewall-1, Cisco, Apache, etc).
- The modification made to Acid enables it to store and search the following types of data that are not normally included:
 - . Accumulated risk to the host at the moment of an attack
 - . Instantaneous risk represented by an alert
 - . Value of the asset at the moment of the attack
 - . Reliability of the event assigned by the correlation engine
- OSSIM allows us to automate administrative tasks for cleanup and creation of histories in order to improve performance.

Risk Monitor

- OSSIM includes a monitor of "accumulated risk" called Riskmeter that utilizes a scoring algorithm called CALM (see "OSSIM: General Description").
- This monitor offers a real time indicator of the security situation of a host, a network, a group of machines, or even the global security situation. The indicator distinguishes between whether the machine may be compromised (or behaves like an attacker) or may be under attack.
- The monitor can graph this indicator over time, send alerts according to defined thresholds, and use them for correlation in logical directives.

Auditing

- OSSIM integrates Nessus for auditing.
- Using Nessus we can obtain a vulnerability index, i.e. the state of network vulnerability, which can be used as an objective or technical assessment of security.
- Vulnerabilities are stored and correlated to prioritize and discard attacks identified by the IDS.

Usage Monitor

- OSSIM includes NTOP, a monitor that collects all traffic data via passive listening and creates a use profile for each machine.
- This information is stored in circular databases that enable us to save detailed information for a long period of time, for example: bytes sent/received, bytes by service, throughput, connections made, time of day, etc.
- The monitor can make graphs of each item, send alerts according to defined thresholds, and use the data for correlation in logical directives.
- OSSIM links this information to security information in order to query them jointly.

Control Panel

- OSSIM integrates, summarizes, and links together all of the above tools in a single Control Panel.
- Its purpose is to enable the user to analyze and interrelate information from the most abstract to the most concrete.
- The control panel allows us to create reports with information cross-referenced from the various tools that make up OSSIM.
- This control panel should be modified and personalized for each organization.