Integrating of SIEM Solutions into Modern Industry Architectures

Evan Riddick

Index

1 Introduction

There are occurrences in the handling of network security implementation where the use of simple monitoring tools can hinder effectiveness. A network hosting public facing servers or sensitive databases can become vulnerable to well-constructed attacks without tools to detect them. While rsyslog has its strong points in log forwarding and storage, but misses key aspects of log analysis and monitoring system integrity.

This puts more of a workload on log analysis services like Splunk or Elasticsearch, which will receive all data from rsyslog unfiltered. It also puts more strain on indexes and causes analysis to take much longer, since logs are not pre-analyzed or filtered before being sent to Splunk. Considering the best use cases for security monitoring in a network architecture is essential to providing an optimal defense system against attackers. One should also consider the services/systems already set in place within the network, since these systems could either hinder or complement the SIEM solution to be implemented. Ensuring that the security information being provided is effectively collected and forwarded is important, however, making sure the information can be used by the log analysis service efficiently is crucial. Since most event management and responses will take place on a log analysis service like Splunk or Elasticsearch, logs should be presented in the simplest and straightforward way to ensure reaction time and techniques are optimal. Modern network security demands more than just the collection and forwarding of logs—it requires tools which are thoughtfully integrated and have a defined role in the detection and response process. This paper explores how optimizing log collection and workflow can significantly improve detection efficiency on SIEM platforms.

1.1 Problem

The Sicilian Biodiversity Observatory Project (ORBS) utilizes rsyslog and Splunk integration to establish a SIEM system.

Architecture

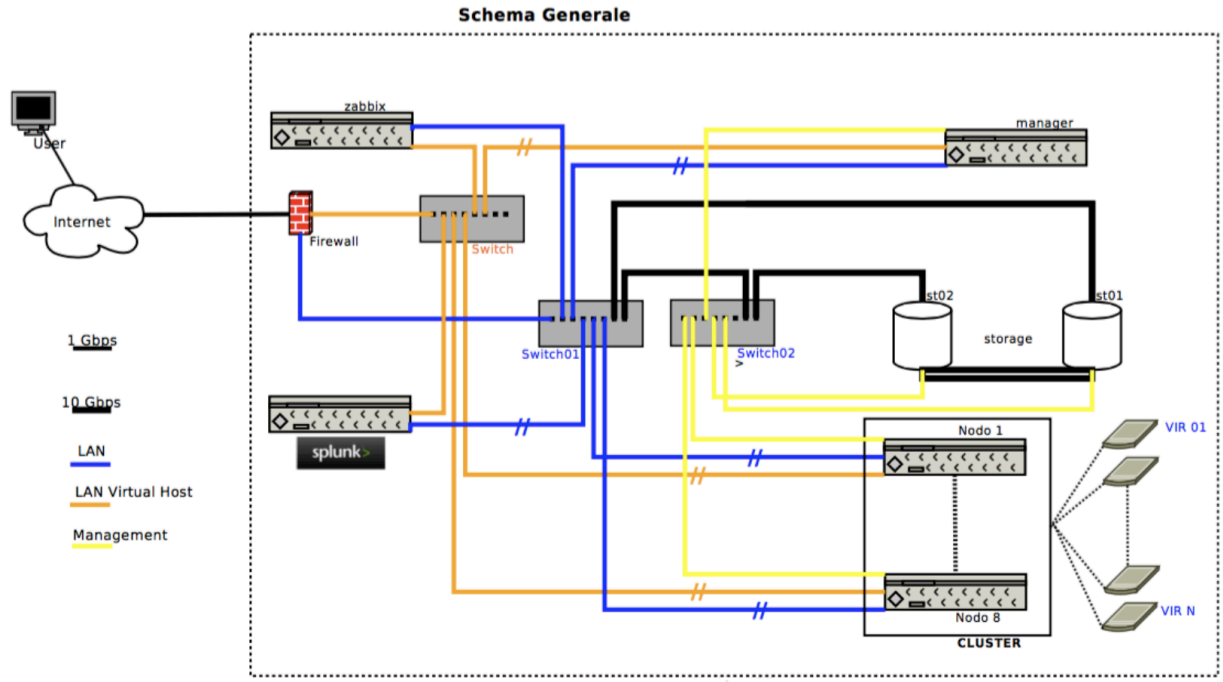

Figure 3 - The main IT infrastructure of ORBS

Zabbix is used to monitor network health and important files. **pfSense** is the centralized firewall service for all machines on the network. There are two **node clusters** which host the public facing servers/applications, and two storage servers.

The limitations of rsyslog are as follows:

- Limits amount of files/directories the user can monitor
- No built-in threat detection. Cannot analyze logs for threats.
- No centralized management options
- Lacks log tampering detection or audit trails. This means rsyslog can't effectively monitor files essential to system integrity.
- manual setup of TLS certificates can be error-prone across systems

Within the context of the ORBS project, the implementation of rsyslog and Splunk presents several issues. One question to ask is how will rsyslog ensure consistent secure connection to Splunk? While rsyslog is an adequate forwarder of logs to Splunk, it requires manual setup of Transport Layer Security (TLS). When relying upon rsyslog to monitor multiple systems within a network, TLS certificate authentication can become tricky and unreliable. Another issue is that Rsyslog will send all syslogs on the targeted system to the index. Doing so without filtering the data first can present challenges for Splunk users. It can make searching and reporting more difficult due to inconsistencies in log format, as well as make alert creation and threat detection more manual and tedious. If there are limits set on an index, the overflow of logs can pose a problem for the indexer. The most outstanding issue with this architecture is the logs which rsyslog cannot read. The service is unable to monitor command inputs for detection of log tampering, and provides no logs of who accessed or changed critical files. This leaves many files pertinent to system integrity exposed, with no way of monitoring them besides Zabbix. While Zabbix does factor some of these variables into system health checks, it is not expressly monitoring for potential attacks/threats.

1.2 Proposal

Instead of Rsyslog, the open-source security platform Wazuh could be used to better fit the network architecture and provide advanced security monitoring. Wazuh agents offer many more security monitoring tools than rsyslog while also providing threat detection, whereas Zabbix does not. This means Wazuh also has the option of monitoring files which Zabbix is already monitoring, securing it even further by proactively searching for threats to the file. Wazuh allows for the utilization of syslog as well as other security monitoring tools like vulnerability detection and osquery. To explain the advantages of Wazuh compared to Rsyslog, a project demonstration has been constructed to reflect the basic ORBS system architecture.

Three virtual machines are used to forward logs from a web server to Splunk via Wazuh. An alert within Splunk then triggers a script to update the system's firewalls remotely.

Why does Wazuh make more sense in this architecture? The network already has Zabbix which monitors health while providing custom scripts and alerts. Having Wazuh monitor for threats and then send them to Splunk offers a similar workflow, but this time handling threat detection.

2 Wazuh

The Wazuh platform offers many services, such as an indexer, dashboard, manager, and agent. For the purpose of monitoring logs and a machine, an agent must be installed onto the machine and given the ability to read these files. Wazuh's modules and built-in utility help securely monitor many types of files on a system, like json, syslog, windows event logs, application/web server logs, etc. This is due to the fact that Wazuh offers a very flexible way to parse a variety of logs, including custom ones. We will go more into detail about this later. For now, we will focus on how secure forwarding of logs from the agent to manager is completed through encrypted TCP communication. Here are the code fragments from both *ossec.conf* files:

Manager:

```
<ossec_config>
  <client>
    <server>
      <address>104.236.239.74</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>ubuntu, ubuntu24, ubuntu24.10</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>
```

Agent:

```
<remote>
  <connection>secure</connection>
  <port>1514</port>
  <protocol>tcp</protocol>
  <queue_size>131072</queue_size>
</remote>
```

2.1 Agent

The Wazuh Agent can log a variety of formats. The *ossec.conf* file for the agent details a lot of the modules and default tools used to effectively monitor the system.

Modules:

```
<wodle name="cis-cat">
  <disabled>yes</disabled>
  <timeout>1800</timeout>
  <interval>1d</interval>
  <scan-on-start>yes</scan-on-start>

  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<!-- Osquery integration -->
<wodle name="osquery">
  <disabled>no</disabled>
  <run_daemon>yes</run_daemon>
  <log_path>/var/log/osquery/osqueryd.results.log</log_path>
  <config_path>/etc/osquery/osquery.conf</config_path>
  <add_labels>yes</add_labels>
</wodle>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>
```

FIM:

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>

  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>

  <scan_on_start>yes</scan_on_start>

  <!-- Generate alert when new file detected -->
  <alert_new_files>yes</alert_new_files>

  <!-- Don't ignore files that change more than 'frequency' times -->
  <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>
```

Log Analysis:

```
<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+[[:digit:]]\+\ \+[[:digit:]]\+\ \+\(.*\):\([[:digit:]
  <alias>netstat listening ports</alias>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>last -n 20</command>
  <frequency>360</frequency>
</localfile>
```

Syslogs:

```
<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/root/juice-logs/logs/access.log</location>
  </localfile>

</ossec_config>
```

Wazuh Agent utilizes modules (wodles) such as **syscollector** and **osquery**, as well as log analysis and vulnerability detection, to monitor system health and security. The location is pointing to a mounted volume which is sourced from a docker container running OWASP Juice Shop, our faulty web server for testing purposes. The docker run command is shown as follows:

```
docker run -d -p 3000:3000  \
-v  /root/juice-logs/logs:/juice-shop/logs  \  --name  juice-shop  \
--user 0  bkimminich/juice-shop
```

The agent also offers audit logs of users who were accessing or changing log files. Searching: index="wazuh-alerts" sourcetype="wazuh-alerts" rule.groups{}"?" in Splunk, and replacing the “?” with the audit log you want to see, such as “adduser” or “config_changed”.

2.2 Manager

With the logs being collected by the agent, the Wazuh Manager is configured with rulesets which trigger alerts based on the log information sent by the agent. There are many rulesets ready to use upon installation, but custom rulesets and decoders can be created to suit the user's needs. Wazuh rule creation is flexible and efficient at custom parsing and analysis. **Decoders** are a way to parse log data for the purpose of generating alerts from them. **URL Match**es are specific to web server logs, and allow the user to specify what rule should trigger based on a url fragment. The **rule id**'s assigned to rules help Splunk understand the threat level as soon as it is sent. This is a custom ruleset I created to work in tandem with other rules -

0246-juice-shop_rules.xml:

```
<group name="juice-shop,access">
  <rule id="100100" level="5">
    <decoded_as>json</decoded_as>
    <description>Juice Shop - Potential unauthorized access</description>
    <match>401</match>
    <match>/rest/user/login</match>
    <group>authentication_failed,juice</group>
  </rule>

  <rule id="100101" level="10">
    <decoded_as>json</decoded_as>
    <description>Juice Shop - Suspicious path access</description>
    <match>/ftp</match>
    <group>web,suspicious,juice</group>
  </rule>

  <rule id="100102" level="3">
    <decoded_as>json</decoded_as>
    <description>Juice Shop - Successful login</description>
    <match>200</match>
    <match>/rest/user/login</match>
    <group>authentication_success,juice</group>
  </rule>

  <rule id="100103" level="10">
    <if_sid>100100,100101</if_sid>
    <url>=select%20|select+|insert%20|%20from%20|%20where%20|union%20|</url>
    <url>union+|where+|null,null|xp_cmdshell</url>
    <description>Juice Shop - SQL injection attempt</description>
    <mitre>
      <id>T1190</id>
    </mitre>
    <group>web,sqli_attempt,juice</group>
  </rule>

  <rule id="100200" level="7">
    <if_sid>31108</if_sid> <!-- override the ignored one -->
    <description>Juice Shop - Access to sensitive admin config endpoint</description>
    <match>/rest/admin/application-configuration</match>
    <group>web,juice,suspicious</group>
  </rule>

  <rule id="100201" level="5">
    <if_sid>31108</if_sid>
    <description>Juice Shop - Accessed application version endpoint</description>
    <match>/rest/admin/application-version</match>
    <group>web,juice,info</group>
  </rule>
</group>
```

While some rules within my custom ruleset are decoded as json, the bottom two rules (id=100200, 100201) use a decoder I borrowed from the web_rules.xml ruleset. These default decoders cover a wide range of log formats and exemplify Wazuh's effectiveness at threat detection.

Comparison of Rsyslog to Wazuh

Rsyslog:

- Insecure (unencrypted) log forwarding by default, and manual setup of TLS certificates can be error-prone across systems
- No auditing capabilities
- No log analysis or threat detection
- No log tampering monitoring
- Light-weight

Wazuh:

- Secure TCP log forwarding from agent to manager
- Auditing of access and changes to log files
- Provides custom decoders and alerts configuration for threat detection
- Log normalization and enrichment
- Medium-weight (Uses Splunk Forwarder)

3 Splunk

Splunk has been configured with SSL security using a certificate signed by Let's Encrypt.

$SPLUNK_HOME/etc/system/local/web.conf :

```
[expose:tlPackage-scimGroup]
methods = GET
pattern = /identity/provisioning/v1/scim/v2/Groups/*

[expose:tlPackage-scimGroups]
methods = GET
pattern = /identity/provisioning/v1/scim/v2/Groups

[expose:tlPackage-scimUser]
methods = GET,PUT,PATCH,DELETE
pattern = /identity/provisioning/v1/scim/v2/Users/*

[expose:tlPackage-scimUsers]
methods = GET
pattern = /identity/provisioning/v1/scim/v2/Users

[settings]
enableSplunkWebSSL = true
privKeyPath = ${KEY_PATH}
serverCert = ${CERT_PATH}

[expose:tlPackage-agent-management-all-endpoints]
methods = GET,POST,DELETE,PATCH
pattern = /agent-management

[expose:tlPackage-agent-management-all-endpoints-webport]
methods = GET,POST,DELETE,PATCH
pattern = /agent-management/**
```

A Splunk App was created to get full customization of alert actions. The app allows users to specify parameters to handle payload conversion and python execution.

$SPLUNK_HOME/apps/juicer/default/alert_actions.conf :

```
[restrict_access]
is_custom = 1
label = Block Attacker IP
description = Blocks IP using ufw-docker over SSH
payload_format = json
python.version = python3
```

Creating a custom app is also the only way to create custom alert actions, which can be included in any alert created through the Splunk UI.

3.1 Splunk Forwarder

Splunk Forwarder is software installed on external machines to monitor files/directories and forward to Splunk Index. In this case, it is paired with our Wazuh Manager to ensure secure log forwarding from the manager to Splunk. A ‘wazuh-alerts’ index was created to store all Wazuh related logs. The index stores the data locally on the Ubuntu-Splunk virtual machine.

Input / Output Configuration

Inputs.conf:

```
[monitor:///var/ossec/logs/alerts/alerts.json]
disabled = 0
host = ubuntu-wazuh
index = wazuh-alerts
sourcetype = wazuh-alerts
```

Outputs.conf:

```
defaultGroup = default-autolb-group
[tcpout:default-autolb-group]
server = 64.225.5.185:9997
[tcpout-server://64.225.5.185:9997]
```

4 Conclusion

Wazuh surpasses the capabilities of Rsyslog in several ways. It can set up alerts to a plethora of log paths, manage agents through a centralized manager, offer modules for specific integrations, offer built-in encrypted log forwarding and receiving, as well as lend more advanced security monitoring tools. This integration offers more efficiency for Splunk instances and easier management of large networks.

In layman's terms, the proposed solution presents…

- An automated SIEM system
- Significant upgrades to security monitoring of network nodes
- Customizable threat detection with Wazuh module integrations
- Support for tasks covered by Zabbix, while providing similar workflow
- Superior management of instances compared to only Rsyslog

The only disadvantages to this proposal is that an architecture would requires more resources than with Rsyslog, and Wazuh requires a mediator (Wazuh Manager) to facilitate log transfer. This means the logs must pass through the agent and manager before being sent to Splunk.