SDMAY22-38

## Members:

- **Alex Nicolellis**
- **Jung Ho Suh**
- **Muhamed Stilic**
- **Pallavi Santhosh**

# 4 Testing

**Testing is an extremely important component of most projects, whether it involves a circuit, a process, power system, or software.**

**The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.**

## 4.1 Unit Testing

**What units are being tested? How? Tools?**

The units we will test are the VM system applications we are using. The VMs are the Attacker, Main IADS, IADS Sensor, and Victim. We will test the Victim and Attacker using snort to check what alerts are being sent through. We will test the IADS sensor and Master  using security onion and elastic search to determine where all of the alerts are coming from..

## 4.2 Interface Testing

**What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?**

The interfaces in our design are SNORT, ElasticSearch, Kibana, SQuil, and Wireshark. They must work together to find and prevent anomalies so when we test the program we are essentially testing their composition. Tools for this include NMAP which we will go into further detail in later sections.

## 4.3 Integration Testing

**What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?**

Critical integration paths for our design are setting up our sensors and the machine learning algorithm. The reason they are so critical is because the sensors need to be created first before they can be tested. The algorithm will take the longest and will go into the next semester to be created. We will test all of these using SNORT, SecurityOnion, SQuil, and Wireshark.

## 4.4 System Testing

**Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?**

Our system level testing strategy includes is black box testing and human testing. This fills up all of the requirements needed because the whole system is acting like a black box. We don't know what attacks will actually be processed in the future. That way using black box testing will use all of our components to see how well they run. Tools that we will use are Nmap, Ping, datasploit.

## 4.5 Regression Testing

**How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?**

We are ensuring that any new additions do not break the old functionality by saving all prior work and testing at each new addition of the project. This way we will be able to pinpoint what additional work causes the issue and will easily be able to revert it by just that portion. Implemented critical features we have are GIT Lab because we can revert back to previous forms. Additionally, with snapshot we can revert back to a previous image.

## 4.6 Acceptance Testing

**How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?**

To demonstrate both non-functional and functional requirements are being met through testing, we will go through and check off all the requirements once we get our results. After we feel confident in this, we will give our client a demonstration and allow them to operate the system themselves. We plan to do this multiple times until the client is absolutely satisfied with the product in all aspects.

## 4.7 Security Testing

**How are you testing the system's security based on your specific requirements?**

A majority of our testing in the second semester will be focused on security testing as this is a cyber security project. In the second semester we plan to focus on developing the master control for all the sensors so in a way the performance testing will be security testing. We plan to use the red team method as the final step of this to ensure the security of our design.

Penetration testing: Scanning tools used to understand the target's resprusion. Then, access is attempted (see image below).

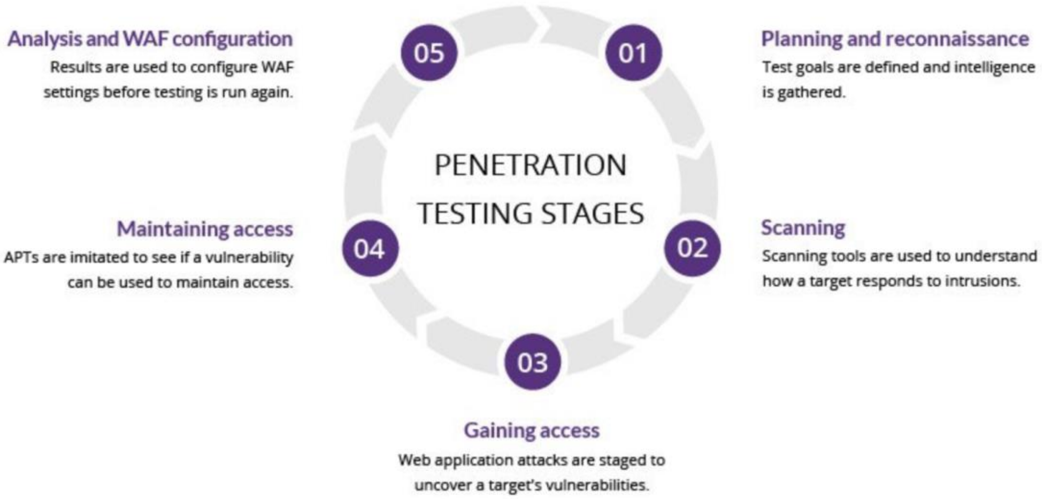
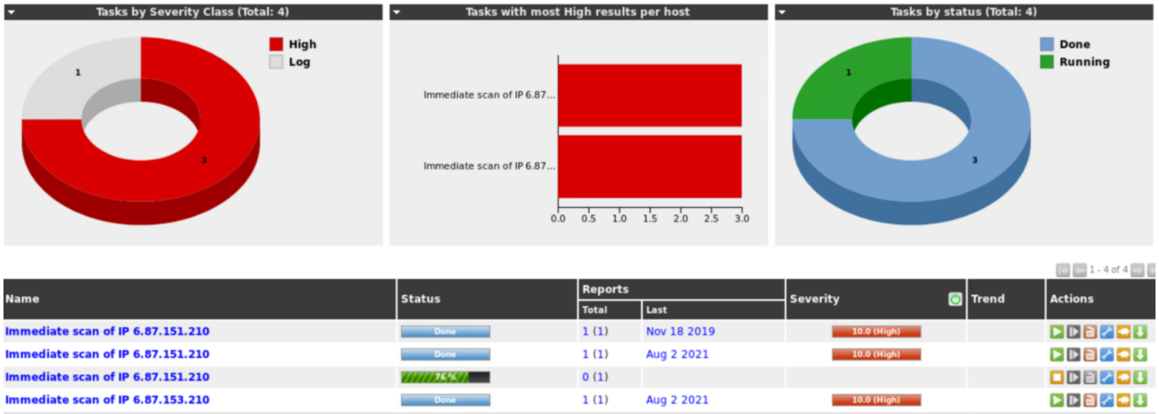

Analysis and WAF configuration
Results are used to configure WAF settings before testing is run again.
05
01
Planning and reconnaissance
Test goals are defined and intelligence is gathered.
PENETRATION TESTING STAGES
Maintaining access
APTs are imitated to see if a vulnerability can be used to maintain access.
04
02
Scanning
Scanning tools are used to understand how a target responds to intrusions.
03
Gaining access
Web application attacks are staged to uncover a target's vulnerabilities.

image from: imperva.com

## 4.8 Results

**What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.**

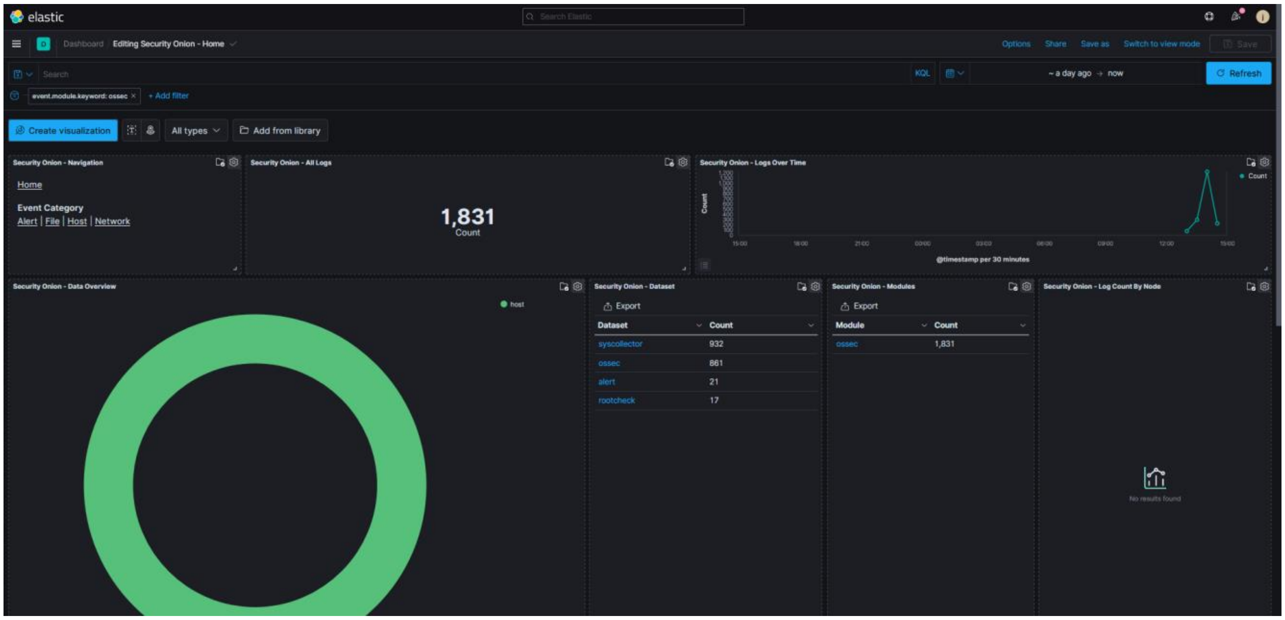
We do not yet have results to test because we have just begun implementing our design but we have done tests on the software we are using including one to test for vulnerabilities in our network using OpenVas which we have included below:


Tasks by Severity Class (Total: 4)
High
Log
1
3
Tasks with most High results per host
Immediate scan of IP 6.87...
Immediate scan of IP 6.87...
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Tasks by status (Total: 4)
Done
Running
1
3
1 - 4 of 4

| Name | Status | Reports | | Severity | Trend | Actions |
|---|---|---|---|---|---|---|
| | | Total | Last | | | |
| Immediate scan of IP 6.87.151.210 | Done | 1 (1) | Nov 18 2019 | 10.0 (High) | | |
| Immediate scan of IP 6.87.151.210 | Done | 1 (1) | Aug 2 2021 | 10.0 (High) | | |
| Immediate scan of IP 6.87.151.210 | 76 % | 0 (1) | | | | |
| Immediate scan of IP 6.87.153.210 | Done | 1 (1) | Aug 2 2021 | 10.0 (High) | | |

As can be seen above: we were successful in getting the software to work and run the programs we needed to.

Here is the screenshot of the Security Onion scanning the traffic of the sample network. It is using Kibana to visualize the traffic such as graphs.



event.module.keyword: ossec

1,831
Count

| Dataset | Count |
|---|---|
| syscollector | 932 |
| ossec | 861 |
| alert | 21 |
| rootcheck | 17 |

| Module | Count |
|---|---|
| ossec | 1,831 |