SIRS
Automatic Vulnerability Detector

T13

October 2019

(a) Inês Albano 87664 (b) Miguel Oliveira 87689 (c) Viviana Bernardo 87709

Figure 1: Group 13

1 Problem

The program in development stores and receives data from group members, which means there is a need for authentication on the behalf of each user. Only the leader of the group should be able to visualize the information stored in a data base relative to the vulnerabilities exploited and fingerprints submitted.

1.1 Requirements

- Confidentiality - sensitive data, such as the exploited vulnerabilities and the fingerprints. Guarantee that only the leader should be able to see the scoreboard and the exploits of each member.
- Integrity - to ensure that the contents should not change state, which means unauthorized people can not change it. For example the SQL databases can not be dropped or deleted. Ensure that the data exchanged between the machines is not changed.
- Authenticity - assurance the vulnerability and fingerprint was submitted by one person in particular, this involves proof of identity.
- Freshness - to avoid rediscovery of the same vulnerability.
- Availability - guarantee that backups are made so that the information is always available.

1.2 Trust assumptions

- There are no loss of data when a team member submits the vulnerabilities exploited and fingerprints.
- The scoring system is always up and running.
- The programs that the vulnerability detector is analysing does not fail.
- The certificates emitted by the authentication server are always trustworthy (self-signed certificates).
- The public key sent by the server (used in custom protocol) is always trustworthy and server cannot be spoofed.

2 Prepared Solution

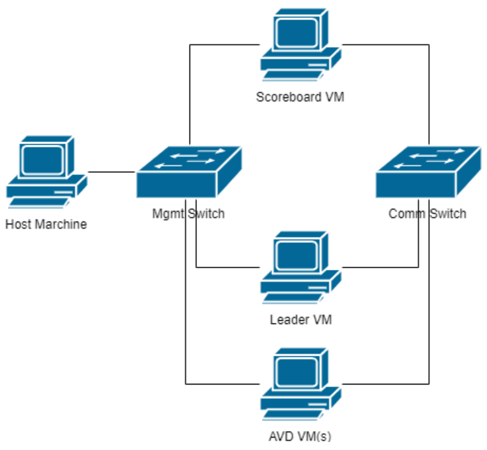
2.1 Deployment

Only two virtual machines (VMs) are needed: one to run the program for Automatic Vulnerability Detector and one to run the scoring system. The two VMs will communicate over a isolated network. The communication between host and VMs will be done over another network.

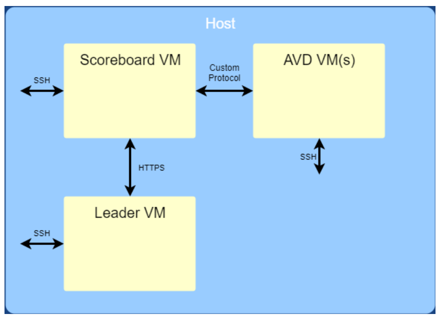
2.2 Secure channels and protocols

The three VMs will have a connection to the host system via SSH. The machine the leader uses communicates with the scoreboard via HTTPS. The machine(s) running AVD will communicate to the scoreboard server via a custom secure protocol:

1. The server sends its public key to the client.
2. The client generates a symmetric key.
3. The client encrypts the symmetric key with the server's public key.
4. The server decrypts the message and stores the symmetric key of the client.
5. The client and the server communicate using the symmetric key.



(a) Network diagram



(b) Machines diagram

Figure 2: Project infrastructure

3 Plan

3.1 Versions

- Basic solution [Weeks 1 and 2] - In this phase of the project we plan to configure both VMs and the database to store the contents of the scoreboard. We pretend to implement the connection between the scoreboard and the database and finally implement the authentication system. Expected to be ready by November 11th.

- Intermediate solution [Weeks 3 and 4] - We pretend to implement the scoreboard inner workings and web page. Implement the custom secure protocol. Planned to be ready until November 24th.

- Advanced solution [Week 5] - At this stage of the project we plan to adapt the Angr [1] tool to connect to the scoreboard and submit the exploits automatically. Ready until December 8th.

3.2 Effort commitments

Week	Inês	Miguel	Viviana
1	Configure VMs and DB; Connect scoreboard to database		
2	Implement HTTPS authentication system and costume protocol authentication system		
3	Implement scoreboard and custom secure protocol		
4	Implement Scoreboard web page; Create simple and rudimentary solution for automatic vulnerability detector.		
5	Create a program that uses Angr tool and connects to the scoreboard		

4 References

References

[1] Angr: Python framework.
`https://angr.io/`