Extended Lateral Movement, Host Control and Protection System

A SYSTEM FOR PURPLE-TEAMING

Mansour Qais Alhmoud | CS499 | Second Semester 2020/2021

FINAL YEAR GRADUATION PROJECT

Computer Sciences Department

Faculty of Information Technology and Computer Science

Yarmouk University

Irbid, Jordan

Graduation Project Report

BSc Project

CS Department

Project ID: YUIT-CS–XX-XXX

Extended Lateral Movement, Host Control and Protection System

A system for purple-teaming

Mansour Qais Alhmoud (2017801019)

Department of Computer Science

Faculty of Information Technology and Computer Sciences

Yarmouk University, Jordan

Contact Information

This project report is submitted to the Department of Computer Science at Yarmouk University in partial fulfillment of the requirements for the degree of Bachelor of Information Technology in Computer Science.

Author(s):
Mansour Qais Alhmoud, (2017801019)
Address: Jordan - Irbid
E-mail: 0doctorvenom9@gmail.com

University supervisor(s):
Dr. Ameera Saleh Jaradat
Department of Computer Sciences

Department of Computer Science
Faculty of Information Technology and Computer Science
Yarmouk University
Jordan

Internet: http://yu.edu.jo
Phone: +962 2 72711111 Ext. 6710
Fax : +962 2 7211111

Intellectual Property Right Declaration

This is to declare that the work done by Mansour Qais Alhmoud under the supervision of Dr. Ameera Saleh Jaradat having title "Extended Lateral Movement, Host Control and Protection System" carried out in partial fulfillment of the requirements of Bachelor of Information Technology in Computer Science, is the sole property of the author and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., only with a written permission of the author.

Date: _____

Authors (s):

Name: Mansour Qais Alhmoud *Signature: _____*

Supervisor(s):

Name: Ameera Saleh Jaradat *Signature: _____*

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of Dr. Ameera Saleh Jaradat having title "Extended Lateral Movement, Host Control and Protection System" is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Author(s):

Name: Mansour Qais Alhmoud Signature: _____

ACKNOWLEDGMENTS

This work is dedicated to my dear parents, the most loving in this world.

I would also like to thank Roman Szydlowski for inspiring me to make this project.

ABSTRACT

Red Team operations should be viewed as a capstone exam of a Corporate Infrastructure Security. It is used to stress-test the blue team's ability to effectively detect, respond to, and recover from an APT (Advanced Persistent Threat) attack. The purpose of a Red Teaming exercise is to understand if the target is prepared to withstand a targeted cyber-attack and what can be done to increase the resilience against such attacks. During a red team operation, after compromising the network and gaining initial foothold, persistence must be established then lateral movement is performed, the “Cyberoracle extended lateral movement, host control and protection system” can collect detailed information across the infrastructure including information about networks, hosts, servers, and other aspects of the infrastructure into a central database to be further processed. It includes various tools that processes the information and generate useful results for purple teamers to help them identify risks and possible attack vectors and facilitate lateral movement and infrastructure control. Each compromised host or device can be used to expand the owned infrastructure by the red teamer and then plan for the next possible targets. The blue teamer can use the same results to plan and employ defensive mechanisms.

Keywords:
Persistence
Lateral Movement
Command and Control
IT Infrastructure Security
Purple Team

Table of Contents

CHAPTER 1 – INTRODUCTION

Purpose of the Project

The goal of this project is to provide one single integrated platform that collects and correlates information across the infrastructure to centralize security operations and allows direct control of devices which will make security operations easier to manage, more effective, and more efficient. This project can help security teams protect the entire attack surface of the enterprise infrastructure. It can also help attacking the enterprise infrastructure. This project is mainly focused on providing a central monitoring and host control, that includes selected capabilities from the whole security stack including selected capabilities of SIEM, EDR, NDR, SOAR systems, and internal recon, vulnerability assessment, exploitation, and data analysis tools.

Purpose of this Document

The purpose of this document is to describe the technical details of this project, the idea behind it, its design, and implementation.

Overview of this Document

This document gives an introduction about penetration testing and Red/Blue/Purple teaming, and security operations in general. Then it describes the implementation of the Cyberoracle system and gives detailed information about its internal logic and operation.

Existing Systems and Solutions

Although most of the features and functionalities provided by this project already exist, but they are being used by professionals as standalone tools. My project includes only selected features from these systems and adds other features and integrate them together into one system that is suitable for both attacking and defending the infrastructure which saves time and effort and makes security professionals more efficient and the security operation more effective. Some of the existing similar systems include SIEM tools like IBM Qradar, Splunk, and SolarWinds, EDR/XDR solutions like Cortex XDR and other solutions like Attivo EDN. But all mentioned systems are built mainly for defense and monitoring, and what makes my project different is that it uses selected features of these systems to get information that can be used for offense planning as well as deriving defensive mechanisms, in addition to that, it includes other offensive tools and features which makes it very useful for purple teamers. So, there are no existing systems like cyberoracle.

Project Overview

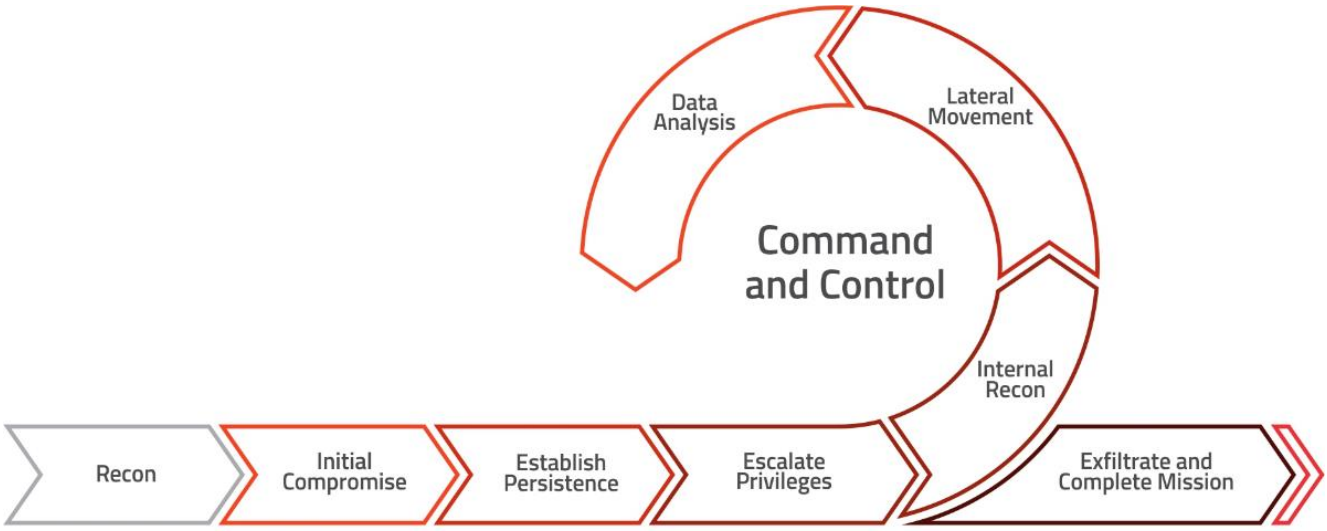
Penetration testing, or pentesting, is the process of simulating cyber-attacks against computer networks and applications to expose security vulnerabilities. It is a technical assurance exercise that has become an integral part of organizations' cybersecurity programs, applying to both physical and virtual infrastructure.[3]

Penetration testing usually involves a combination of manual and automated techniques to identify possible entry points that can be compromised on network devices, servers, web applications, API's, mobile applications, wireless networks and more. Once an initial foothold is gained, the penetration tester is provided with an opportunity to launch further attacks against additional internal or external resources. The aim is to get a point-in-time snapshot of the overall security exposure then demonstrate the depth an adversary can reach by laterally compromising other assets and escalating privileges to resources of higher security requirements. Deeper levels of compromise can help an organization understand the risks they face and what the impact of a breach can look like.[3]

During the penetration test, deep visibility is gained into the organization's security posture, exploitable vulnerabilities are identified, and recommendations are made on how best to fix those weakness. The results of the penetration test are aggregated into an easy to digest format so that leadership is provided with a prioritized list of vulnerabilities to make strategic decisions, enabling IT professionals and developers to make tactical fixes to remediate weaknesses or misconfigurations. Once remediation efforts have taken place, penetration testers will often retest the original findings and validate that they have been adequately fixed or sufficiently mitigated with compensating security controls. By following these steps, organizations receive a level of assurance that the overall security posture is more resilient against cyber-attacks.[3]

In cybersecurity we use the terms "red team" and "blue team" to refer to attackers and defenders, read teamers are doing the penetration testing by following a specific methodology. A red team is a group of offensive security professionals tasked with using real-life adversarial techniques that are available on adversary knowledge bases such as MITRE ATT&CK to help organizations identify and address vulnerabilities across infrastructure, systems, and applications, as well as weaknesses in processes and human behavior. In sophisticated penetration testing engagements, security professionals often conduct red teaming exercises to deliver objective-based assessments of an organization.[9] For instance, an objective might be to determine whether a sophisticated external attacker could gain access to an internal database system and exfiltrate a specific set of sensitive records. In this instance, the red team would simulate an external threat actor and determine whether they could find a series of exploitable vulnerabilities that would cause them to exfiltrate sensitive data from the target database.[3]

The following chart summarizes the execution of a fully-fledged red teaming exercise:



[5]

Each step of a Red Teaming engagement can be briefly summarized as follows:

Reconnaissance – analyzing the target without active attacks. This step involves extensive information gathering, including, but not limited to target organizations employees and their contact details (email, phones, etc.), passive fingerprinting of digital assets (websites, servers, other IT related artifacts), inspecting physical offices/branches, etc[5].

Initial Compromise – based on intelligence gathered during the first step an attack vector for initial intrusion is staged. In most cases it is some sort of a social engineering attack or exploitation of externally facing vulnerable systems. Sometimes, Red Teamers visit target organization's offices and use pre-texting (false pretense of being someone they are not) to install malware on some systems or drop a rogue device onto the network.[5]

Persistence – once a successful initial compromise takes place, the Red Team tries to maintain their access on a compromised system. This is done by installing persistent backdoors on the infected system or hacking into a more stable system on the network.

Privilege Escalation – before spreading across the network Red Teamers try to escalate their privileges on the initially compromised systems as this allows extraction of valuable secrets (credentials) and helps in performing more advanced attacks.[5]

Internal Recon / Lateral Movement / Data Analysis – this step is a continuous process aimed at navigating across the target organization's network and looking for important information or sensitive systems. Lateral movement usually happens via the means of credential extraction or exploitation of internal systems.[5]

Exfiltration – after the holy grail of the company is found, relevant information (defined during the preparation stage) is exfiltrated to finalize the execution of a pre-defined scenario.[5]

While the idea of a Red Teaming exercise is to simulate a real world cyber-attack against the target organization's network the main goal is not to show that it is possible to get inside no matter what, but to identify the weakest points within the organization (whether it's technological or organizational in nature), and to perform the exercise in a way that the malicious activity could be somewhat detected (leaving traces or intentionally raising some alarms to see if defensive mechanisms work correctly), and after finishing the engagement, provide practical guidance on how to better detect and respond to the malicious activity on the network and seal the weak points.[5]

In contrast, a blue team, typically based in a Cyber Security Operations Centre (CSOC), is a group of analysts and engineers responsible for defending organizations from cyber-attacks through a combination of threat prevention, deception, detection and response from real-life adversarial techniques with defensive techniques available in threat intelligence knowledge bases such as MITRE SHIELD and MITRE D3FEND. The blue team are expected to be the defenders. They need to defend against every single attack that is launched by the red team. For the blue team to be effective, they need to be able to defend against all attacks, all the time. Blue teams need access to log data, SIEM data, threat intelligence data and to network traffic capture data.[9] The blue team needs to be able to analyze vast swathes of data and intelligence to detect the proverbial needle in the haystack. A new technology called XDR (Extended Detection and Response) has been developed and many cybersecurity companies such as Palo Alto offer XDR products. XDR is a new approach to threat detection and response that provides holistic protection against cyberattacks, unauthorized access and misuse. It is the evolution of EDR, which optimizes threat detection, investigation, response, and hunting in real time. XDR unifies security-relevant endpoint detections with telemetry from security and business tools such as network analysis and visibility (NAV), email security, identity and access management, cloud

security, and more. It is a cloud-native platform built on big data infrastructure to provide security teams with flexibility, scalability, and opportunities for automation. XDR security is an alternative to traditional reactive approaches that provide only layered visibility into attacks, such as endpoint detection and response, or EDR; network detection and response, or NDR; and user behavior analytics, or UBA, and security information and event management (SIEM).[6]

Blue team exercises become controlled attack simulations that test the effectiveness of a blue team and its capabilities to detect, block, and mitigate attacks and breaches. Blue team exercises model threats that are probable to cause a loss event for an organization today. During the blue team exercise, a red team will begin attacking the organization's assets to exploit vulnerabilities of systems, devices, and applications across the network. As more attacks and actions occur across the business environment, the blue team's goal is to respond to the attacks and perform the necessary measures to isolate infected assets.

At the end of the blue team exercise, the red team will discuss the attack methods and their actions afterward. The blue team later uses this information to evaluate and prioritize changes required to prevent a similar attack from being successful again. In some cases, red teams and blue teams will directly interact during the simulated attacks, measure the effectiveness of attack response, and provide help with how to deal with the threat if the blue team experiences any difficulty. These types of assessments are generally known as purple team exercises.[7]

In recent years, there has been much more discussion in the Infosec industry about purple teaming. Purple teaming is a cybersecurity testing exercise in which a team of experts take on the role of both red team and blue team, with the intention of providing a stronger, deeper assurance activity that delivers more tailored, realistic assurance to the organization being tested.

By sharing intelligence data across the red and blue teams during the purple teaming process, organizations can better understand threat actors' Tactics, Techniques and Procedures (TTPs). By mimicking these TTPs through a series of red team scenarios, the blue team can configure, tune, and improve its detection and response capability. Red and blue teams can provide purple teaming engagements that allow organizations to measure their detection and response capabilities in a way that is much more closely aligned with real world threats. There is increasing recognition that Red Teams and Blue Teams should work together; thus, creating a Purple Team. This purple team isn't necessarily a new 'uber specialized team', but rather a

combination of both existing red team and blue team members coming together. It might be regarded more as a process (that engages red and blue together), as opposed to a unique team.[2]

The red team should be conducting objectives-based assessments that mimic known and quantifiable threat actors. As part of this process, the threat actor's Tactics, Techniques and Procedures (TTPs) should be known.

The blue team must educate themselves around these TTPs and build and configure their detection and response capability in-line with these known approaches. For instance, if a threat actor is known to use spear-phishing as part of a campaign, the blue team must ensure that it can detect and respond to spear-phishing activity. It is no use relying on SIEM technology in the hope that it will alert you to a spear-phishing campaign if the mail servers and relays are not configured to log or alert on specific types of mail content.[2]

If a threat group is known to be trying to exfiltrate sensitive data from a specific industry or market segment, the red team should be attempting to simulate this type of activity. As an approach, this might result in the red team compromising an end-user host, with the intent of reusing their credentials to launch further information gathering campaigns across the internal network infrastructure.[2]

The end objective of the red team might be to escalate their credentials to access a core database before exfiltrating traffic through a web-based protocol into a cloud-based service provider. The blue team needs to have tools and techniques that give them the ability to detect this type of traffic at every hurdle. The blue team needs to be able to respond to the attack and prevent the red team from carrying out its objectives.[2]

By creating a scenario where the Red Team and Blue team work together, Purple Team, organizations will be able to benefit from much more tailored, real-world assurance. The blue team will be able to measure their detection and response capabilities in a way that is much more closely aligned with real-world threats.[2]

Too often an organization gets compromised, and the Blue Team does not see a thing. This is not because of poorly skilled or ineffective people, process, or technology. It is merely the case that the threat actor used a technique that goes undetected. By delivering purple teaming engagements, organizations can address this challenge head on.[2] This project can help purple

teamers in their engagements, to see the internal network infrastructure as would blue teamers see it to find weak points, bypass protections, and discover possible attack vectors, and seal it all.

[10]

CHAPTER 2 – SYSTEM ANALYSIS

Functional Requirements

1. Collect various information from devices on a local network
2. Analyze the collected information and extract useful information
3. Visualize the collected information
4. Perform security assessments and generate reports
5. Identify weak points in IT infrastructure
6. Detect suspicious and malicious activity or programs on the network and on devices
7. Make alerts on certain events and provide details about the incident
8. Provide full control over devices on local network
9. Allow developers to extend the integrated application easily

Non-Functional Requirements

1. Ultrahigh Security
2. High Performance
3. Reliability/Durability/Fault-tolerance
4. Scalability
5. Real-time operation
6. Extensibility

System Description

The system can be generally used by:

1. blue teamers (network infrastructure defenders), SOC team, or network administrators; to constantly monitor all hosts on a network and turn each host into a honeypot or a decoy host that can detect malicious activity on the network and report to the central monitor.

2. red teamers (network infrastructure attackers); to automate many tasks that are usually performed during a penetration test and get better understanding of the network which facilitates the planning of attack vectors.

The system consists of 2 main components:

1. Central monitor
2. Agents

There are 4 types of users in the system which are:

1. Super User: this user can view and modify all the information provided by cyberoracle and manage all other users, parameters, settings, etc...

2. Staff User: this user can download agents to install them on hosts and servers

3. Registrar User: this is an implicit user that is used by agents to register hosts when the agent is run for the first time

4. Agent User: this is an implicit user that owns a host instance in the database, and only that user is allowed to update the information of that host. credentials for this user are generated during the registration process by the agent and are known to the central monitor.

General Use-Case

- *Super User* configures the settings and creates *staff users*
- *Staff users* can download and install agents on hosts
- When the agent runs for the first time, it will go through registration process, during this process *Agent User* credentials are generated by the host and sent to the central monitor where the user is created, an instance of the host will also be created in the database, and it will be bound to the *Agent User* such than only that *Agent User* can perform CRUD on the host instance.
- During the registration, the registrar user is used for the authentication, after registration completes the created Agent User will be used for authentication by the agent.
- After the setup and registration is complete, the agent enters the normal flow and will constantly collect information about the host system and the surrounding network environments and send all collected data to the central monitor through REST API, some information is sent through Cyber Oracle Protocol.
- On the Central Monitor, the data will be processed upon reception and stored in the database.
- The *Super User* can then view the information and perform various operations like cracking the collected password hashes from hosts, etc...
- The operations that the *Super User* can perform are defined by integrated apps in the home page, developers can easily integrate new apps with the platform by implementing a view and a controller (according to the MVC architecture)
- This system will provide the user with knowledge that can help discover vulnerabilities/weak spots and malicious behavior on hosts or the network and use it for their purposes, offensive and defensive.

Design Considerations

Design Constraints

1. **Hardware and Software Constraints**

   The central monitor must run on a capable device that can handle heavy computation; minimal performance requirements depend on the number of hosts that the central monitor is responsible for. The following specs can be used as a reference: 255 hosts require 1TB SSD storage (I/O speed around 2GBps), 64GB DDR4 RAM, a CPU with 16 cores with each core running 2.5Ghz+, stable network connection with 100Mbps bandwidth, and for some applications a discrete GPU is highly recommended but not necessary. The software environment for the central monitor must have PostgreSQL to be installed as a dependency, the program itself is cross platform and requires only python 3.8+ to be installed with the required libraries. The agents are precompiled for specific operating systems, and the installers for these agents install all the required dependencies.

2. **End User Characteristics**

   Although the installation or deployment of the system doesn't require any special skills, but users who are going to use the central monitor must have a solid background in cybersecurity to get the most out of it. Plugin developers must be proficient in python to write plugin apps that can be integrated with the system.

System Architecture

1. **Used Architectural Patterns**

Component	Used Architectural Patterns
Cyber Oracle Protocol	Master-Slave
Central Monitor	Client-Server, MVC, Plugin-Based
Host/Server Agents	Client-Server, Event-Driven

2. **Reuse Of Existing Components**

   Third-party programs and tools will be used to accomplish specific tasks, so instead of reinventing the wheel, efficient open-source and free software is used, these Third-party programs and tools include (PingCastle, Nmap, Hashcat, Impacket, PEASS-NG, Glances)

3. **Project Management Strategy**

   Divide and conquer strategy and requirements prioritization.

4. **Development Method**

   The development method followed in this project is Extreme Programming, the project is developed in small functional chunks, frequently tested, refactored, and easily adapts to changes.

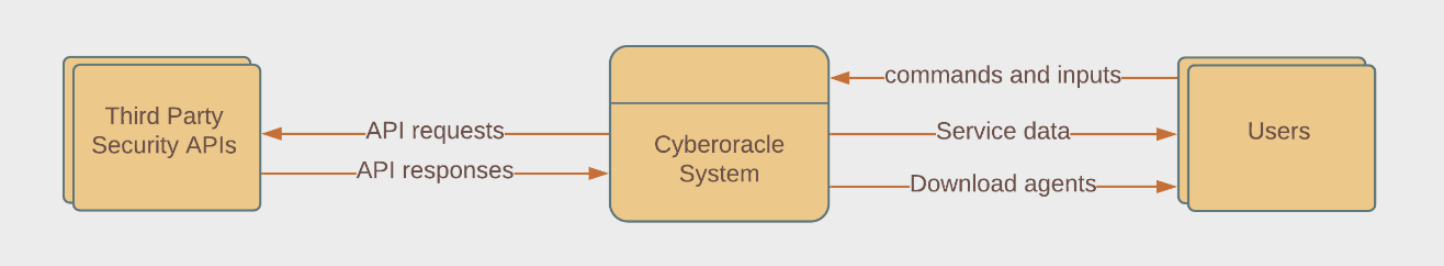
5. **Future Enhancements and Plans**

   In the future, I plan to add more plugin applications and power them with AI, add support for more device types and operating systems, and integrate more tools, APIs, and capabilities, and allow hierarchical chaining of central monitors which will make the system scalable.

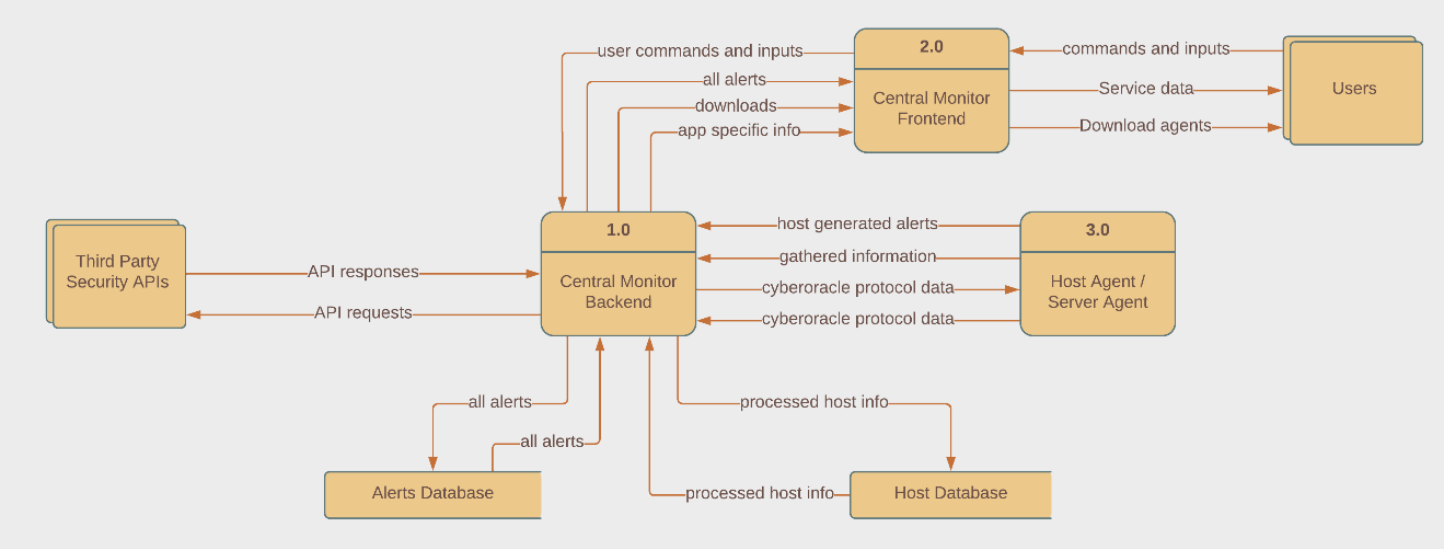
CHAPTER 3 – SYSTEM DESIGN

Data Flow Diagrams

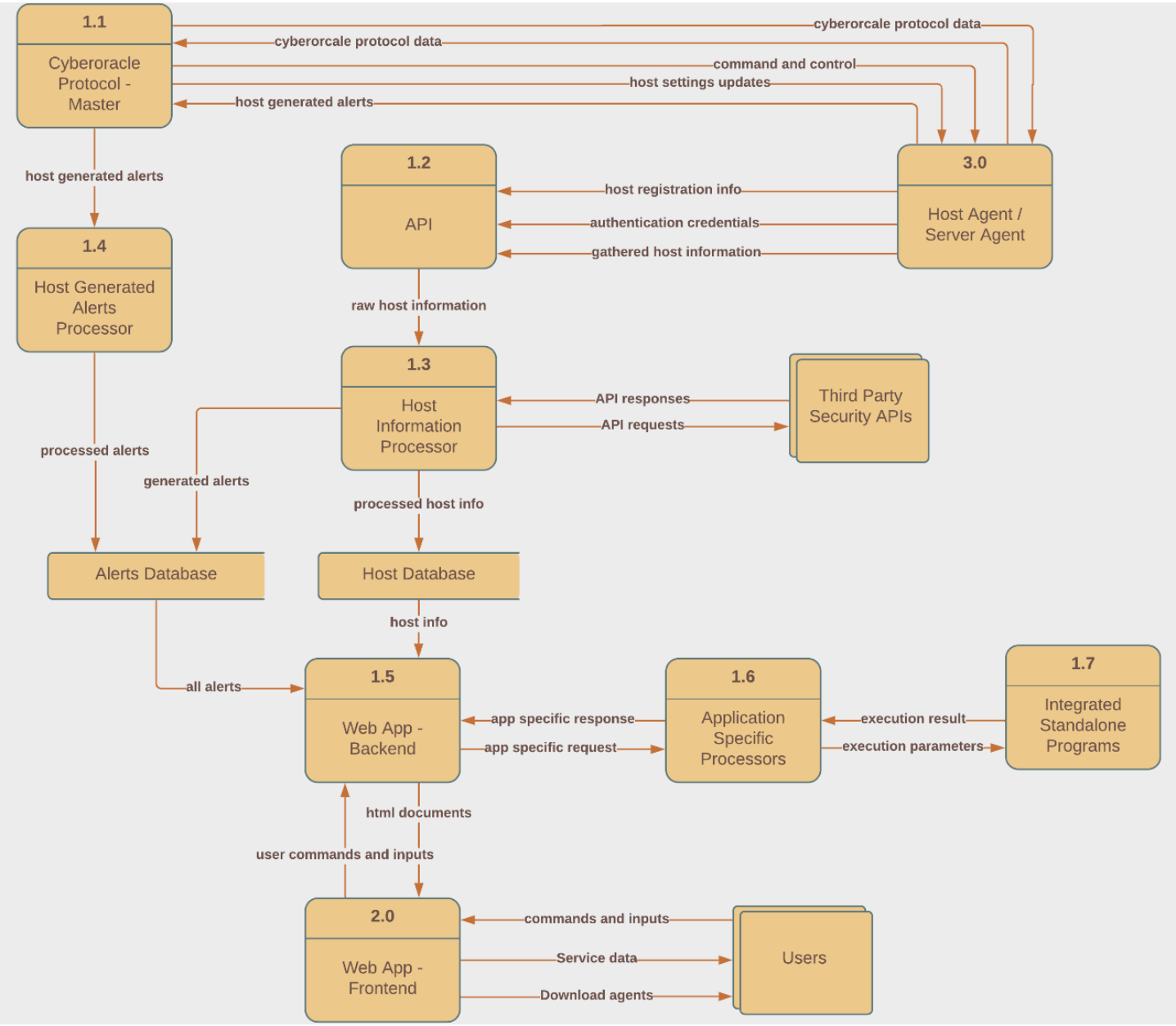
Context Diagram



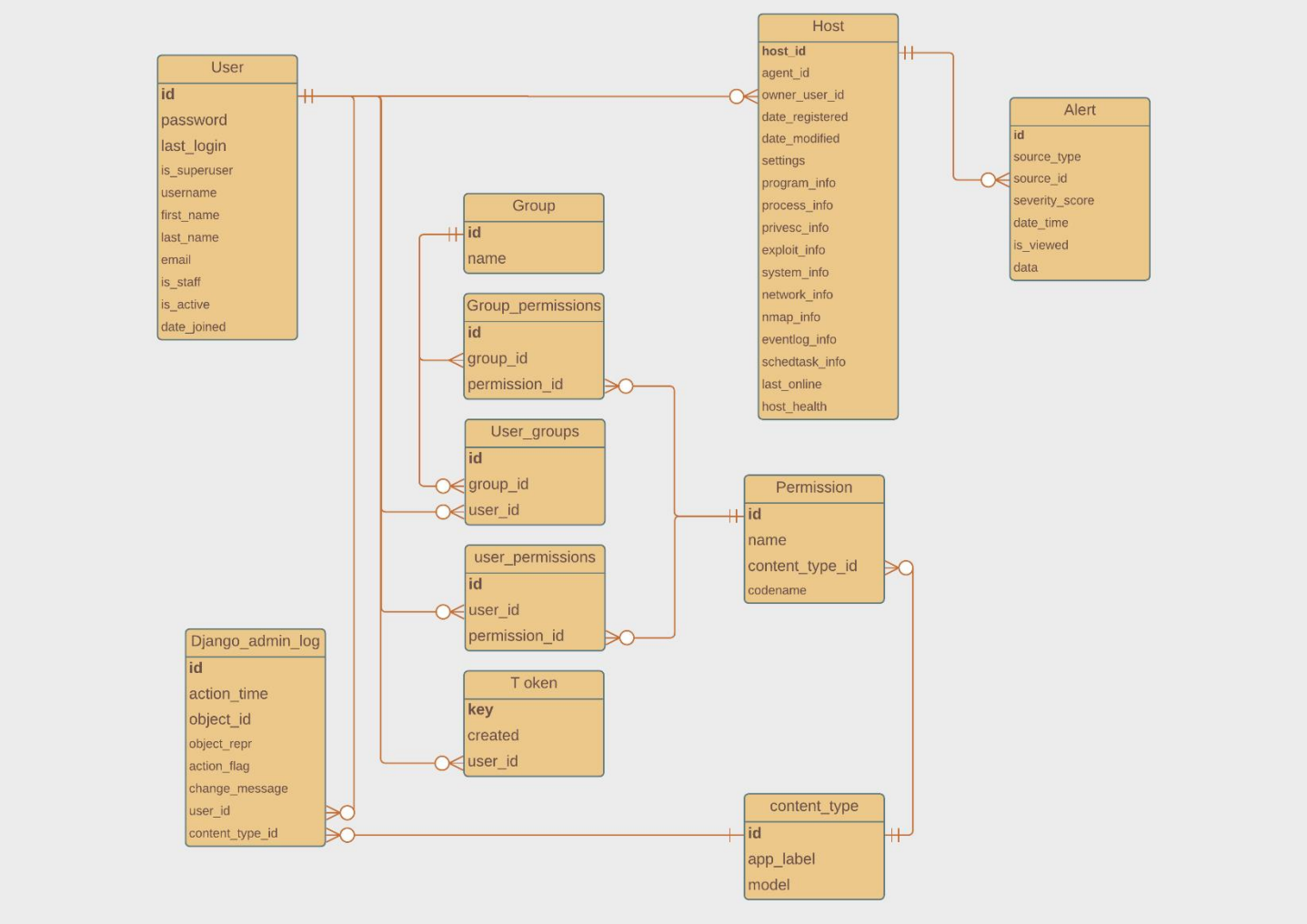
DFD level 0



DFD level 1


1.1
Cyberoracle Protocol - Master
cyberorcale protocol data
cyberorcale protocol data
command and control
host settings updates
host generated alerts
host generated alerts
1.4
Host Generated Alerts Processor
1.2
API
host registration info
authentication credentials
gathered host information
3.0
Host Agent / Server Agent
raw host information
1.3
Host Information Processor
API responses
API requests
Third Party Security APIs
processed alerts
generated alerts
processed host info
Alerts Database
Host Database
host info
all alerts
1.5
Web App - Backend
app specific response
app specific request
1.6
Application Specific Processors
execution result
execution parameters
1.7
Integrated Standalone Programs
html documents
user commands and inputs
2.0
Web App - Frontend
commands and inputs
Service data
Download agents
Users

Entity Relationship Diagram



System Components

1. **Central Monitor**

1.1. Realtime alerting system with email notifications

Alerts of different severity levels will be shown on alerts page, the alerts can be generated by hosts and the central monitor when something malicious or suspicious is detected

1.2. System settings interface

Used to set default settings for agents, and set central monitor settings

1.3. Integrations with third-party APIs

Virustotal API	Get scan reports for installed programs and running processes on hosts
Shodan API	Scan for devices (IP addresses) on local network that are exposed to the public network
Project Honeypot API	Check the IP addresses that hosts communicate with

1.4. Integrated Built-in Applications

Dashboard	Shows an overview of infrastructure security status
Host Monitor	Shows a table of hosts and provides an interface to view host detail and perform actions
Network Monitor	Shows network map
Password Cracker	Cracks the hashes collected from hosts to get plain text credentials
Server Log Analyzer	Analyzes the logs collected from server apps
CVE checker & Version Manager	checks the apps installed on hosts and server for being outdated and shows any CVEs and vulnerabilities they may have
Active Directory Scanner	Scans active directory environment and provides a complete report
Web App Scanner	Scans the target web app for common vulnerabilities and misconfigurations

1.5. Database Management Interface

Can be used to access the database and perform CRUD operations in all tables, and to create/manage users and groups

1.6. REST API

/api-token-auth/	POST	Authentication endpoint
/api/hosts/	POST/PATCH	Registration and information gathering endpoint

1.7. Cyber Oracle Protocol – Master (protocol specs in Appendix B)

keepalive mechanism
interactive shell client
information gathering request invoker
host settings management – send host settings
authentication and encryption mechanisms
anomaly and error reporting, and alert listener

2. Host Agent (Windows/Linux)

2.1. Information Gathering Module

Program Info	Installed programs, their version, and hashes
Process Info	Running processes, their locations, and hashes
Password Info	All locally stored hashed credentials
Privesc Info	Privilege escalation assessment report
System Info	General information about the system
Network Info	Network config, ARP, DNS, routing, open ports
Network Map Info	List of reachable devices on all connected nets
Event Log Info	System event logs
Scheduled Tasks / Cron Jobs Info	Scheduled tasks / Cron Jobs and their statuses

2.2. System Resource Monitor Provider

Upon receiving a request, a web server is started listening on port 61337 (default) and the user can view system resource consumption by processes and system load etc…

2.3. System Shell Provider

2.4. Cyber Oracle Protocol – Slave (protocol specs in Appendix B)

keepalive mechanism
interactive shell server
information gathering request responder
host settings management – request and receive settings
authentication and encryption mechanisms
anomaly and error reporting, and alerting sender

2.5. Decoys and Honeypots

Process Decoy	Intercepts any termination signals to the agent process and sends an alert to central monitor
Network honeypot	Sniffs network packets and parses them, if something malicious/suspicious is detected an alert will be generated
Port honeypot	Listens on specific TCP/UDP ports and generates alert when someone connect to them

3. Host Agent Installer and Uninstaller (Windows/Linux)

Installer	Install dependencies Create installation directory Copy agent and settings files Create local user account Create scheduled task/ cron job
Uninstaller	Remove installation directory Remove local user account Remove scheduled task/ cron job

4. Server Agent (Windows/Linux)

5. Server Agent Installer and Uninstaller (Windows/Linux)

CHAPTER 4 – IMPLEMENTATION AND VALIDATION

Implementation of functional requirements

The whole project is implemented using python 3.8.5, PostgreSQL is used as a DBMS, Django is used as backend framework for web app on central monitor, Jinja web template engine, Bootstrap5, and jQuery are used on the frontend web app. Agents are written in python3.8.5 as well. The code is provided in Appendix A.

Implementation of non-functional requirements

Non-functional Requirement	Implementation method
Ultrahigh Security	using common defensive mechanisms and cryptographic algorithms
High Performance	using caching and efficient algorithms
Reliability/Durability/Fault-tolerance	provided by extensive exception handling, auto restarts of services, and separation of modules and threads
Scalability	yet to be implemented, but planned for future
Real-time Operation	all collected information is fresh and real time interaction is possible
Extensibility	Provided by the plugin-based architecture for app extensions

Validation

Malware samples, automated attack scripts, and manual attacks on the hosts and network were used to validate the correctness of the project. Stress test and penetration tests were used to test non-functional requirements such as performance and ultrahigh security of the central monitor.

Disclaimer

The central monitor has superpower on the network, it can execute any command with the highest privileges on all managed hosts, it also contains critical information such as password hashes which introduces a high risk, to mitigate this risk the central monitor must be maximally secured against any type of attacks, but here a little advantage comes: because central monitor is a perfect target for hackers, you can focus security resources on its protection, set up attack detection and be able to respond proactively.

Appendix A

Attached Folder Structure:

```
.
├── central_monitor
│   ├── _database_initialization.bat
│   ├── _run_cyberoracle.bat
│   ├── alerts
│   ├── api
│   ├── central_monitor
│   ├── config_files
│   ├── downloads
│   ├── home
│   ├── manage.py
│   ├── media
│   ├── settings
│   ├── static
│   └── templates
├── host_agents
│   ├── description.txt
│   ├── some_commands.txt
│   ├── linux
│   └── windows
├── requirements.txt
└── server_agents
    ├── linux
    └── windows
```

- ***requirements.txt*** contains python dependencies (***pip install -r requirements.txt***)
- ***PostgreSQL*** must be installed, and a database must be created
- ***settings.py*** contains all the settings required for running the central monitor, this includes database information
- ***_database_initialization.bat*** creates tables and schemas in the database
- Before running the central monitor, a superuser must be created using the command "***python manage.py createsuperuser***"
- ***_run_cyberoracle.bat*** runs the central monitor
- Source code for agents is in ***host_agents*** and ***server_agents***, agents are compiled then moved to the ***central_monitor/media*** directory where they can be downloaded

Appendix B

Cyberoracle protocol specifications

Cyberoracle PDU structure (JSON format is used):

Version	PDU_ID	timestamp	signature	payload

Payload = [Payload type | payload data]

Payload type [0]: Keepalive
- 0 : general keepalive echo
- 1 : general keepalive reply
- 2 : system monitor keepalive
- 3 : system monitor terminate

Payload type [1]: Command
- 1 : get_shell
- 2 : get_program_data
- 3 : get_process_data
- 4 : get_passwd_data
- 5 : get_privesc_audit
- 7 : run_system_monitor
- 8 : get_system_info
- 9 : get_network_information
- 10 : run_network_mapper

Payload type [2]: Settings
- 0 : settings request
- 1 : settings reply

Payload type [3]: Authentication
- 0 : authentication_request
- 1 : authentication_response

Payload type [4]: Alert
- alert_data

Payload type [5]: Error
- 0 : general/unknown failure while processing the PDU

Payload type [6]: Service Discovery

cyber_oracle_protocol_specs.txt

Appendix References

[1] https://engage.mitre.org/resources/structure

[2] https://www.nettitude.com/us/penetration-testing/purple-teaming/

[3] https://www.nettitude.com/us/what-is-penetration-testing/

[4] https://attack.mitre.org/resources/

[5] https://syntricks.com/red-teaming/

[6] https://www.paloaltonetworks.com/cyberpedia/what-is-xdr

[7] https://www.certitudesecurity.com/blog/analysis-and-assessments/what-are-blue-teams-and-blue-team-exercises/

[8] https://www.packetlabs.net/red-teaming/

[9] https://www.redscan.com/news/purple-teaming-can-strengthen-cyber-security/

[10] https://hackernoon.com/introducing-the-infosec-colour-wheel-blending-developers-with-red-and-blue-security-teams-6437c1a07700

[11] https://www.lucidchart.com