Research Document

Table of contents

[**1. Introduction 2**](#_91myuhraamk1)

[1.1 Context 2](#_jek93zdm4ayq)

[**2. Agreements 3**](#_jiocpgra9a11)

[**3. Requirements 4**](#_kd0nlnn815h1)

[**3.1. What should the system be able to do 6**](#_rxnfura3jzva)

[**4. Network drawing and description 7**](#_qw0l4cfuz977)

[**4.1. Description of the services 7**](#_swr41xetyudd)

[**5. Analysis 8**](#_f11f67lu81ng)

[5.1 Stakeholder analysis 8](#_spd7xmgn2d4l)

[5.2 Goal 8](#_w6nz4udiuavy)

[**6. Technical design 8**](#_opeagtemk8z9)

## 

# **1. Agreements/Requirements**

Network diagram

* Vlan-B is not needed we don’t need client machines
* Management should be accessed through SSH
* IP whitelist/range (fontys/home ips) for secure access within the honeypot
* VPN should reside in VLAN-A

Honeypot project itself

* Webserver will be low-interaction and cowrie will be high-interaction
* Change ssh port from 22. Make sure that pfsense is set to block instead of reject since the former doesn’t tell the attacker that it’s blocking when doing port scans (making the honeypot aspect more secretive).
* Make sure the database isn’t numerated to avoid having to make a lot of convincing dummy data.
* Create a separate ssh node as the entry point running cowrie, having a lot of different entryways for management and attack points

## 1.1. What should the system be able to do

The honeypot system should be capable of effectively emulating various types of systems and services, providing both low and high-interaction honeypots to attract and engage potential attackers. It should have the capability to accurately mimic real-world vulnerabilities and network environments, ensuring that it appears legitimate to attackers while remaining isolated from critical systems to prevent any actual damage. The system must be equipped with robust monitoring and logging mechanisms to capture and analyse network traffic and attacker interactions, allowing for the identification of malicious activities and potential threats.

# 

# 

# 

# **2. Network drawing and description**

## 

* External pfsense connected through seclab
* Three segmented networks
* VPN connection allowing internal access
* Client machines
* Services: Web Server,DB,Mail Server
* Monitoring/logging of the network
* IDS/IPS present

The reasoning behind this design is derived from the Network Separation and Segmentation assignment. This asked for at least two segregated networks and a public router in between granting access to the internet. Looking at the design above, it’s segmented into two vlans,

**VLAN-A** contains the management side/monitoring side of things. This is because we don’t want intruders to know that we are monitoring them so it’s outside the honeypot and we want to be able to manage all the pfsense routers without potentially leaking access to intruders.

**VLAN-C** is the kernel part of the network, this will be the entry point for intruders as it contains services that they might be interested in exploiting. It contains a web server hosting a crypto platform, a database for said webserver and a mail server.

**SSH**

SSH is a commonly targeted protocol by attackers attempting to gain unauthorised access to systems. Here's why SSH makes a good honeypot:

Attractive Target: SSH is often used in modern computing environments, making it a good target for attackers seeking to compromise systems for various malicious purposes, including unauthorised access, data theft, and lateral movement within a network.

Credential Harvesting: SSH honeypots can capture attempted login credentials used by attackers, providing valuable insight into their tactics, techniques, and the prevalence of brute-force attacks or the use of stolen credentials.

Lateral Movement Detection: Since SSH is often used by attackers to move laterally within a compromised network, deploying an SSH honeypot allows organisations to detect and monitor such lateral movement attempts, helping to identify and contain potential intrusions.

**WEB**

Web servers are prime targets for attackers looking to exploit vulnerabilities for various malicious purposes. Here's why deploying a web server honeypot is beneficial:

Common Attack Vector: Web servers are publicly accessible and often host critical applications and data, making them attractive targets for attackers seeking to exploit vulnerabilities such as SQL injection, cross-site scripting (XSS), or remote code execution (RCE).

Early Warning System: Monitoring web server honeypot traffic allows organisations to detect and respond to suspicious activities in real-time, providing an early warning system for potential security incidents and enabling proactive mitigation measures to be implemented swiftly.

In summary, SSH and web server honeypots serve as valuable tools for detecting, analysing, and mitigating cyber threats targeting SSH services and web applications, respectively. By deploying these honeypots within the network architecture, organizations can gain insights into attacker tactics, strengthen their defences, and enhance their overall security posture.

# **3. Analysis**

## **3.1 Stakeholder analysis**

From talking with the client, the main idea is that we want to simulate something very interesting for hackers for them to get interested and try to exploit the honeypot environment. From speaking with the client it seems like a high interaction honeypot is what were going for. We want to see as much attacks as possible and try to see if we are able to identify the type of attack and who’s exactly performing it. Whether it’s a human or a robot perform the type of attack

## **3.2 Goal**

**Main objectives / motivations:**

An environment to learn from what's happening, learn who are our attacks so checking the behaviour behind the attacks

Being able to distinguish the type of attacks

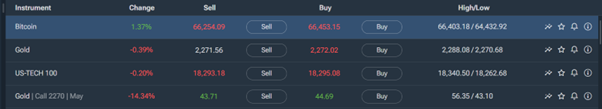
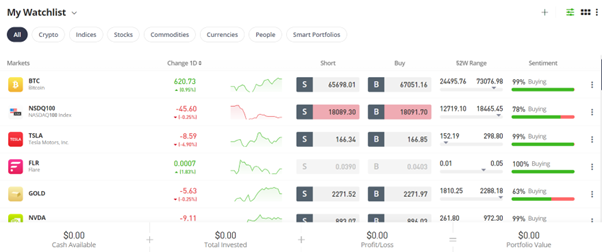
Being able to tell whether it’s a human or a robot (automated attacks)

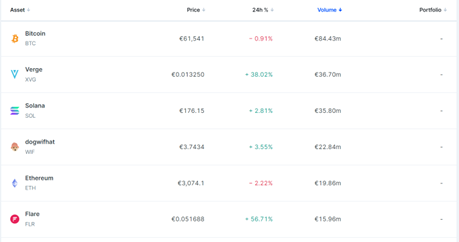
# **4. Research Questions**

**What are design conventions for online trading platforms that can be implemented in a honeypot for a more believable application?**

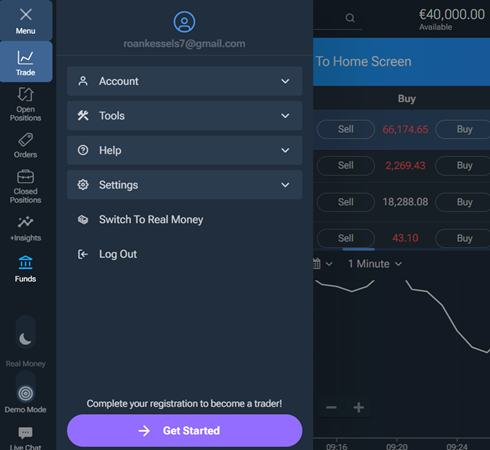
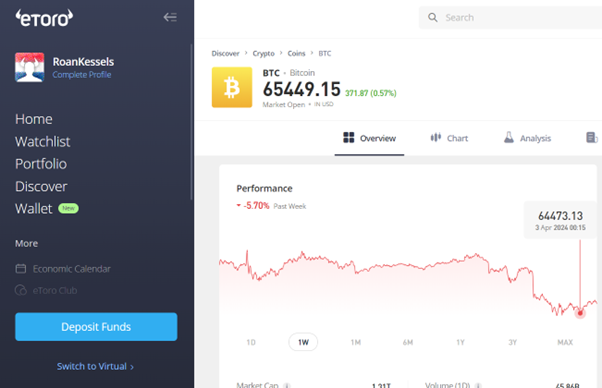
We’ve looked at the two biggest online trading platforms (Plus500 & eToro) as well as a number of smaller sites to discover what similarities and differences exist between them. From this, we’ll find a design that contains all standard aspects and does as good a job as possible in mimicking the look and feel of legitimate websites.

Every single platform had a noticeably similar dashboard page, which included a list overview of every offered currency and data about their market price. Some (like Plus500) also included line graphs in this landing page, but this was not a universal practice. Line graphs were in every case visible from a currency’s detail page, along with live updating prices, buy and sell options, and the ability to see the amount of that currency owned. Other widespread features include a watchlist, search function, portfolio and news page.





As for UI elements, nearly every page made use of a simple sidebar for navigating the different pages. Other repeating elements were the previously mentioned list display of all currencies and the graph + statistic cards in the detailed overview.



Incorporating the found patterns and conventions, our first design looks like this:

1. The dashboard page with currency overview
2. Currency detailed overview and trading functionality
3. Personal Portfolio page
4. Traditional login page