Cyber Adversary Profiling

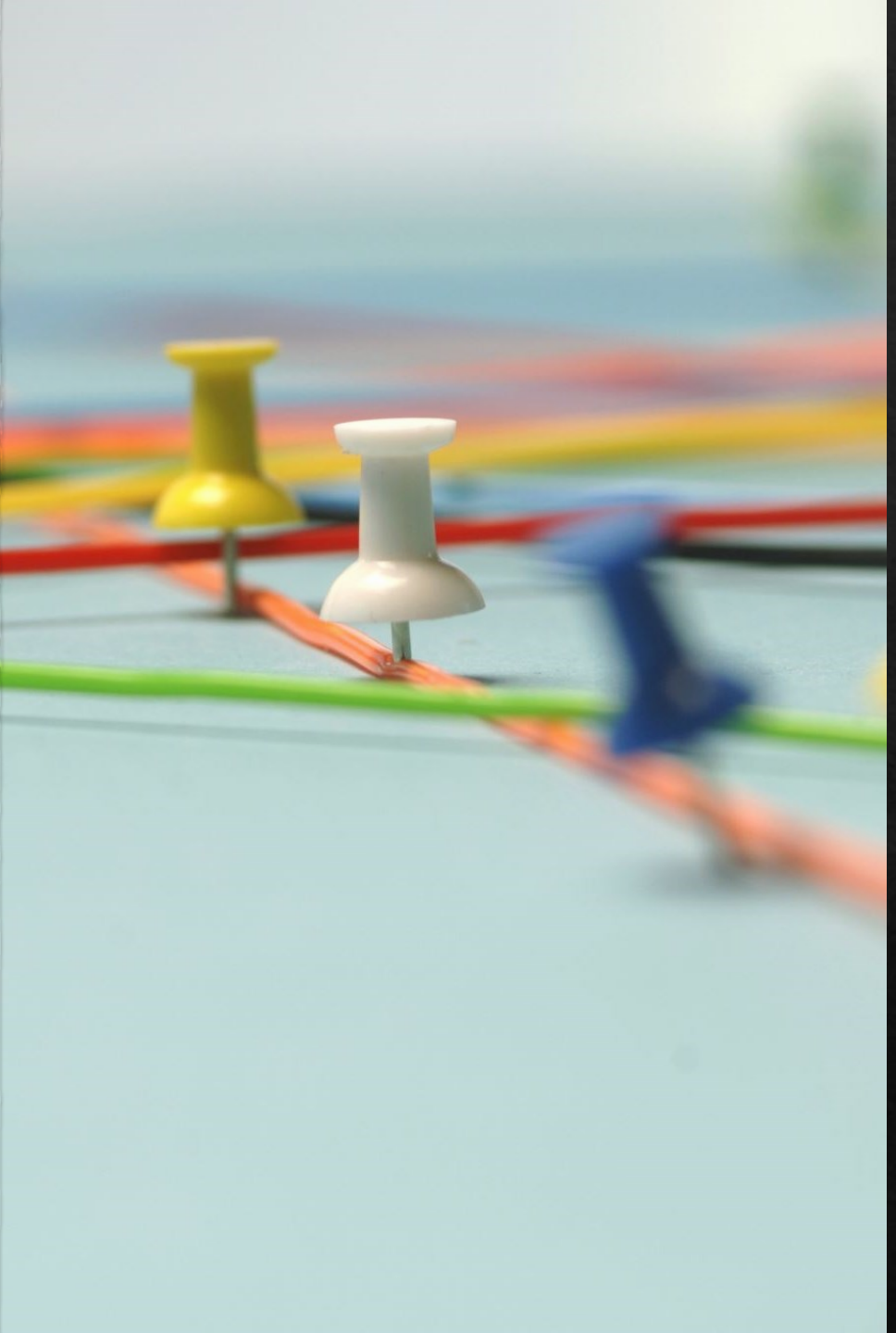
using SSH Honeypot Simulation for Threat Intelligence

-Ramya Manasa (RA2111027010011)
-Akhila Angara (RA2111027010029)
-Vyshnavi Nagella (RA2111027010034)
-Snehal Sukundari (RA2111027010049)
-Seyjuti Banerjee (RA2111027010052)
-Shashi Kumar (RA2111027010053)

What is an SSH Honeypot?

- A specialized cybersecurity tool designed to mimic an SSH server
- Created to attract and detect potential attackers attempting to gain unauthorized access

Why SSH Honeypots?

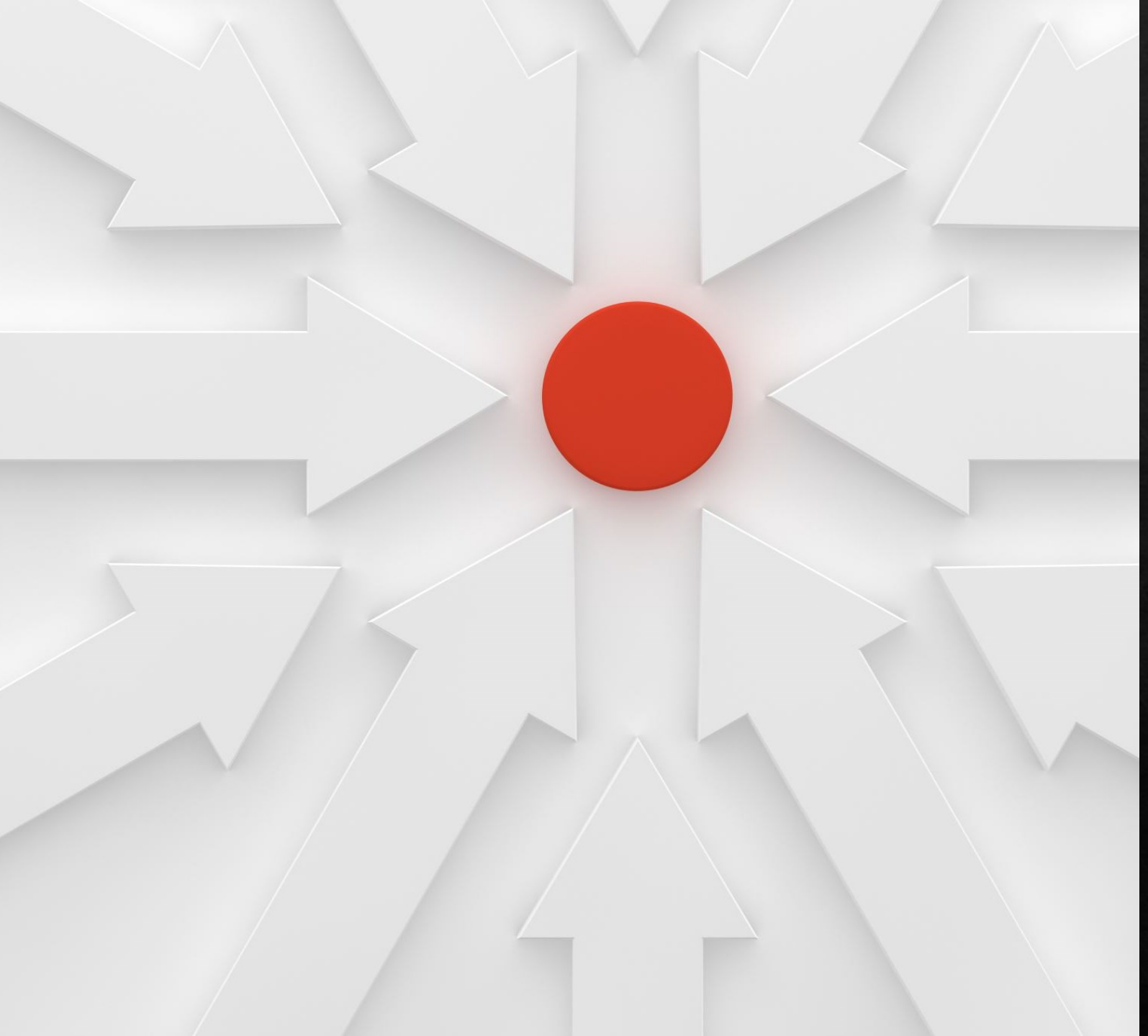


Attracting and Analyzing Attackers

A honeypot is an intentionally created fake system that is designed as a trap for potential attackers. They deviate the attack to the artificial system rather than the original system, and even it helps you detect the malicious traffic and track them.

The function of a honeypot is to represent itself on the internet as a potential target for attackers (usually a server or other high-value asset) and to gather information and notify defenders of any attempts to access the honeypot by unauthorized users

It appears as part of a network but is actually isolated and closely monitored because there is no reason for legitimate users to access a honeypot, any attempts to communicate with it are considered hostile.



Objectives

Project Objectives

- Attract and Monitor
- Attract potential attackers to analyze their behavior
- Monitor and log authentication attempts
- Analyze Attack Patterns
- Identify common attack vectors and techniques
- Enhance threat intelligence for better incident response
- Enhance Cybersecurity
- Contribute to cybersecurity awareness
- Strengthen defense strategies by understanding attack patterns

Significance of SSH Honeypots

Detects Automated and Targeted Attacks

Provides insights into both automated and sophisticated targeted attacks

Collects Data on Attack Vectors

Gathers data on common SSH attack vectors, aiding in proactive defense

Enhances Threat Intelligence

Contributes valuable threat intelligence for the broader security community

How SSH Honeypots Work

Features

SSH Honeypot Interaction

Listens on a specified port, attracting connection attempts

Simulates an SSH server, allowing interaction with potential attackers

Connection Flow

Diagram illustrating the flow of connections, from initiation to authentication

Components

Essential Components

SSH HONEYPOT SERVER

THE MAIN SERVER COMPONENT THAT ACCEPTS INCOMING CONNECTIONS

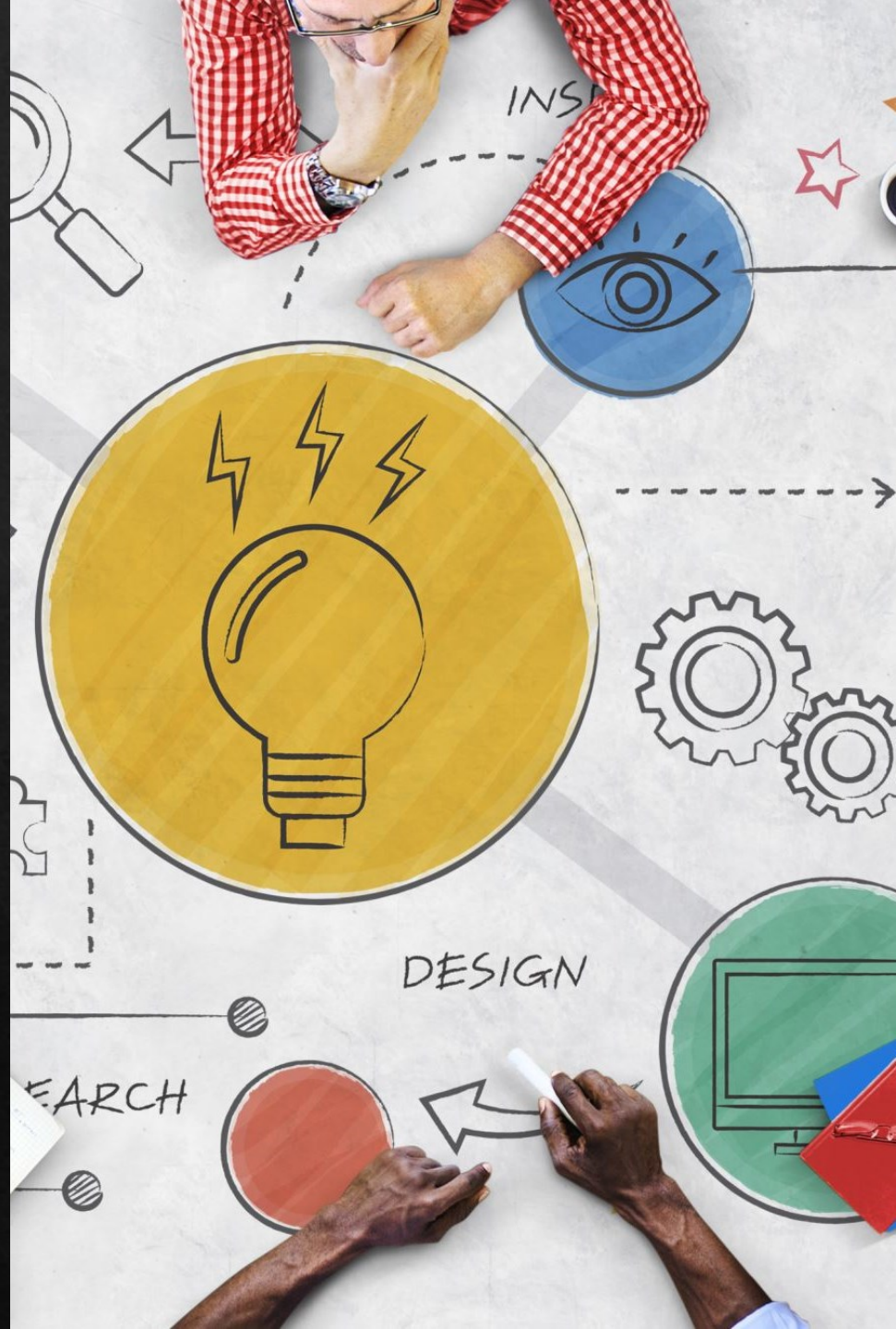
FAKESSHSERVER CLASS

SIMULATES AN SSH SERVER AND HANDLES AUTHENTICATION ATTEMPTS

LOGGING SYSTEM

RECORDS CONNECTION DETAILS AND AUTHENTICATION ATTEMPTS FOR ANALYSIS

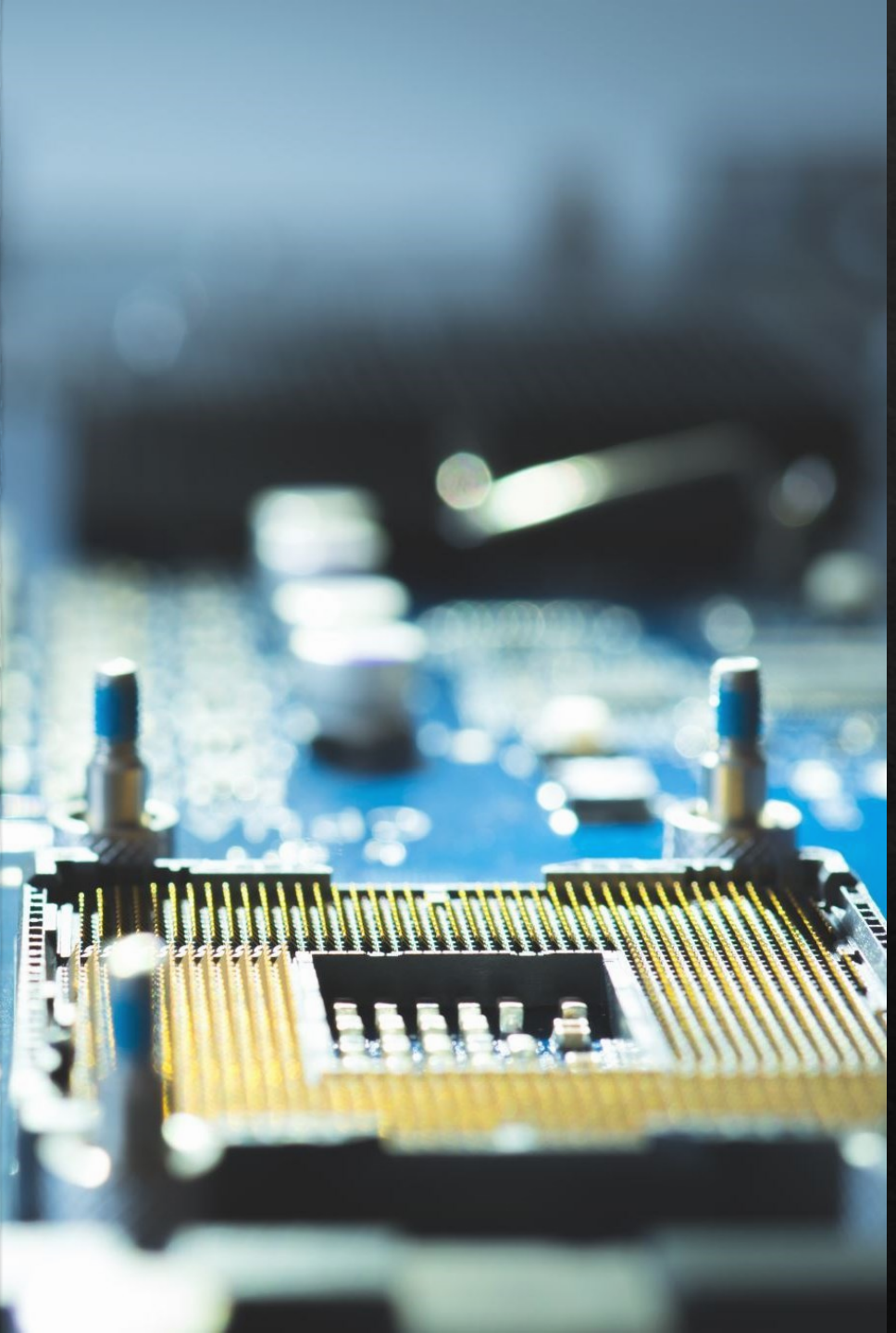
Implementation



Setting Up the SSH Honeypot

- Prerequisites
- Install Python and Paramiko
- Generate an RSA key for server authentication
- Configuration
- Define parameters like the listening port and welcome banner
- Run the script to initiate the honeypot

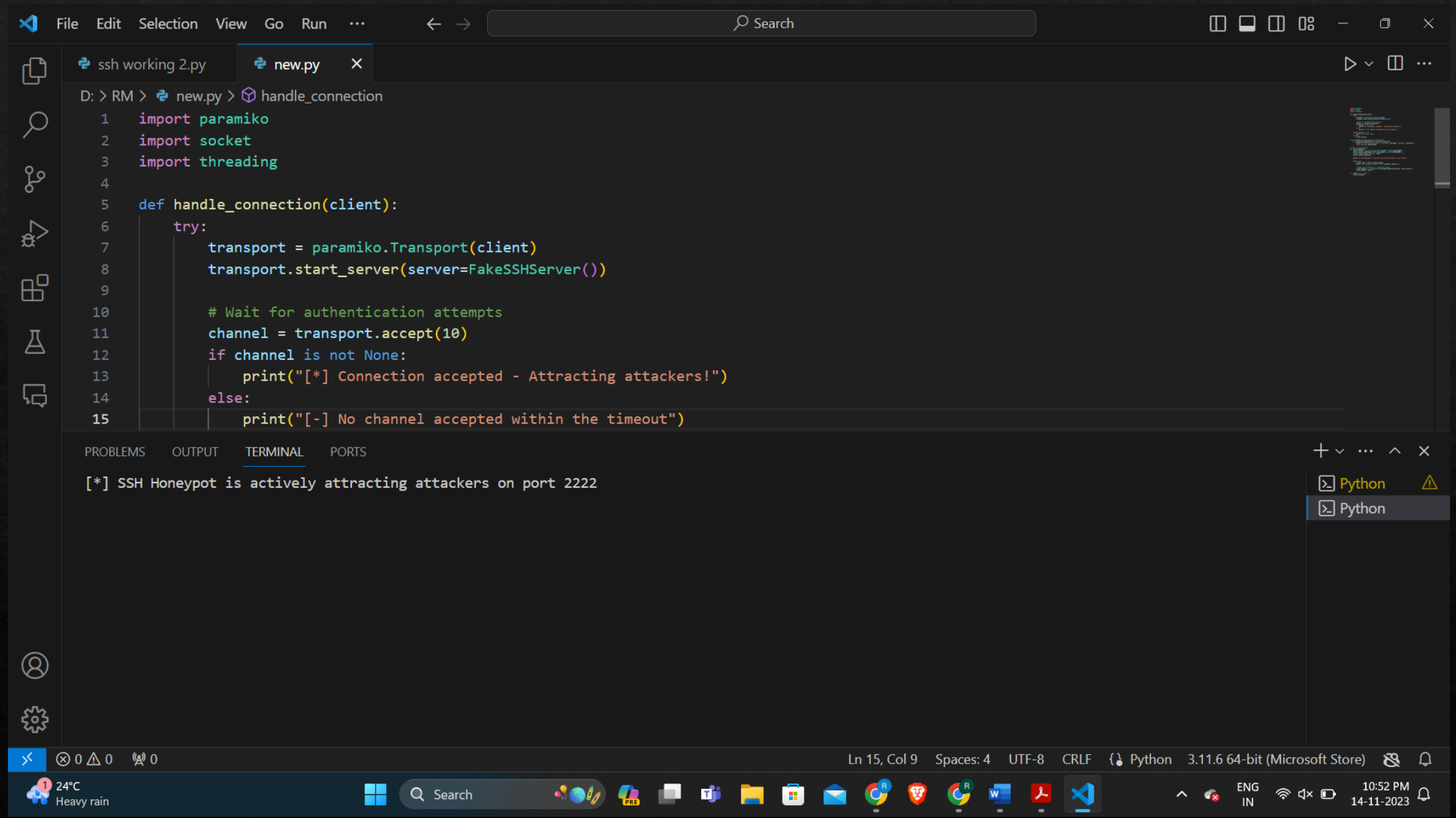
Features

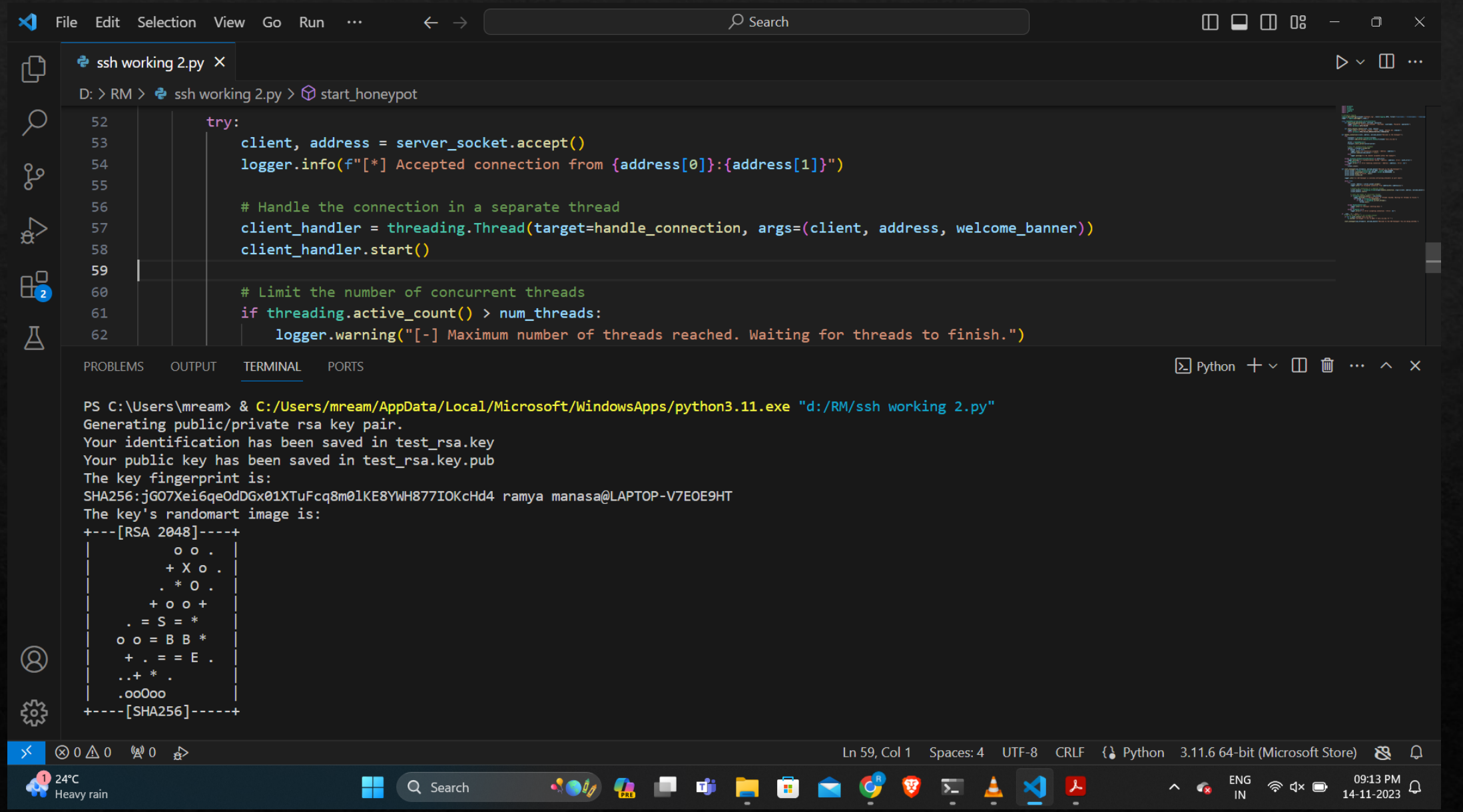


Notable Features

- Logging Authentication Attempts
- Detailed logging of usernames and password attempts
- Handling Multiple Connections
- Ability to manage and analyze multiple concurrent connections
- Customizable Welcome Banners
- Personalize the interaction with potential attackers

Results and Analysis


File Edit Selection View Go Run ···
Search
ssh working 2.py
new.py
D: > RM > new.py > handle_connection
1 import paramiko
2 import socket
3 import threading
4
5 def handle_connection(client):
6 try:
7 transport = paramiko.Transport(client)
8 transport.start_server(server=FakeSSHServer())
9
10 # Wait for authentication attempts
11 channel = transport.accept(10)
12 if channel is not None:
13 print("[*] Connection accepted - Attracting attackers!")
14 else:
15 print("[-] No channel accepted within the timeout")
PROBLEMS OUTPUT TERMINAL PORTS
[*] SSH Honeypot is actively attracting attackers on port 2222
Python
Python
Ln 15, Col 9 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (Microsoft Store)
24°C
Heavy rain
Search
ENG
IN
10:52 PM
14-11-2023


File Edit Selection View Go Run ···
Search
ssh working 2.py
D: > RM > ssh working 2.py > start_honeypot
52 try:
53 client, address = server_socket.accept()
54 logger.info(f"[*] Accepted connection from {address[0]}:{address[1]}")
55
56 # Handle the connection in a separate thread
57 client_handler = threading.Thread(target=handle_connection, args=(client, address, welcome_banner))
58 client_handler.start()
59
60 # Limit the number of concurrent threads
61 if threading.active_count() > num_threads:
62 logger.warning("[-] Maximum number of threads reached. Waiting for threads to finish.")
PROBLEMS OUTPUT TERMINAL PORTS
Python
PS C:\Users\mream> & C:/Users/mream/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/RM/ssh working 2.py"
Generating public/private rsa key pair.
Your identification has been saved in test_rsa.key
Your public key has been saved in test_rsa.key.pub
The key fingerprint is:
SHA256:jGO7Xei6qeOdDGx01XTuFcq8m0lKE8YWH877IOKcHd4 ramya manasa@LAPTOP-V7EOE9HT
The key's randomart image is:
+---[RSA 2048]----+
| o o . |
| + X o . |
| . * O . |
| + o o + |
| . = S = * |
| o o = B B * |
| + . = = E . |
| ..+ * . |
| .ooOoo |
+----[SHA256]-----+
Ln 59, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (Microsoft Store)
24°C Heavy rain
Search
ENG IN
09:13 PM 14-11-2023

Analyzing Attacker Behavior

- Sample Log Entries
- Showcase entries detailing authentication attempts
- Attack Patterns Observed
- Examples of common attack patterns observed during analysis
- Statistics on Connection Attempts
- Graphs or charts displaying key statistics on connection attempts

Challenges and Considerations

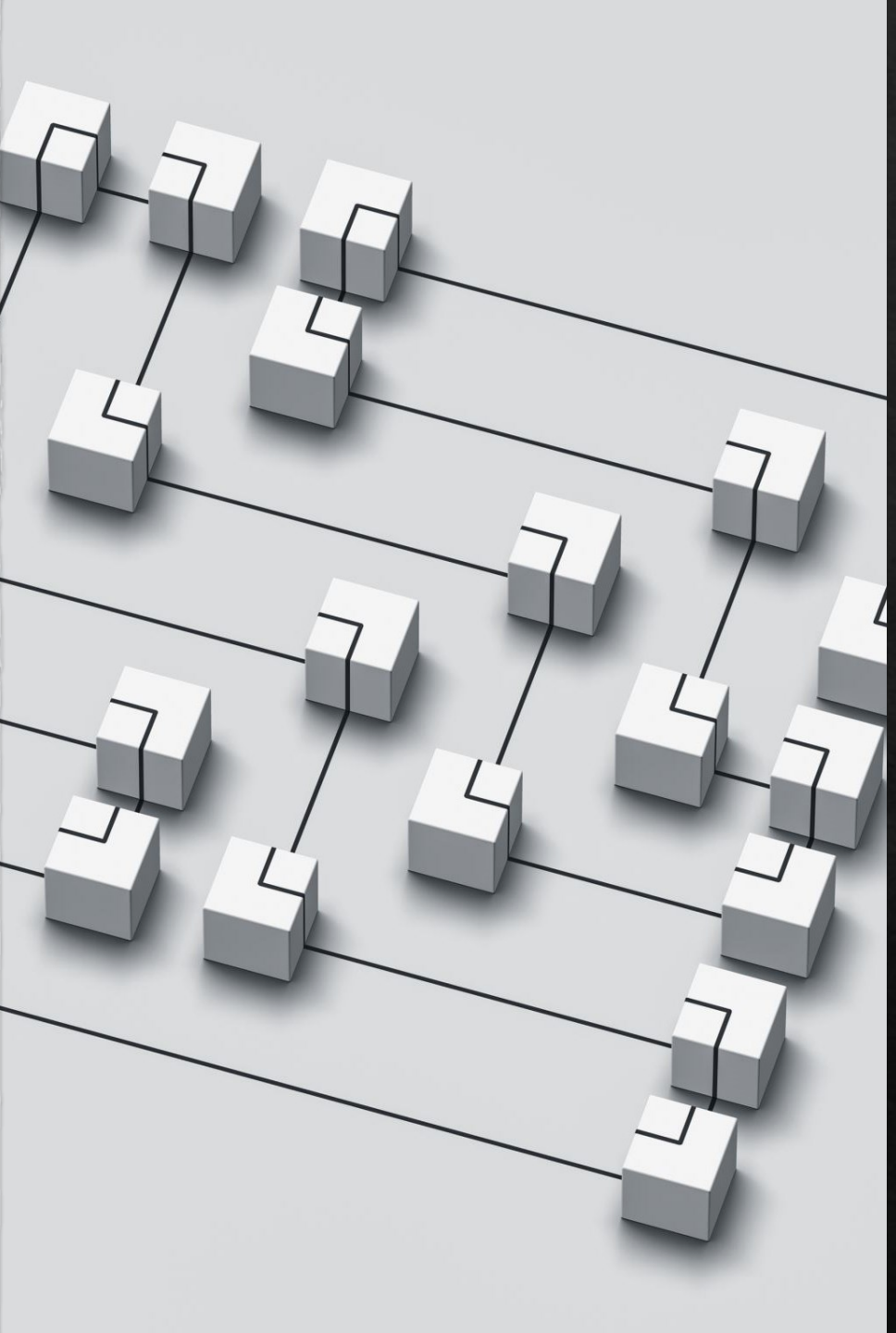
Challenges Faced

- False Positives
- Addressing instances where legitimate actions are mistaken for attacks
- Resource Utilization
- Managing resource consumption, especially in high-traffic scenarios
- Legal and Ethical Considerations
- Navigating legal and ethical considerations related to passive monitoring

Future Enhancements

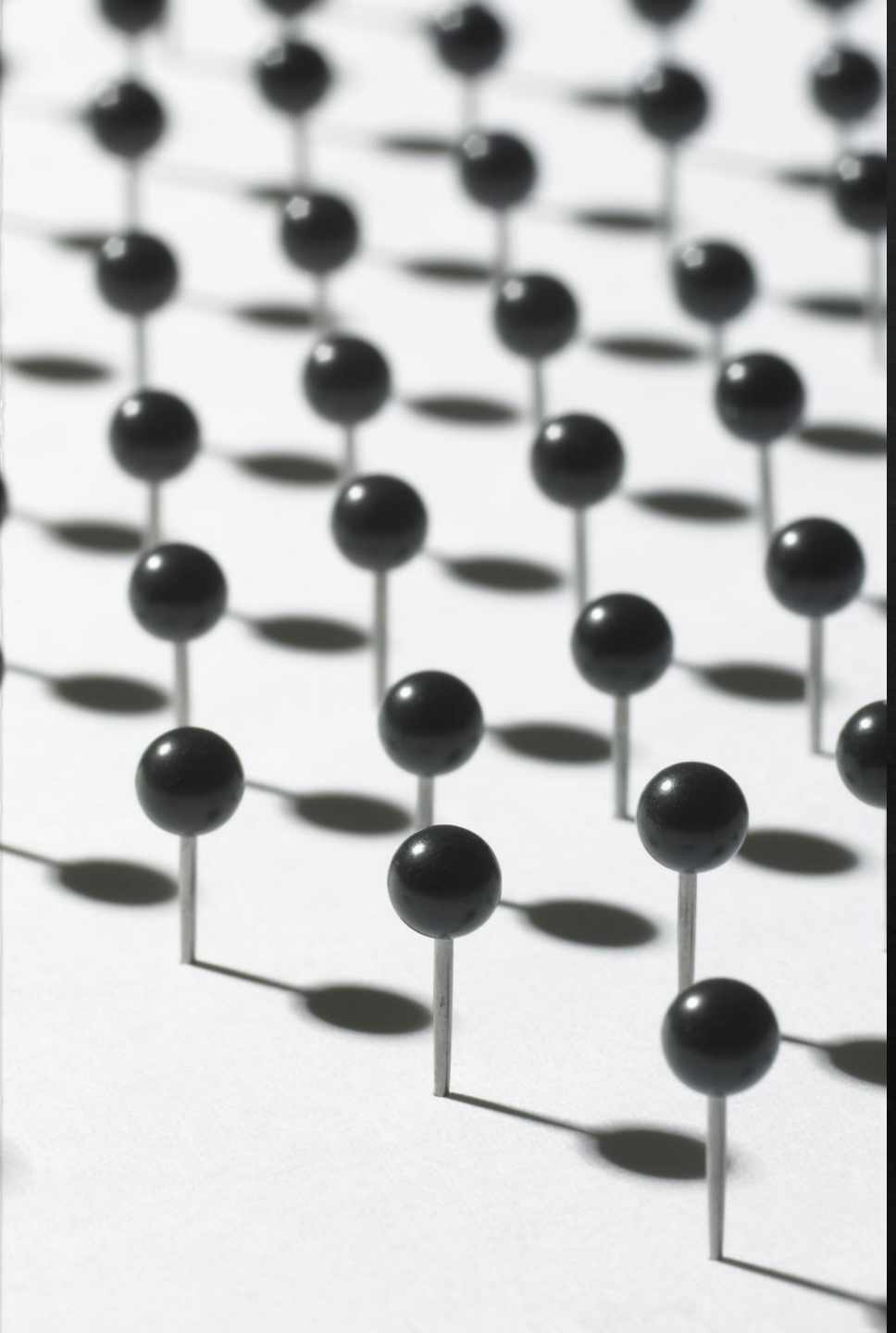
Future Enhancements

- Integration with Threat Intelligence Feeds
- Enhance data analysis by integrating with external threat intelligence
- Advanced Analysis Tools
- Develop or incorporate advanced tools for deeper analysis
- User Interaction Simulation
- Simulate user interactions to study attacker responses



Ensuring Security

- Network Isolation
- Deploy the honeypot in an isolated network segment
- Regular Monitoring
- Regularly review logs and analyze trends
- Secure Log Management
- Implement secure log management practices



Conclusion

- Key Takeaways
- SSH honeypots provide valuable insights into attacker behavior
- Continuous analysis contributes to a stronger cybersecurity posture
- Impact
- Influence on threat intelligence and incident response