SCET-MITWPU

Engaging Attackers with a Highly Interactive Honeypot System Using ChatGPT

Cybersecurity
In-House Project

Group No. B18

1032190103 Ajinkya Nagarkar

1032190071 Chinmay Talnikar

1032191369 Mustafa Mokashi

1032192053 Rishabh Sharma

Project Guide: **Prof. Umesh Raut**

Table of Contents

Introduction

Traditional honeypots rely on isolated systems or specific vulnerabilities and may lack the intelligence to deceive modern attackers effectively.

The project aims to address the limitations of traditional honeypots by developing a dynamic and interactive honeypot using Chat-GPT.

Honeypots attract and deceive potential attackers, enabling security professionals to monitor and analyze their actions.

The proposed honeypot system leverages natural language processing and conversational AI to engage in realistic conversations with attackers.

Chat-GPT allows the honeypot to generate human-like responses and adapt to the attacker's queries and tactics, increasing the chances of deception.

The honeypot system ensures security by running in a separate, controlled environment, providing valuable insights while protecting the organization's infrastructure.

Problem Statement

Develop a highly interactive honeypot system that engages attackers using ChatGPT and provides valuable intelligence on their tactics, techniques, and procedures. This system will help organizations detect and track attackers more effectively, improving their overall cybersecurity posture.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
Yadav, V., & Yadav, S. (2021). Honeypots using deep learning: A comprehensive study. Journal of Intelligent & Fuzzy Systems, 40(4), 7139-7150. doi: 10.3233/JIFS-189423.	2021	When an adversary initiates an interaction with our model, attacks are encouraged to add this predetermined watermark stimulating detection of adversarial examples. HoneyModels offer an alternate direction to secure Machine Learning that slightly affects the accuracy while encouraging the creation of watermarked adversarial samples detectable by the HoneyModel but indistinguishable from others for the adversary	One potential research gap is the limited evaluation of the proposed system. While the authors conducted experiments to evaluate the effectiveness of the proposed honeypot system, they only tested it against a limited number of attack scenarios.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
Huang, X., & Zhao, S. (2021). Chatbot-based honeypot for phishing detection. Journal of Computer Virology and Hacking Techniques, 17(3), 257-268. doi: 10.1007/s11416-020-00448-5.	2021	The proposed system provides an effective and practical solution for detecting and preventing phishing attacks. It can be easily deployed and integrated into existing security systems, making it a cost-effective solution for organizations of all sizes.	One potential research gap is the limited evaluation of the proposed system. While the authors conducted experiments to evaluate the effectiveness of the proposed honeypot system, they only tested it against a limited number of attack scenarios. Further evaluations using a larger and more diverse set of attacks could provide more robust results.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
De Lucia, E., & Zanero, S. (2020). Creating a dynamic honeypot with chatbots. In Proceedings of the 2019 Workshop on Cyber-Physical Systems Security and Privacy (pp. 19-24). ACM. doi: 10.1145/3322518.3323883.	2020	The chatbots act as the honeypot's interface and dynamically adapt to the attacker's behavior, providing more accurate data on their intent and strategies. This approach offers several advantages, including better identification and mitigation of zero-day attacks, as well as improved attacker profiling and threat intelligence gathering.	The paper focuses on a specific type of honeypot and attack (web-based attacks), and it would be interesting to explore the applicability of the proposed approach to other types of attacks and environments.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
C. Sun et al., "Application of Artificial Intelligence Technology in Honeypot Technology," 2021 International Conference on Advanced Computing and Endogenous Security, Nanjing, China, 2022, pp. 01-09, doi: 10.1109/IEEECONF52377.2022.10013349.	2021	Analyze the behavior of incoming traffic and identify malicious activities, reducing response time. AI-based honeypots can dynamically adapt to new attack methods, providing a more robust defense mechanism against evolving cyber threats.	AI systems can be expensive to develop and maintain, which can make it challenging for smaller organizations to implement AI-powered honeypots.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
M. Tsikerdekis, S. Zeadally, A. Schlesener and N. Sklavos, "Approaches for Preventing Honeypot Detection and Compromise," 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 2018, pp. 1-6, doi: 10.1109/GIIS.2018.8635603.	2018	A review of recent approaches that have been found to make honeypots more difficult to detect by attackers. A classification of honeypot characteristics that influence their ability to avoid detection by attackers.	The paper focuses solely on honeypot detection and evasion strategies and does not provide any information on the implementation and deployment of honeypots

Literature Survey

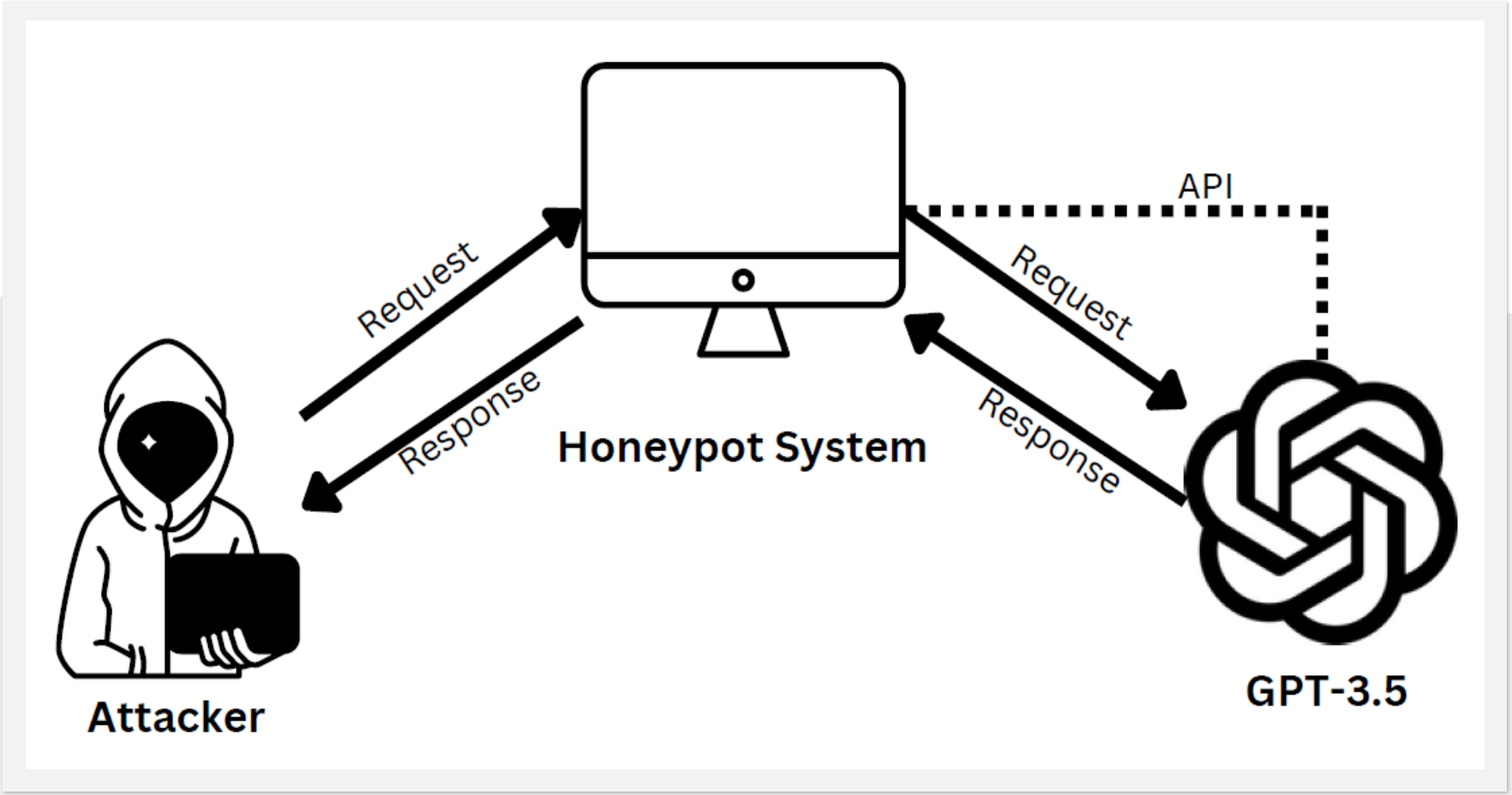
PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
B. -X. Wang, J. -L. Chen and C. -L. Yu, "An AI-Powered Network Threat Detection System," in IEEE Access, vol. 10, pp. 54029-54037, 2022, doi: 10.1109/ACCESS.2022.3175886.	2022	The proposed AI@NTDS system uses the LightGBM algorithm, which results in high accuracy, precision, recall, and F1-score values compared to other commonly used machine learning algorithms such as Naive Bayes, SVM, Random Forest, XGBoost, and Decision Tree.	One of the main disadvantages of AI@NTDS is that it requires large amounts of high-quality data to be effective. This can be a challenge in practice, as obtaining such data can be difficult, time-consuming, and expensive.

Literature Survey

PUBLICATION TITLE	PUBLICATION YEAR	POSITIVE POINTS OF THE PUBLICATION	GAPS IN PUBLICATION WORK
L. Tan, K. Yu, F. Ming, X. Cheng and G. Srivastava, "Secure and Resilient Artificial Intelligence of Things: A HoneyNet Approach for Threat Detection and Situational Awareness," in IEEE Consumer Electronics Magazine, vol. 11, no. 3, pp. 69-78, 1 May 2022, doi: 10.1109/MCE.2021.3081874.	2022	Improved security and resilience: By incorporating both threat detection and situational awareness, the HoneyNet approach enhances the security and resilience of AIoT, making it more resistant to cyberattacks. Effective threat detection. Improved computing and storage resources.	The complexity of designing and deploying a honeynet. There may be privacy concerns associated with collecting and analyzing large amounts of data from AIoT devices.

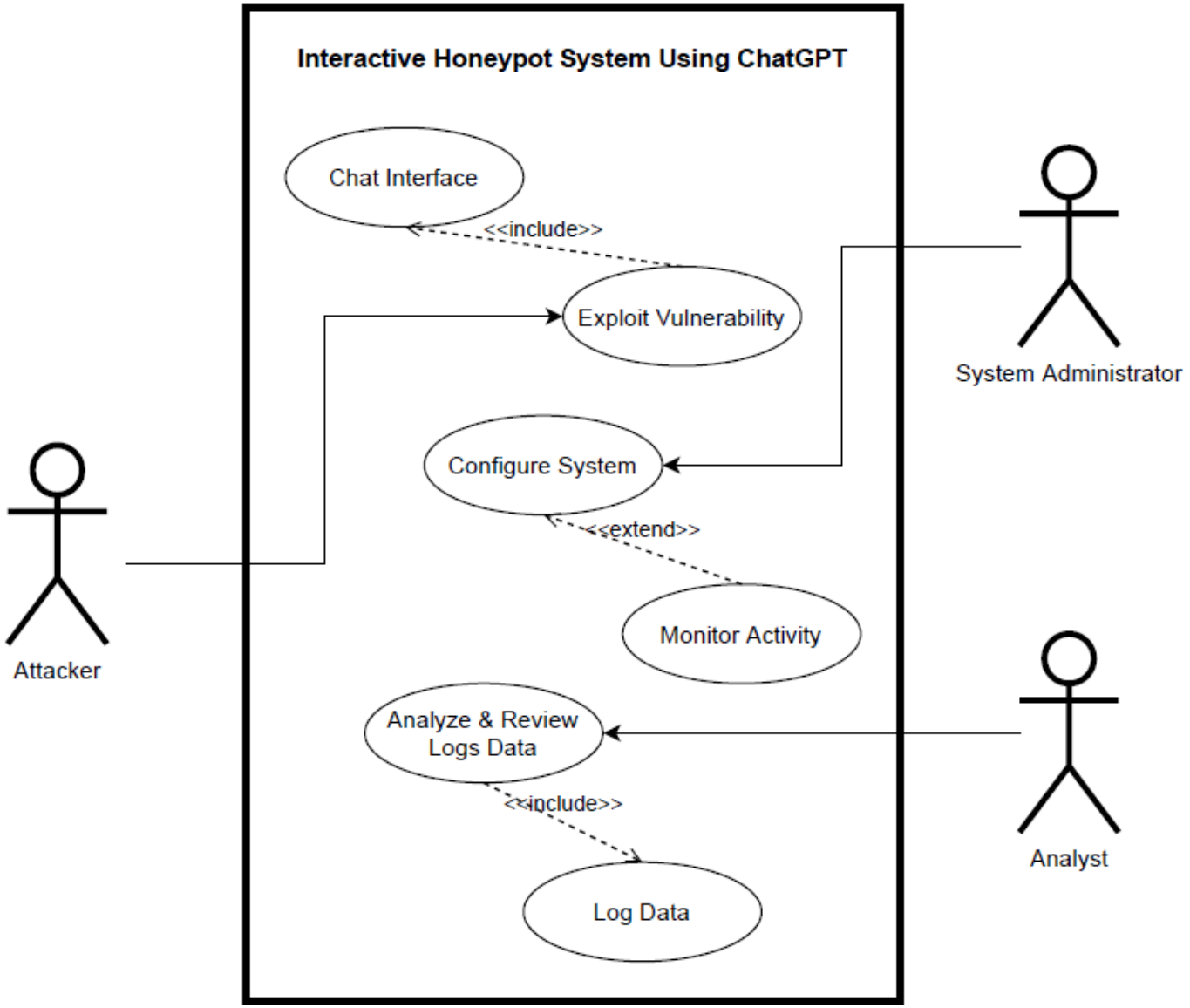
Requirements Gathering

- Hardware Requirements:
  - A computer or server with at least 8GB of RAM and a multi-core processor
  - Sufficient hard disk space to store logs and captured data
- Software Requirements:
  - An operating system that supports virtualization (e.g., Linux or Windows)
  - Virtualization software (e.g., VirtualBox or VMWare) to host virtual machines
  - Docker for containerization
  - Golang
  - Git for version control
  - A text editor or Integrated Development Environment (IDE) such as PyCharm or VS Code
  - ChatGPT language model and its dependencies
  - Web server software such as Apache or Nginx to host the chat interface
  - Network monitoring tools such as Wireshark to capture and analyze network traffic
  - Logging and visualization tools for log management and analysis

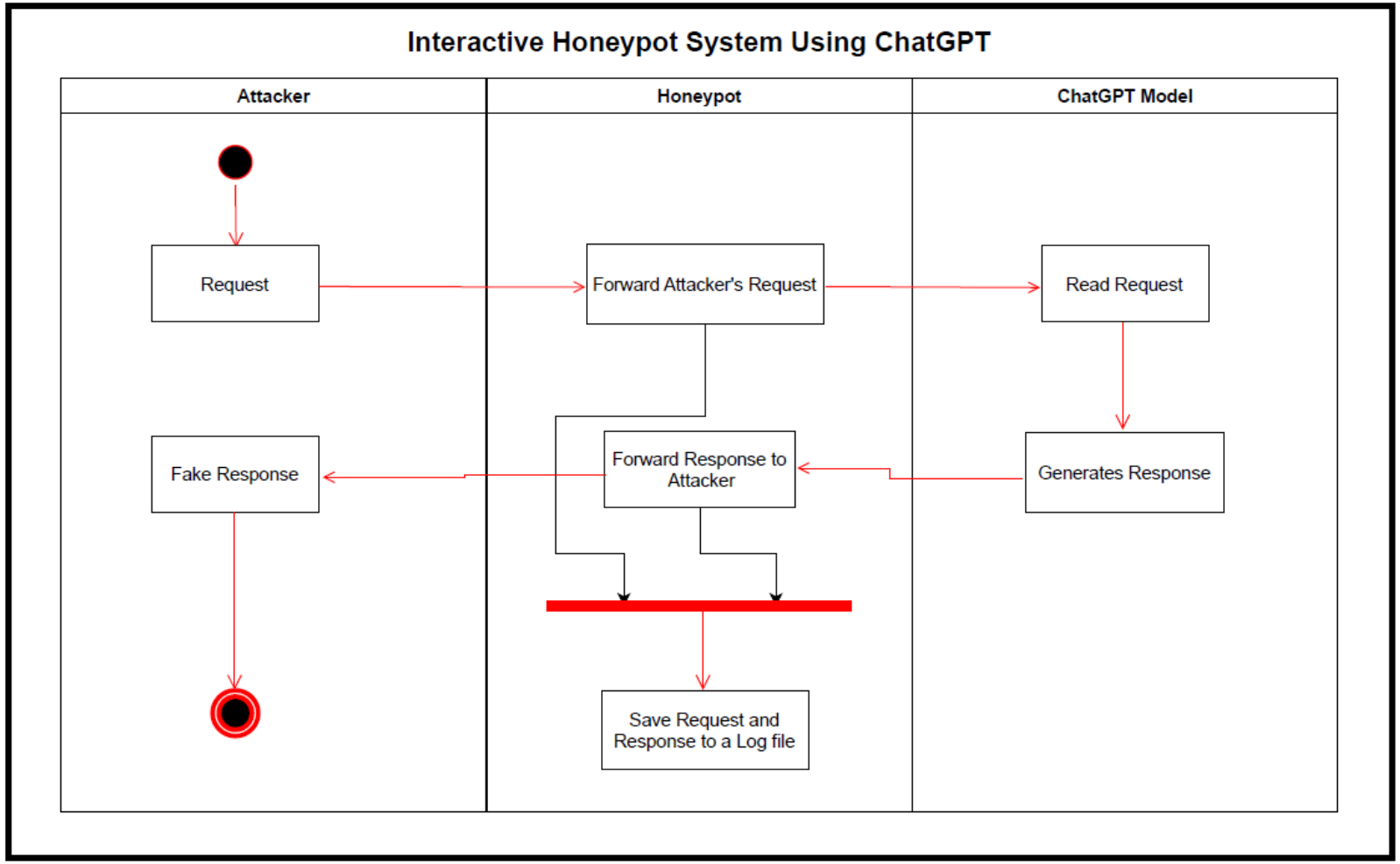


System Design

Architecture Diagram

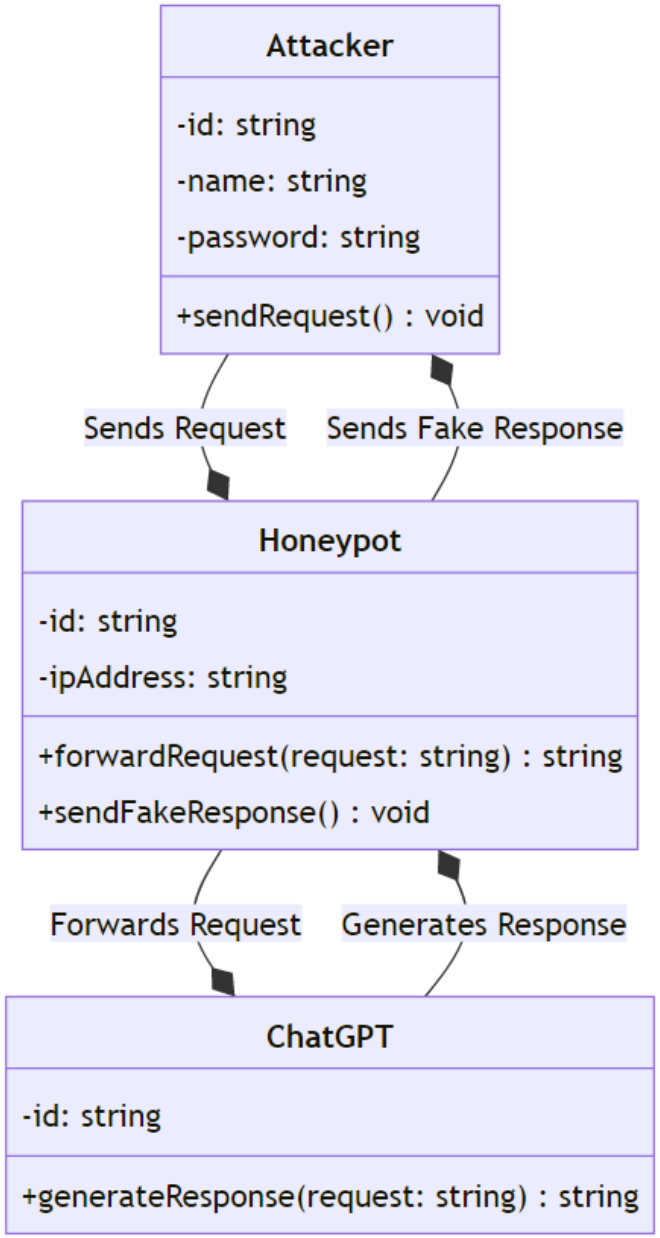


Use Case Diagram

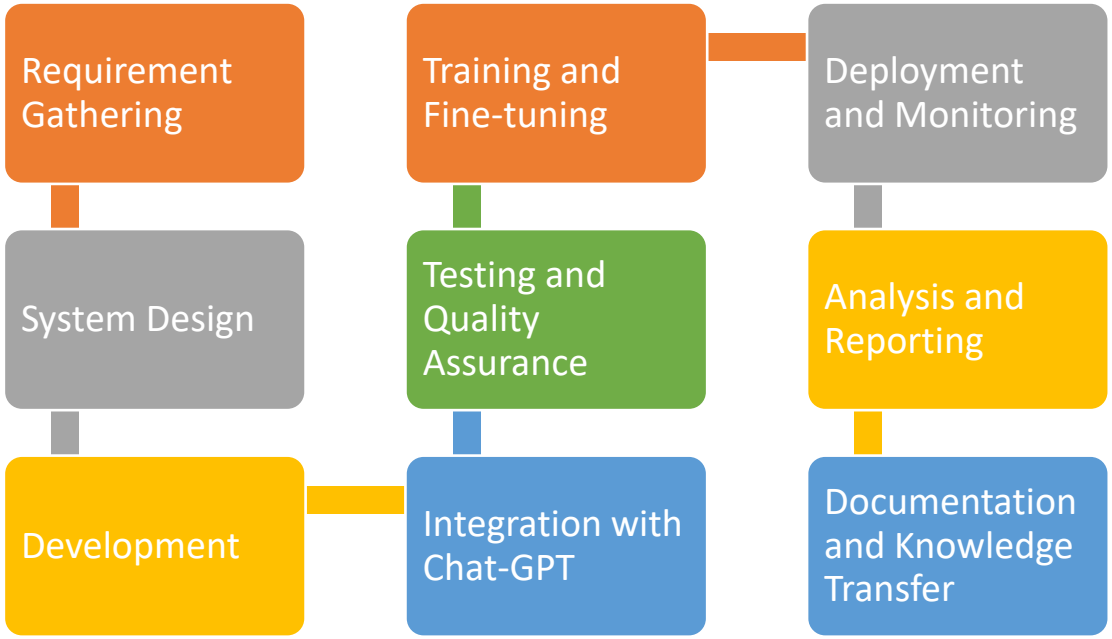


Activity Diagram

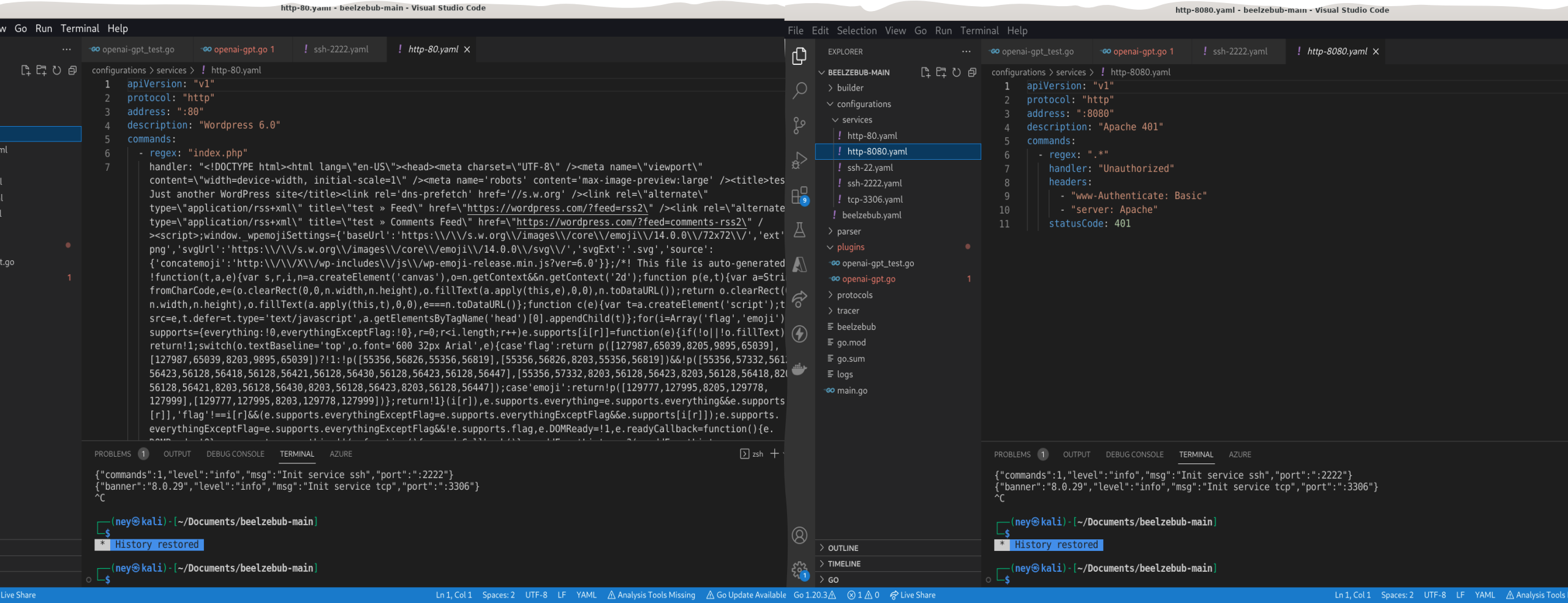
Class Diagram



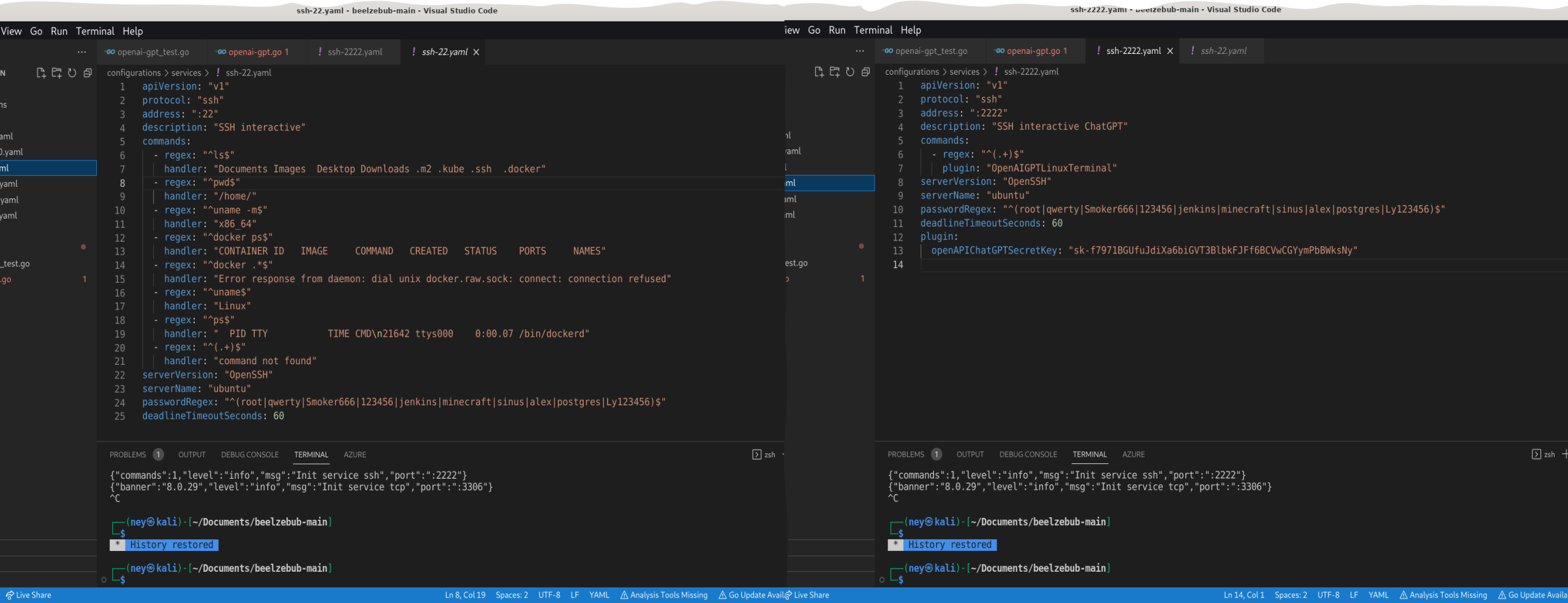
Implementation



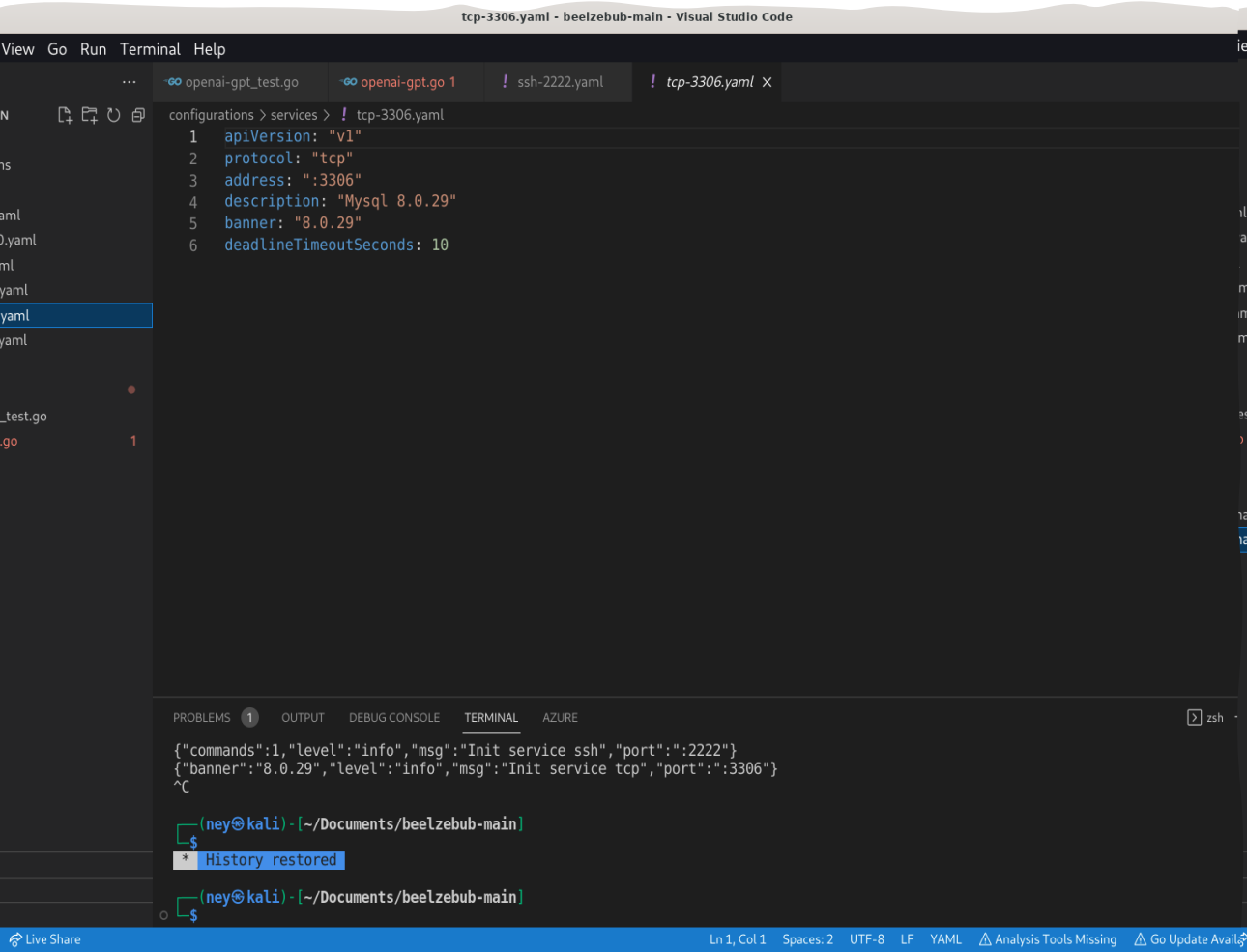
Source Code

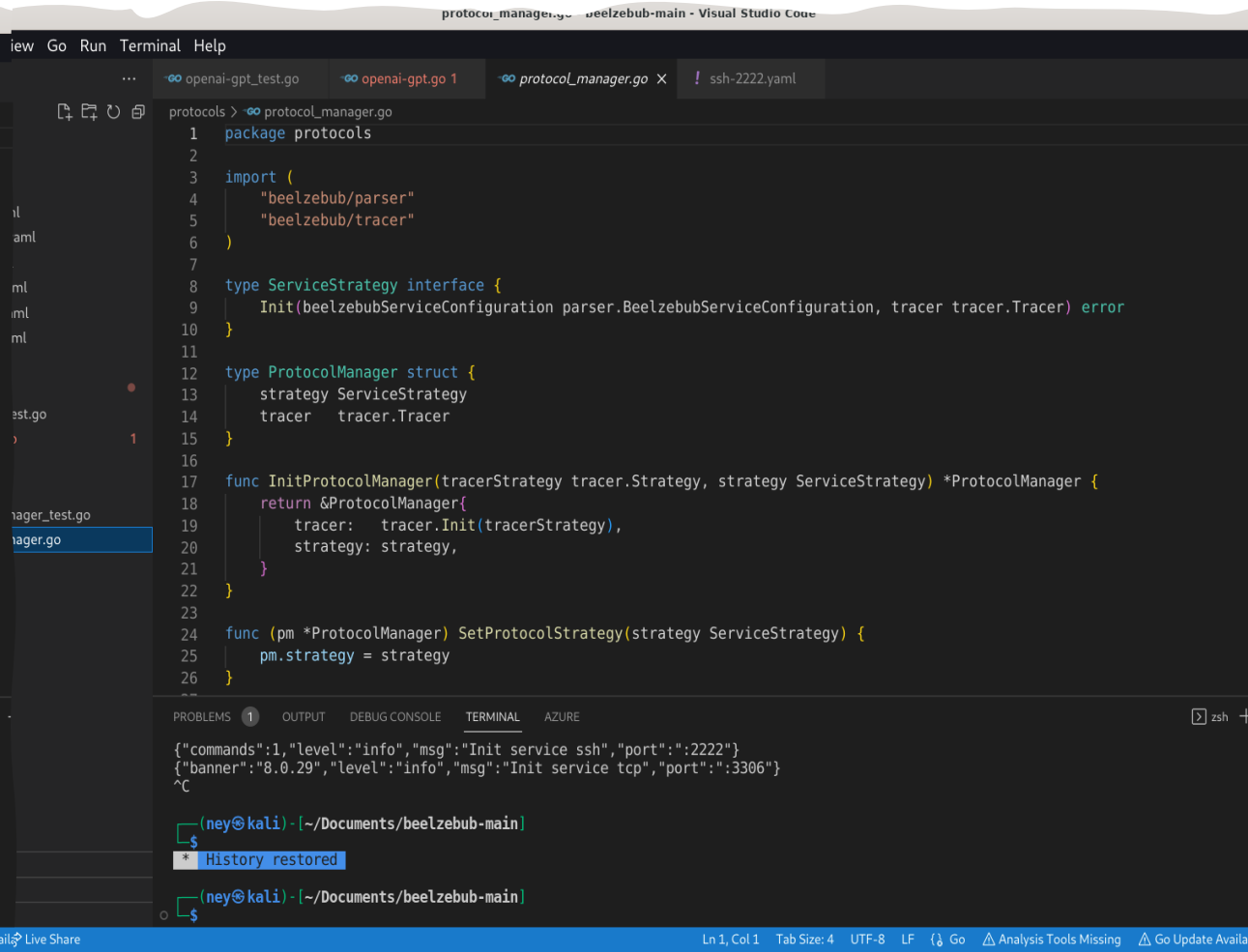

http-80.yaml - beelzebub-main - Visual Studio Code
http-8080.yaml - beelzebub-main - Visual Studio Code
apiVersion: "v1"
protocol: "http"
address: ":80"
description: "Wordpress 6.0"
commands:
apiVersion: "v1"
protocol: "http"
address: ":8080"
description: "Apache 401"
commands:
handler: "Unauthorized"
statusCode: 401
{"commands":1,"level":"info","msg":"Init service ssh","port":":2222"}
{"banner":"8.0.29","level":"info","msg":"Init service tcp","port":":3306"}
History restored

Source Code

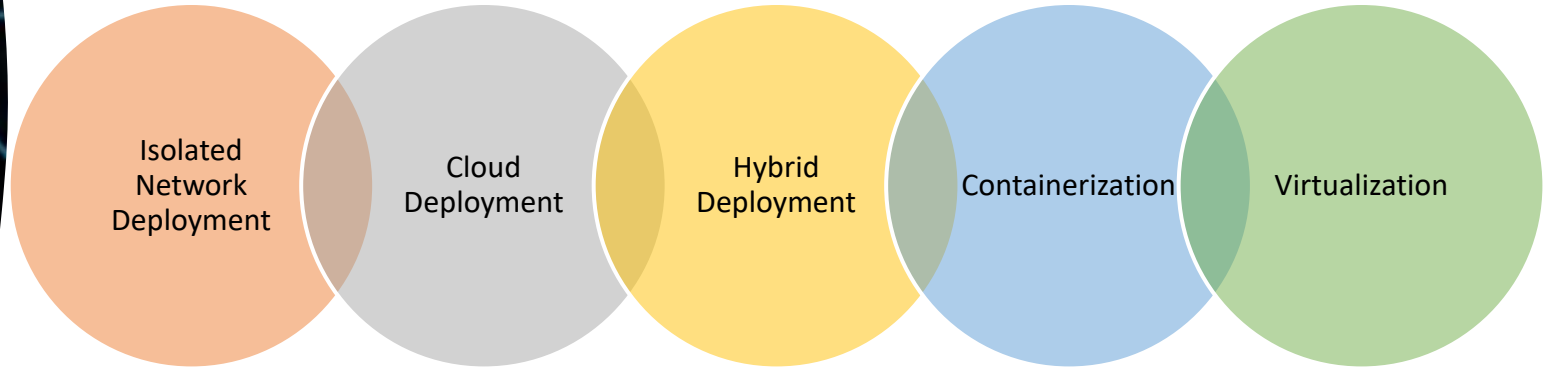

ssh-22.yaml - beelzebub-main - Visual Studio Code
ssh-2222.yaml - beelzebub-main - Visual Studio Code
View Go Run Terminal Help
openai-gpt_test.go
openai-gpt.go 1
ssh-2222.yaml
ssh-22.yaml
configurations > services > ssh-22.yaml
configurations > services > ssh-2222.yaml
apiVersion: "v1"
protocol: "ssh"
address: ":22"
description: "SSH interactive"
commands:
- regex: "^ls$"
handler: "Documents Images Desktop Downloads .m2 .kube .ssh .docker"
- regex: "^pwd$"
handler: "/home/"
- regex: "^uname -m$"
handler: "x86_64"
- regex: "^docker ps$"
handler: "CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES"
- regex: "^docker .*$"
handler: "Error response from daemon: dial unix docker.raw.sock: connect: connection refused"
- regex: "^uname$"
handler: "Linux"
- regex: "^ps$"
handler: " PID TTY TIME CMD\n21642 ttys000 0:00.07 /bin/dockerd"
- regex: "^(.+)$"
handler: "command not found"
serverVersion: "OpenSSH"
serverName: "ubuntu"
passwordRegex: "^(root|qwerty|Smoker666|123456|jenkins|minecraft|sinus|alex|postgres|Ly123456)$"
deadlineTimeoutSeconds: 60
apiVersion: "v1"
protocol: "ssh"
address: ":2222"
description: "SSH interactive ChatGPT"
commands:
- regex: "^(.+)$"
plugin: "OpenAIGPTLinuxTerminal"
serverVersion: "OpenSSH"
serverName: "ubuntu"
passwordRegex: "^(root|qwerty|Smoker666|123456|jenkins|minecraft|sinus|alex|postgres|Ly123456)$"
deadlineTimeoutSeconds: 60
plugin:
openAPIChatGPTSecretKey: "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"
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL AZURE
zsh
{"commands":1,"level":"info","msg":"Init service ssh","port":":2222"}
{"banner":"8.0.29","level":"info","msg":"Init service tcp","port":":3306"}
^C
(ney㉿kali)-[~/Documents/beelzebub-main]
$
History restored
(ney㉿kali)-[~/Documents/beelzebub-main]
$
Live Share
Ln 8, Col 19 Spaces: 2 UTF-8 LF YAML Analysis Tools Missing Go Update Avail
Ln 14, Col 1 Spaces: 2 UTF-8 LF YAML Analysis Tools Missing Go Update Availa

Source Code


tcp-3306.yaml - beelzebub-main - Visual Studio Code
View Go Run Terminal Help
openai-gpt_test.go
openai-gpt.go 1
ssh-2222.yaml
tcp-3306.yaml
configurations > services > ! tcp-3306.yaml
1 apiVersion: "v1"
2 protocol: "tcp"
3 address: ":3306"
4 description: "Mysql 8.0.29"
5 banner: "8.0.29"
6 deadlineTimeoutSeconds: 10
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL AZURE
zsh
{"commands":1,"level":"info","msg":"Init service ssh","port":":2222"}
{"banner":"8.0.29","level":"info","msg":"Init service tcp","port":":3306"}
^C
(ney㉿kali)-[~/Documents/beelzebub-main]
$
* History restored
(ney㉿kali)-[~/Documents/beelzebub-main]
$
Live Share
Ln 1, Col 1
Spaces: 2
UTF-8
LF
YAML
Analysis Tools Missing
Go Update Avail


protocol_manager.go - beelzebub-main - Visual Studio Code
iew Go Run Terminal Help
openai-gpt_test.go
openai-gpt.go 1
protocol_manager.go
ssh-2222.yaml
protocols > protocol_manager.go
1 package protocols
2
3 import (
4 "beelzebub/parser"
5 "beelzebub/tracer"
6)
7
8 type ServiceStrategy interface {
9 Init(beelzebubServiceConfiguration parser.BeelzebubServiceConfiguration, tracer tracer.Tracer) error
10 }
11
12 type ProtocolManager struct {
13 strategy ServiceStrategy
14 tracer tracer.Tracer
15 }
16
17 func InitProtocolManager(tracerStrategy tracer.Strategy, strategy ServiceStrategy) *ProtocolManager {
18 return &ProtocolManager{
19 tracer: tracer.Init(tracerStrategy),
20 strategy: strategy,
21 }
22 }
23
24 func (pm *ProtocolManager) SetProtocolStrategy(strategy ServiceStrategy) {
25 pm.strategy = strategy
26 }
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL AZURE
zsh
{"commands":1,"level":"info","msg":"Init service ssh","port":":2222"}
{"banner":"8.0.29","level":"info","msg":"Init service tcp","port":":3306"}
^C
(ney㉿kali)-[~/Documents/beelzebub-main]
$
* History restored
(ney㉿kali)-[~/Documents/beelzebub-main]
$
Live Share
Ln 1, Col 1
Tab Size: 4
UTF-8
LF
Go
Analysis Tools Missing

Deployment Strategies



Project Security & Maintenance

- Regular Updates and Patching
- Isolated Environment
- Monitoring and Intrusion Detection
- Data Protection and Privacy
- Access Control
- Regular Backups
- Incident Response Plan
- Continuous Monitoring and Testing
- Collaboration with Security Community

Module 1

Objective :

- The honeypot system's implementation, the honeypot's configuration, and real-time attack response.

Implementation :

- Selecting and configuring various types of honeypots based on your project's objectives and target systems.
- Deploying virtual machines or containers that mimic real systems, services, or applications.
- Scope encompasses implementing security measures to isolate the honeypots from the production environment.
- Ensuring regular updates and patching of the honeypot infrastructure to maintain authenticity and security.

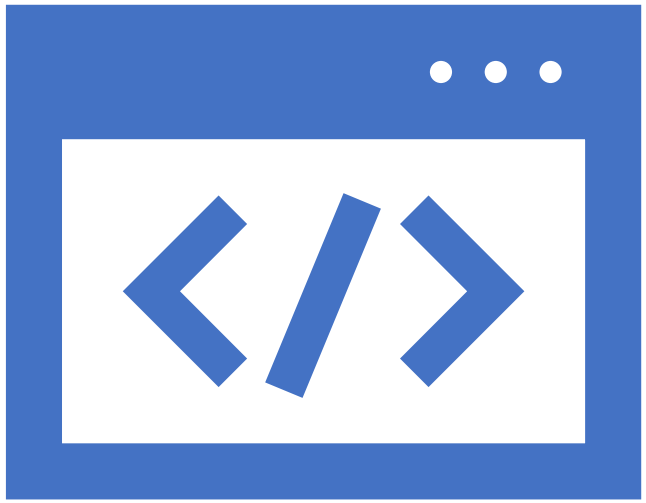
Module 2

- Objective:
  - Integrating the Chat-GPT API for the project and working with the team to create the honeypot.
- Implementation:
  - Integration of Chat-GPT API
  - Learnt and Implemented Chat-GPT API functionality, capabilities, and documentation.
  - Developed the necessary code and configurations to integrate the Chat-GPT API seamlessly into the project's infrastructure.
  - Tested and validated the integration to ensure proper functionality.

Module 3

- Objective:
  - Evaluating the honeypot system, testing the system, identifying technical issues, and conducting thorough testing before deployment.
- Implementation:
  - Conducted various types of testing, such as functional testing, performance testing, security testing, and penetration testing.
  - Identified and resolved any technical issues or vulnerabilities discovered during testing.
  - Validated the system's ability to handle various attack scenarios and interactions with attackers.

Module -4



- Objective:
  - Gathering and analysing data, figuring out the goals of the attacker, displaying the data, and presenting the results.
- Implementation:
  - Collect and capture relevant data from interactions with the attackers within the honeypot system.
  - Analyze the gathered data to extract meaningful insights and patterns.
  - Data Visualization.
  - Attacker Goal Identification

Future Aspects

Machine Learning for Threat Analysis

Behavior Analytics and Anomaly Detection

Threat Intelligence Integration

Collaboration and Information Sharing

Cross-Platform Honeypots

Publication Details

- Paper Title: Engaging Attackers with Highly Interactive Honeypot System Using ChatGPT
- Conference Name: 7th International Conference on Computing, Communication, Control and Automation (ICCUBEA-2023)
- Submission Status: Submitted

Conclusion

- Project offers innovative honeypot system using ChatGPT for engaging attackers and improving cybersecurity.
- Comprehensive security plan includes updates, isolation, monitoring, data protection, access control, and incident response.
- Deployment strategies focus on network isolation, virtualization, configuration management, scalability, logging, and continuous integration.
- Future scope includes advanced deception, improved language processing, machine learning for threat analysis, threat intelligence integration, collaboration, and cross-platform honeypots.
- Project aims to gather threat intelligence, enhance incident response, and foster cybersecurity collaboration.
- Ongoing innovation contributes to a safer digital landscape.

References

- [1] Huang, X., & Zhao, S. (2021). Chatbot-based honeypot for phishing detection. Journal of Computer Virology and Hacking Techniques, 17(3), 257-268. doi: 10.1007/s11416-020-00448-5.
- [2] De Lucia, E., & Zanero, S. (2020). Creating a dynamic honeypot with chatbots. In Proceedings of the 2019 Workshop on Cyber-Physical Systems Security and Privacy (pp. 19-24). ACM. doi: 10.1145/3322518.3323883.
- [3] C. Sun et al. (2022). "Application of Artificial Intelligence Technology in Honeypot Technology." In 2021 International Conference on Advanced Computing and Endogenous Security (ACES), Nanjing, China (pp. 01-09). IEEE. doi: 10.1109/IEEECONF52377.2022.10013349.
- [4] M. Tsikerdekis et al. (2018). "Approaches for Preventing Honeypot Detection and Compromise." In 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece (pp. 1-6). IEEE. doi: 10.1109/GIIS.2018.8635603.
- [5] B.-X. Wang, J.-L. Chen, & C.-L. Yu (2022). "An AI-Powered Network Threat Detection System." IEEE Access, 10, 54029-54037. doi: 10.1109/ACCESS.2022.3175886.
- [6] J. Franco, A. Aris, L. Babun and A. S. Uluagac, "S-Pot: A Smart Honeypot Framework with Dynamic Rule Configuration for SDN," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 2818-2824, doi: 10.1109/GLOBECOM48099.2022.10000682.
- [7] M. S. Tok, M. Dener and M. Demirci, "Processing Honeypot Logs with Big Data and Data Visualization via Hadoop- Power BI Integration," 2022 15th International Conference on Information Security and Cryptography (ISCTURKEY), Ankara, Turkey, 2022, pp. 49-54, doi: 10.1109/ISCTURKEY56345.2022.9931797.

Thank You