

SYNOPSIS

## Project Name :

Honeypot Deployment for Cyber Threat Detection and Deception

*BaitBox*

## Submitted To:

* Mentor Name: Mr. Asheesh Tiwari
* Designation: Associate professor
* Department: [Department of Computer Science]

## Objective:

**BaitBox**, is a honeypot that, stands out by layering a legitimate site, an attacker site, and a honeypot to simulate a realistic environment for capturing hacker activity. Unlike traditional setups, it uses canary tokens for real-time alerts on attacker interactions, enhancing immediate threat detection. With a simulated SSH environment via Paramiko, BaitBox engages attackers with realistic fake command executions, creating a highly effective tool for in-depth threat analysis and proactive cyber defense.

## Scope:

* The project will involve setting up a legitimate site on port 80, an attacker site on port 8080, and a honeypot on port 5000 to interact with attackers. Legitimate ssh server is working on port 22 which is restricted for outsider
* It will focus on capturing and simulating attack attempts, including SSH login attempts, command execution, and file access.
* The honeypot will simulate a fake file system with files like `secret.txt`, `config.json`, and others to lure attackers.
* The system will include canary tokens for real-time alerting when an attacker interacts with the fake files.
* The project will not involve real-world production systems, but rather a controlled environment for observing and analyzing attacks.

# Methodology:

* 1. Technologies Used: AWS, Python, Apache/Nginx web servers, Paramiko (for SSH honeypot), Canary Tokens, and Real-Time Monitoring.
  2. Programming Languages: Python will be used to set up the honeypot and monitor attacker behavior. HTML/CSS will be used for creating the fake attacker website.
  3. Tools:
* Paramiko: For simulating SSH connections and handling fake commands.
* Canarytokens.org: For creating and embedding canary tokens into the honeypot’s fake files.
* Web Servers: Apache or Nginx will serve the legitimate and attacker sites.
* Python Scripts: For managing the honeypot, logging malicious activity, and interacting with the canary tokens.

# Proposed System:

The proposed system is a honeypot deployment that simulates multiple layers of deception:

1. **Legitimate Site on port 80:** Serves real users.
2. **Attacker Site on port 8080:** Deceptive site to lure attackers into interacting with it.
3. **Honeypot on port 5000:** Simulates a vulnerable system with a fake file system to trap attackers.
4. **Real-Time Monitoring:** Provides notifications via canary tokens when an attacker interacts with the fake files.

# Features:

1. **Multi-Site Configuration:** Hosts legitimate and deceptive sites on separate ports (80 for legitimate, 8080 for attackers).
2. **Fake File System:** The honeypot offers a fake file system with mock files like `secret.txt` and `credentials.json`.
3. **SSH Honeypot:** Simulates SSH commands to lure attackers into believing they have access to a real system.
4. **Real-Time Alerts:** Uses canary tokens to alert the administrator when an attacker interacts with the honeypot.
5. **Logging and Monitoring:** Tracks attacker IP addresses, commands executed, and files accessed for analysis.

# Implementation Plan:

### Phase 1: Set up Legitimate Site

* + Set up a simple web server (Apache or Nginx) on port 80 for serving legitimate content.

### Phase 2: Create Attacker Site

* + Implement a deceptive site on port 8080 that mimics the legitimate site.

### . Phase 3: Develop Honeypot System

* + Set up the honeypot listening on port 5000 using Python and Paramiko to simulate a vulnerable SSH service.
  + Create a fake file system within the honeypot.

### . Phase 4: Integrate Real-Time Monitoring with Canary Tokens

* + Generate and embed canary tokens in the honeypot files to track interactions.
  + Set up alerts to notify when malicious activity occurs.

### . Phase 5: Testing and Evaluation

* + Simulate attack attempts to verify the system’s effectiveness in capturing malicious activities.

# Team Members:

* Ritanshu Sharma (2215300032)
* Rajat Yadav (2215300031)
* Shikhar Gupta (2215300033)
* Shraddha Purwar (2215300035)
* Manasvi Bansal (2215300026)

# Resources Required:

1. **Software:**

* Python, Apache/Nginx web servers, Paramiko library, Canarytokens.org.
* Email service for real-time alerts. 2. **Hardware:**
* Virtual Machines or Cloud Instances for hosting the web servers and honeypot.

### 3. Other Resources:

* Network configuration for port forwarding and external access.
* Access to a server or cloud environment for deploying the sites.

# References:

1. Paramiko Documentation
2. [Canarytokens](https://canarytokens.org/)
3. SSH Honeypot with Paramiko
4. AWS Documentation
5. [Youtube](https://www.youtube.com/)

# Expected Outcomes:

By the end of this project, we expect to:

1. Successfully deploy a working honeypot that simulates SSH login attempts and fake file systems.
2. Capture real-time attacker activity using canary tokens.
3. Generate useful logs and insights about attack behavior, such as IP addresses, attempted commands, and file access.
4. Develop a comprehensive understanding of attack patterns to improve defensive strategies.

**Project Supervisor:** Mr. Asheesh Tiwari **Associate professor**

(Department of computer science)

# Conclusion:

This project will provide valuable insights into cybersecurity, particularly in detecting and analysing attack methods using honeypots. The multi-layered approach, including a legitimate site, deceptive attacker site, honeypot, and real-time monitoring with canary tokens, offers an innovative and effective way to capture malicious activities and understand the tactics used by attackers. It will help enhance the security posture of systems and provide crucial information for improving defensive measures.

### Rahul Shandilya Asheesh Tiwari

(Assistant professor) (Associate professor)

**ABSTRACT**

This project, *"****Implementation of Honeypots for Server Security****,"* aims to enhance cybersecurity by deploying honeypots on Amazon EC2 instances. The increasing sophistication of cyber threats has highlighted the limitations of traditional tools such as firewalls and intrusion detection systems, which are often reactive and fail to provide insights into attacker behavior. By leveraging honeypots, this project offers a proactive solution to monitor, analyze, and understand malicious activities, contributing to stronger defense mechanisms.

The project involves creating a simulated server environment with multiple layers of interaction, including a legitimate site hosted on port 80 for genuine users and a deceptive site on port 8080 designed to lure attackers. Additionally, a honeypot environment is established on port 5000, alongside a simulated SSH service on port 2222, mimicking vulnerabilities to attract cyber intrusions. Canary tokens are embedded to trigger real-time alerts upon unauthorized access, ensuring immediate awareness and response.

The implementation phases include setting up the EC2 instance, configuring security rules for access, hosting legitimate and deceptive sites, and integrating honeypot and monitoring tools. The monitoring setup, enhanced by real-time alert mechanisms, provides visibility into attacker strategies such as credential guessing, port scanning, and file exploration. Results reveal the effectiveness of this system in capturing actionable intelligence while safeguarding actual server data.

This project delivers a robust honeypot framework that not only monitors server security but also gathers vital information on attack patterns. It serves as a valuable tool for cybersecurity teams, enabling them to develop adaptive strategies and improve system resilience. Future expansions include integrating machine learning for predictive analytics and extending the honeypot’s capabilities to cover a broader range of attack vectors, fostering a more comprehensive security infrastructure.