King Fahd University of Petroleum & Minerals
College of Computing and Mathematics
Information and Computer Science Department
ICS 344: Information Security – (241)
Project – Final

Offensive Security, Cyber Deception, and SIEM [SSH]

Name	ID	Section
Abdulrahman Aldossari	202036120	03
Abdullah Balqsim	201960830	03
Abdulghani Khayat	202039980	03

Table of Contents

OBJECTIVE

The project aims to build skills in cybersecurity by simulating attacks on vulnerable services, using honeypots for cyber deception, and analyzing security events with SIEM to understand attack patterns. Additionally, it explores defensive strategies to protect against similar threats.

1.GENERAL

1.A. Which service did you target and why?

We chose SSH as our targeted service because it is a critical service frequently used for remote system administration. Its widespread use makes it a popular target for attackers attempting to gain unauthorized access.

1.B. Which honeypot did you target and why?

Opencanary is our targeted honeypot because it provides a lightweight and easily deployable honeypot solution, though it is a low-interaction honeypot. It was chosen as an alternative after discovering that our initial honeypot was outdated and unmaintained.

1.C. Which SIEM did you use and why?

Wazuh is used as the SIEM tool for its comprehensive monitoring capabilities, and because it is open-source software. We installed it using Docker for ease of deployment and configuration.

2. SETUP AND COMPROMISE THE SERVICE

2.A. How did you configure Caldera?

Caldera was installed via cloning the GitHub repository and it requires Nodejs, Python, and Golang. Then it configured using the shipped abilities and custom scripts that grouped in "SSH Compromiser 1" adversary profile. This profile included a series of abilities such as brute-force attacks, privilege escalation, file and directory discovery, and SSH key extraction. The Sandcat agent was deployed on Metasploitable3, and the honeypot host machine to simulate the victim machine, enabling automated attack execution.

2.B. What MITRE ATT&CK TTPs did you select and how were they applied?

MITRE ATT&CK Phase	Test	Description	MITRE ATT&CK Technique ID
Initial Access	Brute Force SSH and Login	Hydra brute-force attacks to find SSH credentials.	T1110.001 – Account Discovery
Privilege Escalation	Using DirtyCow to gain privilege	Exploited DirtyCow vulnerability for root privilege gain.	T1068 – Exploitation for Privilege Escalation
Discovery	Nix File and Directory Discovery	Performed file and directory discovery within the system.	T1083 – File and Directory Discovery
Credential Access	Extract SSH Keys	Extracted SSH keys from the target machine.	T1003 – Credential Dumping

2.C. Which Kali tools did you use, and how were they integrated into the attack?

Tools like Hydra and Metasploit were used for brute-force attacks on SSH. Hydra complemented the automated Caldera TTPs by providing additional manual testing capabilities. While Metasploit helped in identifying possible threats in SSH.

2.D. What custom scripts did you use, and how were they integrated into the attack?

Custom scripts were used for attacks such as Hydra brute-force execution and extracting sensitive information from the target system like the RSA key. These scripts worked alongside Caldera's automation to refine the attack strategies.

2.E. Which tool/approach (Caldera, Kali tools, Custom scripts) was most effective at compromising the service?

Caldera's automation was highly effective for repetitive brute-force attacks, while manual tools like Hydra and custom scripts allowed for detailed customization and exploration of vulnerabilities.

2.F. Were there any limitations in one method that other methods overcame?

Caldera lacked detailed customization for brute-force parameters, which was addressed by using Hydra. Additionally, manual scripting provided greater flexibility for specific attack scenarios.

2.G. How successful were you in replicating a real-world attack scenario?

Using a combination of Caldera and manual tools, we successfully replicated realistic attack scenarios, including brute-force attacks and privilege escalation.

2.H. What challenges or bugs did you encounter during the setup and attack execution? How did you overcome these issues?

We faced some problems in setting up the environment including the tools used which are finding a compatible victim machine (resolved by switching to Metasploitable3). And Caldera set up issues due to dependency conflicts (resolved through troubleshooting and ensuring correct versions of python and Nodejs are installed). Lastly, Sandcat agent configuration issues (resolved by adjusting commands and troubleshooting permissions).

2.I. How easy or difficult was it to use Caldera compared to Kali tools and manual scripting? How did the level of automation in Caldera affect the overall process? Which approach required the most manual intervention or expertise?

Caldera provided ease of use and automation for repetitive tasks but required troubleshooting for the initial setup. Kali tools and manual scripting demanded more expertise and hands-on effort but offered greater control. The automation in Caldera improves the overall time required to compromise the service, and it needs less experience than manual tools.

3. SETUP AND COMPROMISE THE HONEYPOT

3.A. How did you configure Caldera?

Caldera was installed via cloning the GitHub repository and it requires Nodejs, Python, and Golang. Then it configured using the shipped abilities and custom scripts that grouped in "SSH Compromiser 1" adversary profile. This profile included a series of abilities such as brute-force attacks, privilege escalation, file and directory discovery, and SSH key extraction. The Sandcat agent was deployed on Metasploitable3, and the honeypot host machine to simulate the victim machine, enabling automated attack execution.

3.B. What MITRE ATT&CK TTPs did you select and how were they applied?

MITRE ATT&CK Phase	Test	Description	MITRE ATT&CK Technique ID
Initial Access	Brute Force SSH and Login	Hydra brute-force attacks to find SSH credentials.	T1110.001 – Account Discovery

3.C. Which Kali tools did you use, and how were they integrated into the attack?

Hydra was used for brute-force attacks on SSH. Hydra complemented the automated Caldera TTPs by providing additional manual testing capabilities.

3.D. What custom scripts did you use, and how were they integrated into the attack?

Custom scripts were used to simulate SSH credential attacks on the honeypot.

3.E. Which tool/approach (Caldera, Kali tools, Custom scripts) was most effective at compromising the service?

None, as the low-interactive nature of Opencanary prevented a successful compromise.

3.F. Were there any limitations in one method that other methods overcame?

Low-interactive honeypots like Opencanary could only log connection requests and could not simulate a full SSH service and shell for attacks.

3.G. How successful were you in replicating a real-world attack scenario?

The replication was incomplete due to the honeypot's limitations.

3.H. What challenges or bugs did you encounter during the setup and attack execution? How did you overcome these issues?

The initial honeypot was outdated and unmaintained which lead us to switch to Opencanary – based on python –.

3.I. How easy or difficult was it to use Caldera compared to Kali tools and manual scripting? How did the level of automation in Caldera affect the overall process? Which approach required the most manual intervention or expertise?

Caldera's automation was easier for basic tasks, but manual scripting provided better flexibility for testing specific vulnerabilities.

4. COMPARISON BETWEEN REAL SERVICE AND HONEYPOT (REALISM EVALUATION)

Include a detailed evaluation of time, resource usage, and realism in terms of responses to attack scenarios when the honeypot was subjected to the same attacks as the victim service environment. Discuss how closely the honeypot mimicked the real service and any differences observed.

MITRE ATT&CK Test	Tool	Real Service	Opencanary	Matching Analysis	Score	Test Score
SSH Version Test	ssh -v	SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-Zubuntu2.13	SSH-2.0-OpenSSH_5.1p1 Debian-4	Protocol Version: Same	1	0.33
				OpenSSH Version: 6.6.1p1/5.1p1	0	
				OS: Ubuntu/Debian	0	

Key Exchange Algorithms	ssh -Q kex	diffie-hellman-group1-sha1 diffie-hellman-group14-sha1 diffie-hellman-group-exchange-sha1 diffie-hellman-group-exchange-sha256 ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 curve25519-sha256@libssh.org	diffie-hellman-group1-sha1 diffie-hellman-group14-sha1 diffie-hellman-group-exchange-sha1 diffie-hellman-group-exchange-sha256 ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 curve25519-sha256@libssh.org	Presence of basic algorithms: support basic key exchange algorithms	1	0.66
				Outdated algorithms: includes older, weaker algorithms	0	
				Presence of Modern Algorithms: there are advanced algorithms	1	
SSH Port	nmap	22/tcp open	22/tcp open	Both identify correct open ports	1	1

The Reconnaissance similarity is 0.66 indicates that the honeypot partially mimicked the behavior of the real SSH service but with noticeable differences. This score suggests that while the honeypot successfully emulated some characteristics of the real service, such as open port detection and basic SSH protocol support, there were critical discrepancies in areas like supported algorithms and protocol versions.

5. VISUAL ANALYSIS WITH A SIEM DASHBOARD

5.A. How did you configure the SIEM to collect data from the SSH service and honeypot?

Wazuh was configured using Docker for ease of deployment. And its agents are deployed on each machine (metasploitable3 and Debian-honeypot) where it collected logs and monitored events from both the SSH service and honeypot.

5.B. What specific logs or events were forwarded to the SIEM?

Almost all of the MITRE ATT&CK are logged and forwarded to Wazuh.

5.C. What does the SIEM dashboard display about SSH and honeypot activity?

The dashboard displays alerts for evolution over time, rule levels by attack, and MITRE ATT&CK mappings for detected tactics.

5.D. Did the SIEM identify any patterns or anomalies?

It detects MITRE ATT&CK and high activity related to Privilege Escalation, Initial Access, Defense Evasion, and Persistence.

5.E. Which visualizations were most helpful for understanding the data?

The "MITRE attacks by tactic" and "Alerts evolution over time" charts provided valuable insights into attack patterns and activities.

5.F. What challenges or bugs did you encounter during the setup of SIEM? How did you overcome these issues?

First the deployment of Wazuh using pre-built binaries failed, so we switched to docker deployment since it has less dependencies to worry about.

5.F. Any findings observed? For example, differences in visual data between SSH service and honeypot?

There are differences in the attacks logged by the SIEM, which is the attack number and quantity, it is mainly because the honeypot is low-interactive and does not simulate the actual service.

6. Survey Questions

6.A. Based on your experience, what best practices would you recommend for future students working on a similar project?

Focus on automating repetitive tasks using tools like Caldera while learning manual methods with tools like Hydra for a deeper understanding of attack techniques, understand network configurations and permissions to avoid connectivity issues when using agents like Sandcat, and choose an actively maintained honeypot to ensure compatibility and effectiveness.

6.B. How much did you learn from this project? Provide a brief reflection on your experience. Do you recommend this project for use in future course cycles?

We learned how to use Caldera and what MITRE ATT&CK, as well as the usage of honeypots. Honestly, the type of project does not suit the course output, and it puts more emphasis on the tool's setup rather than the actual purpose of them and their usages.

6.C. What learning resources did you rely on during the project? Specify the exact platforms or materials you used, such as edX, Coursera, YouTube, official documentation, or other relevant sources.

Resources included MITRE ATT&CK documentation, official Caldera documentation, and online tutorials from YouTube.

6.D. Which task took the least/most time to execute?

The least is Running automated attacks with Caldera. While the most is troubleshooting setup and configuration issues.

6.E. Feel free to write down any technical details, observations, or feedback.

A more interactive honeypot would significantly improve the realism of attack scenarios. And using updated and maintained tools reduces setup challenges.