HONEYPOT PENTEST REPORT

Version 1.0

1/24/2024

Alexandru Gumaniuc
Corvin Wittmaack
Dimitri Vastenhout
Evaldas Narkevicius

Version Control

Author	*Version*	*Change/Addition*
Corvin Wittmaack	0.1	Front page, Version Control, Introduction
Corvin Wittmaack	0.2	Scope, Management Summary
Corvin Wittmaack	0.3	Methodology Chapter, Styled Doc, Created each chapter
Corvin Wittmaack	0.4	Architecture chapter
Corvin Wittmaack	0.5	Filled out results chapter.
Corvin Wittmaack	0.6	Completed Conclusion and recommendations.
Alexandru Gumaniuc	0.7	Added Information about “PyRDP” and “Rdpy”
Evaldas Narkevicius	0.8	Changed the language used to more official, some minor changes
Corvin Wittmaack	0.9	Fixed references Added results
Alexandru Gumaniuc Corvin Wittmaack	1.0	Final Adjustments

Table of Contents

I. Summary

Introduction

This report was created to explain, visualize, document and analyze the findings/exploits of the penetration group part of the CEH minor from NHL Stenden University of Applied Sciences, Emmen campus, regarding the assignment to digitalize a ships bridge with the help of a virtual environment and 3 honeypots within. The repository for that can be found in the references.

The main focus of this assignment was to support the research of the CEH minor from NHL Stenden in creating a virtual environment able to replicate a vulnerable system, as a bait for hackers targeting breaches in a vessels network. The old and instable systems of the vessels as of today are more than inviting for cyber criminals to infiltrate and manipulate such systems, leading to a great danger to the intercontinental transport system and supply chain. Testing was performed on the following honeypots: Cowrie, DDoSPot and PyRdp.

The pen testing report took place over a lengthened period and was done without any intent of harm, damage or any sort of malevolence. The purpose was to educate about the weaknesses of remote and networking systems.

Scope

Objective

The primary goal was to set up a personalized Honeynet virtual environment and penetrate this to assess the coverage of detection and response capabilities within the virtual environment. The test aimed to identify and classify the types of attack that a cyber-criminal uses, to evaluate the accuracy of the results of the logs and to understand the depth of the collected data it would capture after such attacks. In order to analyze the output generated by each of the honeypots, the research group integrated within the virtual environment a search engine called “Elasticsearch”, which aims to help in analyzing, filtering and sorting the output data.

Software Details

- **System type**: Virtual Machine on a Laptop/PC (using VMWare as a Hypervisor)
- **Operating System**: Ubuntu 20.04.6 LTS (Focal Fossa)
- **Specific hardware requirements:** 6-8 GB RAM and 40 GB storage memory (minimum)
- **Honeypot software:** Cowrie, DDosPot, PyRdp, Logstash, Filebeat, Kibana, Elasticsearch
- **Other software:** Docker, Docker-compose
- **IP Address:** 192.168.x.x
- **Domain Name:** N/A (tested on a private, local network)

**For further information, the project groups GitHub repository can be used as reference.*

In-Scope Testing Methods

The following components of Elastic Stack were included during the testing.

1. **Honeypot Interaction**: Understanding how different types of malicious traffic are logged and responded to.
2. **Data Logging**: Assessing the granularity and accuracy of logged data during an attack.
3. **Alert Mechanisms**: Testing the effectiveness and timeliness of any alert mechanisms in place.

Out-Of-Scope Testing Methods

To maintain the integrity of the test and the system, the following components and actions were considered out of scope:

1. **Physical attacks**: No physical tampering or direct hardware-based attacks on the laptop.
2. **Social Engineering**: No attempts were made to deceive individuals into divulging information or granting access related to the Honeynet system.
3. **Harm or destruct data**: Our intent was purely to observe and analyze, not to harm.

II. Management Summary

During this project's assignment, the group established a virtual environment with 3 honeypot docker containers to create a Honeynet. Moreover, another set of 4 docker containers were added to facilitate an effective logging and analyzation system for the output generated by the honeypots (all related to Elastic Stack). The use of common hacking techniques and methods lead then later to observing the victim machine and the logs to further analyze the content of the newly created logs.

General Summary

The group was able to successfully implement 3 different honeypots and ship all their generated logs to Elasticsearch. Moreover, the entire environment was configured to run in separate Docker containers, inter-connected between them. Every honeypot logs relevant data depending on its purpose, and reacts accordingly to the type of the attack the group used against the whole setup.

All of the logged data is persistent, providing mostly meaningful data, easy to filter, read and understand.

Disclaimer

Please note that the group has limited experience in penetration testing, and as a result, did not conduct any advanced tests.

Moreover, since the test was constrained by the time boundaries of the CEH minor, the group did not have the time an actual ethical hacker would have.

While a thorough set up and examination was conducted, the assessment may not cover all potential security concerns. For a more comprehensive evaluation, it is highly recommended to consult specialized professionals.

III. Methodology

In line with the guidelines presented by the EC-Council's CEH v12, the penetration testing approach adopted a systematic and comprehensive methodology. The aim was to provide a consistent and replicable testing framework, ensuring that every potential vulnerability is uncovered while respecting the boundaries of the defined Scope of the project.

1. Reconnaissance (Passive and Active):

Objective: To gather as much information as possible about the target honeypot system. This helps in identifying potential weak points and understanding the system's architecture.
Passive Techniques: Online search engines were used to identify honeypots matching the assignment's requirements based on their functionality.
Active Techniques: Network scanning.

2. Scanning & Enumeration:

Objective: To identify live hosts, open ports, services, and any running applications. This further aids in pinpointing potential entry points.
Tools Utilized: Nmap

3. Gaining Access (Exploitation):

Objective: To exploit identified vulnerabilities, trying to access the Honeynet system.
Tools and Techniques: Used Metasploit Framework, manual vulnerability exploitation based on findings from previous stages, and custom scripts.

4. Maintaining Access:

Objective: To understand if an attacker can create a persistent presence on the honeypot, emulating advanced persistent threats.
Methods Used: Explored potential backdoors, rootkits, and other persistence mechanisms.
(Note: In real-life assessments, this stage may require explicit permission, given its potentially intrusive nature.)

5. Analysis & Reporting:

Objective: To consolidate findings, evaluate the effectiveness of the virtual environment and its Honeynet in detection and response, and provide actionable recommendations.
Approach: Documented every attack vector tried, the response of the honeypot, evidence in the form of logs/screenshots, and potential mitigation measures.

Tools & Technologies:

Throughout the testing phases, a range of tools were used, both commercial and open source, ensuring the assessment's success.

Penetration Testing Tools

NMAP

Nmap, which is also referred to as "Network Mapper," is a utility that is freely accessible and utilized for network investigation and evaluating security. It is extensively employed by system and network administrators to carry out different activities such as network inventory, managing service upgrades, and monitoring the accessibility of hosts and services.
(*Nmap: The Network Mapper - Free Security Scanner*, 2017)

Metasploit

Metasploit, developed by Rapid7, is a widely utilized framework for penetration testing and exploitation. It serves as a valuable tool for security professionals, ethical hackers, and red teamers to evaluate the security of computer systems and networks. Offering an extensive array of resources and tools, Metasploit aids in the discovery, exploitation, and verification of vulnerabilities within targeted systems.

IV. Honeypot Setup and Architecture

Honeypot Architecture

To effectively evaluate the capabilities of the honeynet environment against potential cyber threats, a controlled virtual environment was set up using VMware Workstation Pro. This simulated environment not only allowed for a safe testing space but also ensured that the potential impact on other external networks or systems was nullified.

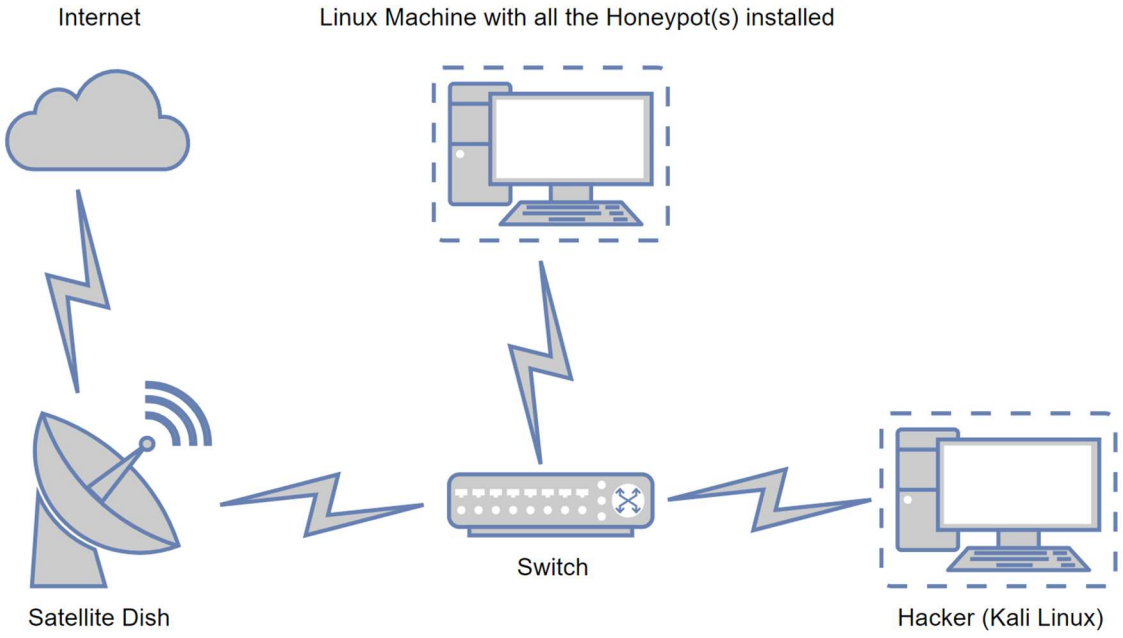

Figure 1: Network structure

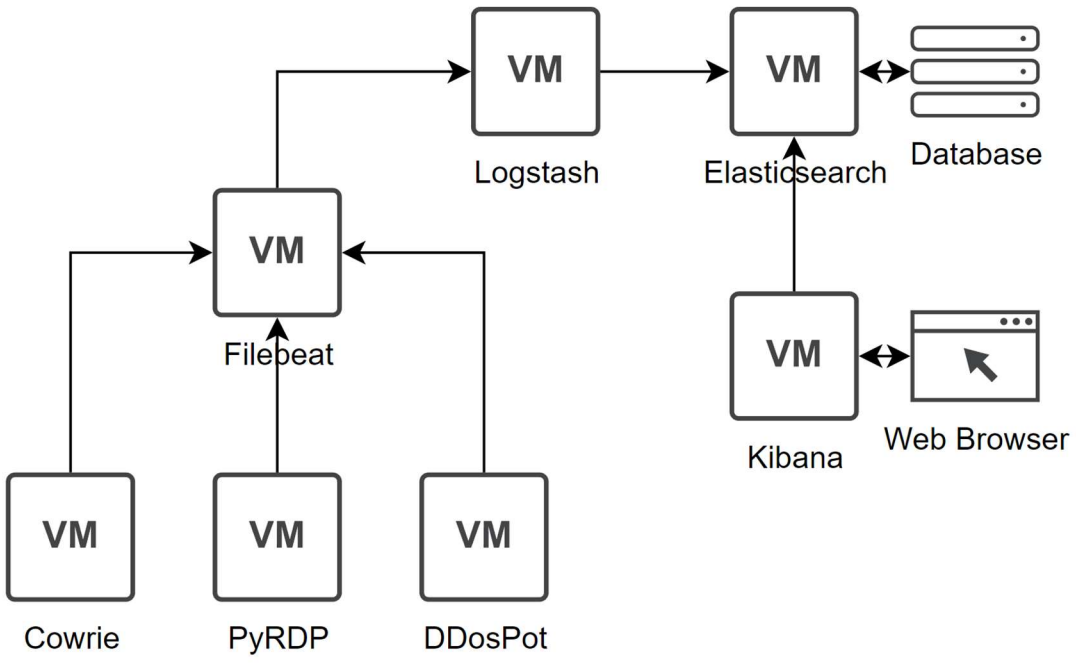

Figure 2: Honeypot network structure

Virtualized Network Architecture

The virtual environment comprised two main components:

Kali Linux / Ubuntu (Attacker VM)

Purpose: This virtual machine, equipped with Kali Linux (or an Ubuntu server with pen-testing tools installed), acted as the attacker's station. Kali Linux, a renowned penetration testing and ethical hacking distribution, provides an array of tools to simulate real-world attack scenarios.

Configuration:

- **Operating System**: Kali Linux (version 2023.3 was used for this assignment) / Ubuntu 20.04
- **Allocated Memory**: 4GB RAM
- **Network Adapter**: Bridged or NAT mode to facilitate communication with the honeynet.

Ubuntu Server (Victim VM)

Purpose: This virtual machine was dedicated to a honeynet which consists of 3 honeypots and 4 other docker containers, all running in a docker network. This will then act as the target for the attacker's actions. It was configured to detect, log, and respond to cyber threats.

Configuration:

- **Operating System**: Debian 11 x64
- **Cowrie Version**: 22.04.0
- **DDoSPot Version:** 1.0
- **RDP Version**: N/A
- **Allocated Memory**: 6-8GB RAM
- **Network Adapter**: Bridged or NAT mode to ensure seamless network interaction with the Attacking machine.

Simulated Network Gateway

A unique aspect of the simulation was the inclusion of a "simulated" satellite dish. While the dish itself was virtually represented, it acted as a crucial network component with specific functionalities:

- **Role as a Router**: The satellite dish played the role of a gateway or router, managing the traffic between the virtual machines and the simulated internet. This added an extra layer of realism to the test environment, mimicking potential latency or data constraints typical of satellite communications.
- **Internet Connection**: While the environment was contained, the satellite dish provided a controlled connection to the internet, enabling updates, downloads, and external interactions when necessary.

Network Flow

The Kali Linux VM, acting as the attacker, initiates network interactions or attacks targeting the honeynet VM.

The honeynet system detects logs, and potentially responds to these interactions based on its configuration and detected threat levels.

All network communications between the VMs, as well as any external interactions, pass through the simulated satellite dish, which manages and routes the traffic accordingly.

Security Precautions

Given the nature of the tests and the potential risks associated with penetration testing tools:

- All external communications, aside from necessary updates or specific tests, were minimized to ensure the containment of the virtual environment.
- Regular snapshots were taken of both VMs to enable quick rollbacks in case of any unintended consequences or to reset the environment between different test scenarios.

Honeynet Components

The build honeynet consists of a total of three honeypots which all focus on a different security aspect, furthermore, countless visualization options using the Elastic Stack, animated live attack maps and lots of security tools to further improve the deception experience. To reach its goal of running as many tools as possible simultaneously and thus utilizing the host's hardware to its maximum it uses docker and docker-compose.

The following docker images are integrated:

Cowrie

A SSH and telnet honeypot made to log brute force attacks and shell interactions done by the attacker.

DDoSPot

A platform for tracking and monitoring UDP-based DDoS attacks for DNS, NTP, SSDP, CHARGEN, and Random/mock UDP servers.

PyRdp

A man in the middle (*mitm*) tool meant to intercept RDP traffic between an attacked and a windows server running RDP (acting as a honeypot).

Logstash

Logstash is a free and open server-side data processing pipeline that ingests data from a multitude of sources, transforms it, and then sends it to your favorite "stash."

Filebeat

Filebeat is a lightweight tool that simplifies the process of collecting and centralizing logs and files from various sources such as security devices, cloud services, containers, hosts, and operational technology (OT) systems. It efficiently forwards these logs and files to a centralized location, making it easier to manage and analyze them.

Kibana

Kibana can run data analytics at speed and scale for observability, security, and search. Powerful analysis on any data from any source, from threat intelligence to search analytics, logs to application monitoring, and much more.

Elasticsearch

Elasticsearch is a distributed search and analytics engine built on Apache Lucene. Since its release in 2010, Elasticsearch has quickly become the most popular search engine and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases.

Partially Integrated

Due to an error in the execution when connecting to the IP. This specific Remote Desktop honeypot is not counted as a component of its own.

Rdpy

A honeypot designed to look like a normal windows Remote Desktop Connection, which reads users data as they use it, because they think they are currently on a normal RDP connection.

V. Technical Findings

Cowrie

Cowrie is a medium to high interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker. In medium interaction mode (shell) it emulates a UNIX system in Python, in high interaction mode (proxy) it functions as an SSH and telnet proxy to observe attacker behavior to another system (Cowrie/Cowrie: Cowrie SSH/Telnet Honeypot Https://Cowrie.readthedocs.io, 2023).

As soon as Cowrie detects SSH probes, it simulates the behavior of an SSH server, to get the attacker to reveal his hacking techniques. Cowrie records all interactions, including login attempts, commands issued, and file transfers, while also emulating a realistic environment to lure attackers to reveal further exploration. This allows researchers, or this CEH group, to gather valuable intelligence on the probing techniques, tools, and intentions.

Additionally, Cowrie implements various countermeasures, such as delaying responses or deploying deception techniques, to stretch the attack and efforts from the attacker to the maximum.

In order to test Cowrie, the group tested specifically with the Nmap testing tool to search for weaknesses in the scope of open ssh or/and telnet ports.

After testing the nmap command “sudo ssh 0.0.0.0 -p 2222”, the terminal shows that a functional connection can be established and that ssh key is being published. Furthermore, asks the console if the attacker wants to continue with the connection. Once typed “yes”, the attacker is asked to input the password to connect to the ssh honeypot and from there, all the commands are logged. Similarly, establishing telnet connection by using “sudo telnet 0.0.0.0 2323” the connection is immediately logged, and the credentials used can be found in the Kibana data view.

DDoSPot

DDoSPot is a honeypot "platform" for tracking and monitoring UDP-based Distributed Denial of Service (DDoS) attacks. The platform currently supports the following honeypot services/servers in form of relatively simple plugins called pots (*Aelth/Ddospot: NTP, DNS, SSDP, Chargen and Generic UDP-Based Amplification DDoS Honeypot*, 2024):

1. DNS server
2. NTP server
3. SSDP server
4. CHARGEN server
5. Random/mock UDP server

Nmap serves as a reconnaissance tool, used to find vulnerabilities within network infrastructures. DDoSPot specializes in collecting distributed denial-of-service (DDoS) attacks, a notorious form of cyber assault aimed at target systems to make them unusable to the owner of that machine and or network.

Therefore, when the Nmap command “sU” is executed, it initiates a UDP scan, probing for open UDP ports on the target system. DDoSPot reacts by analyzing the scan results to identify potential vulnerabilities or exposed services on the target. If vulnerabilities are identified, DDoSPot is going to prepare for its use, such as compromised machines within a botnet, to launch a coordinated distributed denial-of-service (DDoS) attack against the vulnerable services or systems uncovered by the Nmap scan. This attack aims to overwhelm the target with a flood of UDP traffic, exploiting any weaknesses discovered during the reconnaissance phase.

In order to test the DdoSPot, the group tested specifically with the Nmap testing tool to search for weaknesses in the scope of open UDP ports.

After testing the nmap command "-sU", the terminal shows the following ports:

- 161 | udp open | filtered snmp
- 631 | udp open | filtered ipp
- 5353 | udp open | filtered zeroconf

Three ports have been found and with them, the status and service of the given ports.

On the Elasticstack side, the command created a log and collected data from the attacker like the timestamp and message.

RDP Honeypots

The group used two honeypots meant for analyzing RDP attacks. Both honeypots were tested using an external Windows machine, connecting to either honeypot through RDP.

Rdpy

The honeypot had an issue when used. While the remote desktop was supposed to work as intended, by showing a working desktop screen and offer an interface on which the group was able to conduct further penetration form the attacking kali machine, the setup failed after a desktop was visible on the screen. The screen crashed a few seconds into the connection showing only a partial login page before the user was removed and an error was given. The small amount of information given during the initial connection was used and displayed in the elastic stack.

This issue was known amongst the lecturers and no further steps were therefore required.

PyRDP

This man in the middle tool requires an additional windows machine running RDP. The tool is configured to intercept any connection meant for the real server and record it. Afterwards, the tool stores and logs all the actions the attacker takes, such as the keys pressed, passwords attempts, and even provides screen recording videos of the actual sessions. These can be used to see how the attacker was moving the mouse around the screen and what they would click on.

The connections, simple and user specific, worked effortlessly and showed a locked display for the attacker to fiddle with. The simple connection led to a fully controlled interface for the attacker to search around and play for, every action was logged. The user specific connection on the other side, recorded attempts to bypass either the username and/or the password of the victim machine. Therefore, providing insights into the possible tools and strategies of this attack (GoSecure/Pyrdp: RDP Monster-In-The-Middle (Mitm) and Library for Python with the Ability to Watch Connections Live or after the Fact, 2024).

Conclusion

The penetration testing campaign aimed at the tailored honeynet system within the controlled virtual environment has been concluded. The group used specialized tools such as VMware Workstation Pro (*Windows vm | Workstation pro | VMware*, 2024), used to simulate a network. Other tools such as Kali Linux and its pen-testing tools, were used to attack the Honeynet system, acting as the target. The attacker and the target were connected through a virtual gateway, simulating a satellite dish.

The system successfully simulated all the required components in separate virtualized containers. Moreover, it reacts in an expected way depending on the type of the attack, logging all the data in a meaningful and understandable format.

The penetration test has provided valuable insights, despite the relative inexperience of the group and time constraints of the CEH minor course. While the group may not have an extensive amount of information and skills in the cyber security and ethical hacking field, two crucial findings have emerged from our study.

Recommendation

To address possible upcoming issues with future changes upon the project, the group recommends the following things to keep in mind:

- Fix the issues of the remote desktop honeypot to make it easier to implement it in projects, such as this, in order to simulate a real-life scenario and collect data from these attacks. Moreover, the honeypot is using Python 2.7, which is deprecated.
- Keep in mind the versions and maintenance of the honeypots, projects will be stopped in use and/or get major configurations which then need to be updated in the repository that was provided by this group.
- The videos the group provide can be used as a reference on how the honeynet is supposed to behave in case of an issue due to a faulty hosting machine or updates on any of the used containers.

By adhering to these guidelines, the honeynet will adopt a more convincing guise as a genuine target for hackers, thus enhancing its effectiveness as a tool for luring and gainiattacks. Bys into the techniques employed by hackers.

References

aelth/ddospot: NTP, DNS, SSDP, Chargen and generic UDP-based amplification DDoS honeypot. (2024). GitHub. https://github.com/aelth/ddospot

cowrie/cowrie. (2020, October 27). GitHub. https://github.com/cowrie/cowrie

Nmap: the Network Mapper - Free Security Scanner. (2017). Nmap.org. https://nmap.org/#:~:text=Nmap%3A%20Discover%20your%20network,monitoring%20host%20or%20service%20uptime.

Metasploit | Penetration Testing Software, Pen Testing Security | Metasploit. (2019). Metasploit. https://www.metasploit.com/

Docker. (2018). *Enterprise Application Container Platform | Docker*. Docker. https://www.docker.com/

Windows VM | Workstation Pro | VMware. (2024, January 15). VMware. https://www.vmware.com/products/workstation-pro.html

GoSecure/pyrdp: RDP monster-in-the-middle (mitm) and library for Python with the ability to watch connections live or after the fact. (2024). GitHub. https://github.com/GoSecure/pyrdp

Repository

Geniools/ceh-elastic-stack: A honeynet running in docker containers and saving the logs in elasticsearch using logstash and filebeat. (2024). GitHub. https://github.com/Geniools/ceh-elastic-stack

Appendix

DDosPot Result

Figure 3 Elasticstack DdosPot Log

Cowrie Results

```
honey@honey-virtual-machine:~/ceh-elastic-stack$ sudo ssh 0.0.0.0 -p 2222
The authenticity of host '[0.0.0.0]:2222 ([0.0.0.0]:2222)' can't be established.
ED25519 key fingerprint is SHA256:6/f0v6FvSkXtJjocEflaFR4eutdxAOuhHLHiLH+XZ4I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[0.0.0.0]:2222' (ED25519) to the list of known hosts.
root@0.0.0.0's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~# la
-bash: la: command not found
root@svr04:~# ls
root@svr04:~#
root@svr04:~#
root@svr04:~#
root@svr04:~# ls
root@svr04:~# la
-bash: la: command not found
root@svr04:~#
root@svr04:~# exit
Connection to 0.0.0.0 closed.
```

Figure 4 Cowrie Attacker View SSH

Figure 5Elasticstack Cowrie Log For SSH

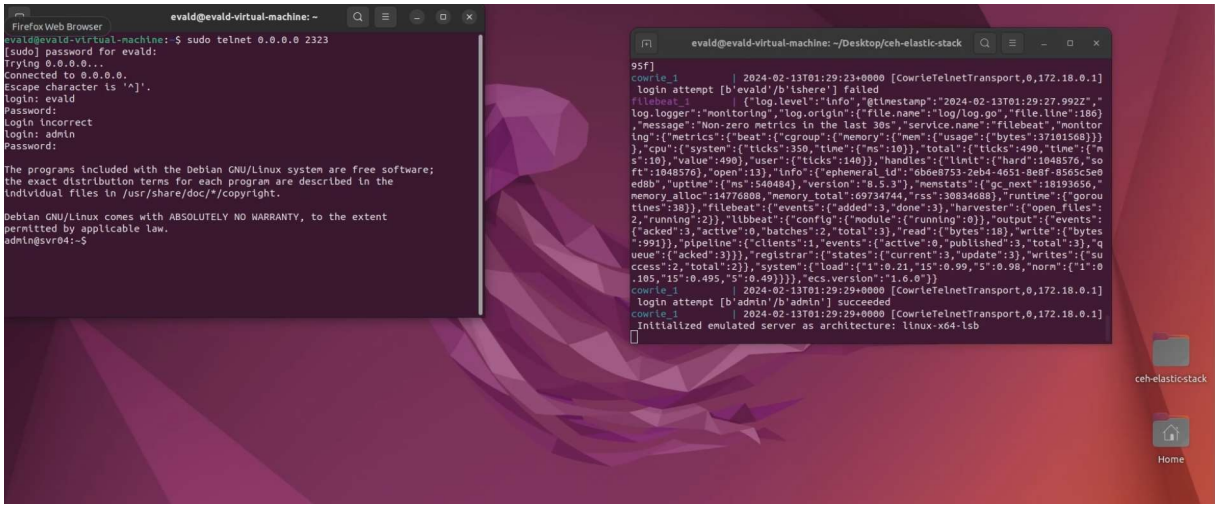

Figure 6 Cowrie Attacker View Telnet

Figure 7 Elasticstack Cowrie Log Telnet