ERSC ENGENHARIA DE REDES E SISTEMAS DE COMPUTADORES ESTG-IPVC

Hackers Activity on the Internet

Submitted on 10/06/2023
authored by
Célio Pina (celiopina@ipvc.pt), Rúben da Luz(rubenluz@ipvc.pt)
supervised by
Prof. Hugo Almeida (halmdeida@estg.ipvc.pt) and Prof. Pedro Pinto (pedropinto@estg.ipvc.pt)

***Index Terms*—Hacking Activity, Security, Network, AWS Cloud, Cyber Threats.**

***Abstract*—As technological advancements progress and our reliance on the Internet deepens, the rise in hacking activity has emerged as a pressing concern in contemporary times. This research aims to analyze the activities of hackers on the internet and assess the effectiveness of using honeypots in the AWS cloud as a proactive security measure, attracting hackers and gathering valuable data about their activities. By deploying and configuring two low-in honeypots in different regions in the AWS cloud, it becomes possible to monitor and analyze the behavior of hackers, identify attack patterns, and develop more effective security solutions. This study aims to contribute to the understanding of hackers activity and enhance the protection of systems and networks against cyber threats.**

I. Introduction

These days, with technological advancement and increasing dependence on the Internet, hacker activity on the network has become a constant concern. Hackers have the ability to exploit vulnerabilities and compromise systems and networks, causing significant harm to businesses, government institutions, and even ordinary individuals. With the constant evolution of technologies and the sophistication of cyber attacks, understanding the activity of hackers on the Internet has become fundamental for the development of effective security solutions.

Current times show us that hackers are constantly adapting and improving their techniques to circumvent existing security measures. This results in a digital arms race, where cybersecurity professionals and organizations need to stay one step ahead of hackers to protect their systems and sensitive data. In this context, studying the activity of hackers on the Internet becomes crucial to understand their motivations, methods and trends.

We seek to configure T-Pot honeypots in the AWS cloud to monitor and record attacks coming from the Internet. Honeypots, security systems designed to simulate vulnerabilities and attract hackers, will allow you to collect valuable information about the activities of these intruders. By analyzing the collected data, it will be possible to identify the most common types of attacks, the geographical origins of the hackers, the techniques used and other relevant factors.

Based on this information, it will be possible to develop more effective security solutions, targeting the most relevant threats and the most susceptible geographic areas. Furthermore, analysis of hackers behavior patterns will provide valuable insights for improving cybersecurity strategies.

Therefore, this project will not only contribute to the advancement of knowledge about the activity of hackers on the Internet, but will also provide practical subsidies for the protection of systems and networks against cyber-attacks. Understanding hackers threats and tactics is a critical step towards strengthening cybersecurity and ensuring the integrity and confidentiality of information in the ever-evolving digital world.

II. Related Work or State-of-art

A. Honeypot

A honeypot is a security mechanism designed to simulate vulnerabilities and attract potential attackers. It acts as a decoy system, luring hackers and capturing information about their activities. Unlike specific security

solutions, a honeypot is a versatile and adaptable tool that does not aim to solve a particular security problem, instead, it serves as a general technology with broad applications in areas such as network forensics and intrusion detection.

They enable security researchers to observe and document the activities of malicious users on a compromised computer without alerting the attacker to their presence. This covert monitoring allows for the collection of valuable intelligence on the attacker's actual strategies and techniques. Honeypots can be classified into two types based on interaction level: low-interaction and high-interaction:

Low-interaction honeypots emulate operating system and port services, providing quantitative information about attackers' motives and techniques.

High-interaction honeypots consist of real applications, operating systems, and devices with dummy data, engaging attackers for extended periods.

Both types offer valuable insights into attackers behaviors and aid in developing effective security measures.

B. T-Pot honeypot Sensors

Dionaea: C and Python based low-interaction honeypot sensor which logging capabilities offer compatibility with log.json, hpfeeds, Fail2Ban and log.sqlite.

Cowrie: Cowrie is a medium SSH honeypot interaction that provides a Debian 5.0-based fake file system, enabling user to include and remove files as their wish.

Honeytrap: Honeytrap is an extensible and open-source framework for honeypots to be run, tracked and maintained.

Tanner: A remote data review and classification service to analyze and compose the response of HTTP queries, then served by Super Next Generation Advanced Reactive Honeypot (SNARE). When offering responses to SNARE, TANNER utilizes several program vulnerability style emulation approaches.

Conpot: Conpot is a low interactive honeypot engineered to be easy to install, change and extend. It encompasses with a set of standard industrial control protocols, capable of emulating complex infrastructures to persuade a competitor that he has just discovered a massive industrial complex.

III. Methodology

A. Experiment Approach

Our primary goal is to setup up two honeypots in different locations on the AWS cloud service, and collect a certain amount of data, in which we will develop a study about the origin, frequency, location, etc, related to attacks, attack techniques, payloads, that are captured by the kibana framework as shown in fig.1, and the develop ways to defend a real server against this possible threats.

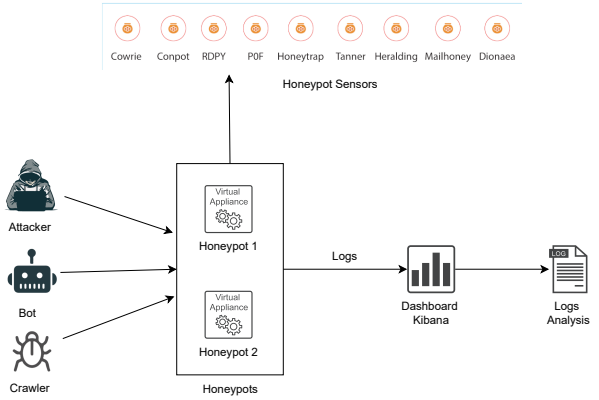


Fig. 1. Research Workflow

B. T-Pot Honeypot

T-POT is a user-friendly open source honeypot system based on the Debian 11 (Bullseye) developed by Deutsche Telekom, that offers effortless deployment and requires minimal maintenance. It integrates a collection of top-notch honeypot technologies into a single system.

T-POT leverages widely recognized honeypot daemons, intrusion detection systems (IDS), and tools for attack submission. The underlying concept of T-POT involves transforming the entire TCP network range, along with key UDP services, into functioning honeypots. This enables the system to efficiently direct incoming attack traffic to the most suitable honeypot daemons for appropriate response and analysis.

C. Amazon Web Services EC2 Instances Creation

With the aim of supervising and gathering global data on newly emerging attack processes, a series of configurations were utilized to establish Amazon Web Services Elastic Cloud Computing (AWS EC2) in two distinct locations. This strategic deployment allows for

comprehensive surveillance and information gathering on evolving attack methods worldwide.

```
Operating System: Debian 11
RAM: 8GB
Storage: 120GB (SSD)
vCPUs: 2
Virtualization: HVM AMI
Instance Type: t2.large
```

The instances that were deployed can be seen in the table.I, along side with there Location, zone and public IP address.

TABLE I
AWS EC2 INSTANCES DEPLOYED WITH T-POT

Name	Location	Zone	IP
Honeypot1	Paris, Europe	eu-west-3b	3.8.155.218
Honeypot2	London, Europe	eu-west-3a	13.37.249.179

D. T-Pot installation

A RSA encrypted private and public key pair were generated for each instances on AWS EC2 and used to SSH into the instances. After successfully using SSH to connect to the instances, both were upgraded in order to be able to install git that were used to clone the repository for the honeypot from GitHub, then proceeded to installation of the honeypot in standard mode.2

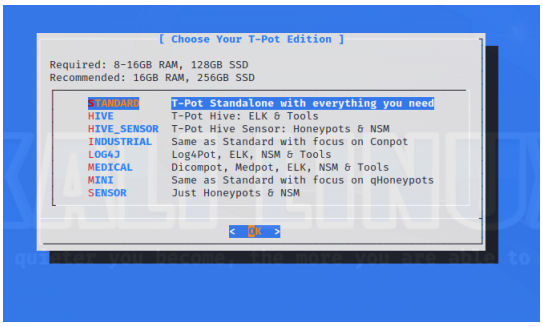


Fig. 2. T-Pot honeypot installation (Standard Edition)

In standard mode with T-Pot Standalone all services, tools, honeypots, etc. were installed on to a single host. During the installation of T-pot we create a username and password that will be used to access the dashboard in the browser. After the installation is complete, we use the following URL format to access the honeypots dashboard: `https://<instance_public_ip>:64297`, after logging we access the kibana option and it shows several

Fig. 3. T-pot Dashboard in Kibana

options of dashboards that analyze the logs, but we access the T-pot dashboard which has a general and detailed analysis of all the T-pot sensors and we enter a dashboard like the one shown in the fig.3.

IV. RESULTS ANALYSIS

Our two honeypots were deploy from 13/05/2023 for the one on Paris and from 20/05/2023 for the one in London to 06/06/2023. We had a total of 258,279 events recorded in both honeypots, as can be seen in fig.4, which for a period of less than 1 month is a very high number and demonstrates that when we are exposed on the internet and in the cloud, the possibility of receiving hundreds or even thousands of attacks or attempts to intrusion per day.

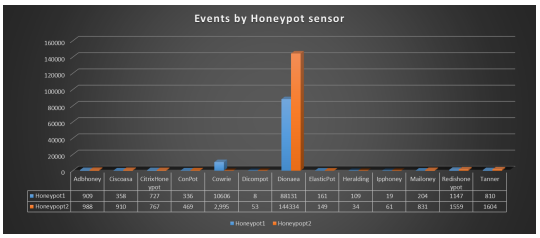


Fig. 4. Events by Honeypot Sensor

A. Attacks Origin and Frequency

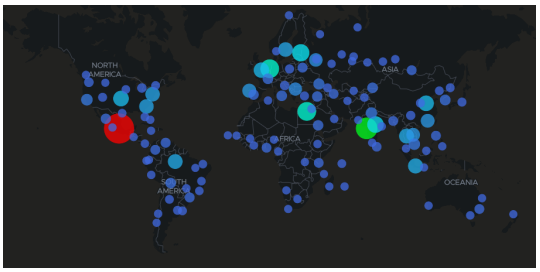


Fig. 5. Attack Map on Honeypot 1

As can be seen in the map shown in fig.5, the honeypot 1 that was the one with most running time, received attacks from places from all across the globe, and as can be verified in table.II, with emphasis in Mexico that registered a number of 36731 attacks on honeypot 1, followed by India with 19842 attacks. On honeypot 2 we can see a complete different scenario as Vietnam and Egypt were the ones with more attacks with 21571 and 15767 attacks respectively. Although we can point that some countries like the USA, China, Russia, are not in the top 3 for both but have a high number of interactions in both of them.

TABLE II
COUNTRIES WITH MORE ATTACKS IN BOTH HONEYPOTS

Honeypot 1		Honeypot2	
Country	Attacks	Country	Attacks
Mexico	36731	Vietnam	21571
India	19842	Egypt	15767
Egypt	846	Ecuador	148
United States	5531	Argentina	14225
China	4528	Russia	11055
Russia	3702	India	7653
Brazil	3264	China	4797
Indonesia	3194	Philippines	4554
Thailand	3185	United States	45
Sweden	3175	Taiwan	4381

Table.III,we have the number of attacks that were targeted to specific ports on both honeypots. And the highest number of attacks captured were in port 445, that is used by SMB, a network file sharing protocol, this port is vulnerable to security assaults, according to security researchers, and should be deactivated or well protected[1].

TABLE III
ATTACKS BY PORT NUMBER

Honeypot1		Honeypot 2	
Port	Attacks	Port	Attacks
445	79922	445	137635
23	1768	1433	1733
6397	1147	80	1532
80	810	6379	1520
443	727	25	820

B. Attacking Scenarios

Table.IV shows the connections received from different type of users and crawlers which are automatically categorized by honeypot sensor based on IP reputation.

TABLE IV
SRC IP REPUTATION

Attack	Honeypot 1	Honeypot 2	Total
known attacker	5 251	5 732	10 983
mass scanner	362	708	1070
bot crawler	12	29	41
tor exit node	7	10	17
form spammer	6	1	7
Total Attacks	5638	6 480	12 118

Among the Suricata alert histograms recorded on the honeypot shown on the table V, two notable categories are:

Attempted Administrator Privilege Gain alerts indicate attackers trying to gain unauthorized administrative access, potentially exploiting vulnerabilities or using brute-force methods.

Generic Protocol Command Decode alerts indicate intercepted network traffic with commands that may be suspicious or malicious.

TABLE V
ATTACKS BY CATEGORY

Category	Honeypot 1	Honeypot2
Attempted Administrator Privilege Gain	72919	15283
Generic Protocol Command Decode	49939	45139
Misc Attack	20559	14634
Attempted Information Leak	6608	9147
Potentially Bad Traffic	5139	6458

In fig.6, we can see a comparison of the HTTP content types most used in both Honeypots. The Content type is used to indicate the data/file that needs to be return from the request[2].

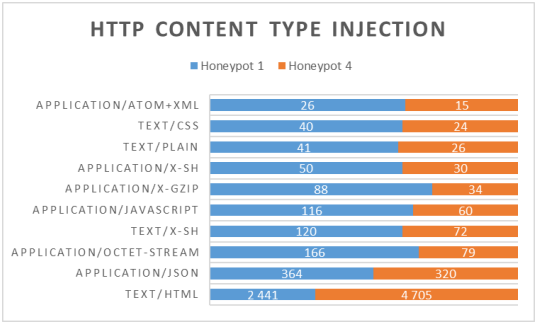


Fig. 6. Comparison of HTTP Content Type Content in both Honeypots

Analyzing Fig.7Attackers performed various HTTP request methods when interacting with the honeypots.

Here are the commonly observed methods and their corresponding frequencies:

- GET: This is the most frequently used method, with attackers making 8562 GET requests. This method is used to retrieve data from a specified resource
- POST: Attackers also utilize the POST method, which involves sending data to the server. In this case, there were 1462 POST requests made by attackers.
- HEAD: This method, with 261 requests, is employed by attackers to retrieve the headers of a specific resource without fetching the actual content.
- OPTIONS: Attackers occasionally use the OPTIONS method, making 76 requests. This method allows them to retrieve the communication options available for a given resource.
- CONNECT: With 50 requests, the CONNECT method is utilized by attackers to establish a tunnel connection to a remote server through an intermediary.
- PUT: Although less commonly used, attackers made 3 PUT requests. This method is typically used to upload or update resources on the server.

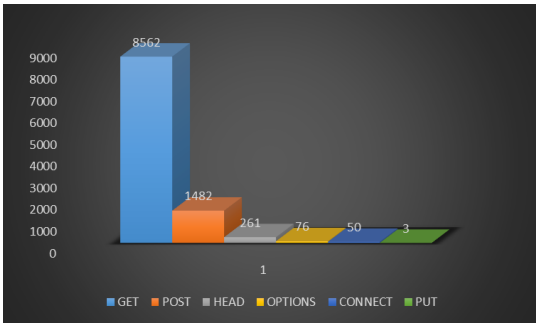


Fig. 7. HTTP Methods Use on both Honeypots

Fig.8 illustrates different exploit types utilized by attackers. Notably, the CVE-2020-11899 exploit was prominently employed. This particular exploit targets an improper input validation vulnerability in the IPv6 component, which allows an unauthorized network attacker to send a malicious request. As a consequence, the target device may experience an out-of-bounds read, potentially leading to a denial-of-service condition. Another significant exploit that stands out is CVE-2019-11500, this exploit targets a specific vulnerability and allows remote attackers to conduct arbitrary code execution. It poses a significant threat as it can potentially compromise the security of the targeted system, leading to unauthorized access and potential data breaches.

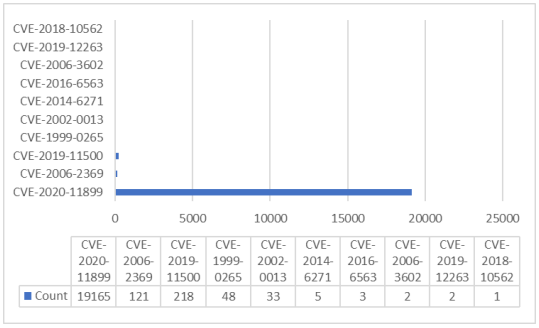


Fig. 8. CVE range used to exploit honeypot network

V. Cloud Server Protection Framework Proposed

As an extension of our work we intend to demonstrate and suggest what would be a safer version of an infrastructure that we intend to place in the cloud, part of what will be suggested has been tested in a practical scenario, others are only suggested based on the results of our study and good security practices in cloud computing. The suggested framework is shown in fig.9 , where we intend to create a scenario that simulates a company that operates in the cybersecurity area and that provides a SaaS (Software as a Service), which is linked in the AWS cloud, and to protect this service we intend to apply a firewall to the server of the company, precisely with key pair authentication practices with public key cryptography, and as a way to monitor the security and exposure of the server, implement a way to monitor the server's security logs.

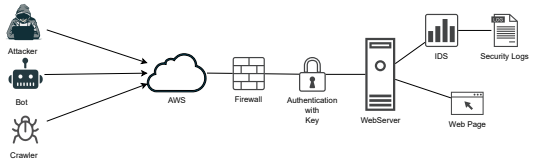


Fig. 9. Proposed framework to Secure Cloud Web Service

Firstly, we create an instance in Aws, with an UBUNTU 20.04 LTS operating system, to which we attribute authorization to receive http and https traffic, then a website was created that simulates the SaaS that the company provides, this website can be consulted at

the following link:http://ec2-35-180-28-203.eu-west-3.compute.amazonaws.com/ and seen in fig.10.

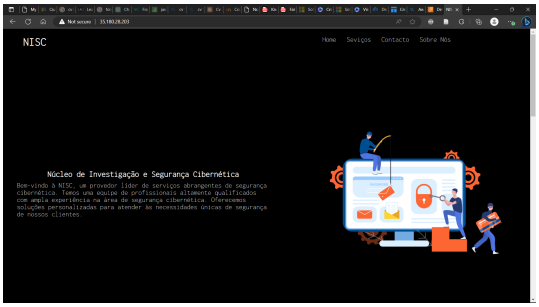


Fig. 10. Company Website

After everything operational, we leave for the security part. The first step was to automate the server's operating system update, through the creation of a script that is executed every day, to prevent attacks that have the ability to discover the server's OS from exploiting vulnerabilities that an old and outdated version you have. Then we created a script that allows monitoring the logs with the help of Shodan, which is a powerful tool that provides information about devices, such as service banners, software versions, open protocols, among other technical details. Then you must configure a firewall that blocks unwanted traffic, a crucial incoming traffic control rule must be placed on port 445, which as we have seen is very exploited by attackers. If our Website was a web-application, to avoid SQL injection, secure coding best practices must be applied, such as avoiding validating and sanitizing user input, avoiding cross-site scripting (XSS), and protecting against brute force attacks. Keeping the web application up-to-date by regularly applying security patches and software updates. Implement additional security mechanisms such as protection against brute-force attacks, limiting request rates, and monitoring application logs for suspicious activity. To secure the connection to the site, is mandatory to create and SSL/TLS certificate to secure your site and users data and privacy.

VI. Conclusions

Honeypots play a crucial role in understanding attackers by attracting interested peers and revealing their methods. Our research provides a comprehensive analysis of threat origin, frequency, interactions, and types of possible attacks that we could be facing on a Cloud environment. This understanding helps in taking protective measures into Computing infrastructures deployed in the cloud. However, limitations arise from default templates and short experiment periods, affecting the results.Future research should consider our findings to develop improved detection and defense strategies, while also addressing the limitations highlighted in this study.

References

[1] *What is an SMB Port + Ports 445 and 139 Explained*, en. [Online]. Available: https : / / www . varonis.com/blog/smb-port (visited on 06/10/2023).

[2] *HTTP headers — Content-Type*, en-us, Section: Web Technologies, Oct. 2019. [Online]. Available: https : / / www . geeksforgeeks . org / http - headers - content-type/ (visited on 06/10/2023).

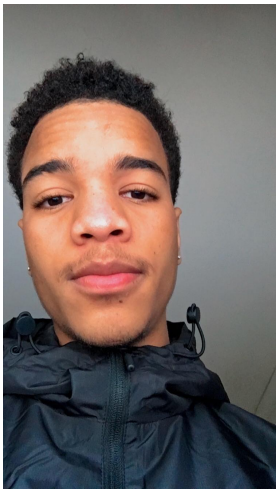
[3] S. M. Z. Ur Rashid, M. J. Uddin, and A. Islam, "Know Your Enemy: Analysing Cyber-threats Against Industrial Control Systems Using Honeypot," en, in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, Dhaka, Bangladesh: IEEE, Nov. 2019, pp. 151–154, ISBN: 978-1-72815-852-5. DOI: 10 . 1109 / RAAICON48939 . 2019 . 69. [Online]. Available: https : / / ieeexplore . ieee . org / document / 9087515/ (visited on 05/11/2023).

[4] S. M. Z. U. Rashid, A. Haq, S. T. Hasan, M. H. Furhad, M. Ahmed, and A. S. S. M. Barkat Ullah, "Faking Smart Industry: A Honeypot-Driven Approach for Exploring Cyber Security Threat Landscape," en, in *Cognitive Radio Oriented Wireless Networks and Wireless Internet*, H. Jin, C. Liu, A.-S. K. Pathan, Z. M. Fadlullah, and S. Choudhury, Eds., vol. 427, Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Cham: Springer International Publishing, 2022, pp. 307–324, ISBN: 978-3-030-98001-6 978-3-030-98002-3. DOI: 10.1007/978-3-030-98002-3_23. [Online]. Available: https://link.springer.com/10.1007/978-3-030-98002-3_23 (visited on 05/11/2023).

[3] [4]

Self-Evaluation

Rúben da Luz - Self-Evaluation: 15
Responsible for the LaTeX report, for processing the the data received on the honeypots, for creating the web server used to implement the security measures, create a logs monitoring system to the server and a script to update the SO, created the website used for simulating a SaaS.



Célio Pina - Self-Evaluation: 15
Responsible for the LaTeX report, for deploying an launching the honeypots on the Aws cloud, for collecting all the necessary data generated by the honeypot, responsible for launching the company example website on the server created to implement the security measures, implement a small firewall in the same server.