SSH Bruteforce attack V2

Attacking a server on a private network from the outside

Realised by Gabriel DECAVE, Maxence BEKHEDDA, Maxence BOUCHADEL, Lucien HALKIN

May 24, 2024

Contents

I Topology

To create the topology, we are going to use Mininet and Ryu for the firewall. To analyze the network and the packets received, we will use wireshark. To block packet we will use Snort, Fail2Ban and firewall.

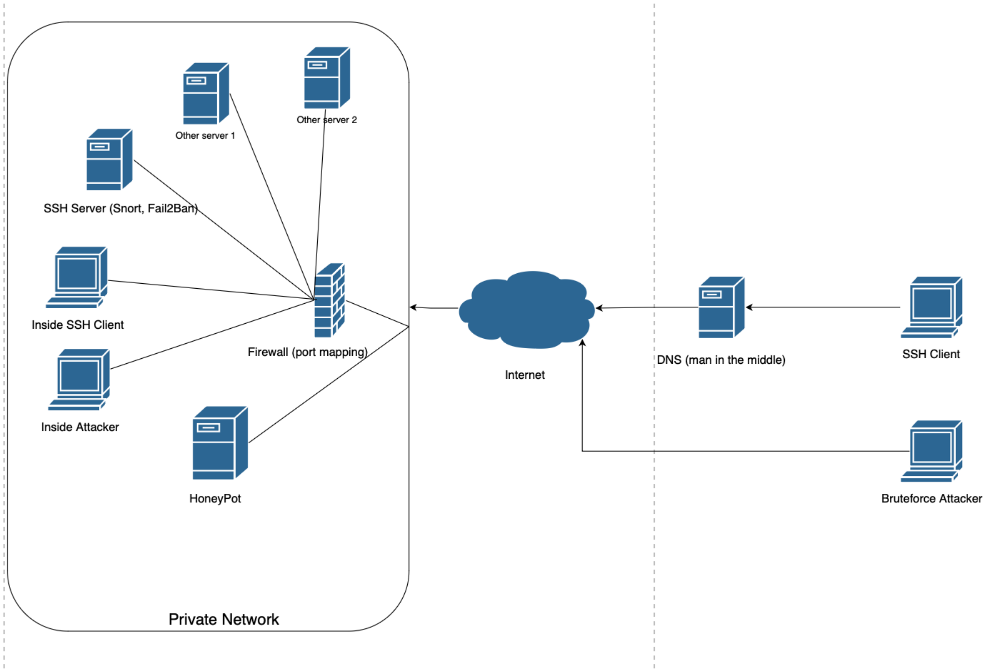

Figure 1: Topology of our simulation

Our topology has a private network with a SSH Server and two others server. All these devices are protected by a firewall connected to the internet. There is a HoneyPot to analyse the comportement of attacker, pre-block some attackant or find vulnerabilities. We have an internal and an external user. There is a DNS that is infected and controlled by an attacker.

II Attack to be performed

The primary objective of the attack is to compromise the security of the network by exploiting vulnerabilities in the SSH server and potentially other services. The attacker aims to gain unauthorized access to sensitive information, disrupt services, or use the compromised servers for further malicious activities, such as launching attacks on external targets or establishing persistence within the network.

a SSH Audit

We want to get the maximum information of the server(OS, ports, ssh versions etc). The attacker will perform different audit to gather information. Also, as part of the defensive strategy, regular audits of the SSH server configuration and logs are conducted to ensure compliance with security best practices and to identify any suspicious activities. This involves reviewing SSH access logs, monitoring authentication attempts, and implementing measures to detect and respond to unauthorized access attempts promptly.

Tools used:

- SSH-audit https://github.com/jtesta/ssh-audit
- nmap
- nc

b Classic Bruteforce

One attack vector the attacker may employ is a classic brute-force attack against the SSH server. This involves systematically trying numerous combinations of usernames and passwords until a successful authentication is achieved. To mitigate this, strong password policies, account lockout mechanisms, and intrusion detection systems are implemented to detect and block repeated failed login attempts. These attack will be perfomed from inside and outside.

Tools used:

- python3
- classic list

c SSH MitM

Another potential attack is a Man-in-the-Middle (MitM) attack targeting SSH communications. In this scenario, the attacker intercepts and potentially alters the encrypted SSH traffic between the client and server, allowing them to eavesdrop on sensitive information or inject malicious commands. Countermeasures against MitM attacks include using encrypted communication channels, such as SSH with strong cryptographic algorithms, and implementing techniques like certificate-based authentication to verify the identity of the server.

Also a MitM attack using the hacked dns server will be perfomed to grab private key, users information or else.

Tools used:

- ssh-mitm https://github.com/jtesta/ssh-mitm
- python3
- others to find

d SSH Snake

An advanced attack strategy, SSH Snake involves exploiting vulnerabilities in SSH client software or leveraging compromised credentials to gain unauthorized access to the server. This could include exploiting zero-day vulnerabilities, conducting phishing attacks to steal credentials, or exploiting weaknesses in SSH configuration settings. To defend against such attacks, regular software patching, user awareness training, and implementing multi-factor authentication are crucial. We will get keys from succesfully compromised server to gain access to other.

Tools used:

- ssh-snake https://github.com/MegaManSec/SSH-Snake
- classic python app or bash script
- others to find

III Detecting and defending the attack

To create best defenses against potential attacks on our network infrastructure, we employ a multi-layered approach utilizing various techniques and tools to detect and mitigate threats effectively.

a Cipher Protection and Private Key Authentication

Implementing robust cipher suites and encryption protocols for SSH communication adds an extra layer of security by protecting data in transit from eavesdropping and manipulation. Furthermore, utilizing private key authentication instead of relying solely on passwords strengthens access controls and mitigates the risk of brute-force attacks.

b DNS Checking

Given the presence of an infected DNS server controlled by an attacker, regular monitoring and integrity checks of DNS configurations and traffic are imperative. Implementing Domain Name System Security Extensions (DNSSEC) can help ensure the authenticity and integrity of DNS data, mitigating the risk of DNS spoofing attacks and unauthorized domain redirection.

c Firewall Port Mapping

Firewalls play a critical role in network security by filtering incoming and outgoing traffic based on predefined rules. By configuring port mapping on the firewall, we can restrict access to essential services such as SSH to specific IP addresses or limit the number of simultaneous connections, thereby reducing the attack surface and thwarting unauthorized access attempts.

d HoneyPot

Deploying a HoneyPot within the network serves as a decoy system designed to lure potential attackers away from critical assets while simultaneously capturing valuable intelligence about their tactics and techniques. By analyzing the interactions with the HoneyPot, we can identify emerging threats, gather threat intelligence, and adapt our defensive strategies accordingly.

e Intrusion Detection Systems (IDS)

Utilizing intrusion detection systems such as Snort or fail2ban enables real-time monitoring of network traffic for suspicious patterns or known attack signatures. These systems can automatically trigger alerts or block malicious IP addresses attempting to exploit vulnerabilities, helping to mitigate the impact of intrusion attempts and prevent unauthorized access to the network.

f Continuous Monitoring and Incident Response

Regularly monitoring network traffic, server logs, and system activity is essential for promptly detecting and responding to security incidents. Establishing incident response procedures and designated response teams ensures a coordinated and effective response to security breaches, minimizing downtime and mitigating the impact on business operations.

By combining these techniques and tools, we create a robust defense-in-depth strategy that enhances our ability to detect, defend against, and mitigate the impact of potential attacks on our network infrastructure. Constant vigilance and proactive measures are crucial in safeguarding the integrity, confidentiality, and availability of our systems and data.