**Mitnick Attack: TCP Session Hijacking Awareness & Defense**

**Prepared by:** Mohammed Fouzan Aamiri

**Date of Completion:** March 14th, 2024

**Executive Summary:** *(Reference from Seed Project Website* [*Link*](https://seedsecuritylabs.org/Labs_20.04/Networking/Mitnick_Attack/)*)*

In this detailed report, we're going to dive into the infamous Mitnick Attack, a pivotal moment in cyber history that reveals just how vulnerable our networks can be. Named after the notorious hacker Kevin Mitnick, this attack takes advantage of weaknesses in the TCP protocol and exploits the trust between connected computers. The showdown between Mitnick and Shimomura in 1994, which has been immortalized in books and movies, serves as a stark reminder of the serious consequences of such exploits.

Our main goal is to help our clients understand the Mitnick Attack better by recreating it in a controlled environment. By carefully mimicking the settings on Shimomura's computers, we aim to show just how dangerous TCP session hijacking can be and the serious threat it poses to organizational security. Successfully replicating this attack will demonstrate how threat actors can take control of compromised systems and execute commands at will, emphasizing the urgent need for strong defense measures.

Understanding the ins and outs of the Mitnick Attack will give our clients invaluable insight into the ever-changing cyber threat landscape. With this knowledge, they can take proactive steps to protect their networks from similar attacks. We strongly recommend implementing a range of defense strategies, including thorough security training, tight access controls, and regular security checks. By following these best practices, organizations can boost their resilience and reduce the risk of falling victim to TCP session hijacking attacks, thereby safeguarding their critical assets, and ensuring smooth operations.

**Attack Description:**

The Mitnick attack is like a cyber magician pulling off a sneaky trick. Instead of just stealing a connection between two targets, it creates its own connection first. Here's the scene: Host A, let's call it X-Terminal, is the target. Host B is a trusted server that can access X-Terminal without a password. Kevin Mitnick, our cyber trickster, wants to get into X-Terminal without a fuss. So, he pretends to be Host B and sets up a secret connection between X-Terminal and the trusted server. Once connected, Mitnick swoops in, taking control of X-Terminal without it suspecting a thing. It's like he's performing a magic act, with four main steps:

1. **Setting the Scene**: Mitnick creates a convincing disguise, making X-Terminal think he's the trusted server.
2. **Creating the Connection**: Mitnick builds a secret line of communication between X-Terminal and the trusted server.
3. **Taking the Stage**: With the connection in place, Mitnick steps in and takes control, fooling X-Terminal into thinking he's the trusted server.
4. **Executing the Act**: Now in control, Mitnick can run his cyber commands on X-Terminal, leaving it vulnerable to his every move.

A diagram of a computer system

Description automatically generated

*Figure 1: Illustration of Mitnick Attack*

Preparing the lab to replicate the Mitnick attack requires careful planning and attention to detail. We'll create a controlled environment with virtual machines representing X-Terminal and the trusted server. These will run on isolated networks to prevent unintended consequences. We'll configure network settings to simulate communication between the two, set up monitoring tools to track activity, and implement security measures like firewalls. With this setup, we'll closely study the attack's mechanics and its implications for network security.

A screenshot of a computer

Description automatically generated*Figure 2: X-terminal*

As seen in Figure 2, we are configuring the X-terminal to have direct access to the trusted server without the requirement of password. To avoid typing password .rhosts file is created and the trusted server’s IP address has been inserted into the file. We must make sure the file is stored at the top level of the user’s home directory in order to connect to the trusted server without the password.

A screen shot of a computer

Description automatically generated

*Figure 3: Trusted-Server*

As seen in Figure 3, we are verifying if the configuration works fine without any authentication failure in Trusted-Server.

This report zeroes in on two key moves in the Mitnick attack:

**Step 1:** Guessing the Sequence Numbers Imagine Mitnick as a codebreaker trying to crack a secret pattern. Before striking, he had to figure out the sequence numbers (ISN) on X-Terminal. These numbers weren't random back then. So, Mitnick sent sneaky SYN requests to X-Terminal and got back SYN+ACK responses. Then, he cleverly cleared any half-open connections from X-Terminal's queue with RESET packets, keeping things tidy. After doing this dance around twenty times, Mitnick spotted a rhythm between the TCP ISNs. This breakthrough let him predict the numbers accurately, setting the stage for his attack.

A screenshot of a computer

Description automatically generated*Figure 4: Guessing the Sequence number from captured packets in Wireshark.*

**Step 2:** Flooding the Trusted Server with SYN Now, picture Mitnick facing a tough challenge. To get the trusted server to talk to X-Terminal, he needed to send a special SYN message. But this caused X-Terminal to respond with a SYN+ACK, confusing the trusted server into sending a RESET to stop the chat. It was like a hiccup in Mitnick's plan. To fix it, he had to quiet down the trusted server. So, before pretending to be it, Mitnick flooded it with SYN messages. This attack was like a battering ram against the server's weaknesses, shutting it down and making it silent for Mitnick's sneaky maneuvers.

A screenshot of a computer

Description automatically generated*Figure 5: This is the result after executing spoof.py*

**Experiment:**

During the Mitnick Attack, operating systems were susceptible to SYN flooding attacks, capable of silencing or completely shutting down the target machine. In a SYN flood attack, the attacker ignores the server's SYN-ACK packets, the response to the initial SYN packets. Consequently, the server's connection table gets inundated with half-open connections, depleting its resources and impeding it from handling genuine connection requests. This culminates in a denial of service (DoS) scenario as the server is inundated and unable to manage legitimate requests.

SYN flooding attacks cause service disruption by overwhelming the server's resources, making it unable to handle genuine connection requests. This leads to the unavailability of services for users. Additionally, the attacks can destabilize the system or network, causing crashes, slowdowns, or performance degradation beyond the targeted services.

**Recommendations:**

Mitnick's attacks served as a wake-up call for the importance of fortifying defenses against SYN flood attacks, which exploit vulnerabilities in the TCP handshake process. To mitigate the impact of such attacks, several measures can be implemented:

1. SYN cookies: This technique helps protect against SYN flood attacks by encoding additional information into the initial SYN-ACK response sent by the server. SYN cookies allow the server to track connection requests without storing extensive state information, thereby mitigating the risk of resource exhaustion.

2. Rate limiting: Implementing rate limiting mechanisms can help control the rate of incoming connection requests to prevent overwhelming the server. By setting thresholds for the number of SYN packets accepted per second, servers can effectively mitigate the impact of SYN flood attacks while still allowing legitimate traffic to access services.

3. Intrusion detection/prevention systems (IDS/IPS): IDS/IPS solutions can monitor network traffic for suspicious patterns indicative of SYN flood attacks. These systems can detect and block malicious SYN packets in real-time, thereby helping to mitigate the impact of such attacks and protect the targeted services.

4. Firewalls: Firewalls play a crucial role in network security by filtering incoming and outgoing traffic based on predefined rules. By configuring firewalls to block or rate-limit SYN packets, organizations can protect their servers from SYN flood attacks and other malicious activities.

**Reflections:**

As we delve into the case study of the Mitnick attack, our attention is drawn to the initial phase of the attack, which involves identifying the sequence pattern. Upon examination, we discover that all three hosts are on the same network. As I undertake this task, I find it simpler to obtain the sequence number when the trusted server is connected to the X-terminal without a password. This raises a pertinent question: how did Mitnick acquire the sequence number if he did not have access to the trusted server?

**Attachments:**

spoof.py

#!/usr/bin/env python3

from scapy.all import \*

x\_terminal\_ip = "10.9.0.5"

x\_terminal\_port = 514

trusted\_server\_ip = "10.9.0.6"

trusted\_server\_port = 1023

ip = IP(src=trusted\_server\_ip, dst=x\_terminal\_ip)

tcp = TCP(sport=x\_terminal\_port, dport=trusted\_server\_port, flags="S", seq=1421334543)

pkt = ip/tcp

ls(pkt)

send(pkt,verbose=0)