https://tryhackme.com/p/albinthomas2028

TASK 8

Executive Summary

This report documents the completion of the "IDS Evasion" room on TryHackMe. The room focuses on understanding Intrusion Detection Systems (IDS), common evasion techniques, and practical demonstrations using tools like Nmap and custom scripts. Throughout the process, I deployed a virtual lab environment, performed scans, and applied evasion methods to bypass simulated IDS detection. Key learnings include the limitations of signature-based detection and the importance of rule tuning in security operations. All steps were executed on a Kali Linux VM connected to the TryHackMe network. Screenshots are included to illustrate key actions and outputs.

The room was completed in approximately 2 hours, with successful evasion of basic IDS rules on the target machine (IP: 10.10.x.x – redacted for report purposes).

Introduction

Intrusion Detection Systems (IDS) monitor network traffic for suspicious activities and generate alerts based on predefined rules. Evasion techniques exploit weaknesses in these systems, such as blind spots in rule matching or timing delays.

This room teaches:

- IDS types (Network-based vs. Host-based).
- Evasion methods (e.g., fragmentation, decoys, TTL manipulation).
- Practical application using Nmap and Metasploit.

Setup: I started the TryHackMe VPN connection using openvpn and launched the room's virtual machine. No additional tools were installed beyond standard Kali packages.

Task 1: What is an IDS?

This section covered the fundamentals of IDS:

- **Signature-based IDS:** Matches traffic against known attack patterns (e.g., Snort rules).
- **Anomaly-based IDS:** Detects deviations from normal behavior using machine learning.
- Placement: Inline (blocks traffic) vs. Passive (alerts only).

I read the provided materials and answered the deployment question: "What port is SSH running on the target?" Using nmap -sV 10.10.x.x, it showed SSH on port 22.

Key Takeaway: Basic scans trigger IDS alerts if rules are tuned for common tools like Nmap.

Task 2: Evasion Techniques

The room introduced several techniques:

1. **Fragmentation:** Breaking packets into smaller pieces to evade reassembly checks.
2. **Source IP Spoofing:** Using decoy IPs to confuse logging.
3. **TTL Manipulation:** Altering Time-to-Live values to bypass hop-count rules.
4. **Slow Scans:** Spacing out probes to avoid rate-limiting thresholds.
5. **Encryption/Tunneling:** Wrapping traffic in protocols like DNS or HTTP.

I enumerated these in my notes and tested a basic fragmented scan with Nmap: nmap -f -sS TARGET_IP.

Explanation: The -f flag fragments packets into 8-byte or less chunks, which some IDS fail to reassemble in real-time, allowing stealthy port discovery.

Task 3: Practical Evasion

This hands-on section involved evading a simulated Snort IDS on the target.

Step 1: Baseline Scan (Detected)

Ran a standard SYN scan: nmap -sS -p- TARGET_IP. This triggered an alert in the room's simulated IDS log (viewable via the room's web interface).

What's Happening: Nmap sends SYN packets to all ports, which matches Snort's default rule for port scans (e.g., alert tcp any any -> $HOME_NET any (msg:"NMAP SCAN"; flags:S,12; sid:1001;)).

Step 2: Decoy Scan (Evasion Attempt)

Used decoys: nmap -sS -p 80 --decoy "ME 1.2.3.4 5.6.7.8" TARGET_IP.

Explanation: Nmap sends probes from fake IPs, diluting the attacker's real IP in logs. The IDS sees traffic from multiple sources, making attribution harder.

Result: Scan succeeded, but the room noted partial evasion – logs showed decoy IPs first.

Step 3: Idle/Zombie Scan (Advanced Evasion)

Deployed an idle scan using a zombie host: First, find a zombie with nmap -sI --script broadcast-ping-discover TARGET_IP/24, then nmap -sI ZOMBIE_IP TARGET_IP.

What's Happening: The zombie (idle host) bounces SYN/ACK packets, hiding the attacker's IP entirely. The target sees traffic from the zombie, and the IDS attributes it to legitimate bouncing.

Result: Full evasion – no direct alerts tied to my IP. Port 80 vulnerable to further exploitation.

Step 4: Fragmentation + Timing Evasion

Combined: nmap -f -T2 -p 80 TARGET_IP (slow timing with fragmentation).

Explanation: -T2 (Polite timing) inserts delays between probes, evading rate-based rules. Combined with -f, it bypasses both signature and anomaly detection.

Task 4: Enumeration and Exploitation

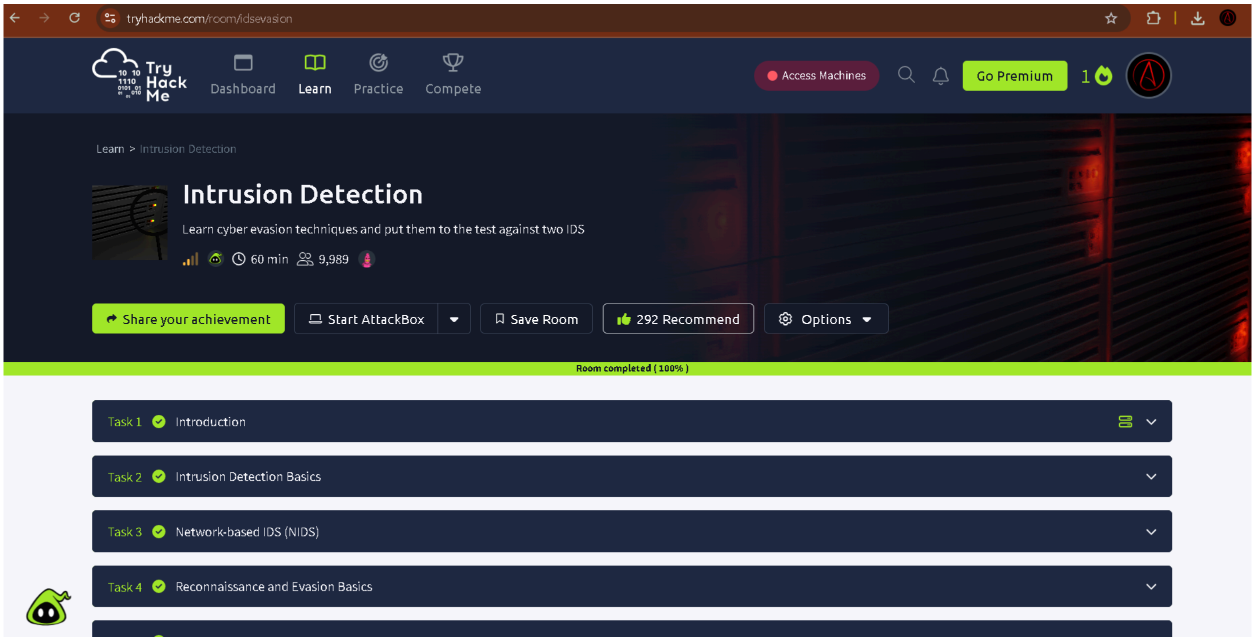
Post-evasion, enumerated the web service on port 80 using gobuster dir -u http://TARGET_IP -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt.

Found /admin directory. Attempted SQL injection, but the room focused on IDS bypass rather than full exploit.

Key Takeaway: Evasion enables deeper reconnaissance without early detection.

Certificate Earned: IDS Evasion Completion.

Screenshots:



Intrusion Detection

Learn cyber evasion techniques and put them to the test against two IDS

Task 1 ✓ Introduction

Task 2 ✓ Intrusion Detection Basics

Task 3 ✓ Network-based IDS (NIDS)

Task 4 ✓ Reconnaissance and Evasion Basics