Simplistic Write up

Target System: Simplistic (CTF Environment) **Assessment Date:** January 31, 2026 **Author:** Manus AI **Classification:** Confidential/Technical

---

1. Executive Summary

This report documents the findings of a security assessment conducted on the "Simplistic" target system. The analysis revealed a critical chain of vulnerabilities that permitted an unauthenticated attacker to achieve **full system compromise** (Root-level access). The attack sequence involved two primary stages: (1) achieving an initial foothold via an **unvalidated command injection** vulnerability in a web-based diagnostic tool, and (2) escalating privileges from the low-privilege web user to the root user by exploiting a **Set User ID (SUID) misconfiguration** on the find utility. Immediate remediation is required to address the fundamental security flaws identified.

2. Initial Foothold: Command Injection Vulnerability

The initial point of entry was identified on the system's web server, hosting a utility named **NetDiag v1.0**. This application was designed to perform network connectivity checks.

2.1. Vulnerability Description

The NetDiag v1.0 application accepts user input, ostensibly an IP address, and passes this input directly to a system shell command, specifically the ping utility, without adequate input sanitization or validation. This design flaw creates a classic **Operating System Command Injection** vulnerability. By introducing shell metacharacters (e.g., ;, |, &&), an attacker can terminate the intended ping command and append arbitrary operating system commands for execution.

2.2. Exploitation Methodology

The exploitation was confirmed by injecting a payload designed to execute a secondary command immediately following the standard network diagnostic operation. The objective of this initial exploit was to retrieve the first-stage flag, confirming the successful execution of arbitrary code.

Injected Payload Structure:

```
[Valid IP Address]; [Arbitrary Command]
```

Specific Payload Used to Retrieve User Flag:

```
127.0.0.1; cat /home/user/user.txt
```

This payload successfully executed the cat command, reading the contents of the user flag file and displaying it within the web application's output stream, thereby granting the attacker an initial foothold on the system.

Stage	Vulnerability Type	Component	Impact
Initial Foothold	Command Injection (CWE-77)	NetDiag v1.0 Web Interface	Execution of arbitrary commands as the web server user.

3. Privilege Escalation Vector

Following the initial compromise, the focus shifted to escalating privileges from the web user to the root user. System enumeration was performed to identify misconfigurations that violate the **Principle of Least Privilege**.

3.1. SUID Misconfiguration Analysis

A systematic search for binaries with the **Set User ID (SUID)** bit set revealed a critical security oversight: the /usr/bin/find utility was configured to run with root privileges.

Enumeration Command:

```
find / -perm -4000 2>/dev/null
```

Critical Finding:

```
/usr/bin/find
```

The SUID bit allows a user to execute a program with the permissions of the file owner, which in this case is root. The find utility is particularly dangerous when SUID is enabled because it possesses the -exec option, which allows it to execute arbitrary commands. This effectively grants any user who can execute find the ability to run any command as root.

3.2. Privilege Escalation Exploitation

The exploitation leveraged the SUID-enabled find binary to spawn a new shell with inherited root privileges.

Exploitation Command:

```
/usr/bin/find . -exec /bin/sh -p \; -quit
```

Command Breakdown:

- find .: Start the search in the current directory.
- -exec /bin/sh -p \;: Execute the /bin/sh shell. The -p flag is crucial as it ensures that the shell runs in **privileged mode**, preserving the effective user ID (EUID) of the file owner (root) rather than reverting to the real user ID (RUID).
- -quit: Terminates the find process immediately after the first execution of the command.

This command successfully instantiated a persistent shell with root privileges, confirming the full compromise of the target system.

4. Conclusion and Mitigation Recommendations

The "Simplistic" challenge demonstrates a classic attack vector involving the chaining of two distinct, yet common, security flaws. The successful exploitation underscores the importance of rigorous input validation and strict adherence to the Principle of Least Privilege.

4.1. Mitigation Strategies

The following table outlines the identified vulnerabilities and provides specific, actionable recommendations for remediation:

Vulnerability	Technical Cause	Recommended Mitigation
Command Injection	Unsanitized user input passed to ping utility.	Implement **strict input validation** (whitelisting) to ensure only valid IP address formats are accepted. Alternatively, use programming language-native functions for network diagnostics instead of invoking system shells.
SUID Misconfiguration	SUID bit set on /usr/bin/find.	**Remove the SUID bit** from all non-essential binaries, especially powerful utilities like find, nmap, vi, and less. Command: chmod u-s /usr/bin/find.

Vulnerability	Technical Cause	Recommended Mitigation
General Principle	Violation of Least Privilege.	Regularly audit system permissions and SUID configurations. Ensure that applications and users only possess the minimum set of privileges necessary to perform their required functions.