



Training DLL Side-Loading for Red Team Ops

Por Jesús Domínguez

22/11/2022

¡Hola a todos, gracias por tomarse el tiempo de asistir a este taller!

Cualquier duda que tengas siéntete libre de hacérmela saber y con gusto la responderé de la mejor forma posible

\$~: whoami

Offensive security researcher en Ocelot team by
Metabase Q

Infosec streamer en twitch.tv/notmalafama directo todos
los jueves

Offensive Security Exploit Developer



¿Qué hace un security researcher?

Análisis de malware

Analizar el comportamiento de malware para identificar IOCs TTPs

Analizando el ransomware Darkside que atacó al oleoducto estadounidense Colonial

Por: Miguel Gonzalez y Jesus Dominguez de Ocelot Offensive Security Team

Ploutus está de regreso, atacando cajeros de Itaotec en Latinoamérica

Por: Jesús Domínguez, Equipo de Seguridad Ofensiva Ocelot

Informe sobre amenaza Janeleiro.mx

Por Jesús Domínguez del Equipo de Seguridad Ofensiva, Ocelot.

Quién es y cómo opera Raccoon

Por Ramsés Vázquez y Jesús Domínguez del Equipo de Ocelot

Detección de vulnerabilidades

Detectar e implementar de vulnerabilidades durante ejercicios de red team

Filezilla DLL Side-Loading

By Jesús Domínguez, Ocelot Offensive Security Researcher

Identificando una DLL vulnerable



TODO LO QUE NECESITAS SABER PARA ENCONTRAR ESTA VULNERABILIDAD

¿Qué es una DLL?

Dynamic Link Library (DLL)

Como su nombre nos indica Dynamic Link Library (DLL) son archivos conocidos como bibliotecas de enlaces dinámicos. Estas bibliotecas ofrecen mediante sus "exports" funciones específicas a las aplicaciones que así lo requieren.



LoadLibrary Windows API

Esta API de Windows es muy popular para cargar en la memoria de un proceso una DLL, esta API recibe el nombre de la DLL que se quiere cargar en la memoria del proceso.

C++

```
HMODULE LoadLibraryA(  
    [in] LPCSTR lpLibFileName  
);
```

Orden de búsqueda de las DLL

Es la forma en que standar que Microsoft define para la búsqueda de una DLL que se quiere cargar mediante LoadLibrary.

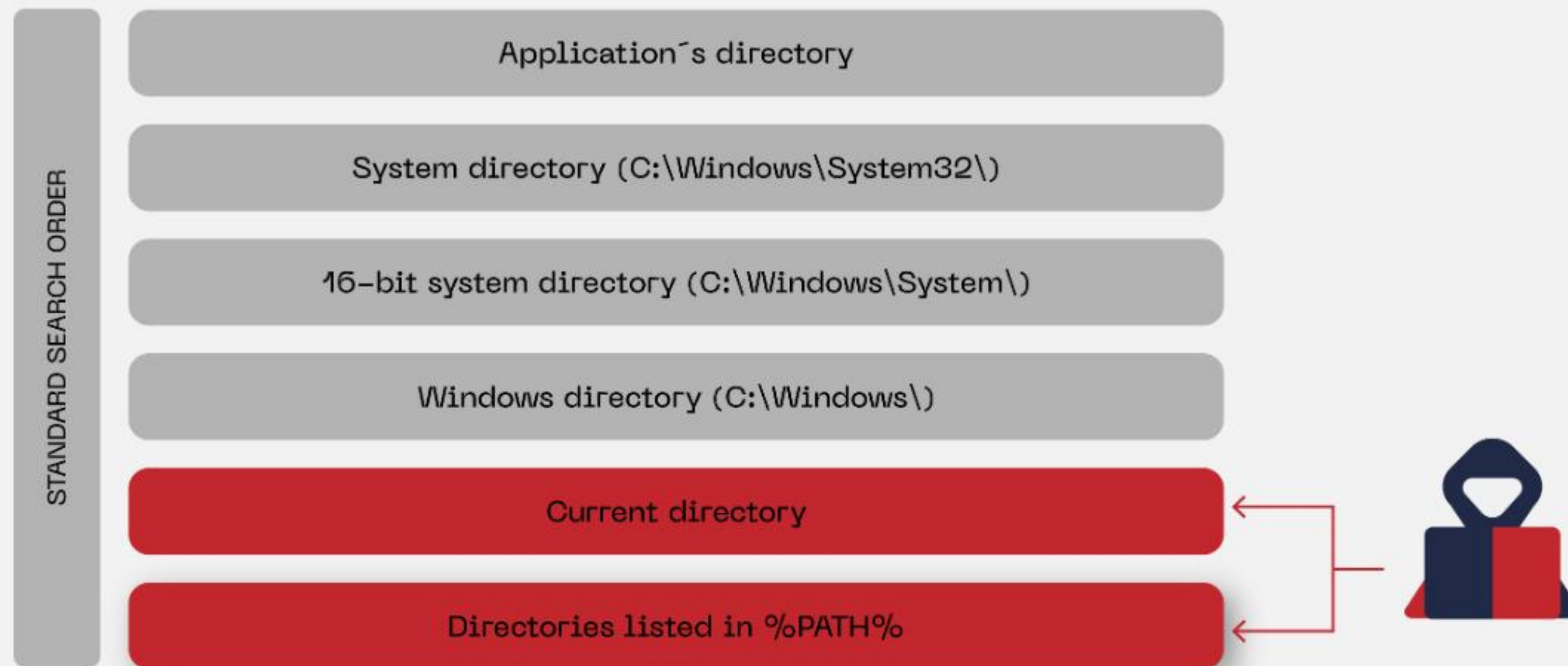


Figure 1. Traditional DLL search

Usando ProcMon para encontrar una DLL vulnerable

Process Monitor aka procmon es una herramienta de Windows que nos muestra en tiempo real el sistema de archivos, llaves de registro y process/thread activity.





Explotación

TODO LO QUE NECESITAS SABER PARA EXPLOTAR ESTA VULNERABILIDAD

Arquitectura básica de la DLL

La parte más importante de la DLL se le conoce como Entry-Point.

La DLL cuenta con distintos escenarios donde la DLL va a ser llamada, estos son:

- * DLL_PROCESS_ATTACH - El proceso carga la DLL
 - * DLL_THREAD_ATTACH - El proceso crea un nuevo thread
 - * DLL_THREAD_DETACH - El thread termina
 - * DLL_PROCESS_DETACH - El proceso cierra el handle de la DLL
-

Analizando los exports de la DLL

Además del DLL Entry-Point una DLL tiene funciones que expone para su uso por otras aplicaciones, estas funciones son mejor conocidas como "exports".

CFF Explorer VIII - [uxtheme.dll]

File Settings ?

uxtheme.dll

File: uxtheme.dll

- Dos Header
- Nt Headers
- File Header
- Optional Header
- Data Directories [x]
- Section Headers [x]
- Export Directory**
- Import Directory
- Resource Directory
- Exception Directory
- Relocation Directory
- Debug Directory
- TLS Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor

Member	Offset	Size	Value
Characteristics	000881E0	Dword	00000000
TimeDateStamp	000881E4	Dword	4B037C22
MajorVersion	000881E8	Word	0000
MinorVersion	000881EA	Word	0000
Name	000881EC	Dword	00089B06
Base	000881F0	Dword	00000001
NumberOfFunctions	000881F4	Dword	000000C0
NumberOfNames	000881F8	Dword	00000055
AddressOfFunctions	000881FC	Dword	00089608

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000031	00021CB0	N/A	N/A	N/A
00000032	0004E170	N/A	N/A	N/A
00000033	00004C70	0032	00089BD2	BufferedPaintStopAllAnimations
00000034	00004FB0	0033	00089BF1	BufferedPaintUninit
00000035	0000D820	0034	00089C05	CloseThemeData
00000036	00037700	0035	00089C14	DllCanUnloadNow
00000037	000377C0	0036	00089C24	DllGetActivationFactory
00000038	00037890	0037	00089C3C	DllGetClassObject
00000039	0000C020	0038	00089C4E	DrawThemeBackground
0000003A	00002AA0	0039	00089C78	DrawThemeEdge
0000003B	0004CA50	003A	00089C86	DrawThemeIcon

Caso de uso: FileZilla DLL side- loading



FileZilla DLL side-loading

Para este caso usaremos la última versión del popular software de FTP FileZilla, a continuación vemos los pasos a seguir:

1. Mover la DLL a un directorio diferente al original
2. Agregar FZ_DATADIR con el path C:\Program Files\FileZilla FTP Client a las variables de entorno de Windows
3. Ejecutar procmon para ver las DLL que se cargan
4. De las DLL que salen seleccionamos libfilezilla-32.dll
5. Revisamos exports con CFF explorer
6. Con el script en python obtenemos los exports que vamos a pegar en el proyecto de visual studio
7. Abrimos un nuevo proyecto de DLL en Visual Studio 2022
8. Pegamos los exports
9. Compilamos y probamos



Mitigación

MITIGANDO DLL SIDE-LOADING

Recomendaciones para la mitigación

- * Usar el path completo para llamar a la DLL
 - * Emplear herramientas automatizadas para detectar esta vulnerabilidad
 - * Usar alguna solución que sea capaz de bloquear DLL maliciosas cargadas por software legítimo
-

¡Muchas gracias por haber llegado al fina!

Me puedes encontrar así

Twitch / Youtube



LinkedIn

Jesús Rey Domínguez Domínguez

Offensive Security Researcher en Metabase Q

Área metropolitana de Ciudad de México · [Información de contacto](#)

Twitter @chucho_domz

Chucho

@chucho_domz

Sr Developer 🖥️ | CTF player 🚩 | Malware Analyst | Offensive Security Researcher |
Infosec streamer twitch.tv/notmalafama

📅 Joined March 2010