

# Handleiding: Threat **CVE-2024-6387** Uittesten

Deze handleiding beschrijft hoe je de kwetsbaarheid **CVE-2024-6387** kunt uittesten in een kritieke Debian-omgeving (32bit versie 11 Bullseye), die wordt aangevallen vanaf een Kali Linux-machine. Beide virtuele machines (VM's) worden opgestart via **VBoxManage**, en de benodigde scripts worden automatisch geïnstalleerd via PowerShell-scripts.

## 0. Voor het stappenplan

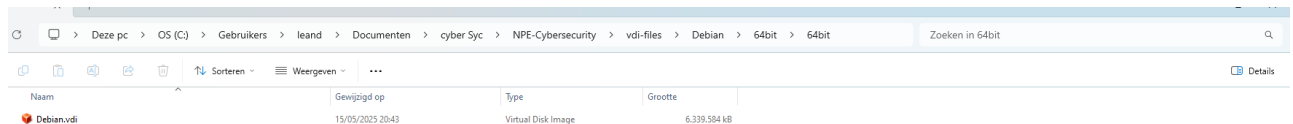
1. Clone de github-repo op: <https://github.com/JoranVanGoethem/NPE-Cybersecurity>
2. Voer in deze repository het stappenplan uit, hierin vind u ook de nieuwste handleiding en extra informatie over de aanval. Daarnaast staan hier ook de meest recente scripts in voor het testen van deze aanval.

## 1. Stappenplan

### 1.1 VM's Aanmaken

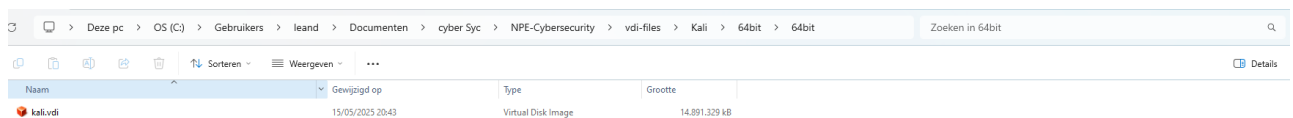
#### 1. Download de benodigde VDI-bestanden:

- **Kali Linux:** [Kali.vdi](#)
- **Debian 11 Bullseye:** [Debian.vdi](#)



A screenshot of a Windows File Explorer window. The address bar shows the path: 'Deze pc > OS (C:) > Gebruikers > leand > Documenten > cyber Syc > NPE-Cybersecurity > vdi-files > Debian > 64bit > 64bit'. The search bar contains 'Zoeken in 64bit'. The file list shows one file: 'Debian.vdi' with a size of 6.339.584 kB and a type of 'Virtual Disk Image'.

Naam	Gewijzigd op	Type	Grootte
Debian.vdi	15/05/2025 20:43	Virtual Disk Image	6.339.584 kB



A screenshot of a Windows File Explorer window. The address bar shows the path: 'Deze pc > OS (C:) > Gebruikers > leand > Documenten > cyber Syc > NPE-Cybersecurity > vdi-files > Kali > 64bit > 64bit'. The search bar contains 'Zoeken in 64bit'. The file list shows one file: 'kali.vdi' with a size of 14.891.329 kB and a type of 'Virtual Disk Image'.

Naam	Gewijzigd op	Type	Grootte
kali.vdi	15/05/2025 20:43	Virtual Disk Image	14.891.329 kB

#### 2. extract de zip files

- extract Kali in de locatie: **/vdi-files/Kali**
- het pad noemt: **/vdi-files/64bit/64bit/Kali Linux 2024.4 (64bit).vdi**
- extract Debian in de locatie: **/vdi-files/Debian**
- het pad noemt: **/vdi-files/32bit/32bit/Debian 11 (32bit).vdi**

```
$ Build-VM.sh M X Handleiding.md .gitignore
VM-Scripts > $ Build-VM.sh
1  #!/bin/bash
2
3  #-----
4  # Bash settings
5  #-----
6  set -o errexit  # abort on nonzero exitstatus
7  set -o nounset  # abort on unbound variable
8  set -o pipefail # don't mask errors in piped commands
9  set -u          # Stop het script bij een onbestaande variabele
10
11 #-----
12 # Variables
13 #-----
14 # VM-namen
15 DEBIAN_VM_NAME="vulnerable-debian"
16 KALI_VM_NAME="attacker-kali"
17
18 # Pad naar VDI-bestanden (vervang deze door jouw correcte paden met forward slashes)
19 DEBIAN_VDI_PATH="../../vdi-files/Debian/64bit/64bit/Debian.vdi"
20 KALI_VDI_PATH="../../vdi-files/Kali/64bit/64bit/Kali.vdi"
21
22
23 # Naam van het interne netwerk
24 INTNET_NAME="intnet"
25
26 #-----
27 # Functions
28 #-----
29
30 function create_VM(){
31     create_Debian
32     create_Kali
33 }
34
```

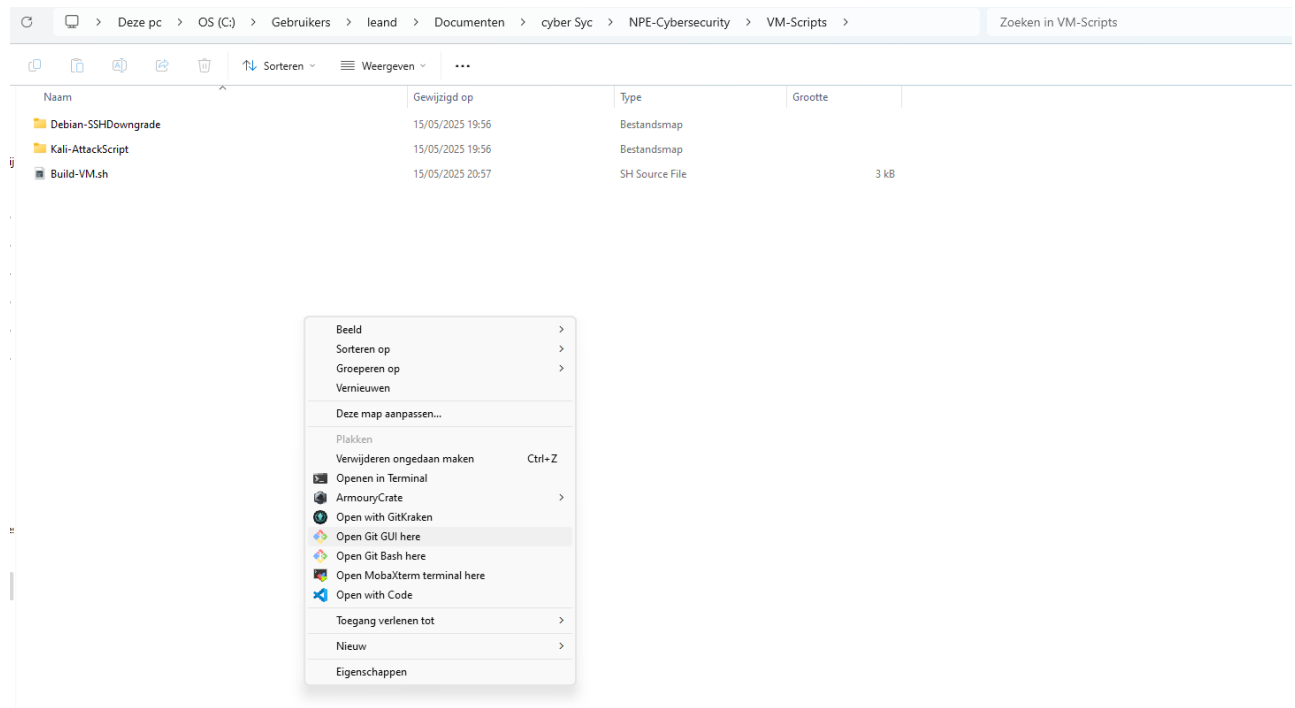
### 3. Pas de **Build-VM.sh** aan INDIEN vorige stap niet gevolgd:

- Vervang **DEBIAN\_VDI\_PATH=""** met:

```
DEBIAN_VDI_PATH="/jouw/pad/naar/Debian.vdi"
```

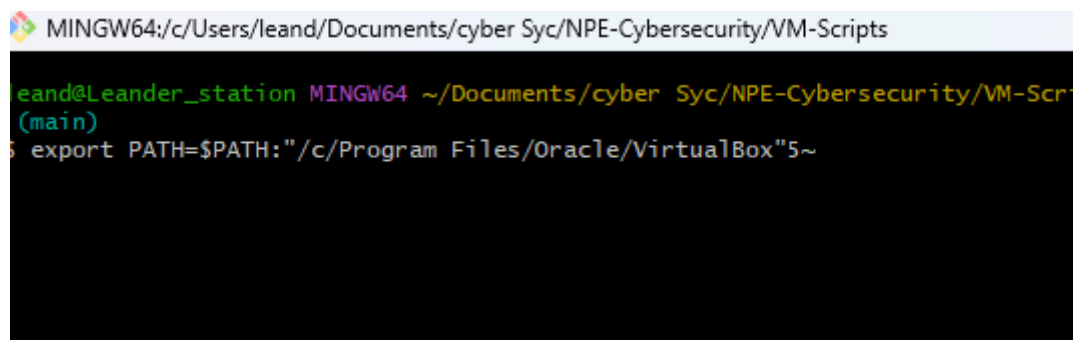
- Vervang **KALI\_VDI\_PATH=""** met:

```
KALI_VDI_PATH="/jouw/pad/naar/Kali.vdi"
```



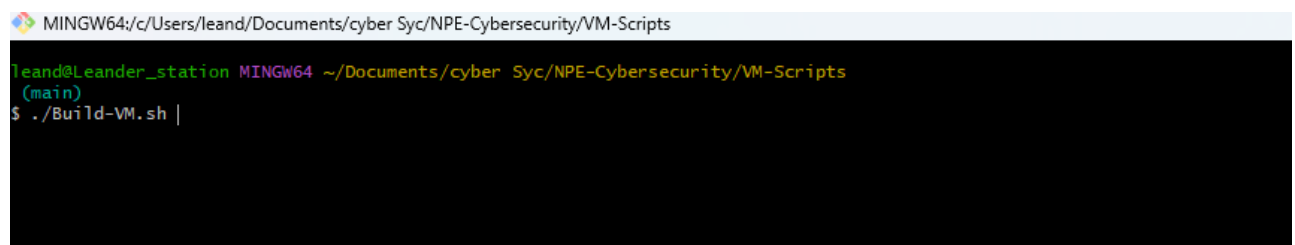
#### 4. Voeg VBoxManage toe aan de terminal

- voer het commando: `export PATH=$PATH:"/c/Program Files/Oracle/VirtualBox"` uit



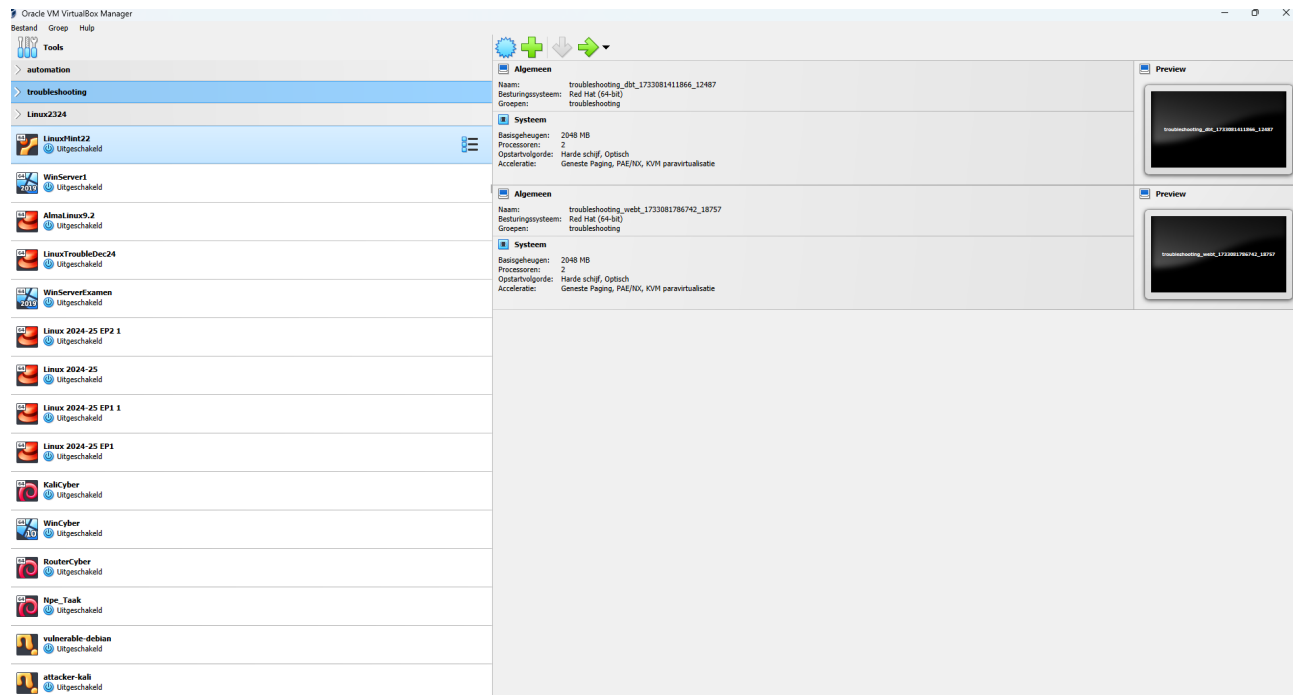
#### 5. Voer het script Build-VM.sh uit.

- open een git bash terminal in de map `/src/VM-Scripts/`
- Voer dit commando uit om kali & Debian aan te maken: `./Build-VM.sh`
- De virtuele machines worden aangemaakt in VirtualBox.



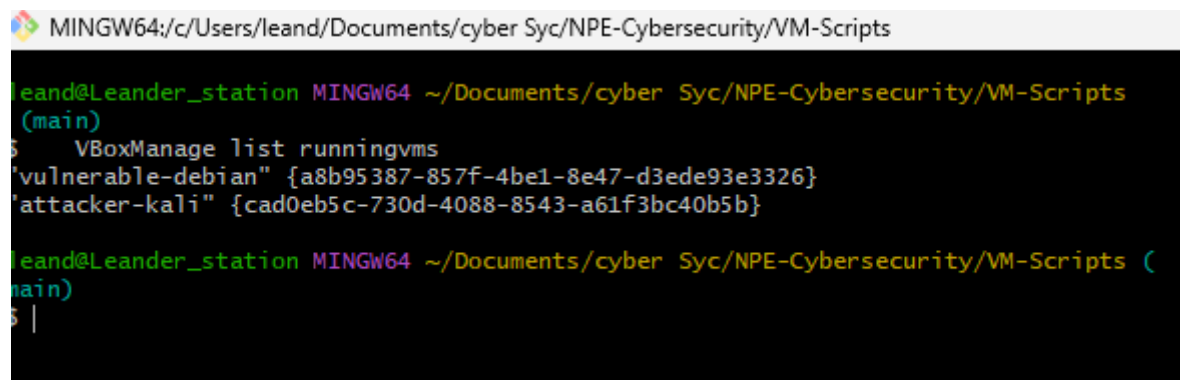
#### 6. start de VM's

- open virtualbox en start beide VM's



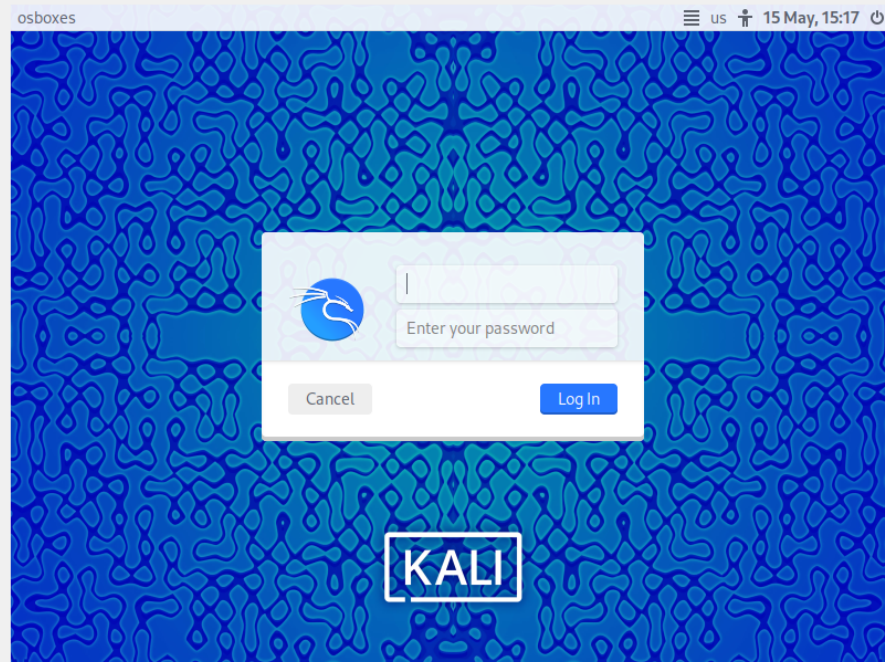
## 7. Controleer of de VM's correct zijn opgestart via de VirtualBox GUI of via:

```
VBoxManage list runningvms
```



## 8. Log in op de VM's

- Log in met het wachtwoord: [osboxes.org](https://osboxes.org) op de kali VM
- Log in met het wachtwoord: [osboxes.org](https://osboxes.org) op de Debian VM



```
Debian GNU/Linux 11 osboxes tty3
osboxes login: osboxes
Password:
Linux osboxes 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 15 14:11:35 EDT 2025 on tty3
osboxes@osboxes:~$
```

### 1. Maak een SSH-verbinding met de Debian VM:

```
ssh -p 2222 osboxes@127.0.0.1

# wachtwoord = osboxes.org
```

```
C:\Users\camma>ssh -p 2222 osboxes@127.0.0.1
The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:enbsCjQvNP3vJ60tiKuLsjlgmmv9TPkJ6GozRs3Exbg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:2222' (ED25519) to the list of known hosts.
osboxes@127.0.0.1's password:
Linux osboxes 5.10.0-8-686 #1 SMP Debian 5.10.46-4 (2021-08-03) i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 16 09:47:03 2025
osboxes@osboxes:~$
```

### 2. Maak het install script aan:

```
nano install_openssh_8.5p1.sh
```

```
osboxes@osboxes:~$
osboxes@osboxes:~$ sudo ./install_openssh_8.5p1.sh |
```

### 3. Maak het script uitvoerbaar en voer het uit:

```
chmod +x install_openssh_8.5p1.sh
sudo ./install_openssh_8.5p1.sh
```

Na het uitvoeren moet je onderstaande afbeelding verkrijgen. Ook kun je controleren of de juiste versie van OpenSSH is geïnstalleerd.

```
✓ Stap 4: Configuratie aanpassen...
✓ Stap 5: sshd starten...
🚀 Installatie voltooid. Controleer met: ssh -V
osboxes@osboxes:~$ ssh -V
OpenSSH_8.5p1, OpenSSL 1.1.1w 11 Sep 2023
osboxes@osboxes:~$ |
```

Sluit hier de SSH-

verbinding met Debian zodat poort 22 niet bezet is tijdens de aanval.

## 1.3 Aanval Uitvoeren vanuit Kali op de Debian VM

### 1. Test de verbinding tussen Kali en Debian.

```
ping 192.168.56.101
```

```
(osboxes@osboxes)-[~]  
$ ping 192.168.56.101  
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.  
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=3.16 ms  
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.945 ms  
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=2.77 ms  
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=1.76 ms  
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=1.29 ms  
^C  
— 192.168.56.101 ping statistics —  
5 packets transmitted, 5 received, 0% packet loss, time 4023ms  
rtt min/avg/max/mdev = 0.945/1.984/3.161/0.850 ms
```

### 2. Start sshd in debugmodus (voor live logging tijdens aanval):

```
sudo /usr/sbin/sshd -ddd
```

```
osboxes@osboxes:~$ sudo /usr/sbin/sshd
```

Als je een fout zou krijgen zoals: **Bind to port 22 failed: Address already in use** Stop dan de actieve SSH-service:

```
sudo systemctl stop ssh  
sudo systemctl disable ssh
```

Na het uitvoeren van deze commando's zou alles correct moeten verlopen.

### 3. Voer de exploit uit

op de doelmachine via netwerkinterface eth1 worden met 200 gelijktijdige verbindingen geconecteerd om de race condition te triggeren.

```
python3 CVE-2024-6387.py exploit -T 192.168.56.101 -p 22 -n eth1 -s 200
```

```
File Actions Edit View Help  
(osboxes@osboxes)-[~/Downloads]  
$ python3 CVE-2024-6387.py exploit -T 192.168.56.101 -p 22 -n eth1 -s 200 -t 3
```

### 4. Monitor ondertussen de debug-output in Debian:

Let op onderstaande signalen in de debug-output, die wijzen op succesvolle exploitatie.

- padding error
- ssh\_dispatch\_run\_fatal
- message authentication code incorrect
- killing privsep child

```
debug3: send packet: type 20 [preauth]
debug1: SSH2_MSG_KEXINIT sent [preauth]
padding error: need 37 block 8 mod 5 [preauth]
debug3: send packet: type 1 [preauth]
ssh_dispatch_run_fatal: Connection from 192.168.56.102 port 36164: message authentication code incorrect [preauth]
debug1: do_cleanup [preauth]
debug1: monitor_read_log: child log fd closed
debug3: mm_request_receive: entering
debug1: do_cleanup
debug1: Killing privsep child 1910
osboxes@osboxes:~$ _
```

2. Cheatsheet

Handige commando's en referenties:

Commando	Omschrijving
ssh -p 2222 osboxes@127.0.0.1	Maakt verbinding met de Debian VM via poort 2222 (port forwarding)
nano install_openssh_8.5p1.sh	Opent een nieuw scriptbestand om OpenSSH 8.5p1 te installeren
chmod +x install_openssh_8.5p1.sh	Maakt het script uitvoerbaar
sudo ./install_openssh_8.5p1.sh	Voert het installatiescript uit als root
sudo /usr/sbin/sshd -ddd	Start sshd in debugmodus voor live monitoring van de aanval
sudo systemctl stop ssh && sudo systemctl disable ssh	Stopt en schakelt de standaard SSH-service uit om conflicten te vermijden
ping 192.168.56.101	Test de netwerkverbinding vanaf Kali naar Debian
python3 CVE-2024-6387.py exploit -T 192.168.56.101 -p 22 -n eth1 -s 200	Voert de exploit uit vanaf Kali met 200 gelijktijdige verbindingen

Zeker, hier is een nog duidelijker en verder opgesplitste versie:

3. Mogelijke gevaren en oplossingen

3.1 Gevaren



- **Race condition in OpenSSH** Een fout in de `sshd`-component veroorzaakt een race condition die remote code execution zonder authenticatie mogelijk maakt.
  - **Volledige roottoegang** Aanvallers kunnen hiermee volledige controle over het systeem krijgen.
  - **Massale exploitatie** De aanval kan automatisch en op grote schaal plaatsvinden, vooral op publiek toegankelijke servers.
  - **Moeilijke detectie** Door het karakter van de race condition blijven sporen in logbestanden vaak uit, waardoor detectie lastig is.
- 

## 3.2 Oplossingen

- **Toegangsbeperking**
    - SSH-toegang beperken tot VPN of specifieke IP-adressen.
    - Firewallregels toepassen om verbindingen te reguleren.
  - **Configuratie-aanpassingen**
    - Verlaag `LoginGraceTime` in `sshd_config`.
    - Beperk `MaxStartups` om overbelasting tegen te gaan.
  - **Beveiligingsmaatregelen**
    - Gebruik monitoring om verdachte SSH-activiteit te signaleren.
    - Activeer SELinux of AppArmor voor extra systeembeveiliging.
    - Zet fail2ban in om brute-force aanvallen te blokkeren.
  - **Voorbereiding en beheer**
    - Maak regelmatige back-ups en sla deze extern op.
    - Gebruik een extern logging systeem (SIEM) voor analyse en detectie.
    - Automatiseer patchbeheer om updates snel toe te passen.
  - **Strategische aanpak**
    - Implementeer een Zero Trust beveiligingsmodel voor een gelaagde verdediging.
- 

## 4. Samenvatting

Deze handleiding beschrijft het opzetten en testen van de kwetsbaarheid **CVE-2024-6387** op een Debian 11 (32-bit) VM, aangevallen vanuit een Kali Linux VM. Het doel is om de race condition in OpenSSH 8.5p1 te demonstreren.

### 4.1 Aanmaken

1. **Download** de Kali Linux (64-bit) en Debian 11 Bullseye (32-bit) VDI-bestanden van [osboxes.org](https://osboxes.org).
2. **Pak** de gedownloade .7z-bestanden uit naar een bekende map, bijvoorbeeld `/vdi-files/Kali/` en `/vdi-files/Debian/`.

3. **Pas** in het script `Build-VM.sh` de paden aan naar de uitgepakte VDI-bestanden.
4. **Zorg** dat `VBoxManage` via de terminal toegankelijk is (voeg het pad toe aan je PATH).
5. **Voer** het script `Build-VM.sh` uit om de VM's automatisch aan te maken en configureren in VirtualBox.
6. **Start** de VM's handmatig via de VirtualBox GUI.
7. **Log in** met gebruikersnaam en wachtwoord `osboxes` / `osboxes.org`.

## 4.2 Aanval

- Start beide VM's (Kali & Debian) in VirtualBox
- Installeer OpenSSH 8.5p1 op Debian via script
- Sluit SSH-verbinding met Debian zodat poort 22 vrij is voor de aanval
- Start `sshd` in debugmodus op Debian

```
sudo /usr/sbin/sshd -ddd
```

- Voer exploit uit vanaf Kali

```
python3 CVE-2024-6387.py exploit -T 192.168.56.101 -p 22 -n eth1 -s 200
```

- Observeer in Debian de volgende signalen (debug-output):
  - `padding error`
  - `message authentication code incorrect`
  - `ssh_dispatch_run_fatal`
  - `killing privsep child`

---

## 5. Opmerkingen

- De gebruikte PoC werkt enkel op **32-bit Linux-systemen** met **glibc** en de kwetsbare versie van OpenSSH (8.5p1). De meeste moderne 64-bit systemen zijn hiermee niet kwetsbaar.
- Voor een succesvolle demonstratie is het cruciaal dat `sshd` **manueel wordt gestart** (bijv. met `sshd -ddd`) zodat je live kunt observeren wat er gebeurt.
- Tijdens de aanval mogen er **geen actieve SSH-sessies** zijn, anders kan het exploit-effect (zoals race condition of crash) uitblijven.
- De CVE richt zich voornamelijk op **pre-auth heap corruptie**, met aangepaste payloads zou dit potentieel tot **Remote Code Execution** kunnen leiden.

**!!! Deze GitHub-repository is uitsluitend bedoeld voor educatieve doeleinden en mag niet worden gebruikt voor kwaadwillige of illegale activiteiten.**

---

## 6. Makers van het Project

Deze handleiding is gebaseerd op de resources en scripts die beschikbaar zijn op de GitHub-pagina van het project:

- GitHub Repository: [NPE-Cybersecurity](#)
  - Auteurs: Joran Van Goethem, Leander Counye en Vincent Cammaert
- 

## 6. Bron van het Python-script

Het Python-script CVE-2024-6387.py is geïnspireerd door de repository van de volgende auteur:

- GitHub Repository: [CVE-2024-6387 Exploit](#)
- Auteur: Karmakstylez