



Delinea

# IoT Security Challenges: The Risks and How to Minimize Them

An ethical hacker's guide to IoT security risks



By Joseph Carson

WHITEPAPER

# IoT Security Challenges: The Risks and How to Minimize Them

## An Ethical Hacker's Guide to IoT Security Risks

Welcome to the world of IoT (Internet of Things). More devices get connected every minute, with over nine-billion devices automatically performing all sorts of tasks to improve productivity.

Each day, your employees power up their devices and connect to the internet. They check the news, receive and respond to emails, chat with colleagues, pay invoices, work, shop, listen to music, and stream the news; the list goes on and on.

## The Internet of Things in business, health, and infrastructure

In the past few years, new technologies connect to the internet, collect vast amounts of data and send it across the world to be analyzed, monetized, used to improve daily life, and sometimes—stolen. IoT tech includes medical and health devices, engines, power stations, wind turbines, transportation, financial applications, CCTV, and consumer technology.

Smart cities rely on the industrial internet to improve service delivery. For example, autonomous vehicles communicate with transportation infrastructure to help manage traffic lights, send weather condition alerts, and ensure the most efficient traffic flow.

The challenge facing organizations that use IoT devices to support their operations, such as Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA), sensors and Programmable Logic Controllers (PLCs), is that these systems are typically designed to work over a long production life cycle, usually between seven to 20 years. Ease of use and robust design take priority over security. In almost all scenarios, security is an afterthought.

IoT security is also a concern for consumer technology companies. When vendors stop supporting their smart devices and security updates end, devices become

vulnerable to cyber attacks. This has already happened with webcams, mobile phones, tablets, and smart home devices that typically only have a few years of warranty and support. A recent security vulnerability in Western Digital MyBook Live devices resulted in many people having their devices remotely wiped and sensitive data lost.

Many IoT systems and devices are running on legacy operating systems, including old Linux versions, Windows 7 and even Windows XP. Firmware often contains hard-coded passwords. Web interfaces are running over insecure HTTP. Security controls involve only a simple PIN and no authentication integration or encryption.

This lack of security might have worked when a traditional perimeter could be controlled and secured from outside abuse. However, with today's cloud, mobile and always-on connectivity, it's nearly impossible to protect systems that are constantly exposed to the public internet.

Risks to IoT devices and systems are very high. If you're deploying IoT, it's critical to address the security challenges along with enjoying the productivity benefits.

**In this eBook, you'll learn how you can assess the security of your IoT systems by viewing them through the lens of an ethical hacker.**

## IoT and the balance of security

The types of functions and tasks IoT-connected devices carry out have changed dramatically. In the past, a single computer could be programmed to perform multiple, complex functions, such as managing web applications, financial applications or databases. Today, devices and their associated hardware are designed for very specific functions, often simple tasks. As a result, many more microsystems and microprocessors have been put in place and expand your attack surface.

Each IoT device and system has a different security risk profile. To determine the risk, first, look at the actual role it's meant to perform and ask: Is it a data processor? Is it a data collector? Is it a data correlator?

Beyond looking at their function, to determine the level of risk, you also need to assess:

- Is this device something that could potentially attack the network?
- Is it something that could be vulnerable to data poisoning?
- Can the data it's generating be manipulated?
- Is it providing an access point for an attacker to gain access to the larger network?

It's crucial to make sure you understand the type of data devices collect, if data can be modified, any function changes the devices enable, if devices could be used in a DDoS attack, or if an attacker could abuse the device to gain access to the wider network.

Bottom line: You should always perform a risk analysis of any devices that you deploy and use.

## Top risks of IoT

The Internet of Things introduces several new risks that make them vulnerable to compromise and open the door to new techniques for cyber attack.

To help you make effective security decisions when creating, deploying, or using IoT devices, The Open Web Application Security Project (OWASP) created the OWASP Internet of Things Project. The OWASP Top 10 for IoT helps you structure your risk assessments by focusing on the top IoT risks.

Mapping your IoT devices to the [OWASP Top 10](#) is an important first step that will help you determine what type of security controls you need to put in place for each device and system.

### OWASP Top 10 Security Risks for IoT

1. Weak, Guessable, or Hardcoded Passwords
2. Insecure Network Services
3. Insecure Ecosystem Interfaces
4. Lack of Secure Update Mechanism
5. Use of Insecure or Outdated Components
6. Insufficient Privacy Protection
7. Insecure Data Transfer and Storage
8. Lack of Device Management
9. Insecure Default Settings
10. Lack of Physical Hardening

## IoT Through the Eyes of a Hacker

To secure IoT devices against cyber attack, it helps to think like a cyber criminal. The ethical hacking community has collected numerous tools and techniques to help organizations reduce IoT-related risk.

Let's explore the strategies and resources ethical hackers use at each stage of an ethical hack:

Stage 0: Pre-Engagement

Stage 1: Passive Recon

Stage 2: Hardware

Stage 3: Firmware Boot

Stage 4: Firmware Analysis and Reverse Engineering

Stage 5: Firmware Flashing

Stage 6: Network and Radio Frequencies

Armed with this knowledge, you'll be able to try your hand at an ethical hack to test your own organization's defenses. Most importantly, you'll also learn IoT security strategies that block malicious hackers from reaching their goals.

### STAGE 0: Pre-Engagement

The first step in an ethical hack is to determine the goals, target, and scope of your activities. You should always ensure you do no harm. So, make sure your plan adheres to an ethical code and stays within legal boundaries.

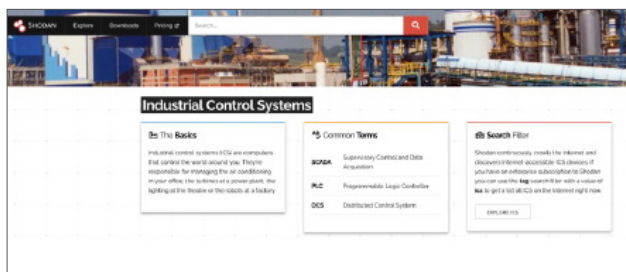
Confirm you have the proper permission from your organization to run any hacking tools you'll use. Are you permitted to target employees, or only systems and applications? This typically depends on how closely you want to simulate a real-world attack.

If your organization has a lab environment, test your toolset there first, before you begin the active hack. If

your actions trigger alarms, you'll know security controls are doing their job. If not, there may be misconfigurations that you will want to investigate further.

### STAGE 1: Passive Recon

This next step applies to practically all engagements. For IoT, this usually means learning about what devices are being used, versions used, locations and configurations. Open-Source Intelligence (OSINT) is a technique to gather publicly available information.



OSINT is critical to any risk assessment of a company's security, especially the hardening of systems and devices. With OSINT, a hacker can obtain publicly available information to gain network access. The more information you gather, the better prepared you'll be. At times I have found that when one attack path doesn't work, a solid OSINT review allows me to adapt quickly.

Shodan is another great tool you can use to discover devices connected to the public internet and what ports are open.

### STAGE 2: Hardware - Opening the Devices to Discover What's Inside

Once you have acquired a solid digital footprint on the devices being used, the next step is to learn as much about the devices as possible.



## Images and Design

You can learn a lot just by looking at the devices or, if you know the FCC ID, by searching the FCC.io. This can sometimes show you more information on the images, such as UART (Universal Asynchronous Receiver-Transmitter) or JTAG (Joint Test Action Group) ports.

Let's take a closer look at several devices and some methods to gather information:

- Documentation
- FCC Filings
- Online Reviews
- Patents
- Hands-On

One of my handy tools is an iFIXIT toolkit, which includes several magnifying glasses, and a device holder to keep things steady. (Tip: Make sure you know how to open the toolkit without breaking the plastic clips.)



*Toolkit to help open cases*

A good magnifying glass is helpful to see the small details. Looking for possible interface ports, chips, and memory can provide information on how a device is made.

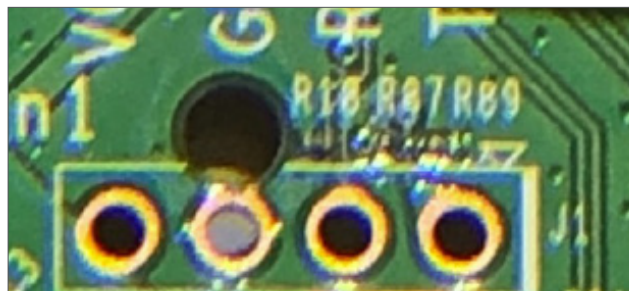
I tend to look for how I can connect directly to the device. Look for G = Ground, Tx = Transmit, Rx = Receive and V = Voltage. The power supply is typically 3.3v or 5v, but it's best to start with the lowest voltage; otherwise, you could burn out the device.

The image below shows an example of a webcam that has a UART interface identified by "console." However, the manufacturer of this device has made it a little more difficult to connect, and this one requires some soldering skills.



*UART Interface for a Webcam*

Some devices display better labeling and are easier to connect to without requiring soldering. This example shows how you can simply connect without soldering by using clips or gator grips to ensure you have a strong connection.



*A Router that easily identifies UART*

Some devices don't easily identify their interface, so you'll have to conduct some trial and error using a Logic Analyzer or an oscilloscope. I find the Kingst Logic Analyzer sufficient, though there are many choices available. An example of this is shown below.



*Connecting Logic Analyzer to the device*

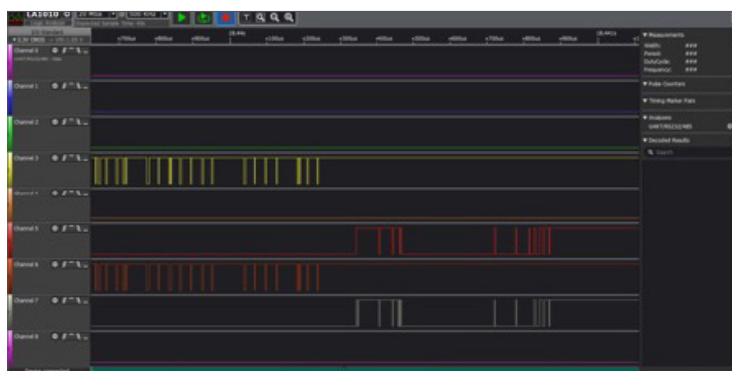
The example below shows the data collected from a router UART interface.

- Channel 3 = Yellow = VCC
- Channel 5 = Red = Rx
- Channel 6 = Orange = Tx
- Channel 7 = Grey = Ground

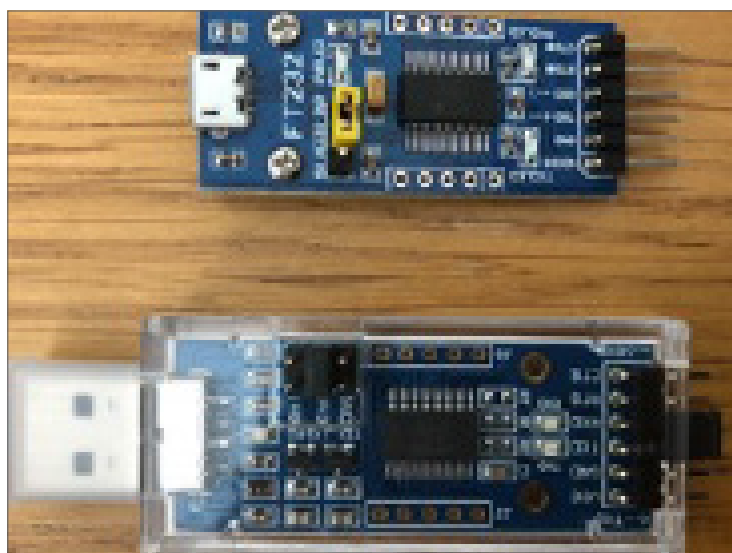
Once you've identified the interfaces, you can now connect them to an FDTI USB device such as the FT232RL. Connect to the correct pins that you will use to communicate with the device.

In the example to the right, I connected my FT232 to the interface on the router after using the Logic Analyzer to ensure I have the correct interfaces. Once connected, I communicate with the device using the command "sudo minicom -device /dev/ttyUSB0." The ttyUSBx may be different on your machine depending how many USB devices you have connected.

Once connected, I can now observe the device getting booted.



Logic Analyzer collecting sample data



FT232 USB Serial Interface

## STAGE 3: Firmware Boot

Once booted, I can interact with the device via the command prompt. I can check configuration files, versions, history, and log files, etc. For example, to the right I have searched the directories for interesting files and found that I can read the passwd file for users and passwords.

```
U-Boot 1.1.3 (Jun 14 2018 - 11:06:28)

Board: Ralink APSoC DRAM: 32 MB
relocate_code Pointer at: 81fc0000
flash manufacture id: 1c, device id 70 16
Warning: un-recognized chip ID, please update bootloader!
=====
Ralink UBoot Version: 4.3.0.0
=====

ASIC 7628_MP (Port5<->None)
DRAM component: 256 Mbits DDR, width 16
DRAM bus: 16 bit
Total memory: 32 MBytes
Flash component: SPI Flash
Date:Jun 14 2018 Time:11:06:28
=====
icache: sets:512, ways:4, linesz:32 ,total:65536
dcache: sets:256, ways:4, linesz:32 ,total:32768

#### The CPU freq = 580 MHZ ####
estimate memory size =32 Mbytes
RESET NT7628 PHY!!!!!!
continue to starting system.
```

Reading Device Directories and passwd file

Knowing the device users and password hashes, I can take those hashes. If they're weak or previously compromised credentials, they'll be easy to crack.

Let's move those to our "kracken" machine and see if we can crack them.

Using John, I can target the discovered hash for the admin user.

As you can see, the password was easy to crack since it was a weak credential which should never have been used in production.

User = Admin

Password = 1234

```
~ # pwd
/
~ # ls
web      usr      sbin     mnt      lib      dev
var      sys      proc     linuxrc  etc      bin
~ # cd etc
/etc # ls
services      init.d
samba         group
resolv.conf   fstab
reduced_data_model.xml  default_config.xml
ppp           TZ
passwd.bak    SingleSKU_CE.dat
passwd        RT2860AP.dat
iptables-stop MT7628_EEPROM_20140317.bin
inittab       MT7628_AP_2T2R-4L_V15.BIN
/etc # cat passwd
admin:$1$iC.dUsGpxNNJGeOm1dFio/:0:0:root:/:/bin/sh
dropbear:x:500:500:dropbear:/var/dropbear:/bin/sh
nobody:*:0:0:nobody:/:/bin/sh
```

*Reading Device Directories and passwd file*

```
kali@kali:~/lab/endpoint$ sudo john rtrhash
[sudo] password for kali:
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3)) $1$ (and variants) [MD5 128/128 AVX 4x3]
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 41 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234 (admin)
1g 0:00:00:00 DONE 2/3 (2021-08-04 07:13) 3.225g/s 17141p/s 17141c/s 17141C/s 123456..knight
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

*Using John to Crack the Password*

## Recommendation:

Far too many organizations make the mistake of not changing default credentials, using easy passwords that can be cracked, or using the same credentials on all devices to make them easier to manage.

Manage IoT device credentials using strong, unique passwords that are different for all devices. You can make it easy on yourself by using a password manager or a Privileged Access Management solution to help manage IoT devices throughout your environment.



## STAGE 4: Firmware Analysis and Reverse Engineering

Another technique for compromising an IoT device involves downloading the firmware from the vendor's download site(s). By getting access to the firmware binary, you can analyze it and extract it.

In this example, I've used a Netgear D6000 router firmware.

First, I download the firmware and extract the zip. I can then run a check on the file details:

```
kali@kali:~/hardware/firmware$ file D6000-V1.0.0.41_1.0.1.bin
D6000-V1.0.0.41_1.0.1.bin: data
kali@kali:~/hardware/firmware$
```

I can also run strings to extract the readable characters from the binary.

```
kali@kali:~/hardware/firmware$ strings D6000-V1.0.0.41_1.0.1.bin > strings.txt | head strings.txt
2RDH
VERSION:V1.0.0.41_1.0.1
HW_ID:NETG01
REGION:WW
MODELNAME:D6000
```

I can now run binwalk, which will search binary images of embedded files and code.

```
kali@kali:~/hardware/firmware$ binwalk -e -M D6000-V1.0.0.41_1.0.1.bin

Scan Time:      2021-09-21 08:15:28
Target File:    /home/kali/hardware/firmware/D6000-V1.0.0.41_1.0.1.bin
MD5 Checksum:  5be7bba89c9e249ebef73576bb1a5c33
Signatures:    391
```

DECIMAL	HEXADECIMAL	DESCRIPTION
264704	0x40A00	Certificate in DER format (x509 v3), header length: 4, sequence length: 2
674064	0xA4910	Certificate in DER format (x509 v3), header length: 4, sequence length: 32
2280400	0x22CBD0	Certificate in DER format (x509 v3), header length: 4, sequence length: 4
3407888	0x340010	Linux kernel version 2.6.36
3471968	0x34FA60	CRC32 polynomial table, little endian
3853156	0x3ACB64	Neighborly text, "NeighborSolicitstunnel6 init(): can't add protocol"
3853176	0x3ACB78	Neighborly text, "NeighborAdvertisementst add protocol"
3855047	0x3AD2C7	Neighborly text, "neighbor %.2x%.2x.%pM lostrename link %s to %s"
4034736	0x3D90B0	Intel x86 or x64 microcode, pf_mask 0x100, 1C00-17-30, rev 0x0100, size 2048

After running binwalk I now see a squashfs-root folder.

```
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted$ ls
100 100.7z 15A6D2.squashfs squashfs-root
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted$ cd squashfs-root/
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root$ ls
bin boardroot dev etc firmware_version lib linuxrc proc sbin sys tmp userfs usr var
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root$
```



From the squashfs-root I can search for files named 'passwd.' See the results below.

```
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root$ find . -name passwd
./usr/etc/passwd
./usr/bin/passwd
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root$
```

I can now check the contents of the usr/etc/passwd file to see if I find anything interesting that I can use to exploit credentials.

```
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root/usr/etc$ cat passwd
admin:$1$iC.dUsGpxNNJGeOm1dFio/:0:0:root:/:/bin/sh
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root/usr/etc$
```

I can now use John to try again and crack the password.

```
kali@kali:~/hardware/firmware/_D6000-V1.0.0.41_1.0.1.bin.extracted/squashfs-root/usr/etc$ sudo john passwd
[sudo] password for kali:
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 84 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234 (admin)
lg 0:00:00:00 DONE 2/3 (2021-08-04 08:16) 10.00g/s 55440p/s 55440c/s 55440C/s 123456..larry
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

As you can see, this is another weak credential that has resulted in a cracked password for the admin account.

The credentials in this example are the same:

User = Admin

Password = 1234

## Recommendation:

Always avoid using default or weak credentials in production environments on IoT devices since attackers can easily gain access.

## STAGE 5: Firmware Flashing

Firmware is software that enables control of a device's hardware. Most hardware requires firmware to function, such as the BIOS for most personal computers and other firmware that can include a complete operating system.

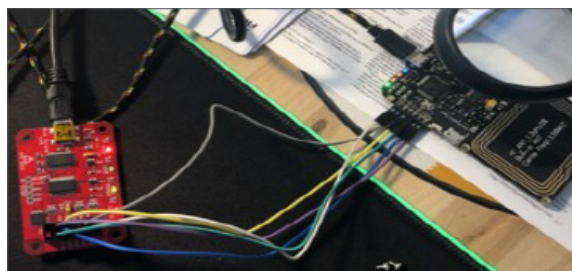
Unfortunately, it's easy to modify firmware using devices such as the Bus Pirate. Flashing Firmware of the Proxmark v3 using Bus Pirate is shown below. This can allow an attacker to modify functionality, change configurations and embed backdoors.

The Proxmark is an RFID device used to interact with many RFID tags and systems that use RFID.

Bus Pirate is an open-source multipurpose tool for programming, analyzing, and debugging many IoT devices. It's simply a protocol emulator that supports many different interfaces, including UART, SPI, I2C and JTAG.

Bus Blaster is a JTAG debugger. The Bus Pirate also supports JTAG, but it is extremely slow. The Bus Blaster is much better at debugging JTAG.

[GreatFET One](#) is a hardware hacking open-source tool that includes a programmable digital I/O, serial protocols supported such as SPI, I2C, UART, and JTAG. It includes many more features and is readily customizable.



*Flashing Proxmark using Bus Pirate*



*Bus Pirate, Bus Blaster and GreatFET One*

## STAGE 6: Network and Radio Frequencies

Most IoT devices need to communicate online, whether it's to collect and process sensor data, receive commands via an API, get new software updates, or update configuration settings. Most devices communicate via WIFI, 4G, 5G, Bluetooth, RFID, or Radio.

An attacker can easily intercept data communicated via these channels, so it's extremely important that IoT devices use encryption. However, many devices don't. For example, the screenshot below illustrates how a tool called Wireshark captures the traffic between an IoT device and the vendor for a software update.

Thus, it's critical that you know whom these devices communicate with and how they get updated. Attackers could infiltrate the supply chain or redirect the communication to send malicious updates that include backdoor malware. The infamous SolarWinds Security Incident from 2020 is an example of this kind of hack.

No.	Time	Source	Destination	Protocol	Length	Info
23	19.448125	10.10.10.11	54.244.232.183	TCP	74	47806 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4294916059 TSecr=0
24	19.611848	54.244.232.183	10.10.10.11	TCP	74	80 → 47806 [SYN, ACK] Seq=0 Ack=1 Min=14488 Len=0 MSS=1386 SACK_PERM=1 TSval=219502
25	19.613691	10.10.10.11	54.244.232.183	TCP	66	47806 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=4294916101 TSecr=21950274
26	19.614375	10.10.10.11	54.244.232.183	HTTP	203	GET /router/firmware/query.aspx?model=DCR-S160_Ax_Default_FW_0120_6C7220C5175A HTTP
28	19.777884	54.244.232.183	10.10.10.11	TCP	66	80 → 47806 [ACK] Seq=1 Ack=138 Win=15616 Len=0 TSval=21906315 TSecr=4294916101
29	19.782866	54.244.232.183	10.10.10.11	HTTP	267	HTTP/1.1 404 Not Found (text/plain)
30	19.786161	10.10.10.11	54.244.232.183	TCP	66	47806 → 80 [ACK] Seq=138 Ack=202 Win=5840 Len=0 TSval=4294916144 TSecr=21950316
36	20.618512	10.10.10.11	54.244.232.183	TCP	66	47806 → 80 [FIN, ACK] Seq=138 Ack=282 Win=5840 Len=0 TSval=4294916352 TSecr=21950316
38	20.782873	54.244.232.183	10.10.10.11	TCP	66	80 → 47806 [FIN, ACK] Seq=202 Ack=139 Win=15616 Len=0 TSval=21950566 TSecr=42949163
39	20.783223	10.10.10.11	54.244.232.183	TCP	66	47806 → 80 [ACK] Seq=139 Ack=203 Win=5840 Len=0 TSval=4294916393 TSecr=21950566

```

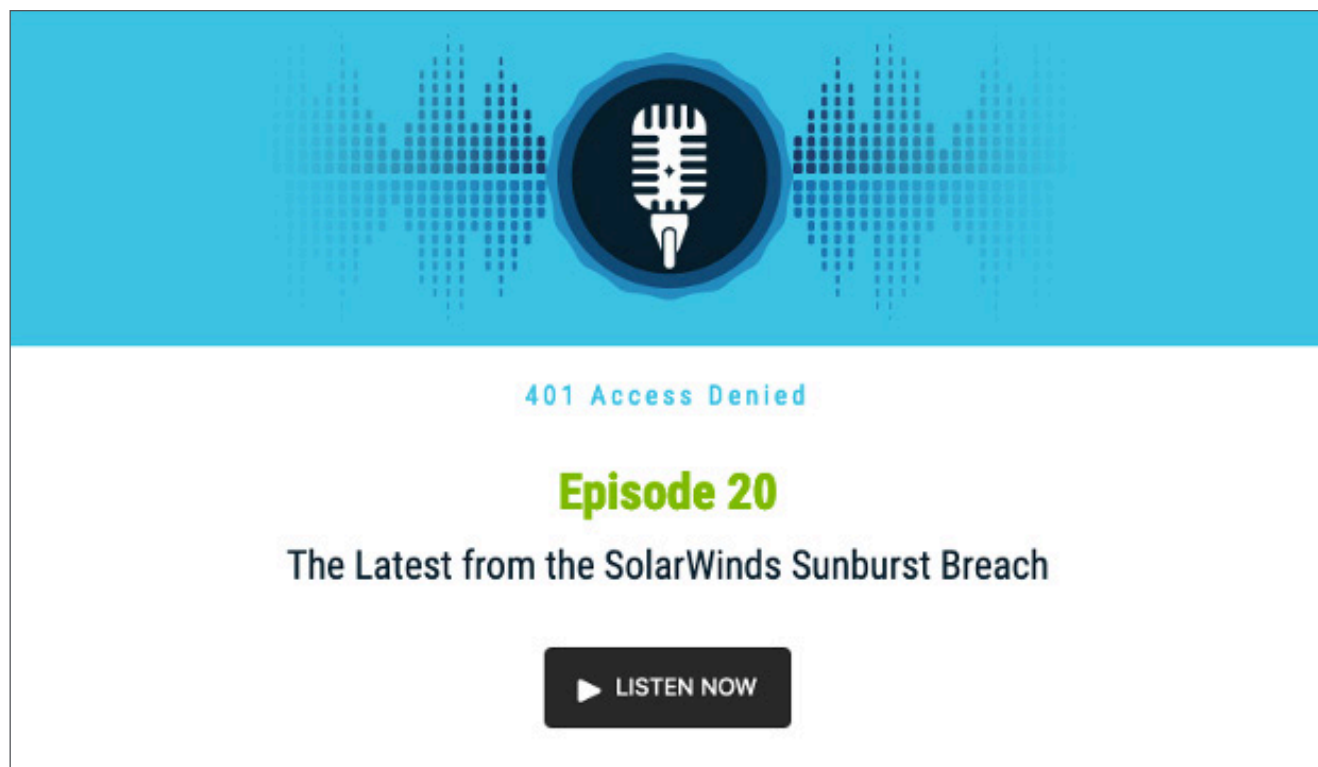
Wireshark - Follow HTTP Stream (tcp.stream eq 0) - MysteryIoTnew.pcap
GET /router/firmware/query.aspx?model=DCR-S160_Ax_Default_FW_0120_6C7220C5175A HTTP/1.1
Host: wrpd.dLink.com
Connection: Keep-Alive

HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
Date: Tue, 08 Sep 2016 18:57:34 GMT
ETag: W/"9-b1748d4d"
X-Powered-By: Express
Content-Length: 9
Connection: keep-alive

```

*IoT Water Sensor checking for Firmware Updates*

You can learn more by listening to [my podcast](#) on the SolarWinds Sunburst breach that affected FireEye, the US government, and thousands of other organizations:



401 Access Denied

**Episode 20**

The Latest from the SolarWinds Sunburst Breach

▶ LISTEN NOW

On the FCC.io page you can also find the frequencies devices communicate. You can then use devices such as the HackRF One to capture the data, which also allows you to replay those signals that could allow opening doors, etc.





*HackRF One with PortaPack*

Other techniques are to capture the WIFI data or modify the data using devices which allow Monitor Mode.



*Wi-Fi Adapters that can Sniff IoT Traffic*

Another common communication radio frequency used is RFID and is typically used for tracking packages, monitoring sensors, door access keycards, and many more features. Several devices such as the Proxmark can be used to capture and read the tags data.

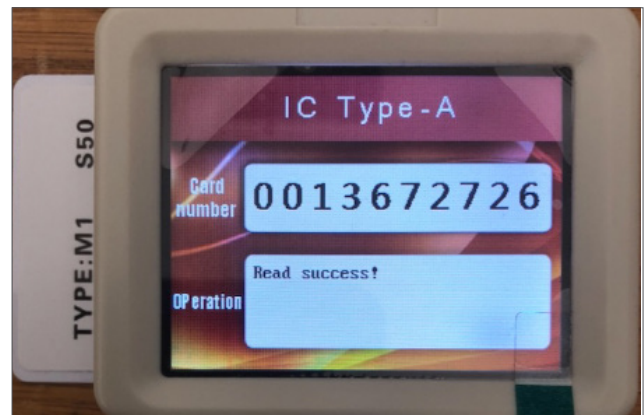


*RFID-Capturing Devices*

Once the data has been captured, it allows the attacker to write or create duplicate cards such as those shown below:



*Cards used to create Duplicate Tags*



*Reading and Cloning RFID Tags*

## IoT Risks and Summary

While rapidly becoming a part of all our professional and private lives, IoT devices are a security risk to most organizations and individuals. To protect your devices and users, you must perform an IoT risk impact assessment for devices you have deployed, are currently using, or planning to deploy. You need to understand exactly what the device's functionality enables and consider turning off specific functionality that isn't being used.

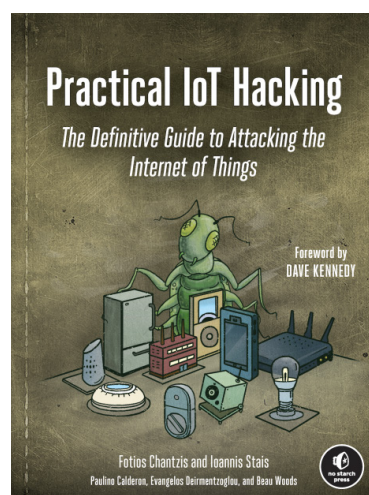
Review the OWASP Top 10 risks to IoT and apply the proper security controls to reduce those risks.

1. Weak, Guessable, or Hardcoded Passwords
2. Insecure Network Services
3. Insecure Ecosystem Interfaces
4. Lack of Secure Update Mechanism
5. Use of Insecure or Outdated Components
6. Insufficient Privacy Protection
7. Insecure Data Transfer and Storage
8. Lack of Device Management
9. Insecure Default Settings
10. Lack of Physical Hardening

Become familiar with the techniques described here and try them on your own devices. Check how your devices get updated—especially from where and how. Know how your devices communicate and whether they are encrypted.

Finally, always change default credentials and consider investing in a privileged access security solution to manage IoT access, create complex, unique credentials for each device, and ensure auditability.

For further learning, I highly recommend reading [Practical IoT Hacking](#) by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods.



## About Joseph Carson

- Chief Security Scientist at Delinea
- Over 25 years' experience in enterprise security
- Author of *Privileged Access Management for Dummies* and *Cybersecurity for Dummies*
- Cyber security advisor to several governments, critical infrastructure, financial and transportation industries
- Speaker at conferences globally

## About Delinea

Delinea is a leading provider of privileged access management (PAM) solutions that make security seamless for the modern, hybrid enterprise. Our solutions empower organizations to secure critical data, devices, code, and cloud infrastructure to help reduce risk, ensure compliance, and simplify security. Delinea removes complexity and defines the boundaries of access for thousands of customers worldwide, including over half of the Fortune 100.

[Delinea.com](https://delinea.com)



Defining the boundaries of access

Delinea is a leading provider of privileged access management (PAM) solutions that make security seamless for the modern, hybrid enterprise. Our solutions empower organizations to secure critical data, devices, code, and cloud infrastructure to help reduce risk, ensure compliance, and simplify security. Delinea removes complexity and defines the boundaries of access for thousands of customers worldwide. Our customers range from small businesses to the world's largest financial institutions, intelligence agencies, and critical infrastructure companies.

Learn more about Delinea's solutions at [delinea.com](https://delinea.com).

© Delinea