A Large-Scale Analysis of the Security of Embedded Firmwares

Andrei Costin, Jonas Zaddach, Aurélien Francillon, Dave Balzarotti, *Eurecom*

Presented by: Zach Day

Introduction and Background

- **Firmware**
  - A “combination of a hardware device and computer instructions or computer data that reside as read-only software on the device.”[†]
  - Software embedded on the device
  - Typically stored on ROM (or more recently, EPROM)
- The entire system *depends* on the firmware
  - Any security flaws affect the entire system
- Pretty much everything electronic nowadays has firmware
  - ⇒ It is Very Relevant to IoT

[†]IEEE Std 610.12-1990

Introduction and Background: Security

Reminder: *security* is many things

Confidentiality: Data is unavailable to third parties

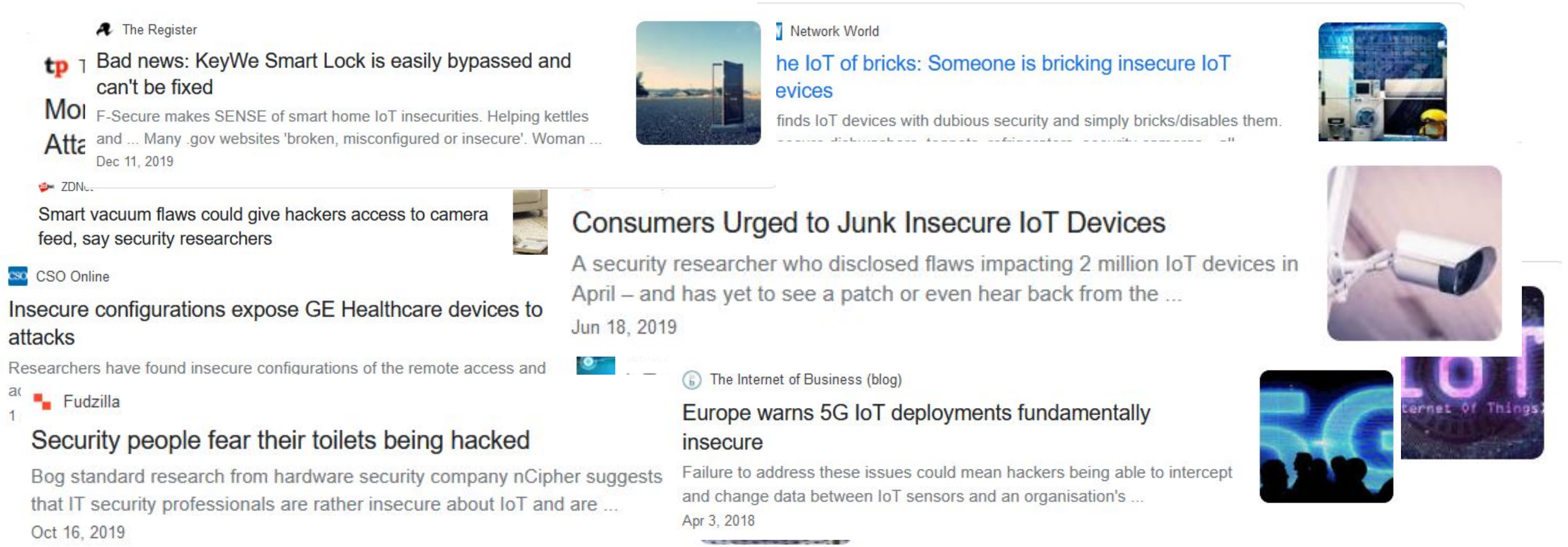
Integrity: Data is not inappropriately modified

Availability: The system works when expected

With this in mind, let’s talk about how security relates to IoT.

Introduction and Background: Security

- IoT firmware has a reputation of being insecure



tp T
Moı
Atta

The Register
Bad news: KeyWe Smart Lock is easily bypassed and can't be fixed
F-Secure makes SENSE of smart home IoT insecurities. Helping kettles and ... Many .gov websites 'broken, misconfigured or insecure'. Woman ...
Dec 11, 2019

Network World
he IoT of bricks: Someone is bricking insecure IoT evices
finds IoT devices with dubious security and simply bricks/disables them.

ZDNe
Smart vacuum flaws could give hackers access to camera feed, say security researchers

Consumers Urged to Junk Insecure IoT Devices
A security researcher who disclosed flaws impacting 2 million IoT devices in April – and has yet to see a patch or even hear back from the ...
Jun 18, 2019

CSO Online
Insecure configurations expose GE Healthcare devices to attacks
Researchers have found insecure configurations of the remote access and
ac
1

Fudzilla
Security people fear their toilets being hacked
Bog standard research from hardware security company nCipher suggests that IT security professionals are rather insecure about IoT and are ...
Oct 16, 2019

The Internet of Business (blog)
Europe warns 5G IoT deployments fundamentally insecure
Failure to address these issues could mean hackers being able to intercept and change data between IoT sensors and an organisation's ...
Apr 3, 2018

But *how* insecure?

Introduction and Background: Security

We lack a solid understanding of the state of IoT security

Most issues are reported through individual security issues

There are many issues that OS patches can't fix: Vulnerable configs, exposed private keys, etc.

Introduction and Background: Surveying IoT

Automating the gathering of general data on IoT firmware is hard

- Lack of representative data set
  - Various operating systems, instruction sets, and custom components
- Firmware identification
  - Difficult to consistently get metadata from devices
  - No version numbers ⇒ harder to track latest version

Introduction and Background: Surveying IoT

Automating the gathering of general data on IoT firmware is hard

- Unpacking firmware is hard
    - Varying formats of data make unified analysis impossible

(important stuff)			
Machine code formats	**Groups of files**	**Resources**	**etc.**
ELF, PE	ZIP, TAR	Configuration files, scripts, images	

We have some solutions but it is overall still a challenge

Introduction and Background: Surveying IoT

Automating the gathering of general data on IoT firmware is hard

- Scalability and computational limits are a thing
  - Comparing every file to every other file results in quadratic scaling on ilfie count

- Once the firmware is collected, confirming vulnerabilities is hard
  - Costly to acquire units to confirm vulnerability
  - Requires manual observation

Introduction and Background: Hacking

How do we unpack arbitrary firmware images?

- Extraction
  - There are many formats for storing data
  - Vendors may invent their own custom storage format
    - Custom compression algorithms
    - Obfuscation
- Observation
  - Forensic strategies
    - *Carving*: perform pattern matching on various offsets within a file to guess where files are
- Other issues
  - If we can't recognize a file format, it's impossible to say whether it's just a resource file (i.e. an image) or code

Introduction and Background: Hacking

How are issues found?

- Dynamic analysis
  - Observing the output or running state of a program to find problematic actions
- Static analysis
  - Looking for problematic patterns in the source or machine code
- Comparison to files with known issues
  - A subtype of static analysis

Introduction and Background: Risk Analysis

How do we compare files?

- String and pattern matching
- Fuzzy hashing
  - Works like a hashing algorithm
  - Similar files will have similar hashes
  - Implementations include *ssdeep*, *sdhash*

Introduction and Background: Risk Analysis

Other important risk analysis techniques

- Password cracking
  - Taking hashed password and finding corresponding plaintext
  - *John the Ripper*
- Certificate observation
  - Certificates and private keys are crucial to securing communication
  - If they are baked into the firmware, an attacker can decrypt communications!

Problem Definition

- There’s a lot of IoT devices
  - There’s a lot of firmwares

- IoT firmwares are known for being insecure

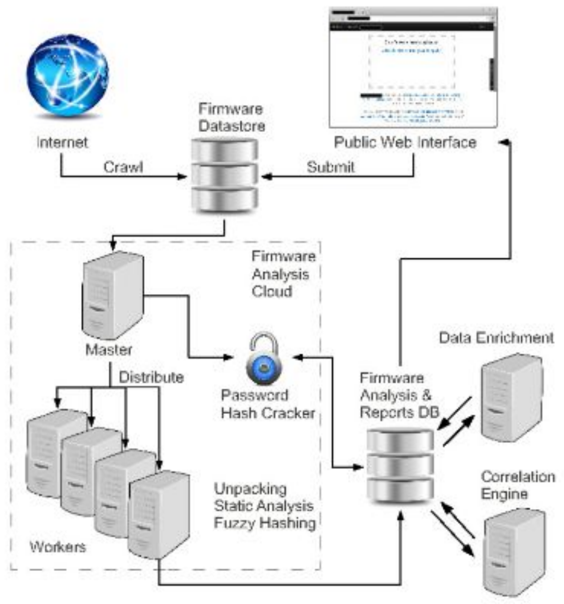
- There is no pre-existing survey of the status of security

Problem Definition

We need to survey the landscape of IoT firmware security.

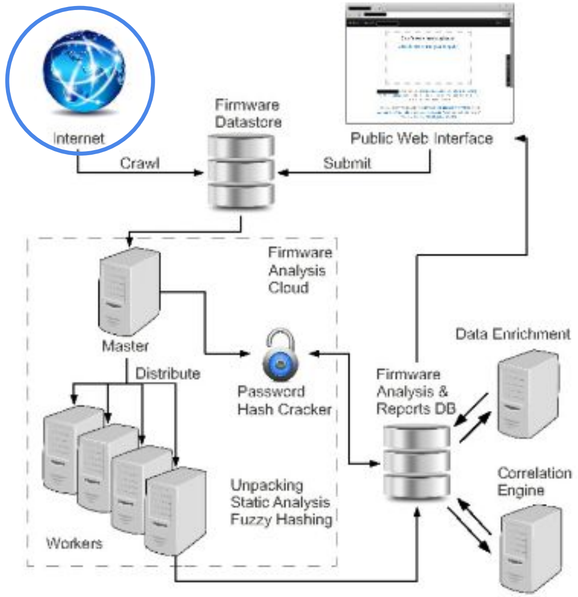
System Design

- Acquisition
  - Web crawler finds firmware files
- Extraction
  - Master node distributes files to worker nodes
- Analysis
  - Worker nodes perform fuzzy hashing and other static analyses
  - Correlation engine draws comparisons between files



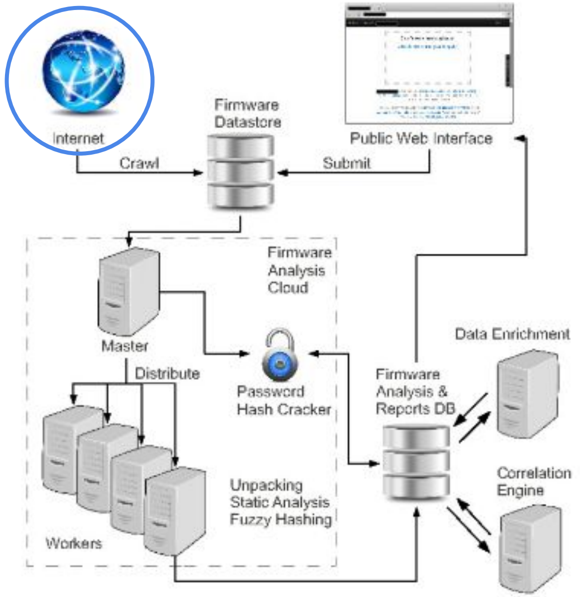
System Design: Web

- Primary goal: automate the retrieval of firmware files
- Obvious solution: a web crawler
  - Seed the crawler with support pages of manufacturers
- Other ideas
  - Public FTP indexing engines
  - Google Custom Search Engines



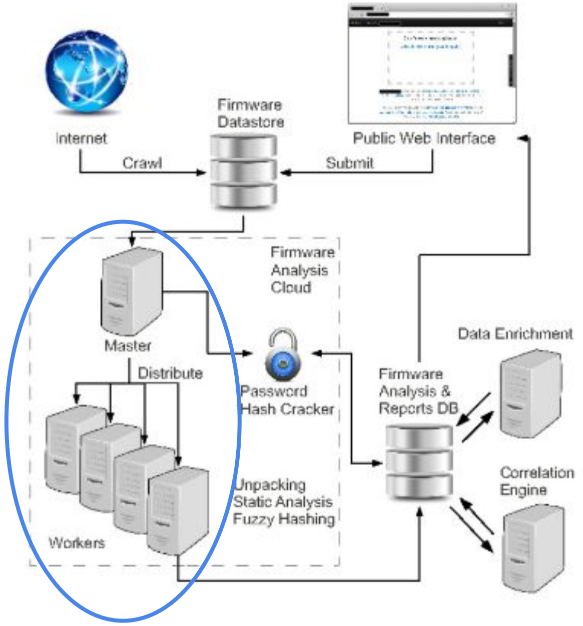
System Design: Web

- Crowdsourcing is a valid technique
- Public submission form
  - This grants access to files that the web crawler can't find



System Design: Distributed computation

- One or more *master nodes*
  - Pick data from the datastore to be processed
- Master nodes send data to *worker nodes*, which perform the computation



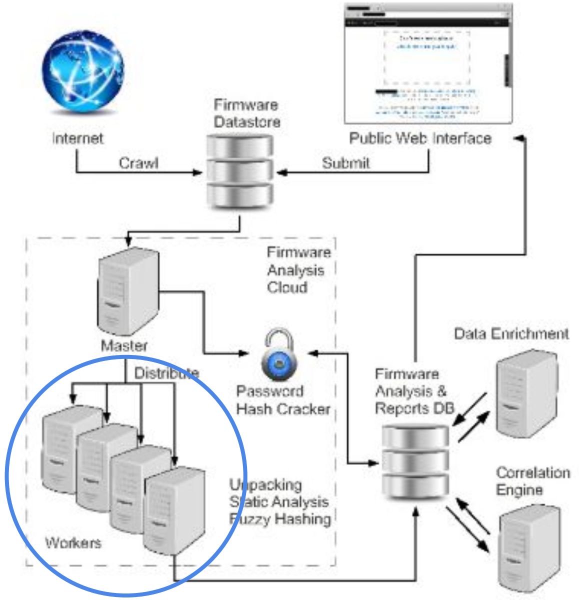
System Design: Unpacking

The unpacking process

- Given a binary blob, split it into files

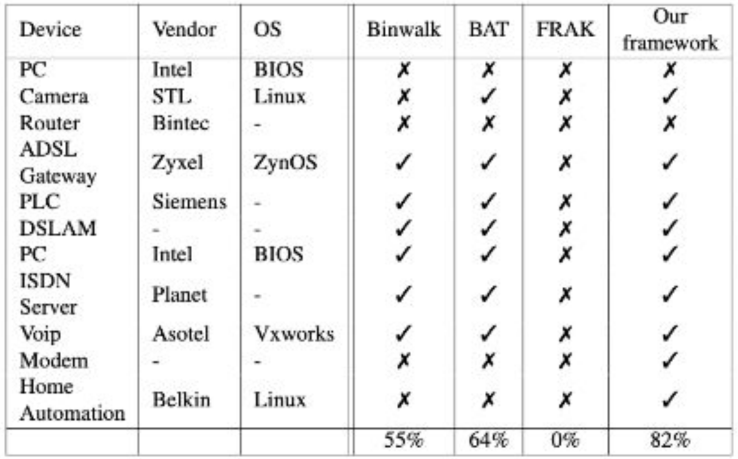
`firmware.bin` $\xrightarrow{\textit{BAT}}$

```
/boot/
  bootloader.efi
  …
/etc/
  ssh/
    authorized_keys
  apache/
    apache.conf
  …
```

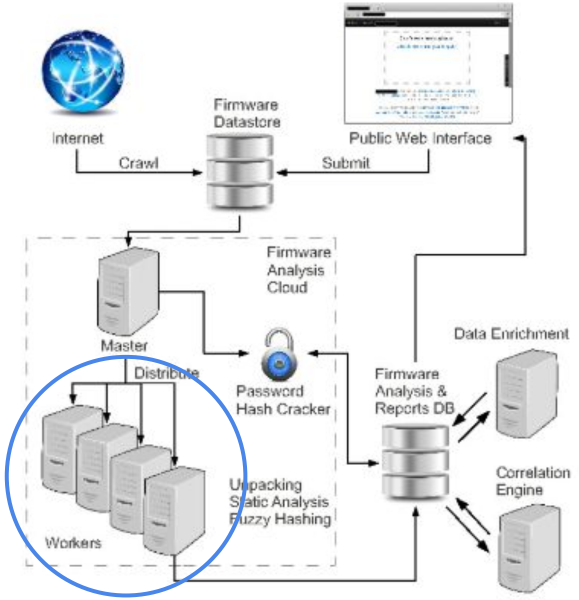


System Design: Unpacking

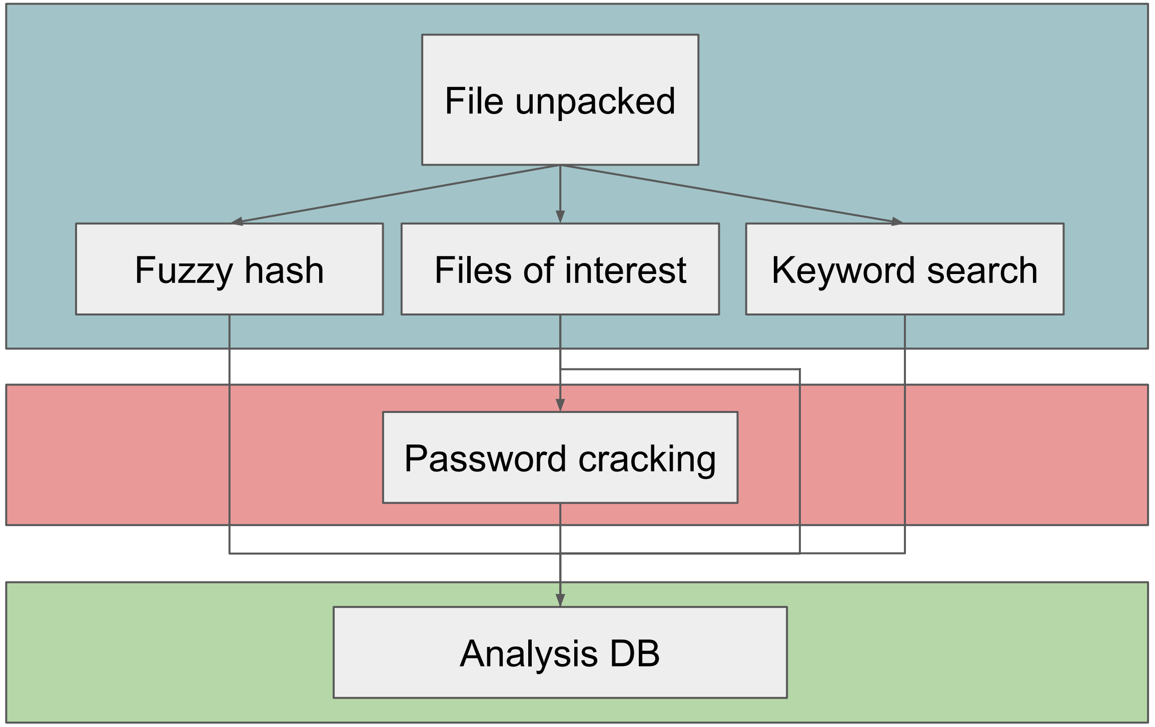
- There are three popular tools for unpacking firmware
  - *binwalk*
  - *FRAK*
  - *Binary Analysis Toolkit (BAT)*

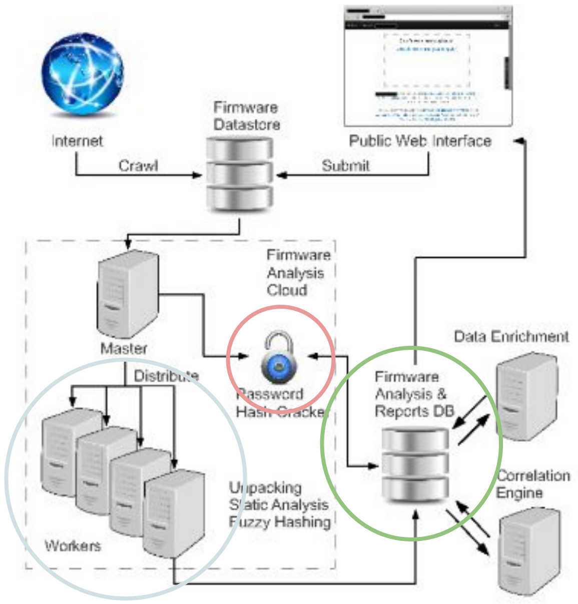


Device	Vendor	OS	Binwalk	BAT	FRAK	Our framework
PC	Intel	BIOS	✗	✗	✗	✗
Camera	STL	Linux	✗	✓	✗	✓
Router	Bintec	-	✗	✗	✗	✗
ADSL Gateway	Zyxel	ZynOS	✓	✓	✗	✓
PLC	Siemens	-	✓	✓	✗	✓
DSLAM	-	-	✓	✓	✗	✓
PC	Intel	BIOS	✓	✓	✗	✓
ISDN Server	Planet	-	✓	✓	✗	✓
Voip	Asotel	Vxworks	✓	✓	✗	✓
Modem	-	-	✗	✗	✗	✓
Home Automation	Belkin	Linux	✗	✗	✗	✓
			55%	64%	0%	82%



System Design: Unpacking

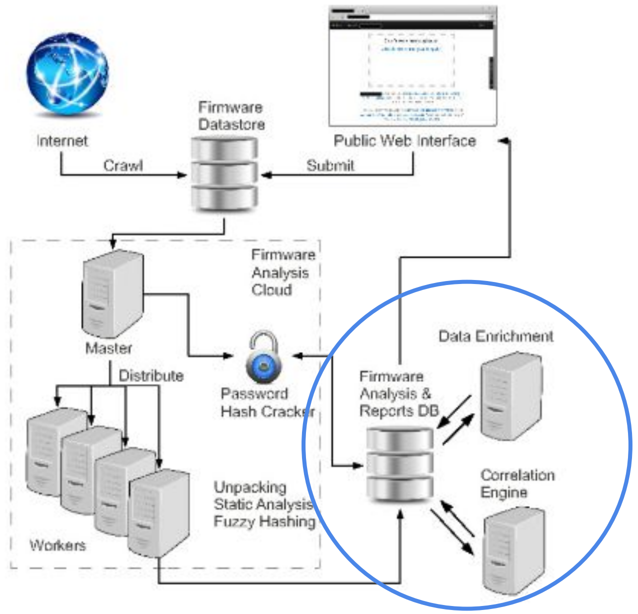




System Design: Correlation

The system correlates on three major axes:

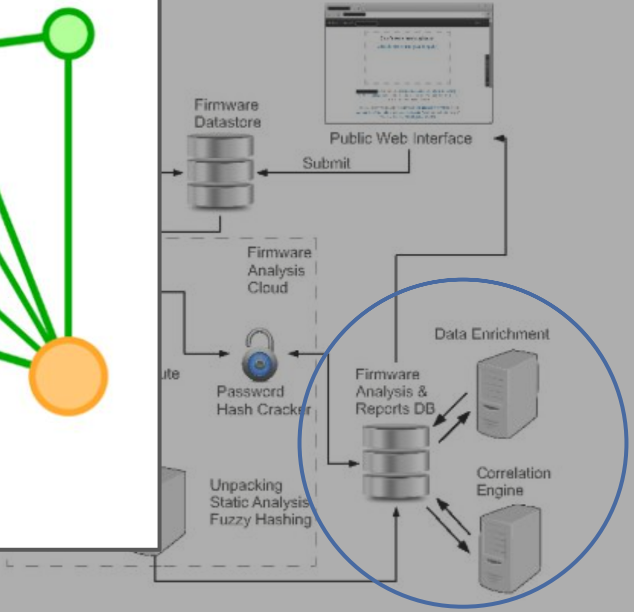
1. Shared credentials
   - Vulnerabilities have been found *across vendors* via correlating on a shared self-signed cert!
2. Keywords
3. Fuzzy hash triage
   - Files with similar fuzzy hashes may have the same vulnerabilities



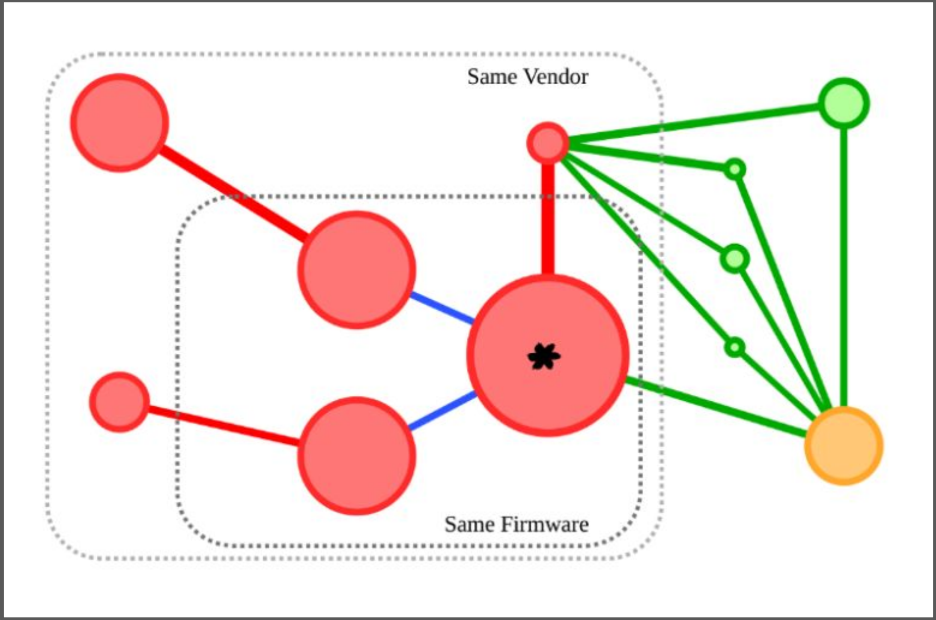
System Desi

The system correla

1. Shared creden
   - Vulnerabilitie
     correlating or
2. Keywords
3. Fuzzy hash tria
   - Files with sim
     vulnerabilities



Fuzzy Hash Correlation



Evaluation

Scale of the project

- 759,273 files, totaling 1.8TB of possible firmware packages
  - 34% (±8%) of the data in that set is actual firmware
- 26,275 / 32,356 images successfully unpacked
- This is a pretty decent sample size, in my opinion

Evaluation: Areas of Failure

The authors note several points of failure for IoT firmware developers

- *Software design*
- *Release management*
- *Infrastructure management*
- *Build management*

Evaluation: Software Configuration

Configuration files of web servers within firmware were analyzed

- **More than 80% were configured with `user=root`**

Other issues include backdoors

- Setting a user agent string to `xmlset_roodkcableoj28840ybtide` (read backwards, “edit by 04882 joel backdoor”) enables remote access
- Half a million users have downloaded the app for this router!

Evaluation: Software configuration

- Compilation banners from firmware images
  - i.e. `root@ubuntu (gcc version 4.2.0)`
- 10 of the 267 unique hostnames resolved to public IP addresses

Evaluation: Passwords

- 100 distinct password hashes acquired
  - 687 images, across 27 vendors
- 58 of those password hashes were cracked
  - Affecting 538 images
- Some of the most popular passwords:

  - `<empty>`
  - `helpme`
  - `pass`
  - `logout`

Evaluation: Certificates and Keys

- Some vendors have been using self-signed certificates
  - 56 self-signed certificates extracted, 41 with their private RSA keys
  - They were able to find 35,000 exposed online devices that use these certificates

Critique

- No machine learning
- Not enough statistics about found issues
  - In general, poor or vague framing of results
- Insufficient description of vulnerabilities found through static analysis

Conclusions

The authors developed...

- an engine for crawling the web for IoT firmware images
- major contributions to BAT
  - Making BAT into an effective tool for security analysis
- a methodology for large-scale surveys of IoT security

Which resulted in…

- the largest known dataset of IoT firmware images
- validation of the poor security reputation held by IoT devices

Questions

- Graham: The paper mentions firmware update sites. How do we update firmware? I thought that it is baked into hardware.
    - That was the original method of storing firmware, but flash and EEPROM have become more viable, allowing for rewriting of the data
- Henry: How does this static analysis work. Once the firmware is unpacked properly other than the correlation engine what is programmatically being done to identify bugs?
    - To my understanding, the did not examine any code to find vulnerabilities (outside of the fuzzy hashing)
- Mike: Do companies really have the word "backdoor" in their backdoor strings that isn't just leftover from testing and is unusable? That seems absolutely ridiculous to ship a product with that still in there.
    - Yeah, sure does...