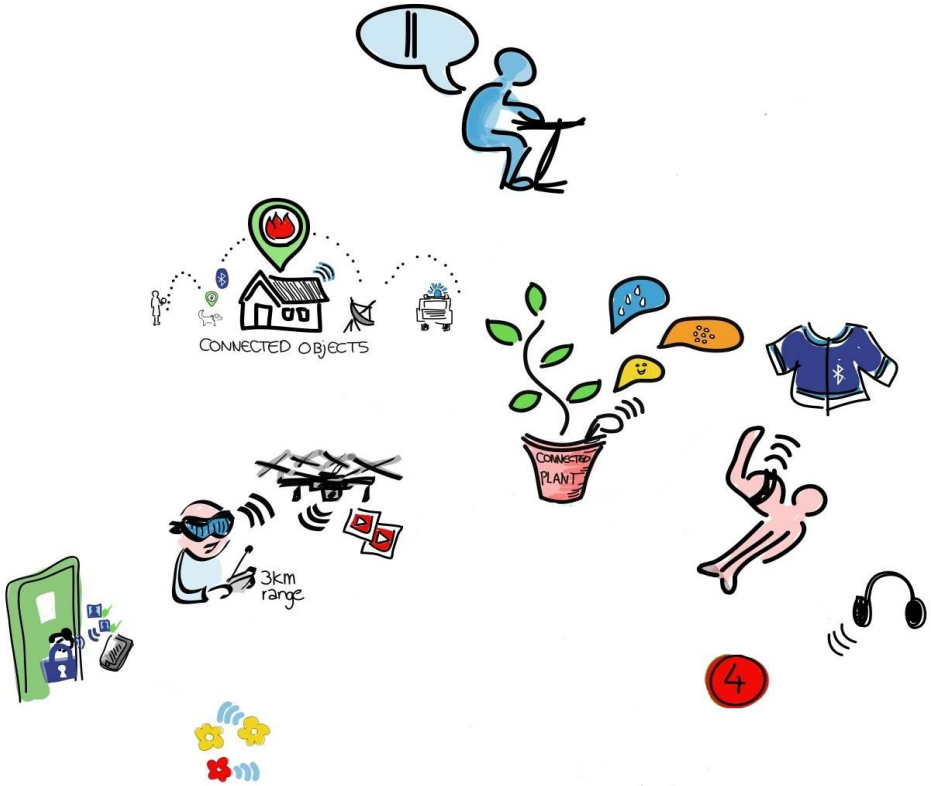
black hat EUROPE 2014

A Large Scale Analysis of the Security of Embedded Firmwares

A. Costin, *J. Zaddach*, A. Francillon, D. Balzarotti
•EURECOM, France

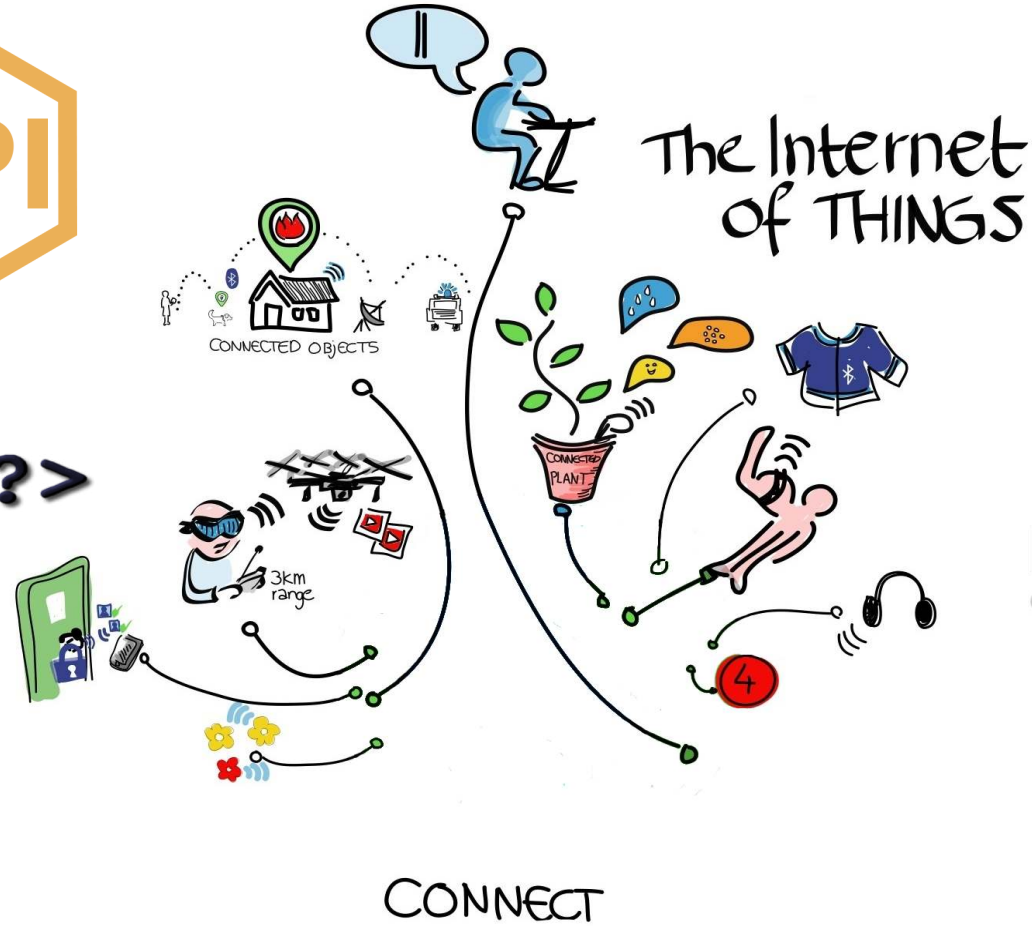
Embedded Systems Are Everywhere



Smarter & More Complex

API

<?xml?>

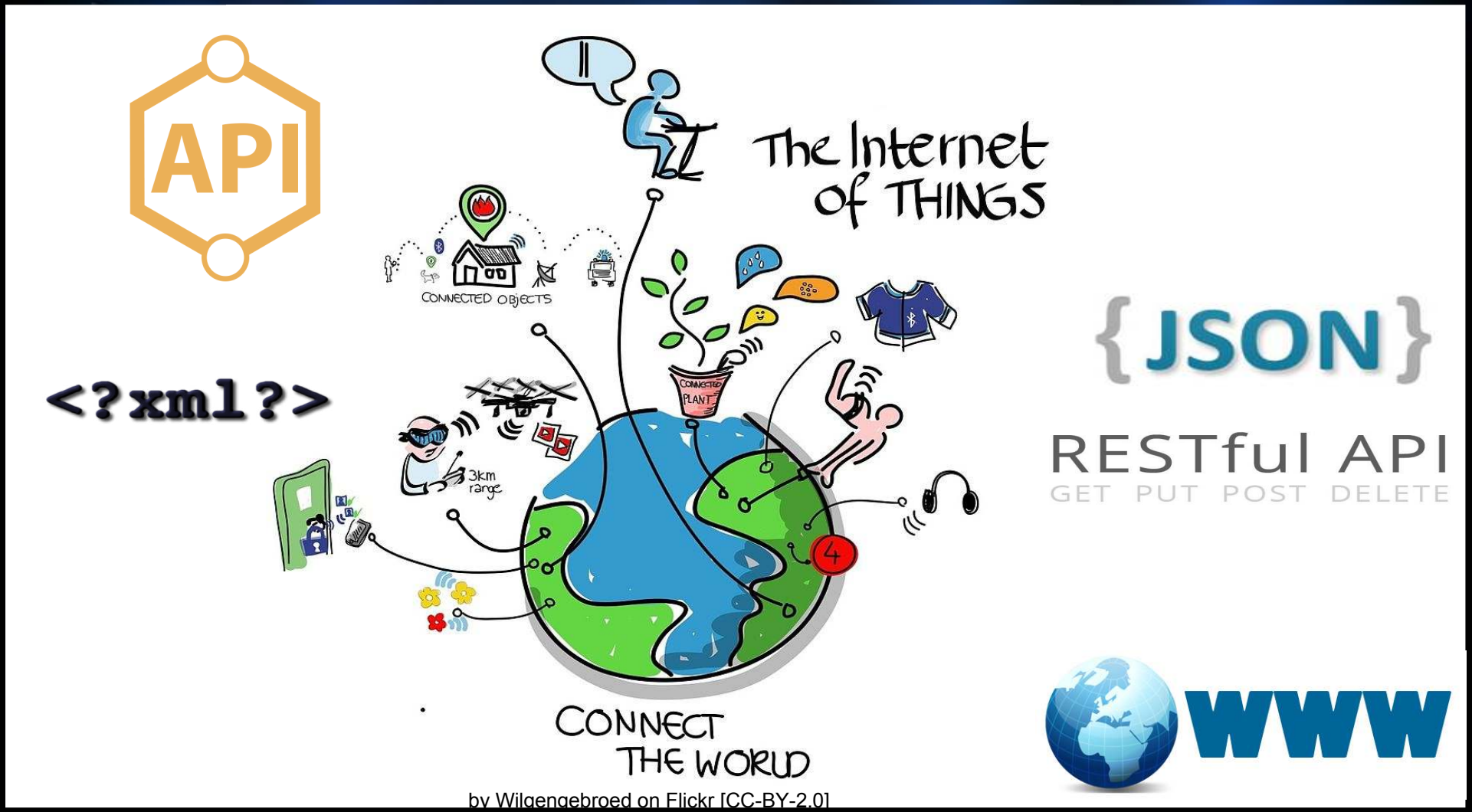


{JSON}

RESTful API

GET PUT POST DELETE

Interconnected



by Wilgengebroed on Flickr [CC-BY-2.0]

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars

Hackers Reveal Nasty New Car Attacks – With Me Behind The Wheel (12/08/2013, Forbes)

Many Examples of Insecure Embedded Systems

- Routers

Firefox | Reverse Engineering a D-Link B...

www.devttys0.com/2013/10/reverse-engineering-a-d-link-backdoor/

Based on the source code of the HTML pages and some Shodan search results D-Link devices are likely affected:

- DIR-100
- DIR-120
- DI-624S
- DI-524UP
- DI-604S
- DI-604UP
- DI-604+
- TM-G5240

Additionally, several Planex routers also appear to use the same firmware:

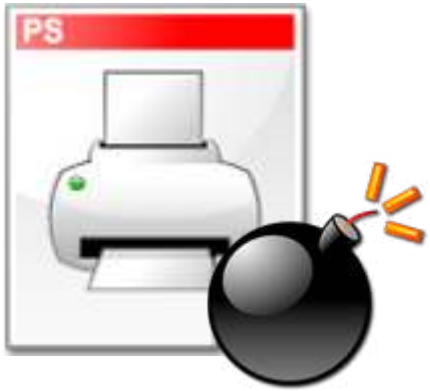
- BRL-04R
- BRL-04UR
- BRL-04CW

You stay classy, D-Link.

Many Examples of Insecure Embedded Systems

- Routers
- Printers

Networked printers at risk
(30/12/2011, McAfee Labs)



Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP

Cisco VoIP Phones Affected By On Hook Security Vulnerability (12/06/2012, Forbes)

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones

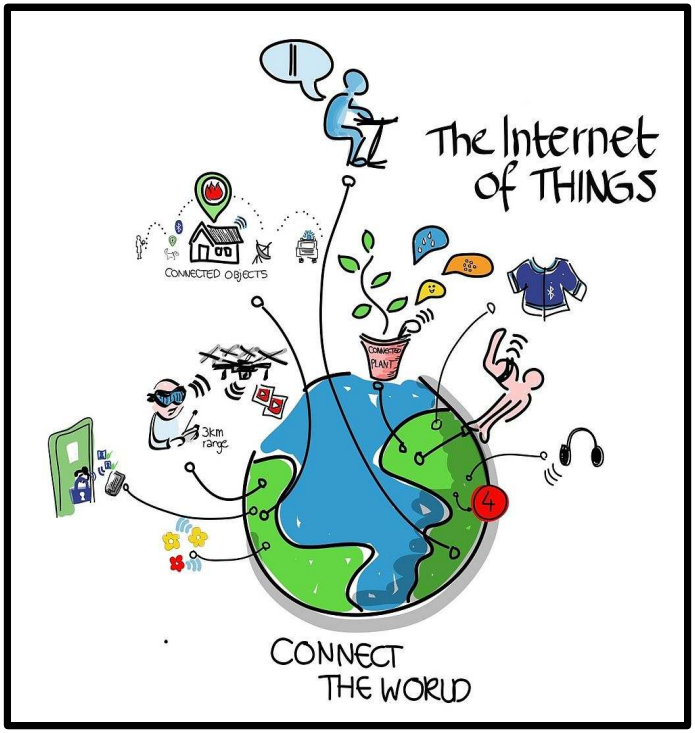
Hacker Releases Software to Hijack Commercial Drones

by BRYANT JORDAN on DECEMBER 9, 2013

Like 489 people like this. Be the first of your friends.

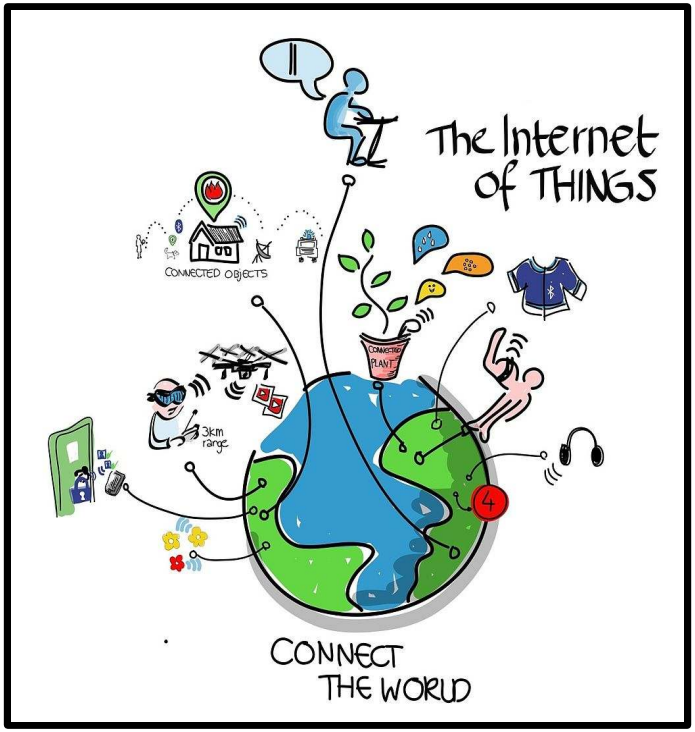
Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones
- ...



Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones
- …



- **Each of the above is a result of individual analysis**
- **Manual and tedious efforts → Does not scale**

The Goal

Perform a large scale analysis to gain a better understanding of firmware problems

The Problem With Large Scale Analysis

- Heterogeneity of
  - Hardware, architectures, OSes
  - Users, requirements
  - Security goals

The Problem With Large Scale Analysis

- Heterogeneity of
  - Hardware, architectures, OSes
  - Users, requirements
  - Security goals
- Manual analysis does not scale, it requires
  - Finding and downloading firmware
  - Unpacking and initial analysis
  - **Re**-discovering a similar bugs

Previous Approaches

- Test on real devices [Bojinov09CCS]
  - Accurate results
  - Does not scale well

Previous Approaches

- Test on real devices [Bojinov09CCS]
  - Accurate results
  - Does not scale well
- Scan devices on the Internet
  - Large scale testing [Cui10ACSAC]
    - Can only test for known vulnerabilities
    - Blackbox approach
  - More is too intrusive [Census2012]

Our Approach to The Large Scale Analysis

- Collect a large number of firmware images

Our Approach to The Large Scale Analysis

- Collect a large number of firmware images
- Perform broad but simple static analysis

Our Approach to The Large Scale Analysis

- Collect a large number of firmware images
- Perform broad but simple static analysis
- Correlate across firmwares

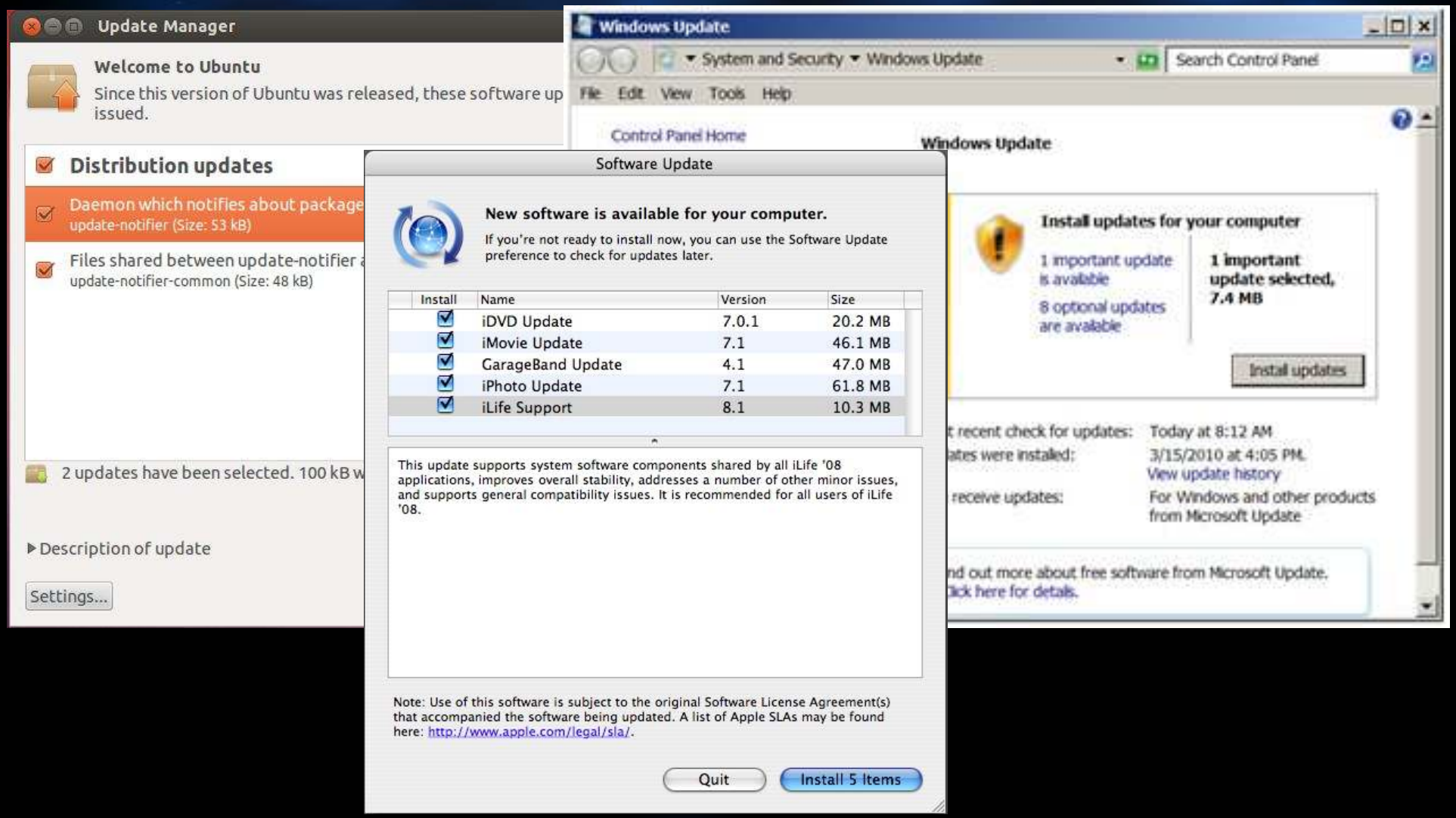
Our Approach to The Large Scale Analysis

- Collect a large number of firmware images
- Perform broad but simple static analysis
- Correlate across firmwares
- Advantages
  - No intrusive online testing, no devices involved
  - Scalable

Our Approach to The Large Scale Analysis

- Collect a large number of firmware images
- Perform broad but simple static analysis
- Correlate across firmwares
- Advantages
  - No intrusive online testing, no devices involved
  - Scalable
- Many challenges remain

Mainstream Systems Have Centralized Updates



EURECOM
Sophia Antipolis

black hat
EUROPE 2014

Challenge: Embedded Systems Update Sources are diverse

- Public site
  - Manufacturer web site
  - FTP site
- Hidden site
  - Accessed by firmware update utility
- Restricted site
- Request-only updates
- Delivery on other media (CD-Rom, …)
- Firmware only delivered on device

Challenge: Embedded Systems Update Mechanisms are diverse

Collecting a Dataset

- No large scale firmware dataset yet
  - As opposed to existing datasets in security or other CS research areas

Collecting a Dataset

- No large scale firmware dataset yet
  - As opposed to existing datasets in security or other CS research areas
- We collected a subset of the firmwares available for download

Collecting a Dataset

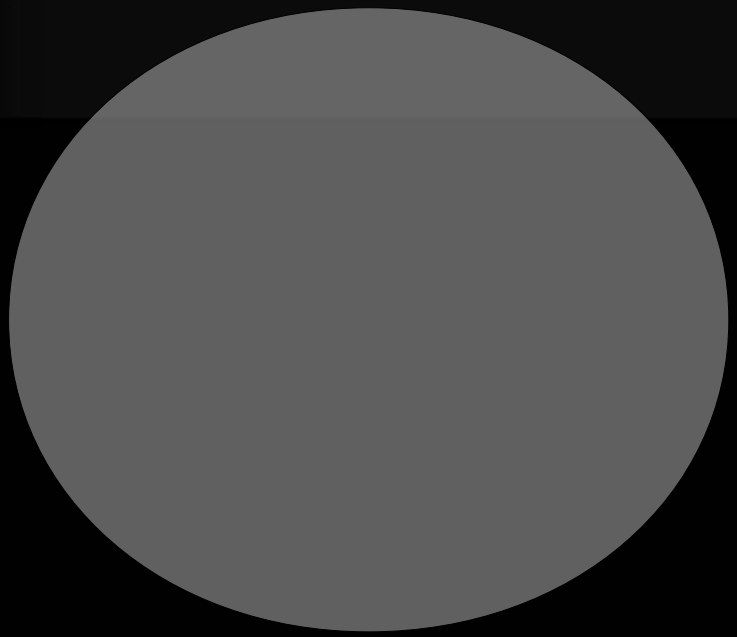
- No large scale firmware dataset yet
  - As opposed to existing datasets in security or other CS research areas
- We collected a subset of the firmwares available for download
- Many firmwares are not publicly available
  - Not intended to have an upgrade
  - Require product purchase and registration

Collecting a Dataset

- No large scale firmware dataset yet
  - As opposed to existing datasets in security or other CS research areas
- We collected a subset of the firmwares available for download
- Many firmwares are not publicly available
  - Not intended to have an upgrade
  - Require product purchase and registration
- www.firmware.re project

Challenge: Firmware Identification

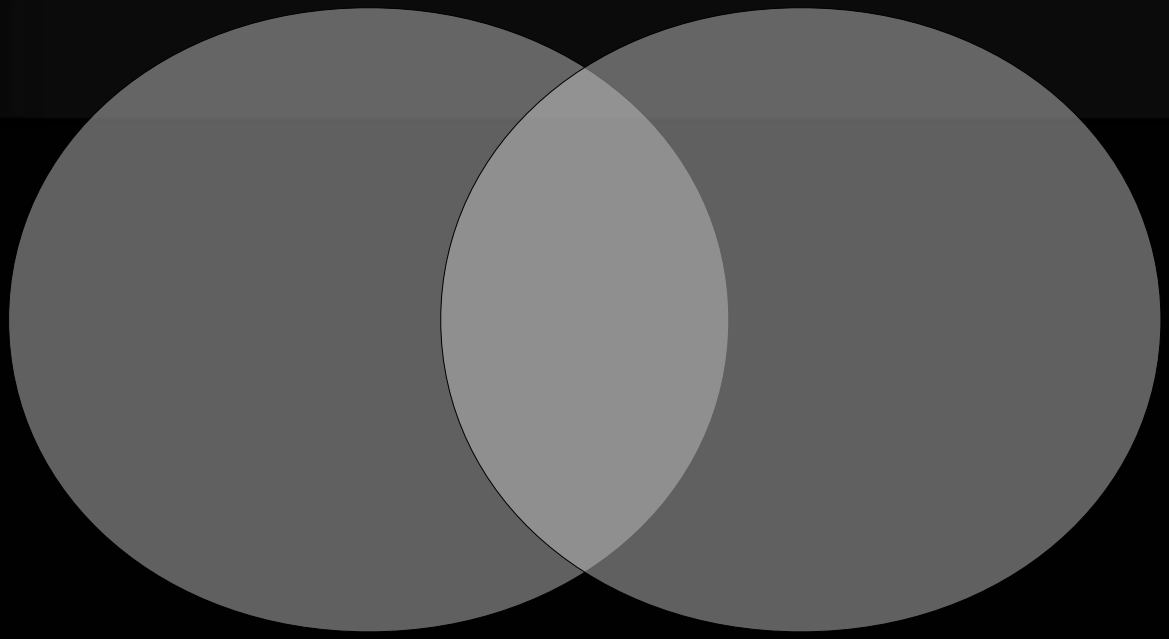
← Clearly a Firmware



Challenge: Firmware Identification

← Clearly a Firmware

Clearly not a Firmware →



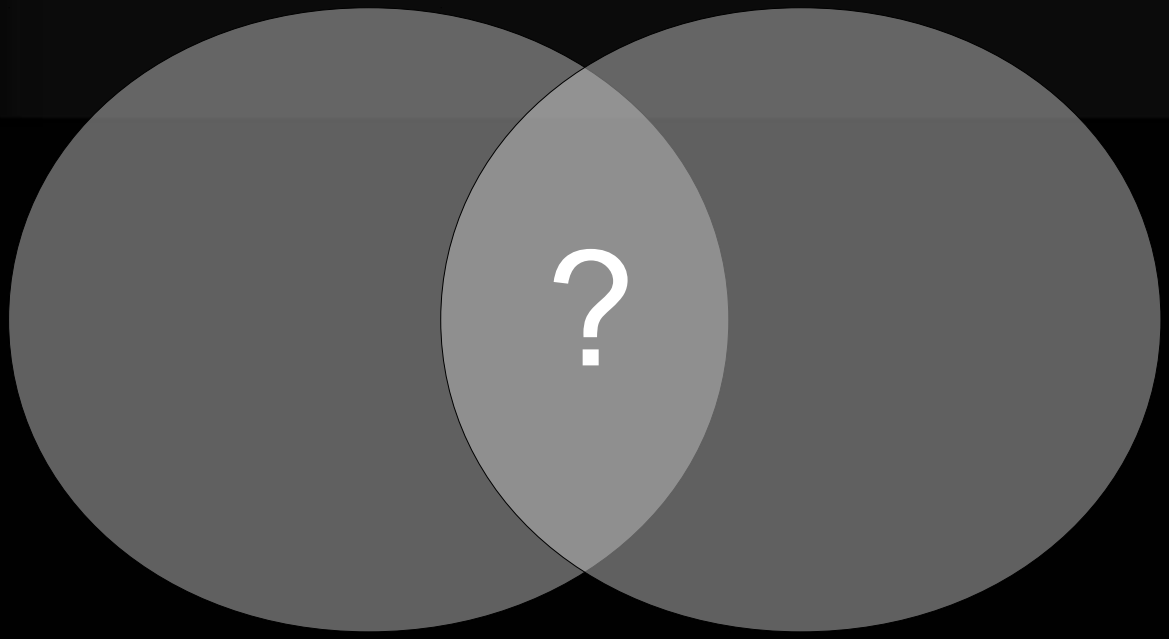
EURECOM
Sophia Antipolis

black hat
EUROPE 2014

Challenge: Firmware Identification

← Clearly a Firmware

Clearly not a Firmware →



Challenge: Firmware Identification

- E.g., upgrade by printing a PS document

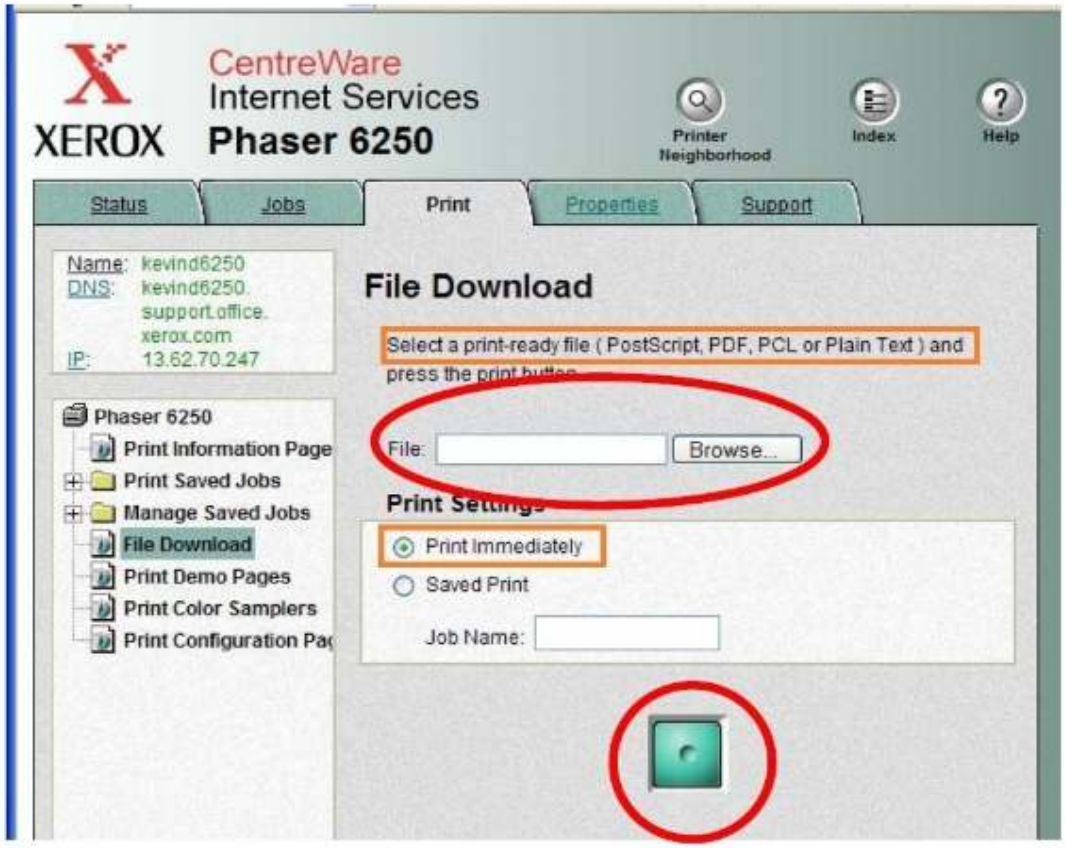


Figure 4: Select the firmware update file and press the green button to send it.

Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?

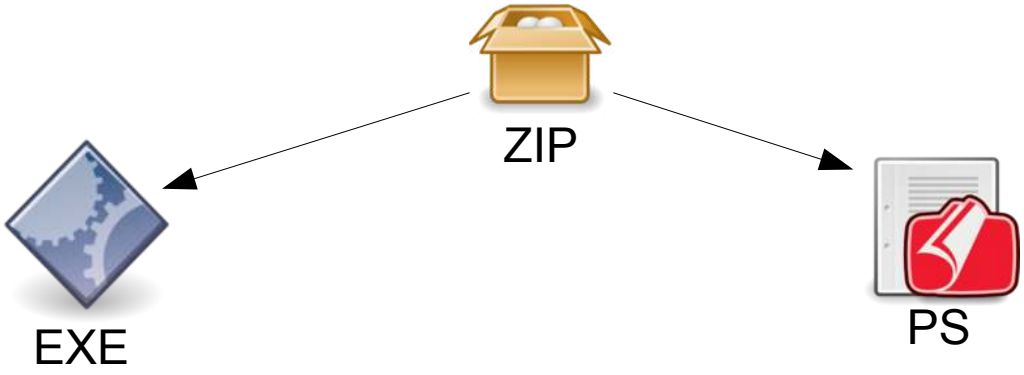
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?

ZIP

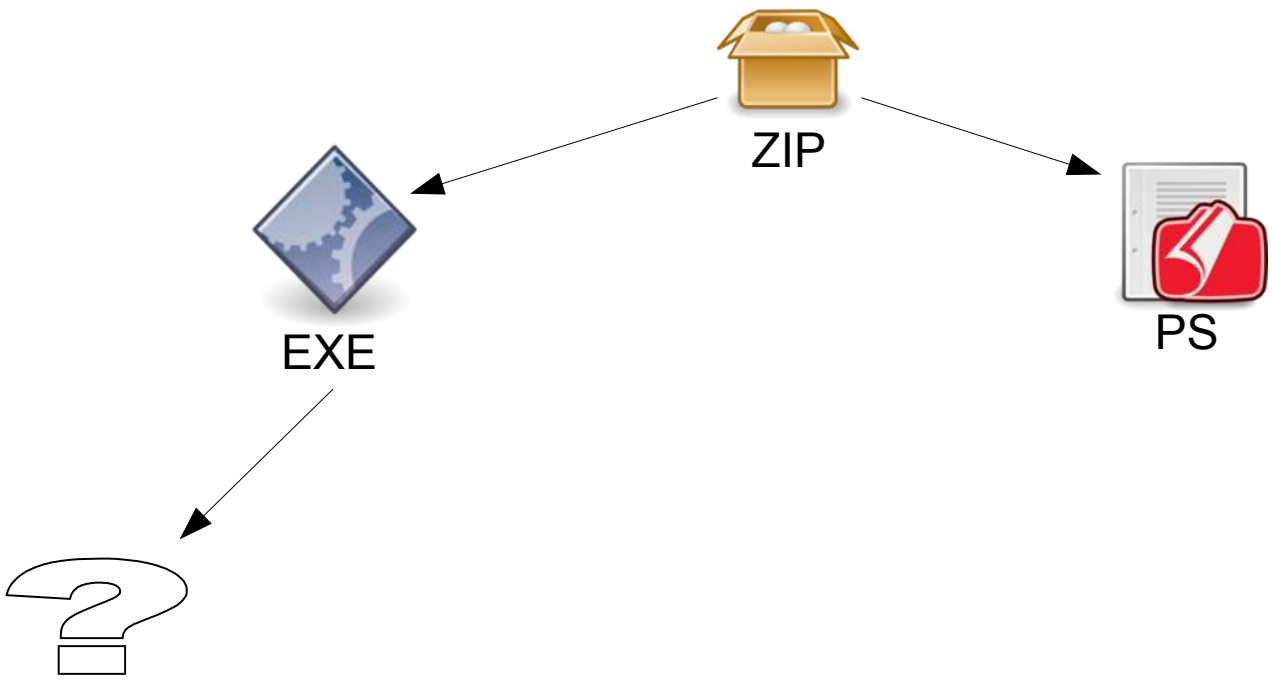
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



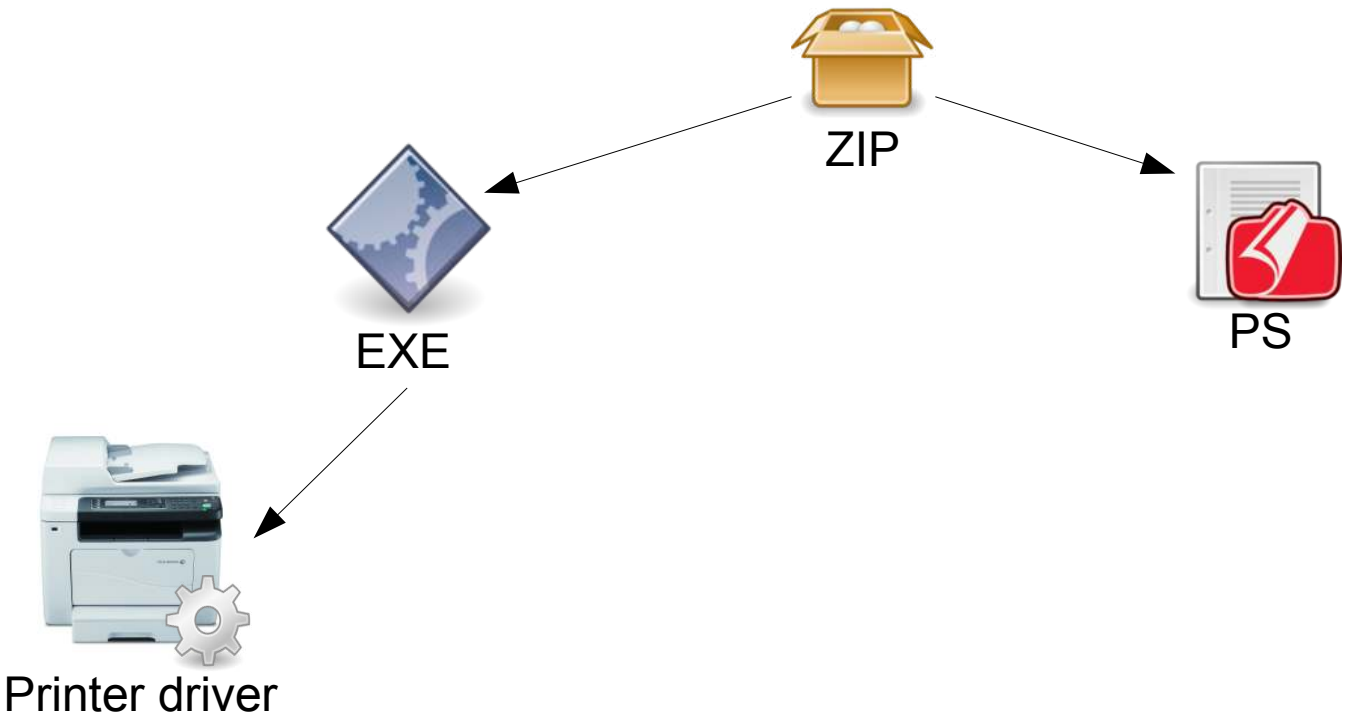
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



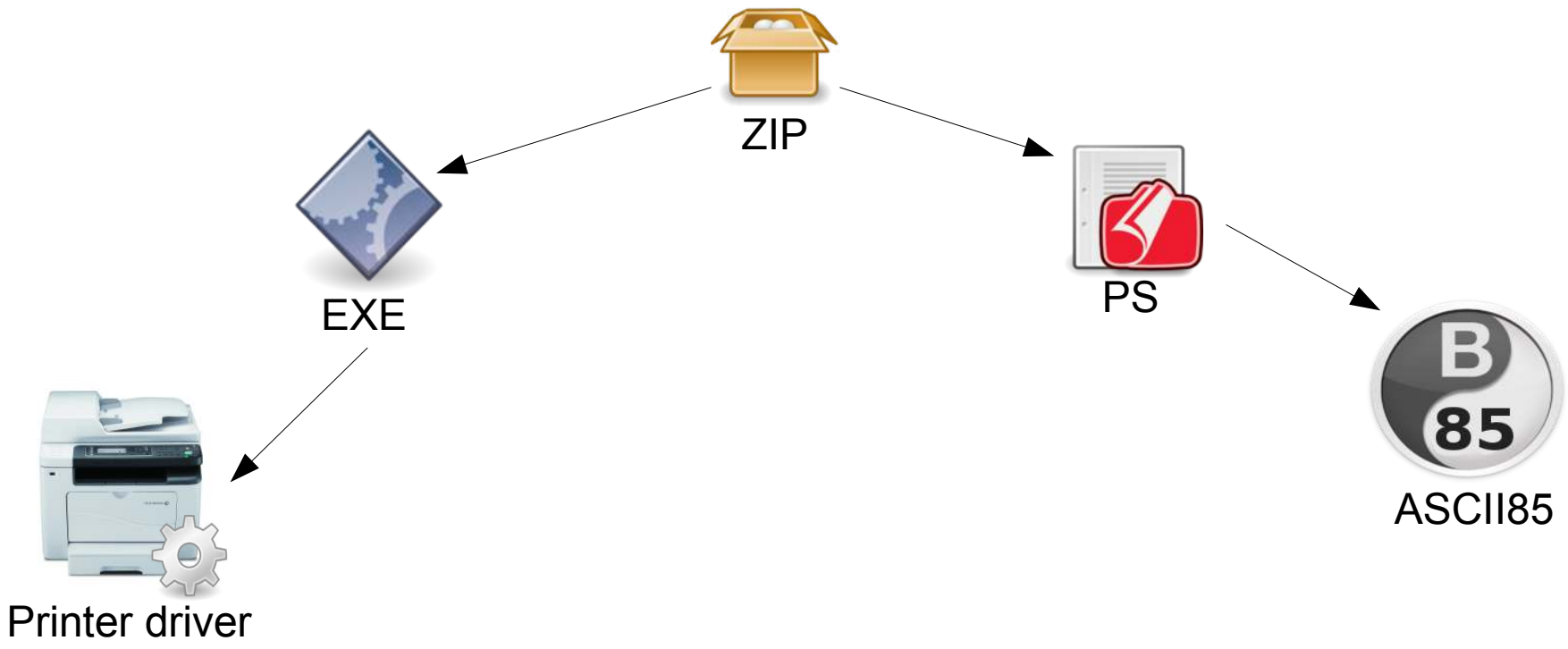
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



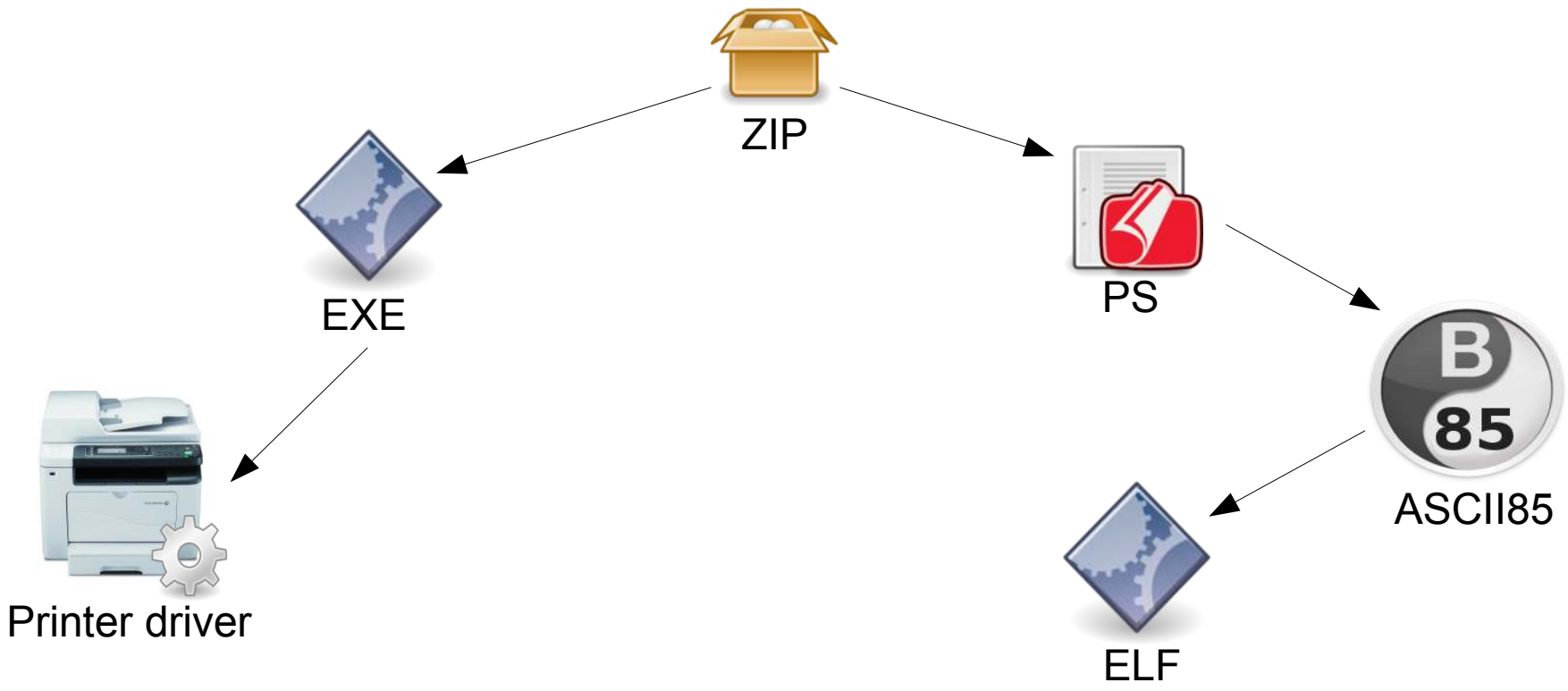
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



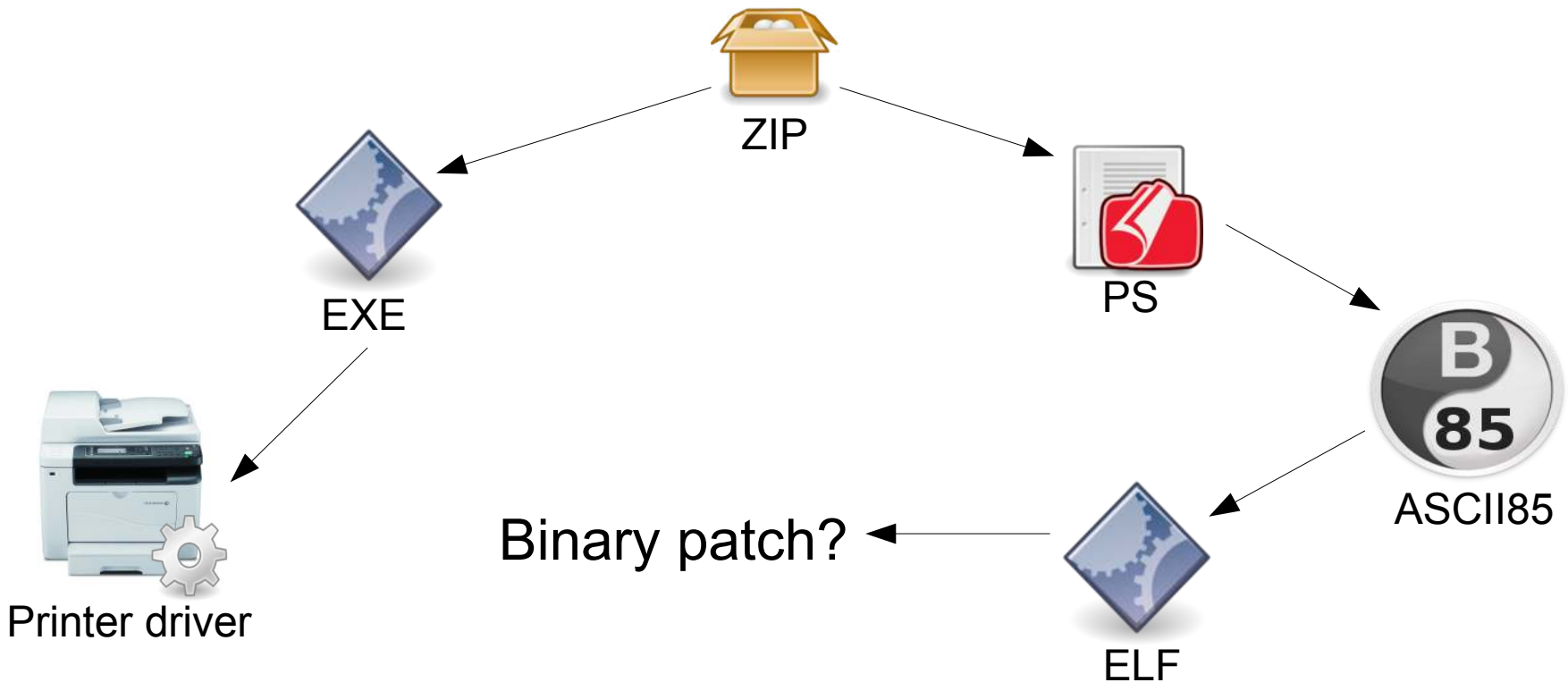
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



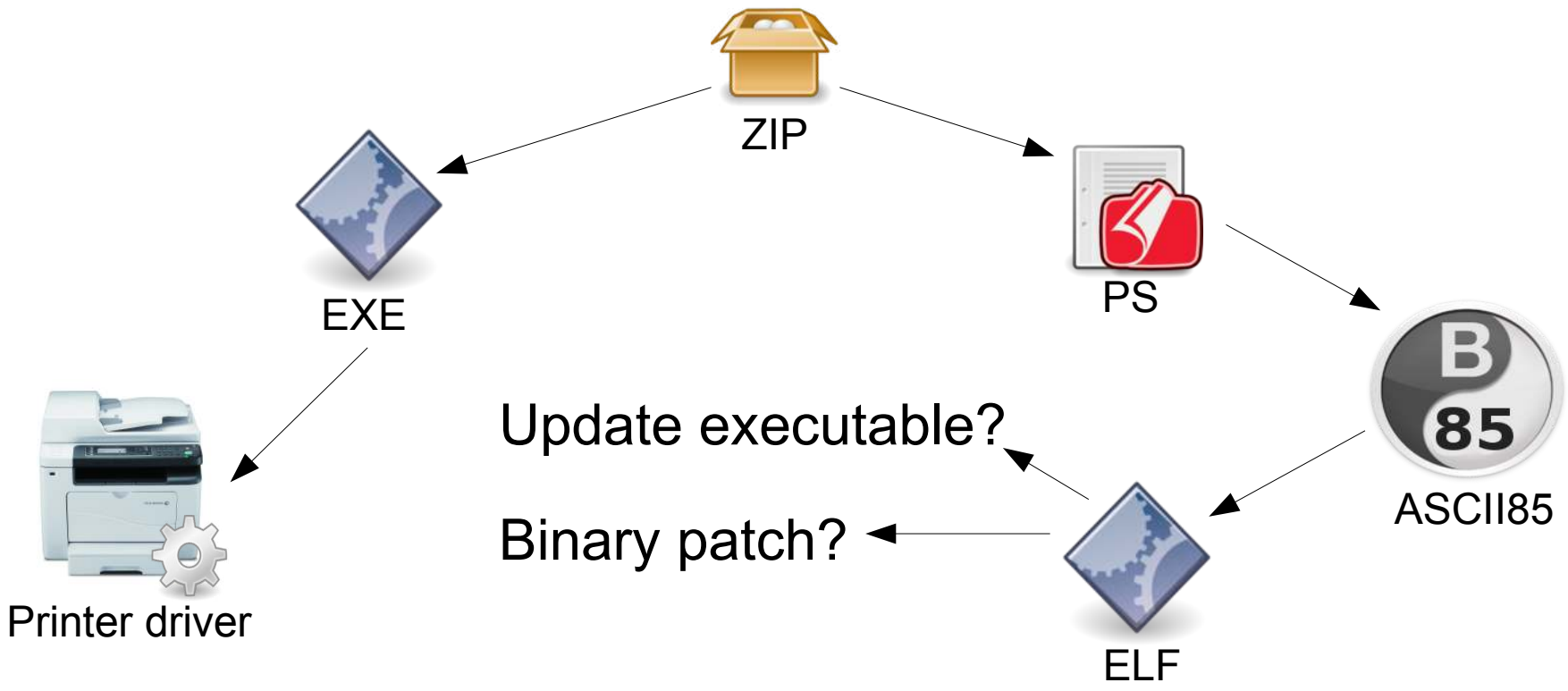
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



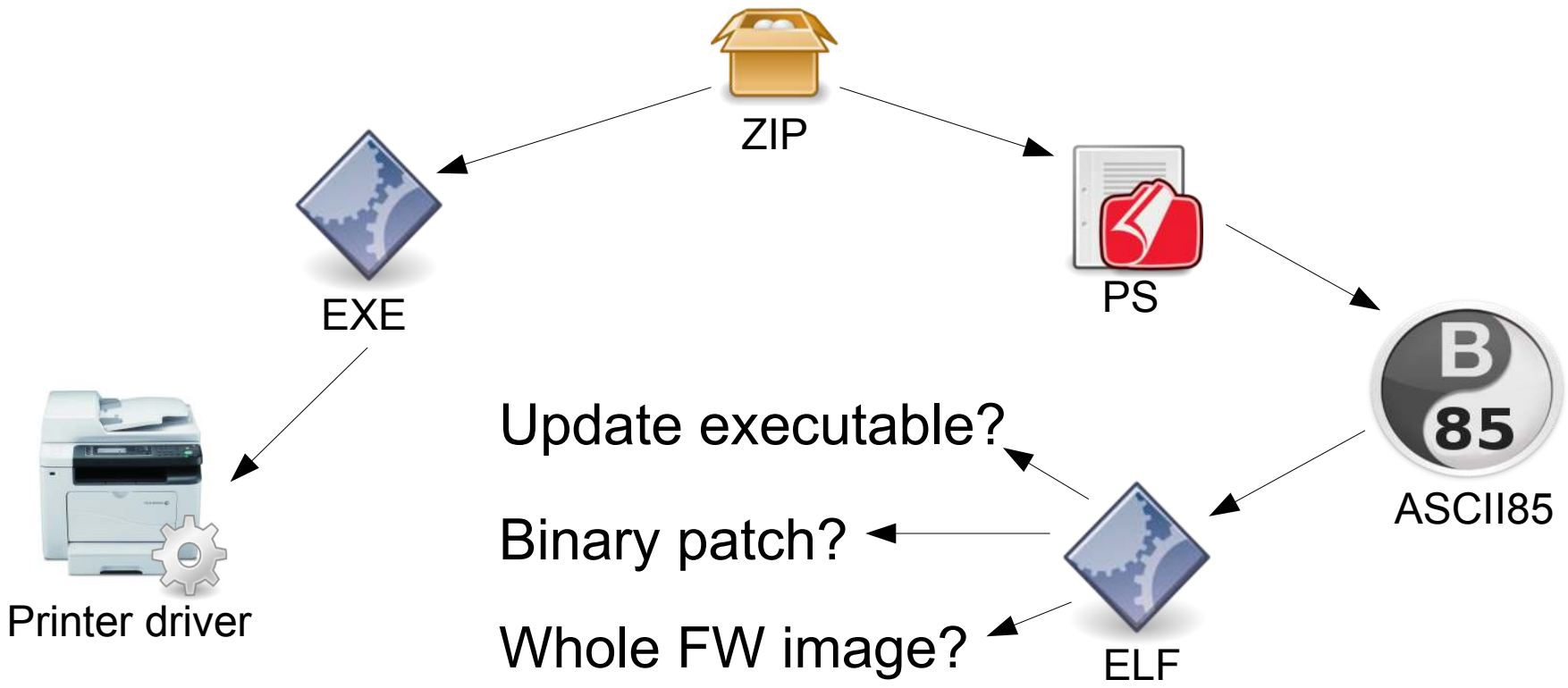
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



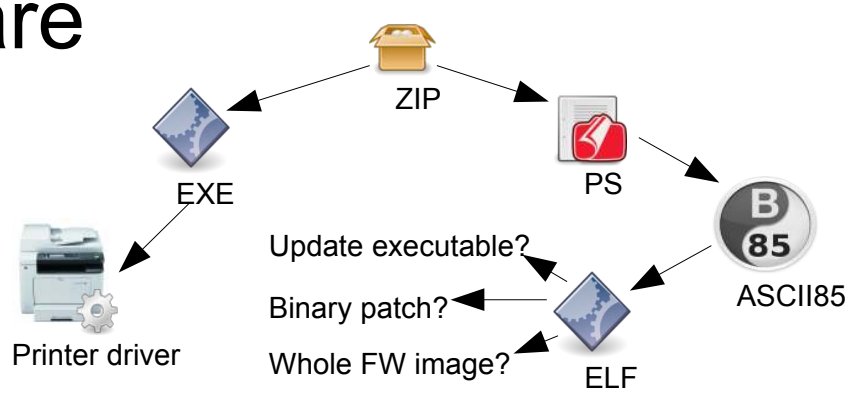
Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?



Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?
- Firmware updates often are ”russian dolls”
- Sometimes result of unpacking is just a binary data blob



Our Approach to Unpacking & Custom Formats

- We compared existing tools
- Used BAT (Binary Analysis Toolkit)
  - Extended it with multiple custom unpackers
  - Continuous development effort

Our Approach to Unpacking & Custom Formats

- We compared existing tools
- Used BAT (Binary Analysis Toolkit)
  - Extended it with multiple custom unpackers
  - Continuous development effort
- Often, a firmware image→just 'data' binary blob
  - File carving required
  - Bruteforce at every offset with all known unpackers

Our Approach to Unpacking & Custom Formats

- We compared existing tools
- Used BAT (Binary Analysis Toolkit)
  - Extended it with multiple custom unpackers
  - Continuous development effort
- Often, a firmware image→just 'data' binary blob
  - File carving required
  - Bruteforce at every offset with all known unpackers
- Heuristics for detecting when to stop

Challenge: Scalability & Computational Limits

- Unpacking and file carving is very CPU intensive

Challenge: Scalability & Computational Limits

- Unpacking and file carving is very CPU intensive
- Results in millions of unpacked files
  - Manual analysis infeasible
  - One-to-one fuzzy hash comparison is CPU intensive

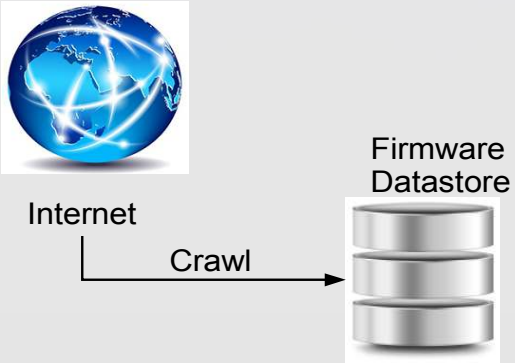
Challenge: Results Confirmation

- An issue found statically
  - May not apply to a real-device
  - Cannot guarantee exploitability
  - E.g., vulnerable daemon present but never started

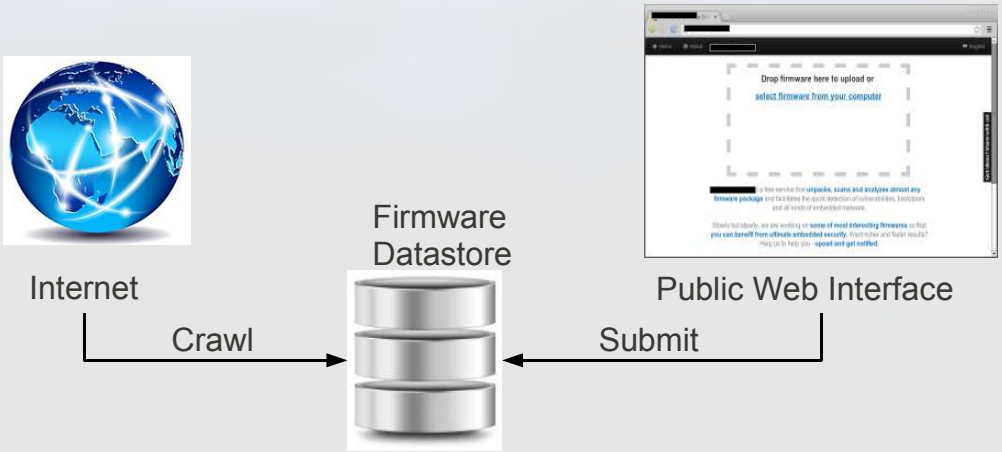
Challenge: Results Confirmation

- An issue found statically
  - May not apply to a real-device
  - Cannot guarantee exploitability
  - E.g., vulnerable daemon present but never started
- Issue confirmation is difficult
  - Requires advanced analysis (static & dynamic)
  - Often requires real embedded devices
  - Does not scale well in heterogeneous environments

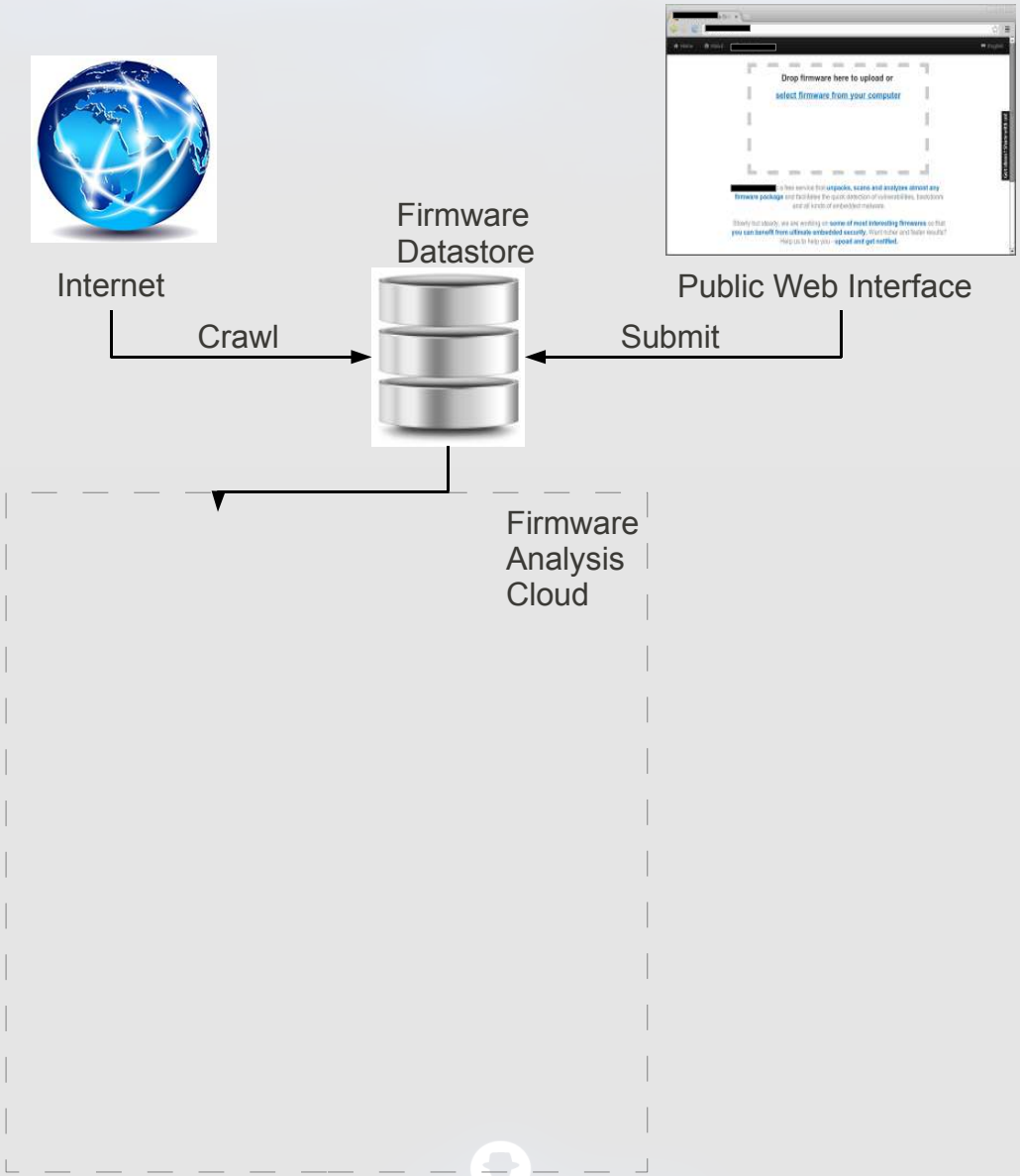
Architecture



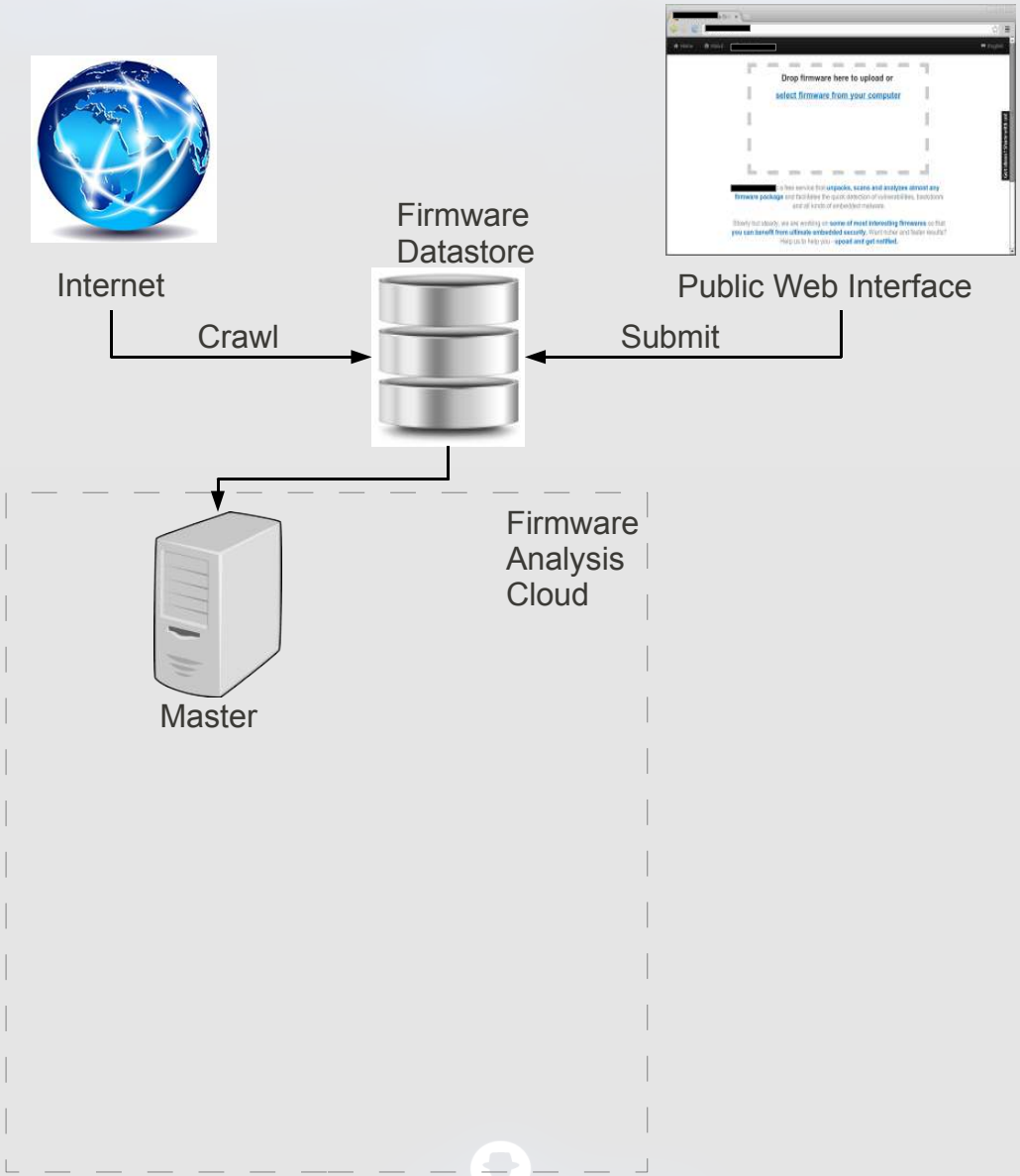
Architecture



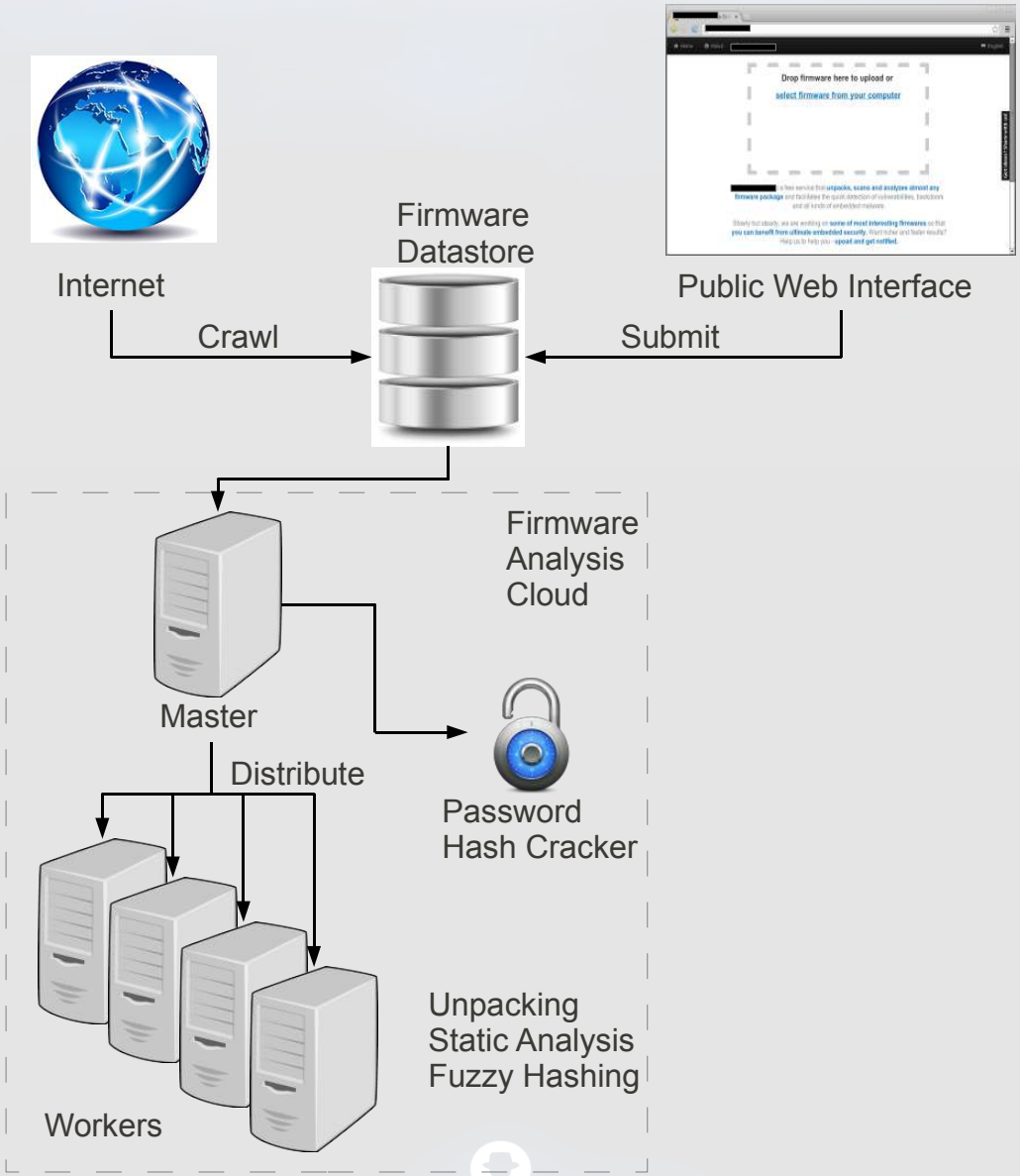
Architecture



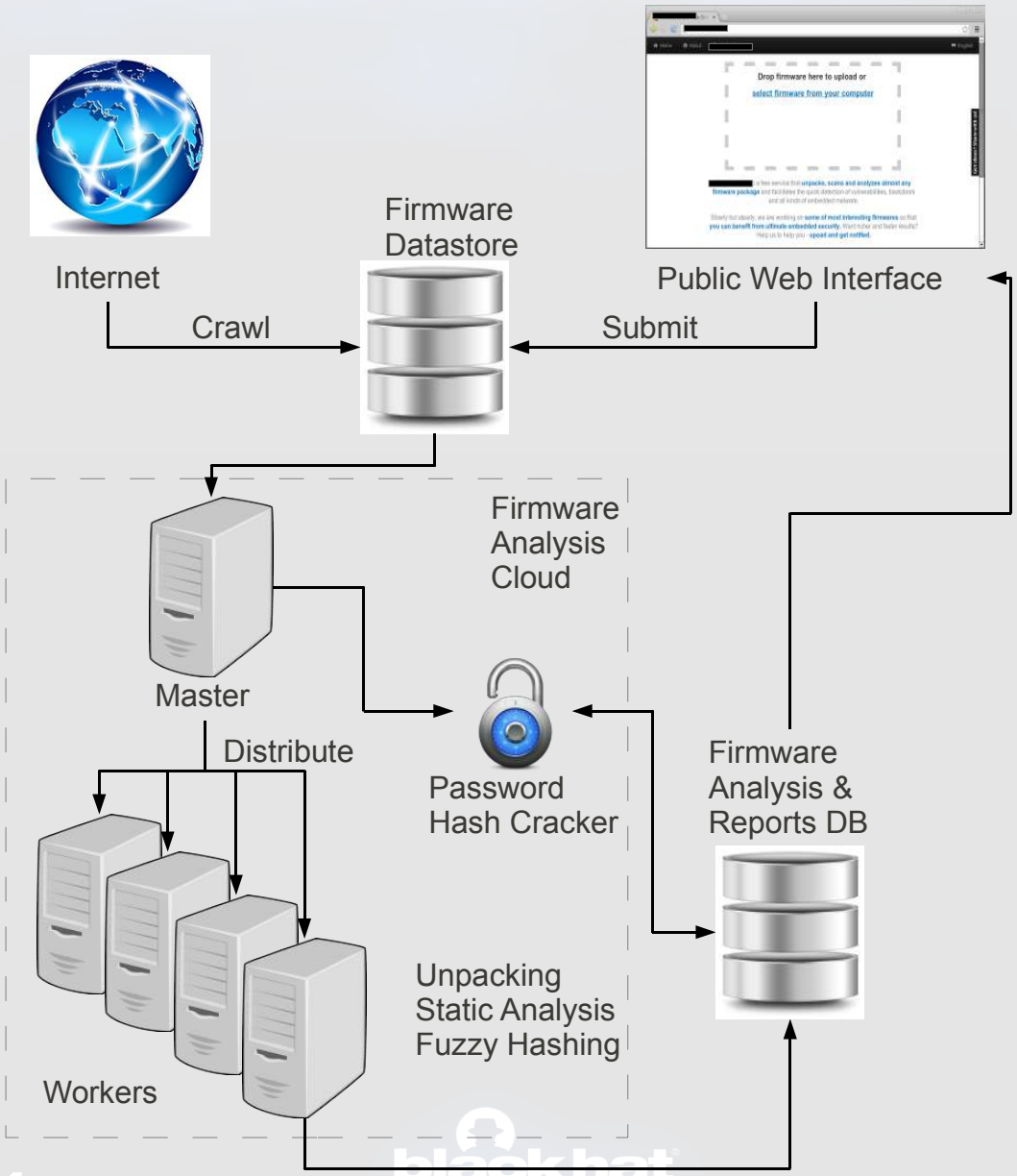
Architecture



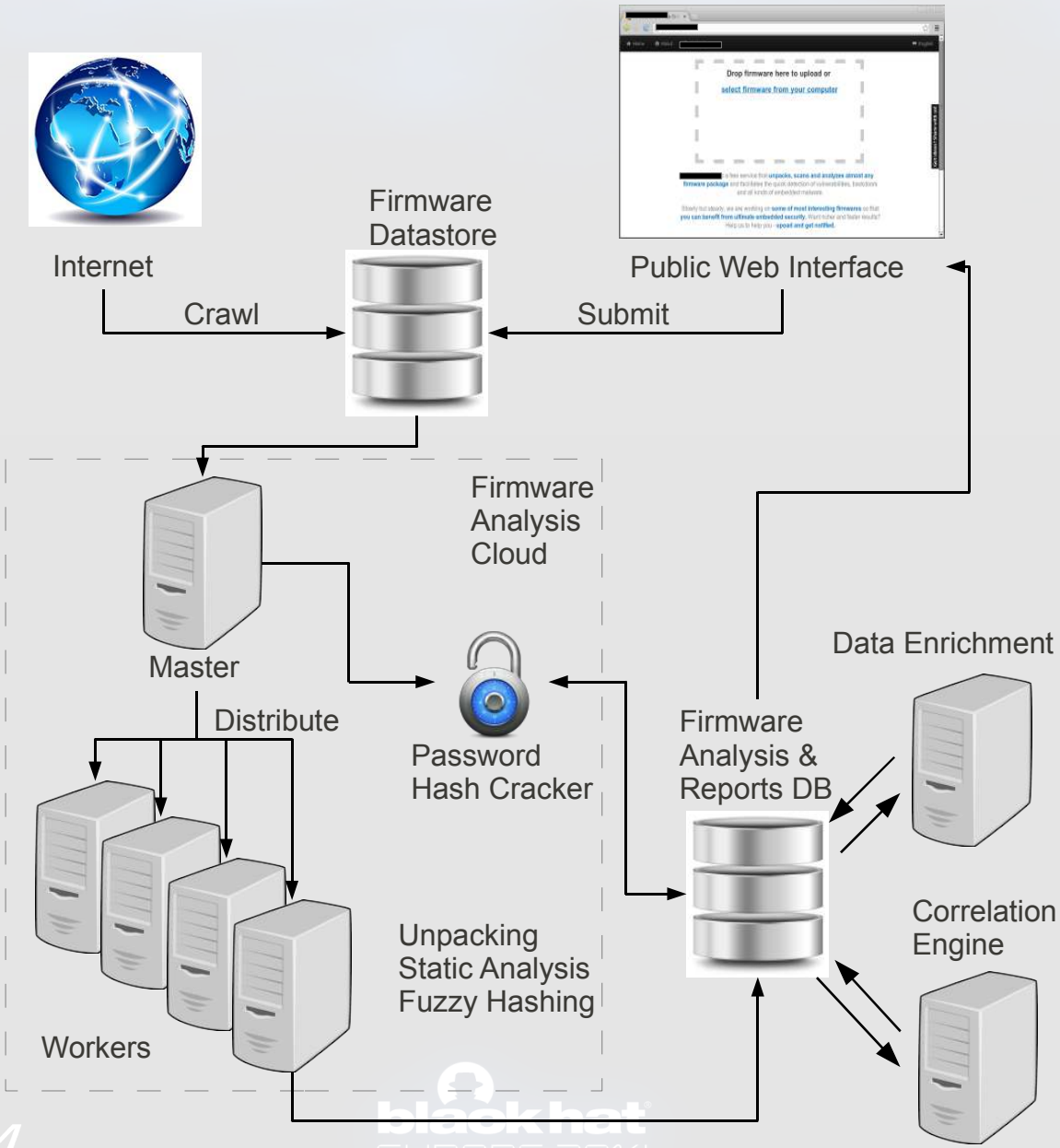
Architecture



Architecture



Architecture



Crawler

- Multiple seeds
  - FTP-index engines
  - Google Custom search engines
- Several download techniques
  - WGET scripts
  - Beautiful Soup scripts
- 759 K collected files, 1.8 TB of disk space

www.Firmware.RE (beta)
Will provide Unpacking and Analysis

Firefox | firmware · ǝɹ - Free Online Firm...

www.firmware.re

firmware · ǝɹ (beta)

USENIX Security '14 | BH13US | About | English

Upload

Info

Examples

To start, drag-n-drop firmware here or

select firmware from your computer

Got ideas? Share with us!

Blog | Twitter | contact@firmware.re | Google groups | ToS | Privacy policy

Unpacking

- 759 K total files collected

  ↓ Filter non firmware

- 172 K filtered interesting files

  ↓ Random selection

- 32 K analyzed

  ↓ Successful unpack

- 26 K unpacked (fully or partially)

  ↓ Unpacked files

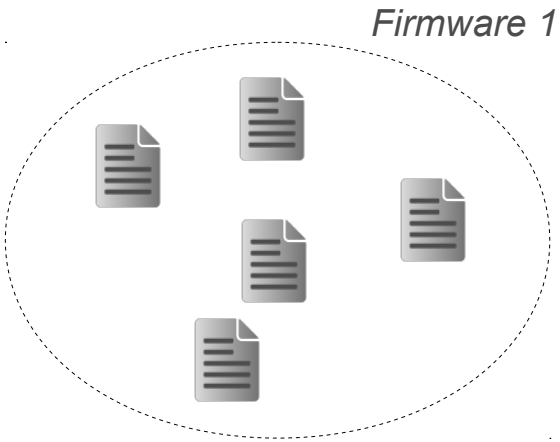
- 1.7 M resulted files after unpacking

Static Analysis

- Correlation/clustering
  - Fuzzy hashes, Private SSL keys, Credentials
- Misconfigurations
  - Web-server configs, Credentials, Code repositories
- Data enrichment
  - Version banners
  - Keywords (e.g., telnet, shell, UART, backdoor)

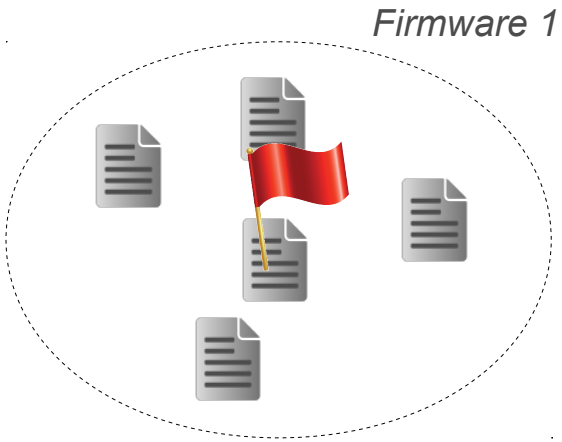
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)



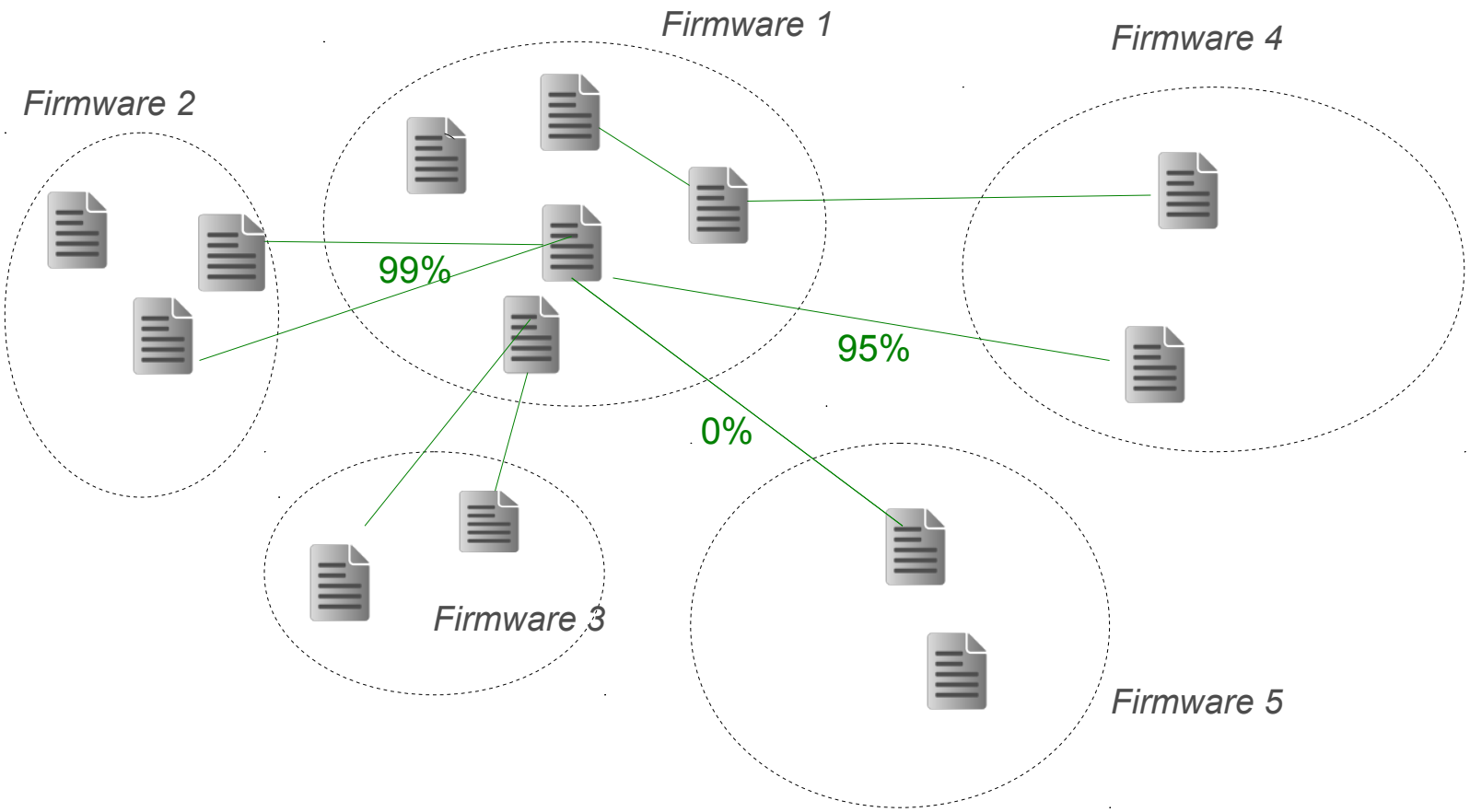
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)



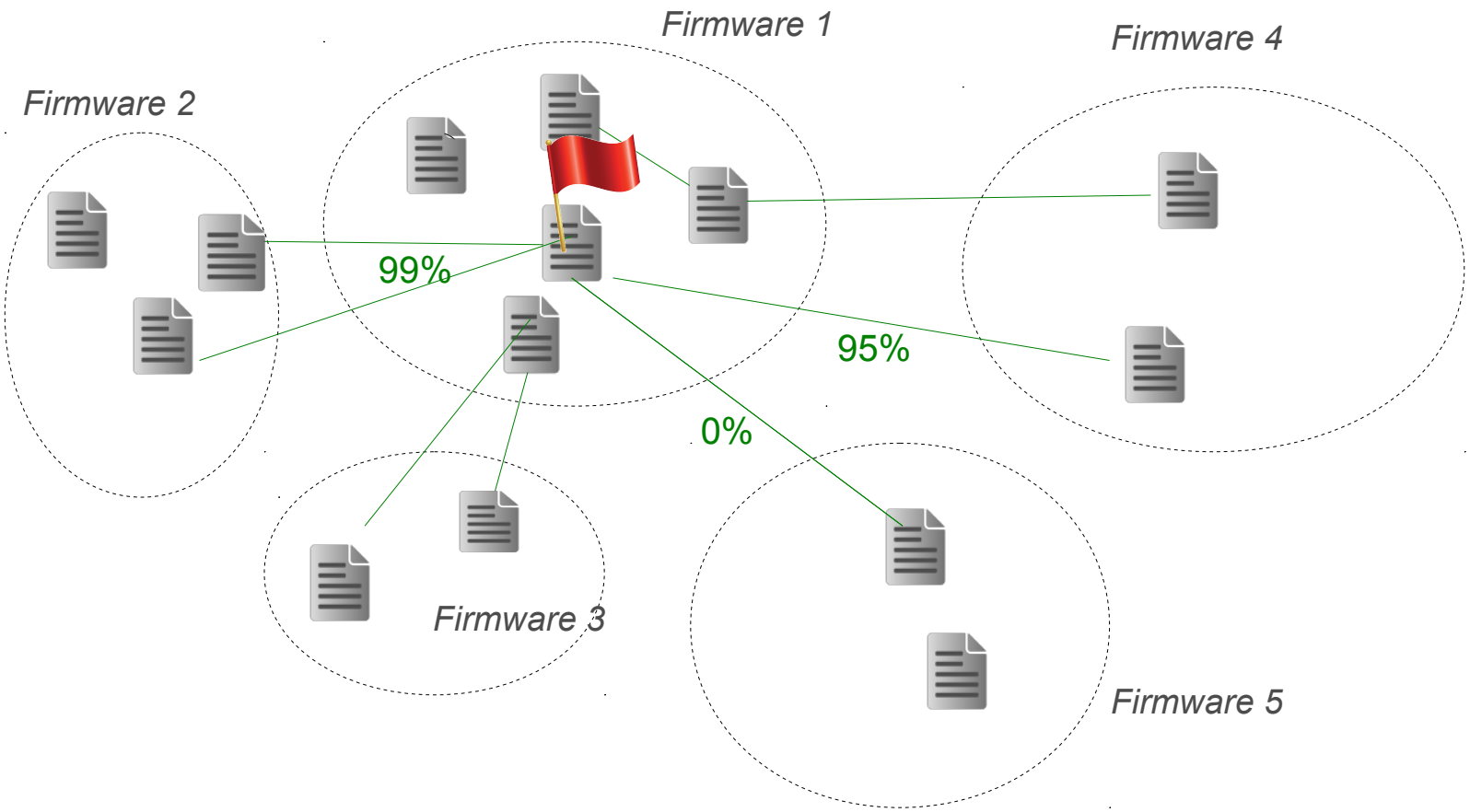
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)



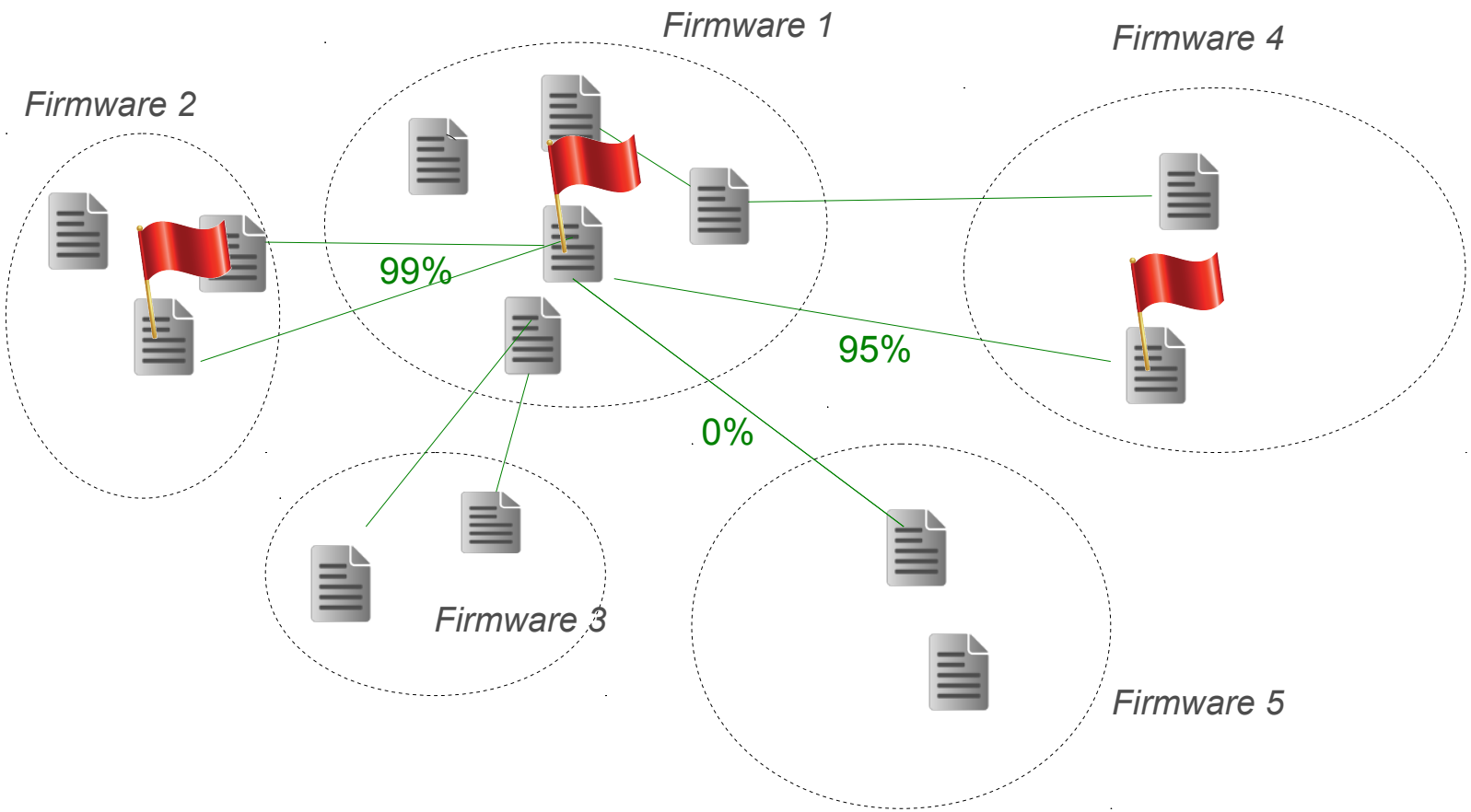
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)



Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)

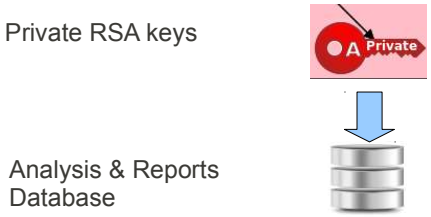


Example: RSA Keys

- SSL keys correlation + vulnerability propagation

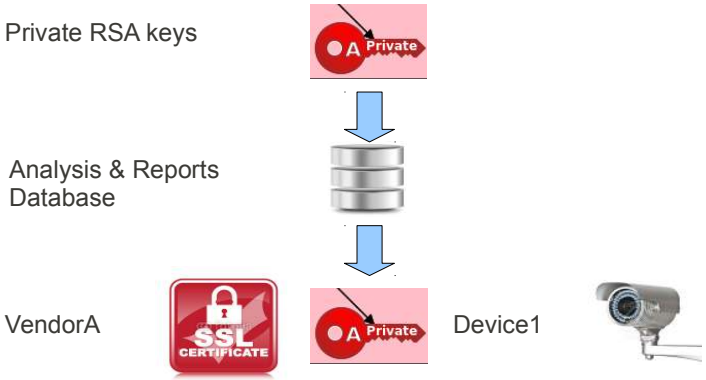
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



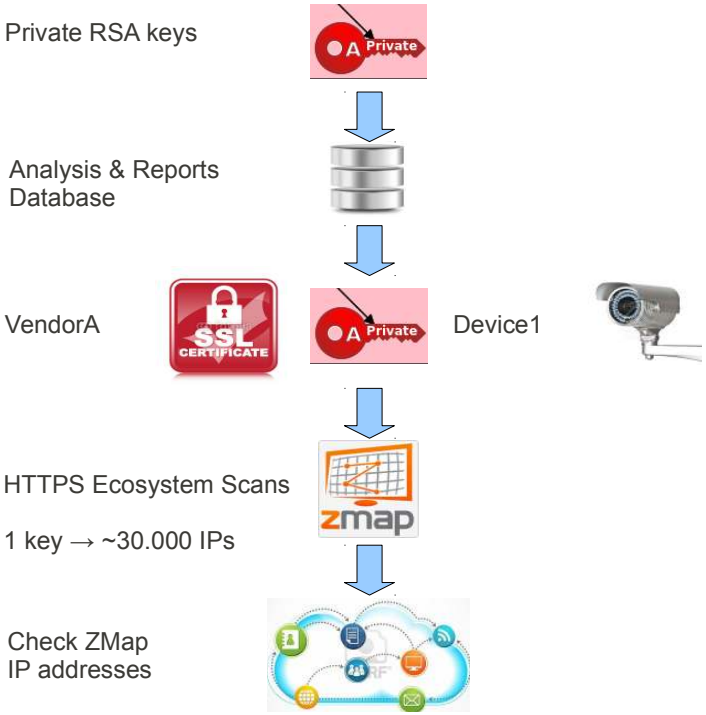
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



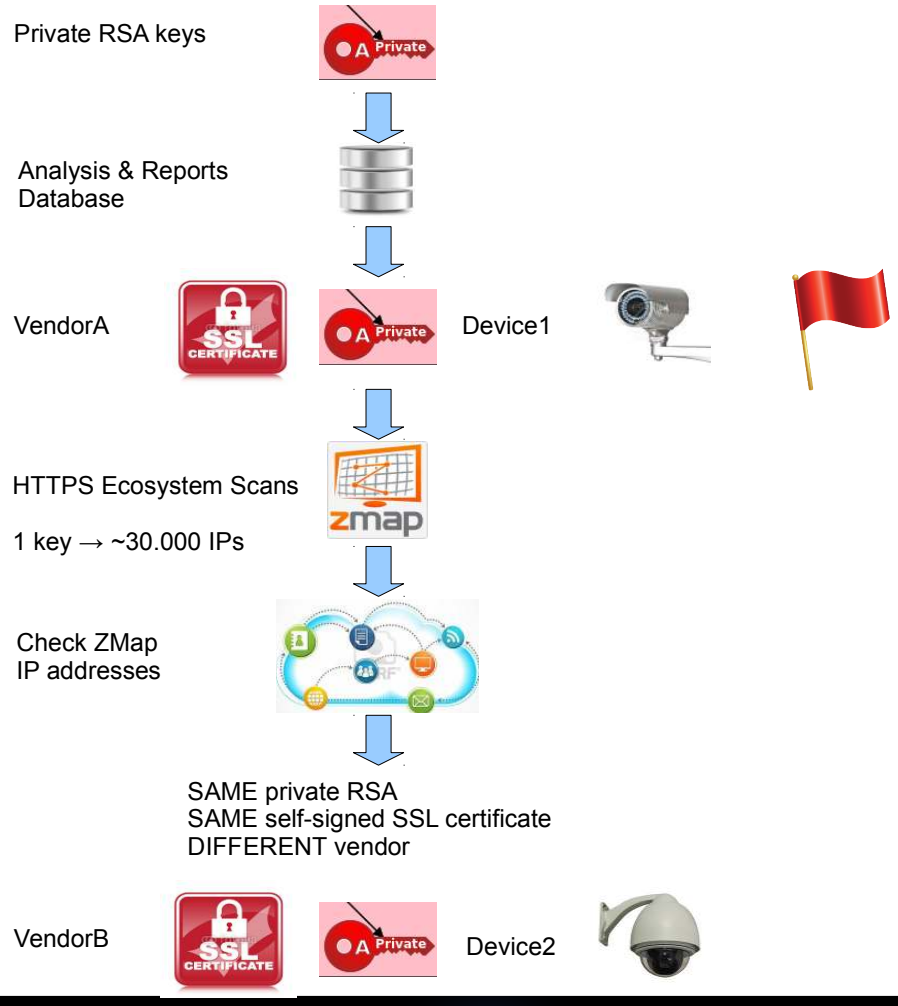
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



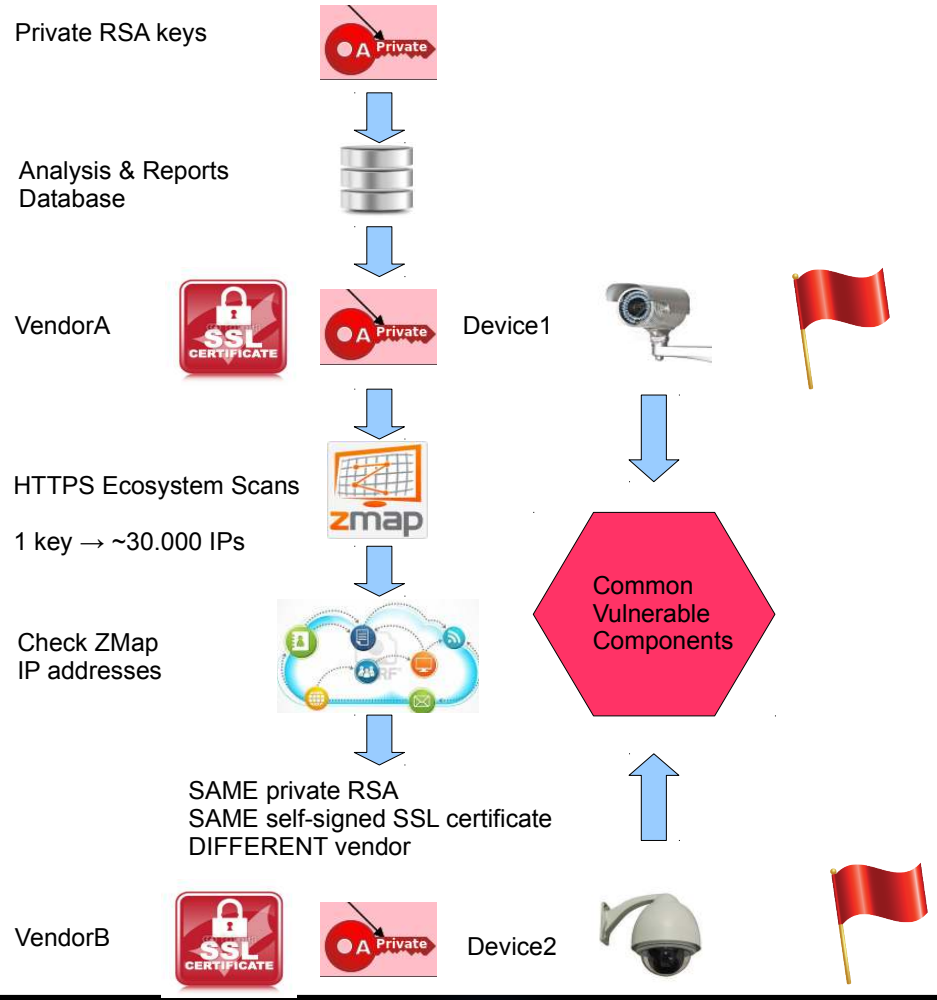
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



Example: RSA Keys

- SSL keys correlation + vulnerability propagation



Results: Summary

- 38 new vulnerabilities (CVE)
- Correlated them to 140 K online devices
- Affected 693 firmware files by at least one vuln

”Chamber of Horrors”

- Several recently build images with linux kernels, busybox older than 9 years
- Similar ”debug” backdoor daemon in networking, home automation equipment
- Forgotten or backdoor entries in `authorized_keys` files

”Chamber of Horrors”

- Linux kernel older than 4 years compiled by root on a machine with public IP accepting SSH connections (GPS/Aerospace manufacturer)
- Discovered vulnerability in wireless fireworks system, implemented PoC attack [3]

Contributions Summary

- First large-scale static analysis of firmwares
- Described the main challenges associated
- Shown the advantages of performing a large-scale analysis of firmware images
- Implemented a framework and several efficient static techniques

Conclusions

- A broader view on firmwares
  - Not only beneficial
  - But necessary for discovery and analysis of vulnerabilities
- Correlation reveals firmware relationship
  - Shows how vulnerabilities reappear across different products
  - Could allow seeing how firmwares evolve

Conclusions

- There are plenty of latent vulnerabilities
- Security
  - Tradeoff with cost and time-to-market
  - Clearly not a priority for some vendors

Thank You!

Questions?

{name.surname}@eurecom.fr

EURECOM
Sophia Antipolis

black hat
EUROPE 2014

References

- [1] A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, *"A Large-Scale Analysis of the Security of Embedded Firmwares"*, In Proceedings of the 23rd USENIX Conference on Security (to appear)
- [2] A. Costin, J. Zaddach, *"Poster: Firmware.RE: Firmware Unpacking and Analysis as a Service"*, In Proceedings of the ACM Conference on Security and Privacy in Wireless Mobile Networks (WiSec) '14
- [3] A. Costin, A. Francillon, *"Short paper: A Dangerous 'Pyrotechnic Composition': Fireworks, Embedded Wireless and Insecurity-by-Design"*, In Proceedings of the ACM Conference on Security and Privacy in Wireless Mobile Networks (WiSec) '14