A Large Scale Analysis of the Security of Embedded Firmwares

A. Costin, **J. Zaddach, A. Francillon, D. Balzarotti**

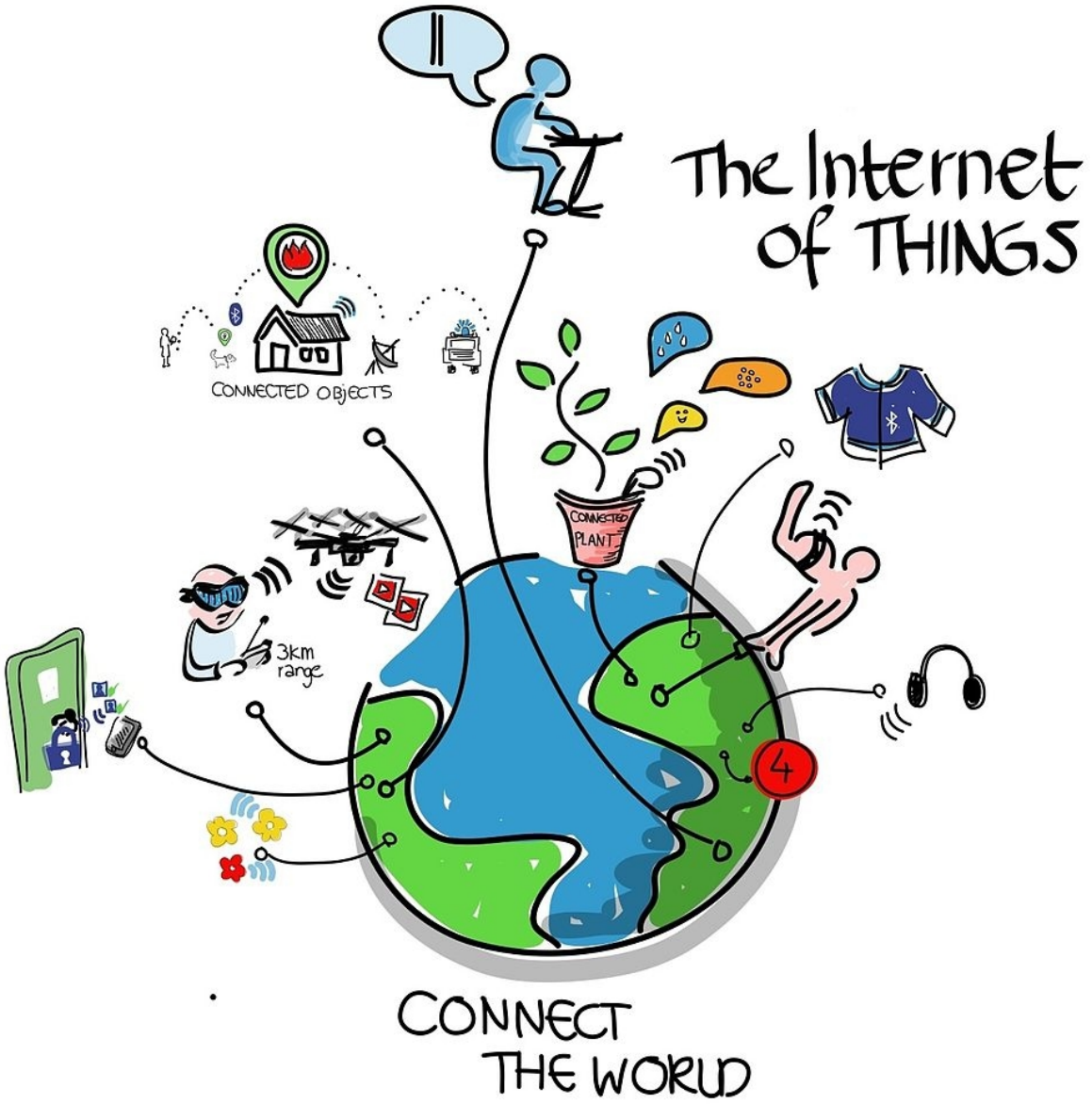
EURECOM, France

20th August 2014

USENIX Security '14 – San Diego, USA

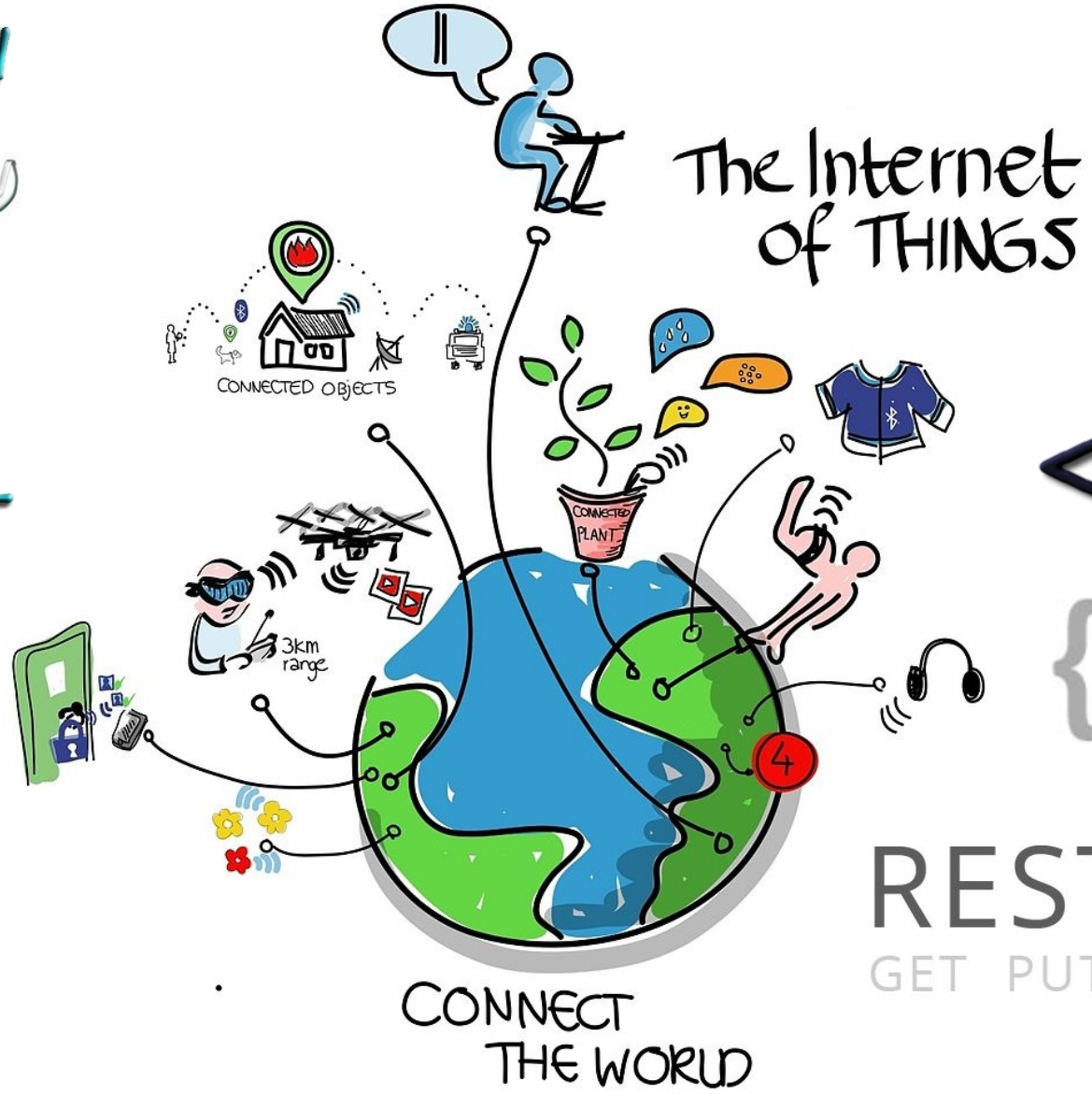
EURECOM
Sophia Antipolis

Embedded Systems Are Everywhere



By Wilgengebroed on Flickr [CC-BY-2.0]

Smarter & More Complex



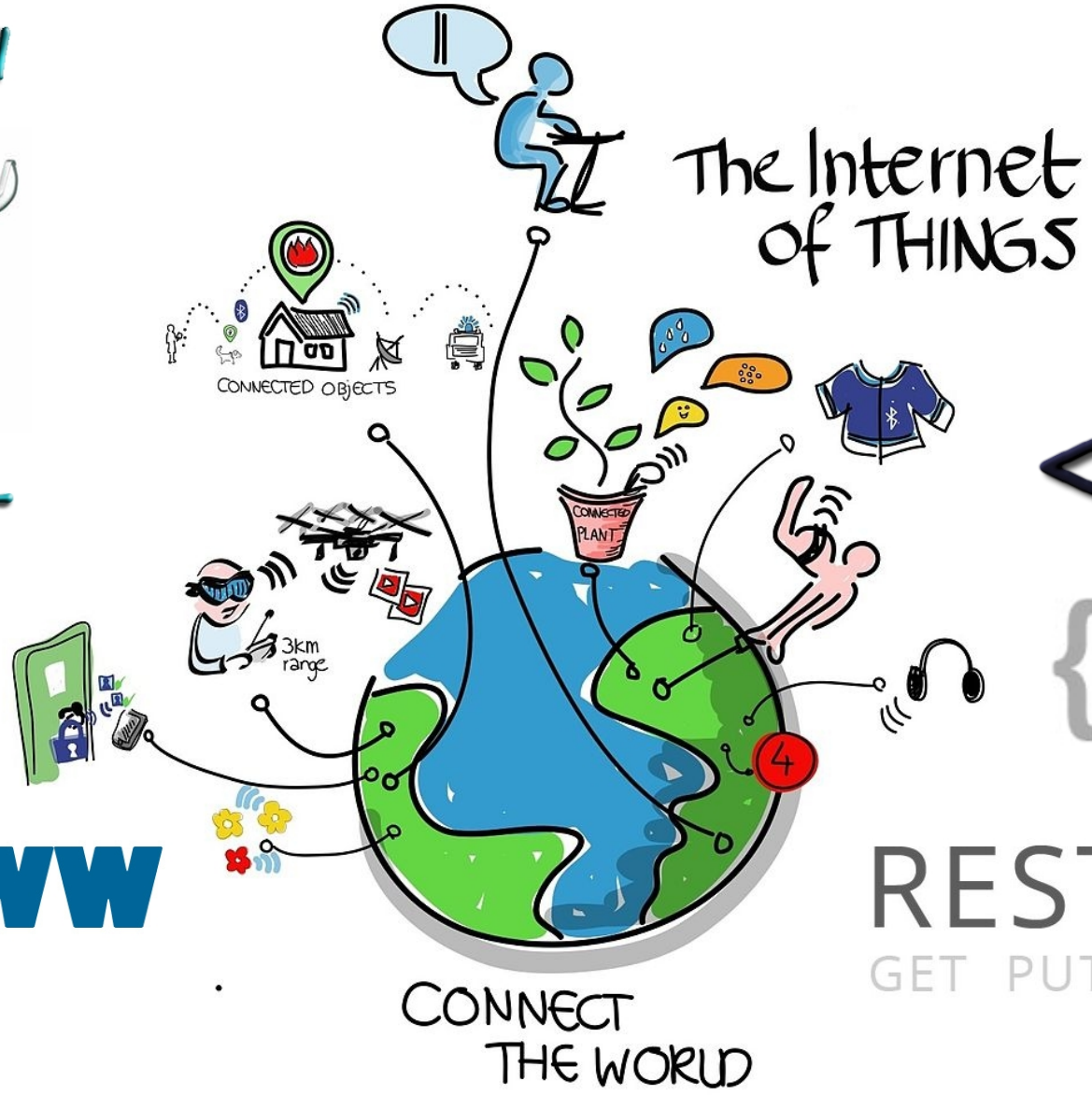
<?xml?>

{JSON}

RESTful API

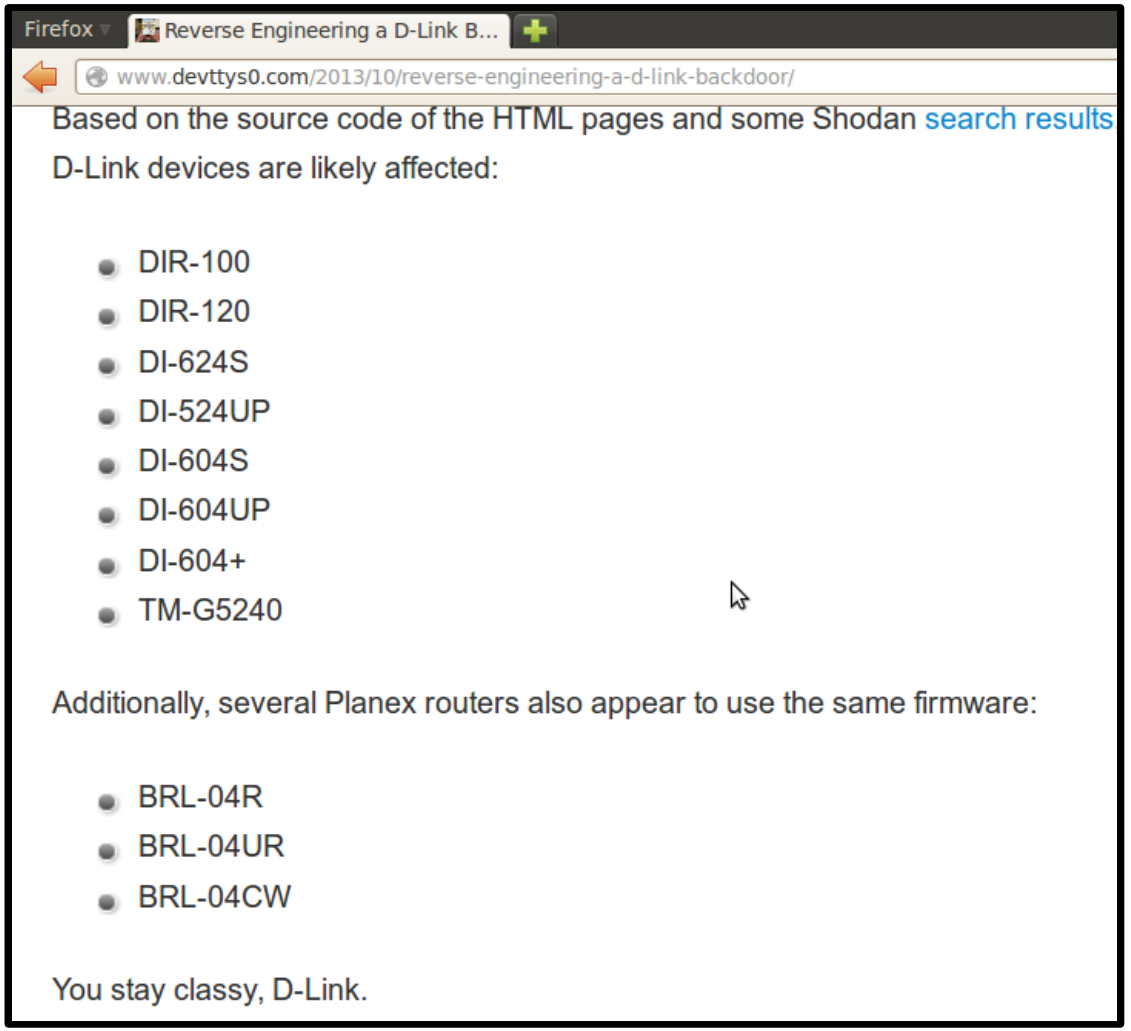
GET PUT POST DELETE

Heavily Interconnected



Many Examples of Insecure Embedded Systems

- Routers



Many Examples of Insecure Embedded Systems

- Routers
- Printers

Networked Printers at Risk

By Jimmy Shah on Dec 30, 2011

Like 0 Share 0 g+1 0 Tweet 0

Multifunction printers (MFPs) have been common in offices for years. They let employees print, scan, and copy documents. Two separate talks at the 28th Chaos Communications Congress (28c3) show how attackers can infect these trusted office devices.

Hacking MFPs

In Andrei Costin's presentation "Hacking MFPs," he covered the history of printer and copier hacks from the 1960s to today. The meat of the talk concerned executing remote code on an MFP using crafted PostScript. Just printing a particular document can get code to run on the machine. Previous research proof of concepts have done exactly that, once with a specially designed Word document and once with a Java applet.

EURECOM
Sophia Antipolis

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP

FBI: Criminals Auto-dialing With Hacked VoIP Systems

By Robert McMillan, IDG News Service Dec 5, 2008 5:50 PM

Criminals are taking advantage of a bug in the Asterisk Internet telephony system that lets them pump out thousands of scam phone calls in an hour, the U.S. Federal Bureau of Investigation warned Friday.

The FBI didn't say which versions of Asterisk were vulnerable to the bug, but it advised users to upgrade to the latest version of the software. Asterisk is an open-source product that lets users turn a Linux computer into a VoIP (Voice over Internet Protocol) telephone exchange.

In so-called vishing attacks, scammers usually use a VoIP system to set up a phony call center and then use phishing e-mails to trick victims into calling the center. Once there, they are prompted to give private information. But in the scam described by the FBI, they apparently are taking over legitimate Asterisk systems in order to directly dial victims.

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars

FORBES 7/24/2013 @ 9:00AM | 506,516 views

Hackers Reveal Nasty New Car Attacks--With Me Behind The Wheel (Video)

This story appears in the August 12, 2013 issue of Forbes.

+ Comment Now + Follow Comments

EURECOM
Sophia Antipolis

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones

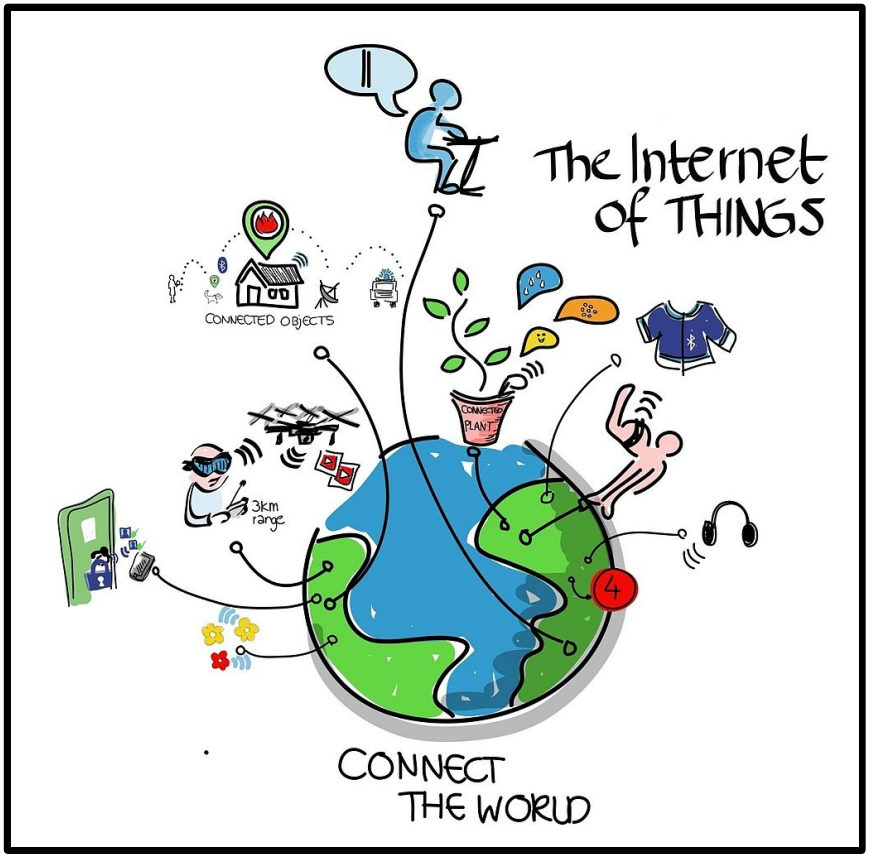
Hacker Releases Software to Hijack Commercial Drones

by BRYANT JORDAN *on* DECEMBER 9, 2013

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

Many Examples of Insecure Embedded Systems

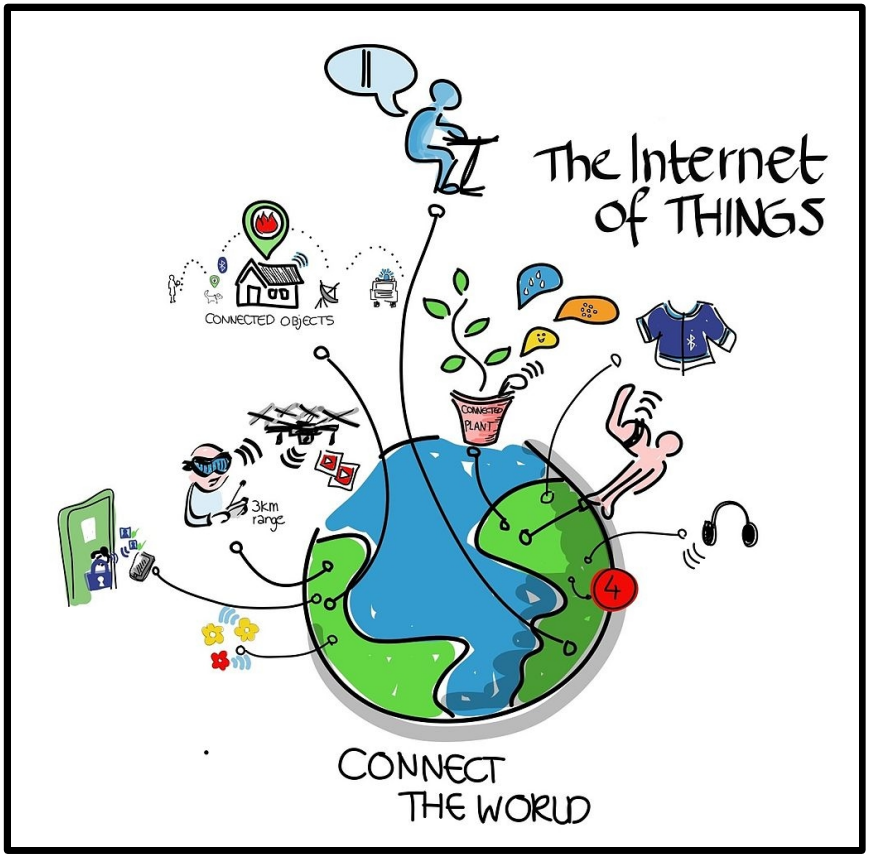
- Routers
- Printers
- VoIP
- Cars
- Drones
- ...



EURECOM
Sophia Antipolis

Many Examples of Insecure Embedded Systems

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



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

The Goal

Perform a large scale analysis to provide a better undestanding of the problem

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 the firmwares
  - Unpacking and performing initial analysis
  - **Re**-discovering the same or similar bug in other firmwares

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

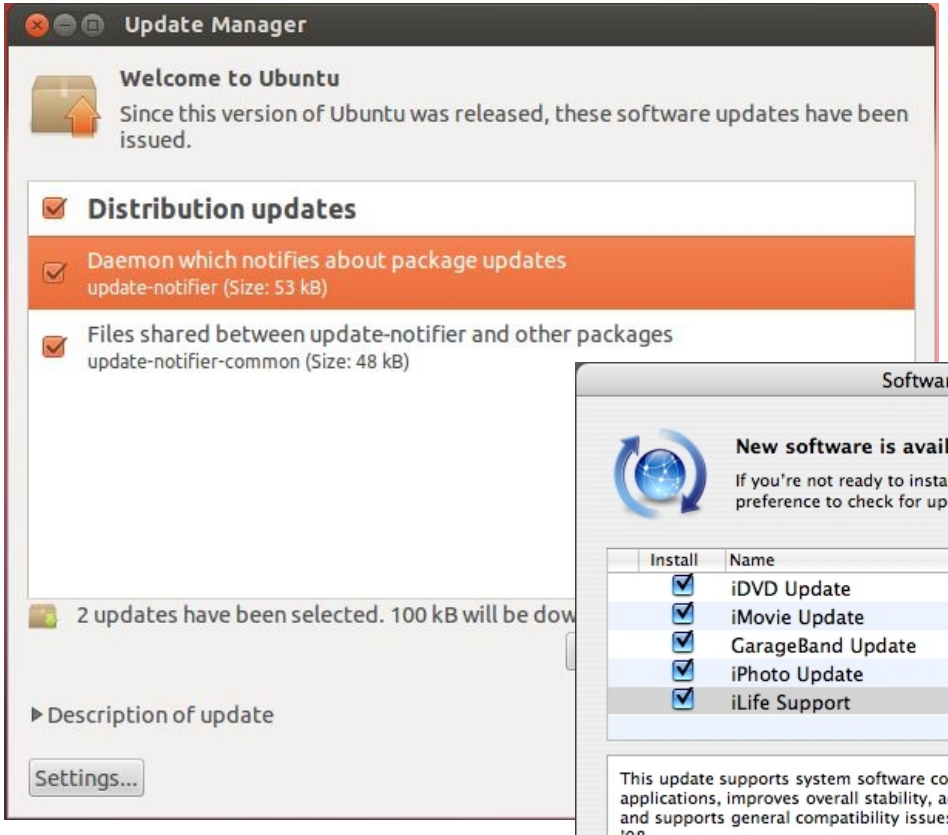
Our Approach to The Large Scale Analysis

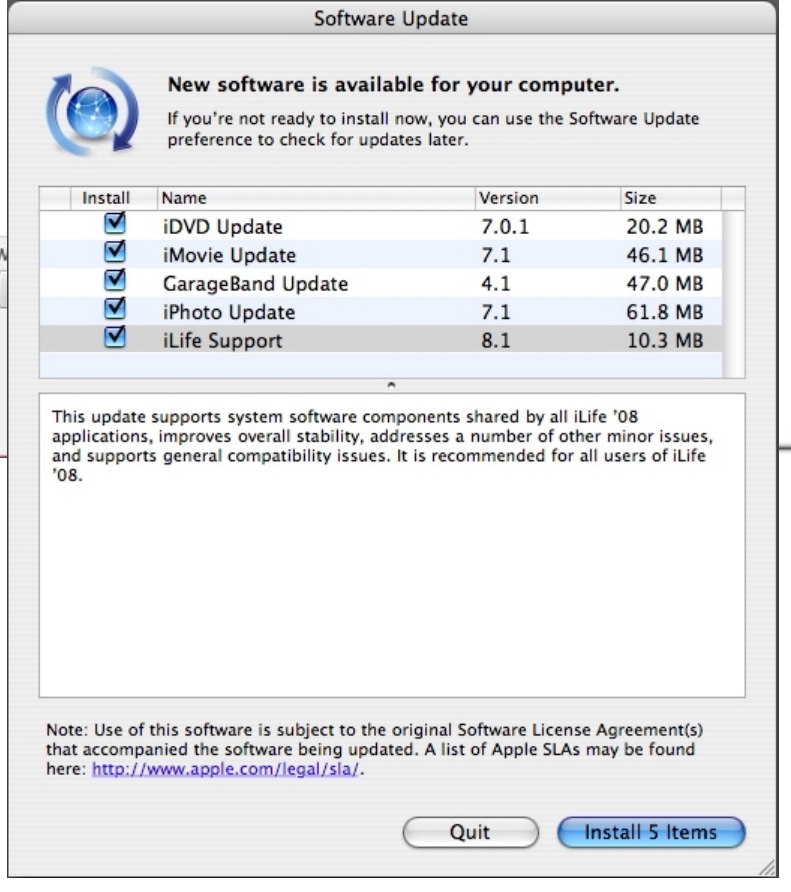
- Collect a large number of firmware images
- Perform broad but simple static analysis
- Correlate across firmwares
- Advantages
  - No intrusive online testing, no devices involved
  - Scalable

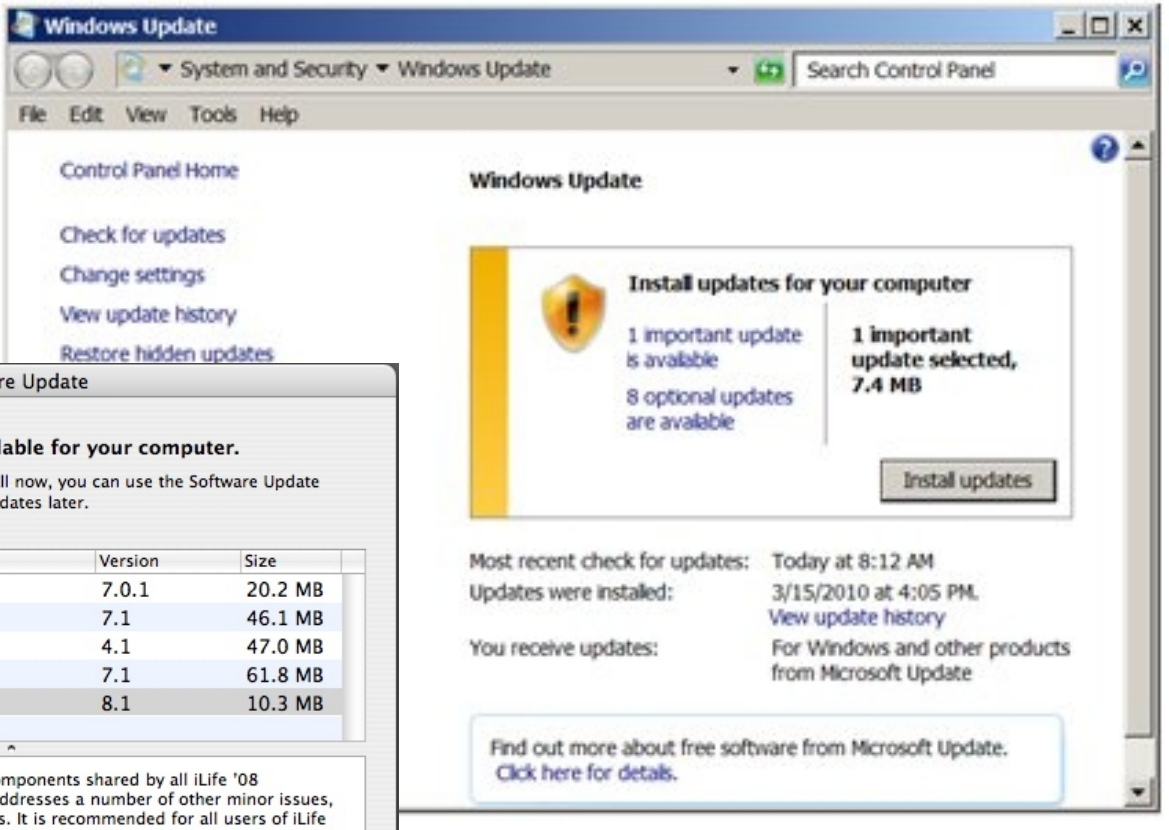
Our Approach to The Large Scale Analysis

- 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
- But many challenges

Mainstream Systems Have Centralized Updates

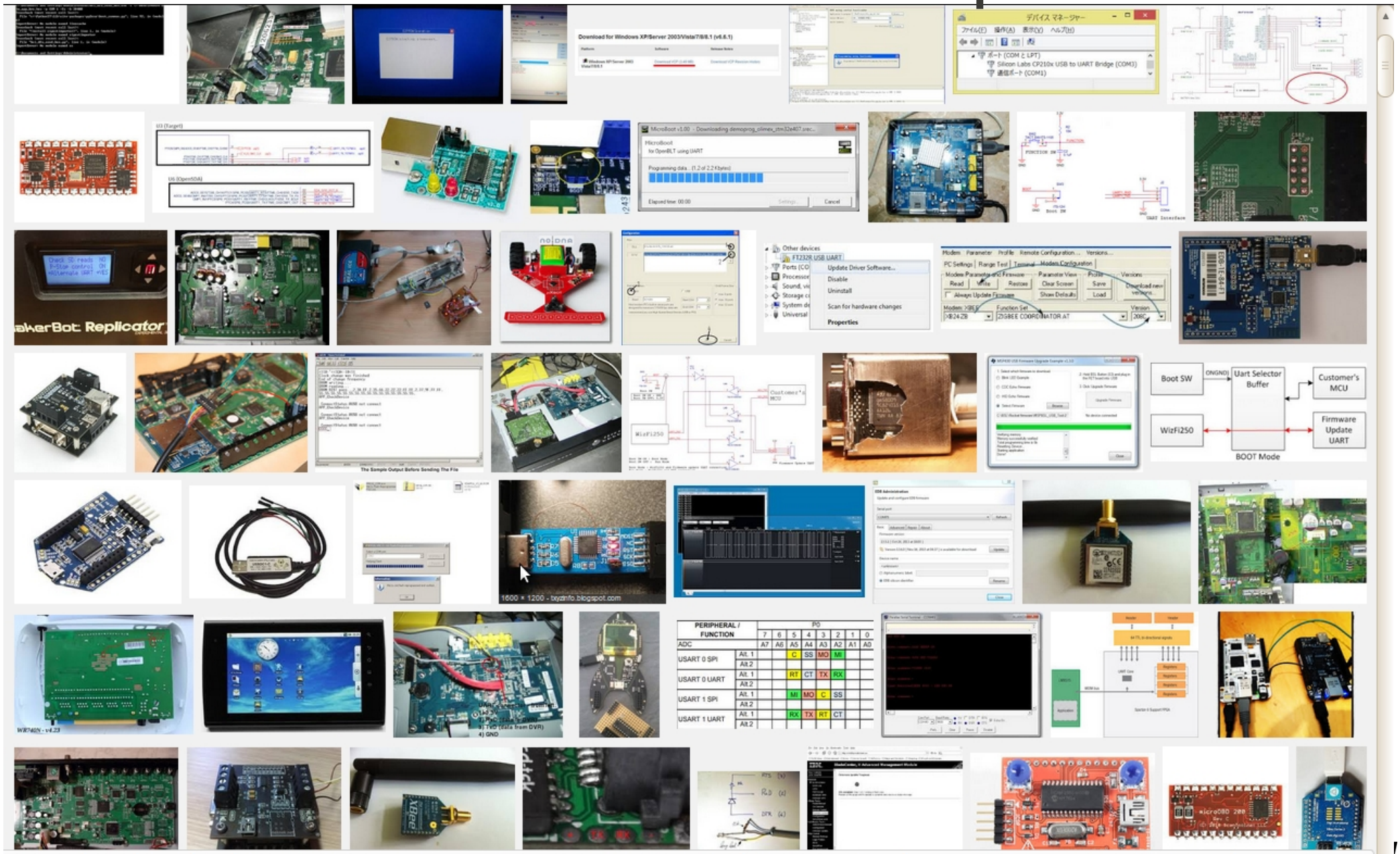






EURECOM
Sophia Antipolis

Challenge: Embedded Systems Have No Centralized Updates



www.google.com/imgres?imgurl=http://3.bp.blogspot.com/-eoyP93DWXHM/ULGN-zu9vpI...om=1&docid=5yINoC5G55MiVM&ei=E73oU4egAsim0QXZ14CICQ&tbm=isch&ved=0CEoQMygjMCM

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

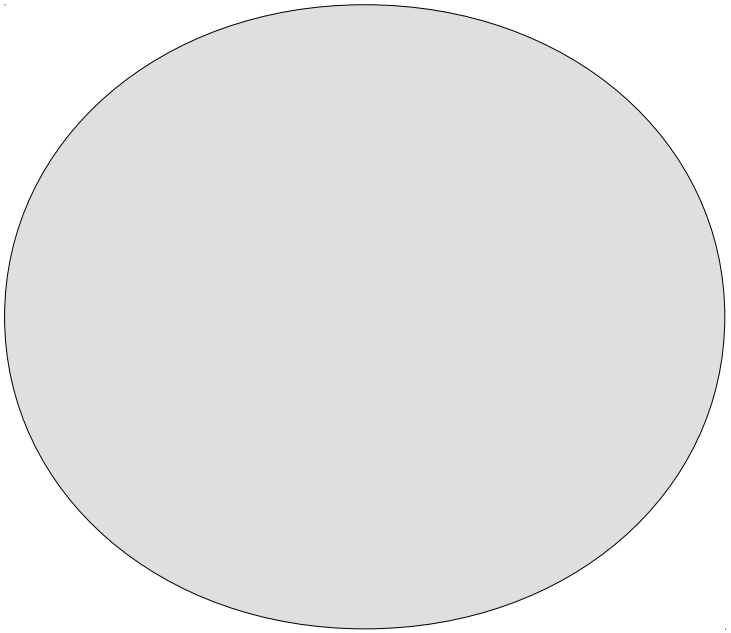
- No large scale firmware dataset yet
  - As opposed to existing datasets in security or other CS research areas
- We collected a subset of the firmwares available for download
- Many firmwares are not publicly available
  - Not intended to have an upgrade
  - Require product purchase and registration

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

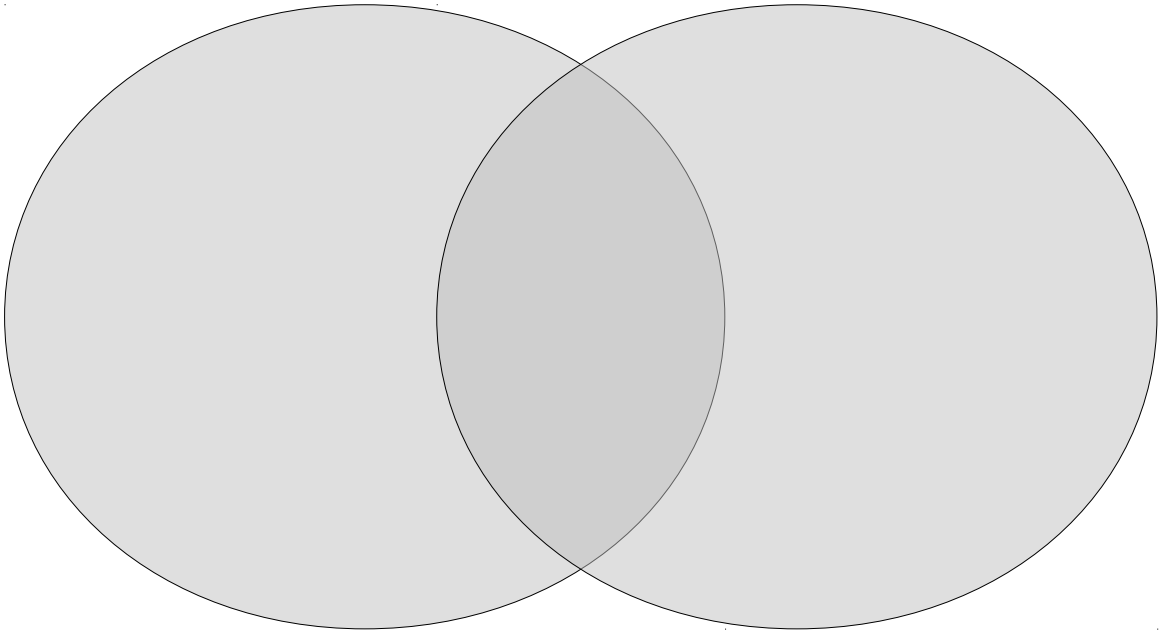
Clearly a Firmware



Challenge: Firmware Identification

Clearly a Firmware

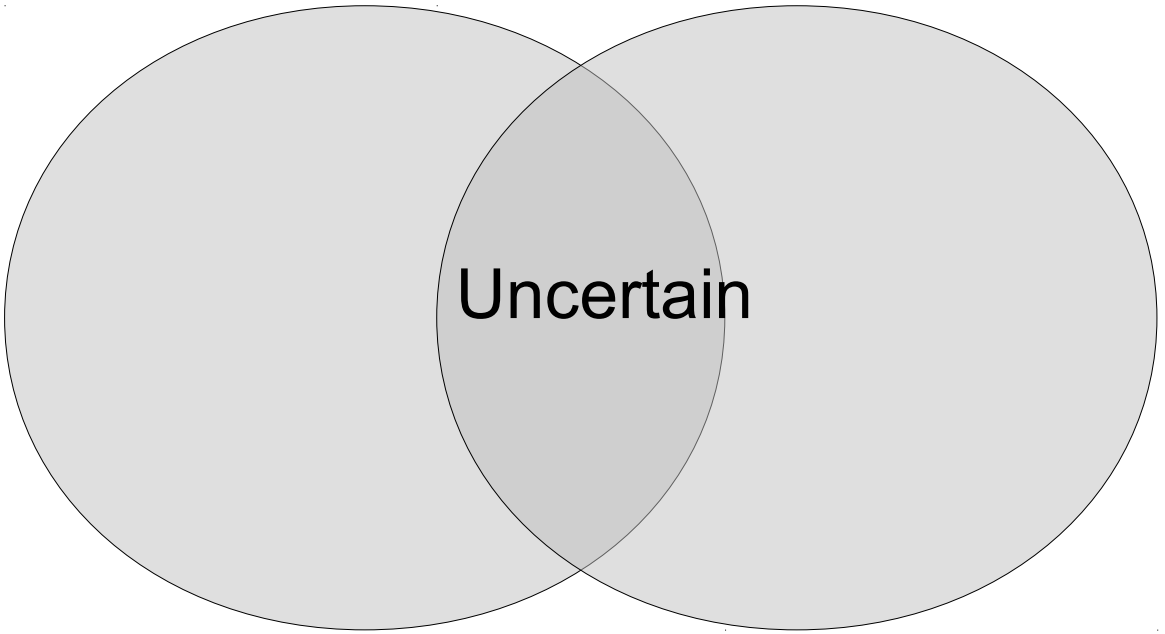
Clearly not a Firmware



Challenge: Firmware Identification

Clearly a Firmware

Clearly not a Firmware



Challenge: Firmware Identification

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

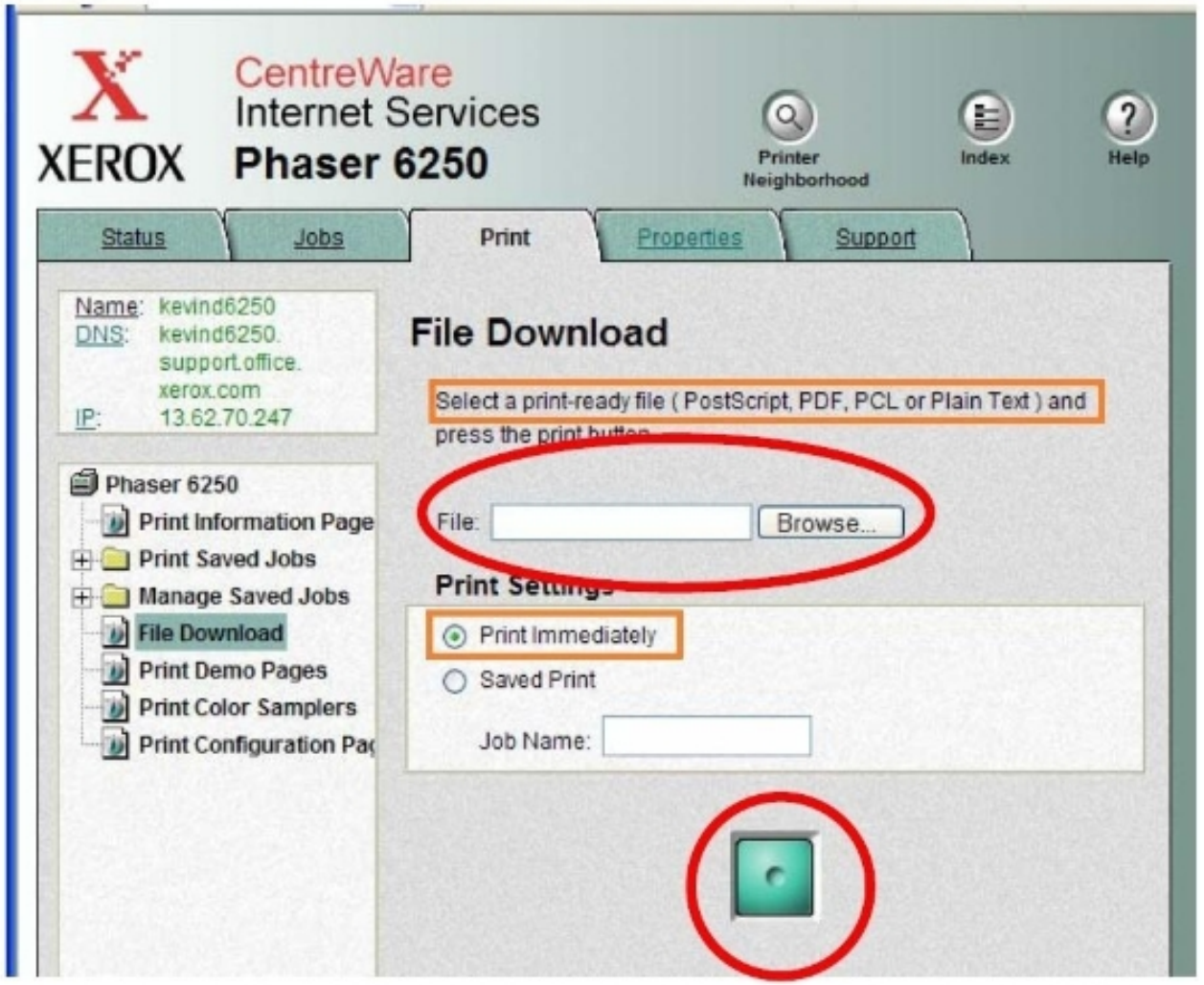


Figure 4: Select the firmware update file and press the green button to send it.

Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?

Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?
- E.g., vendor provides a .ZIP 'firmware package'
  - .ZIP→.EXE+.PS
    - .EXE→self-extracting archive
      - Extract more or not?
      - Turns out to contain a printer driver inside

Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?
- E.g., vendor provides a .ZIP 'firmware package'
  - .ZIP→.EXE+.PS
    - .EXE→self-extracting archive
      - Extract more or not?
      - Turns out to contain a printer driver inside
    - .PS→ASCII85 stream→ELF file that could be:
      - A complete embedded system software
      - An executable performing the firmware upgrade
      - A firmware patch

Challenge: Unpacking & Custom Formats

- How to reliably unpack and learn formats?
- E.g., vendor provides a .ZIP 'firmware package'
  - .ZIP→.EXE+.PS
    - .EXE→self-extracting archive
      - Extract more or not?
      - Turns out to contain a printer driver inside
    - .PS→ASCII85 stream→ELF file that could be:
      - A complete embedded system software
      - An executable performing the firmware upgrade
      - A firmware patch
- Often, a firmware image→just 'data' binary 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

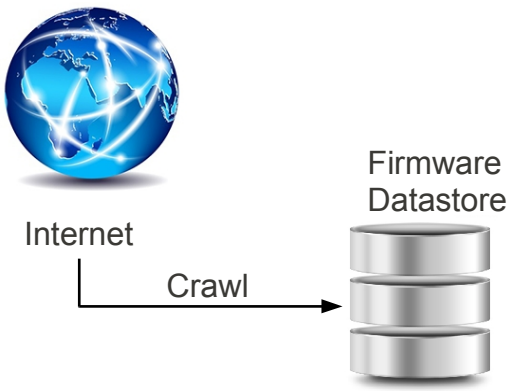
Challenge: Results Confirmation

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

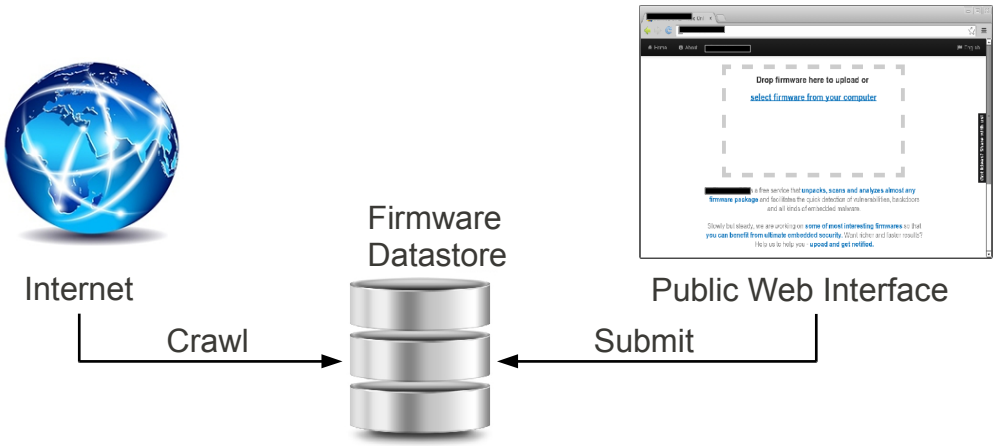
Challenge: Results Confirmation

- 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

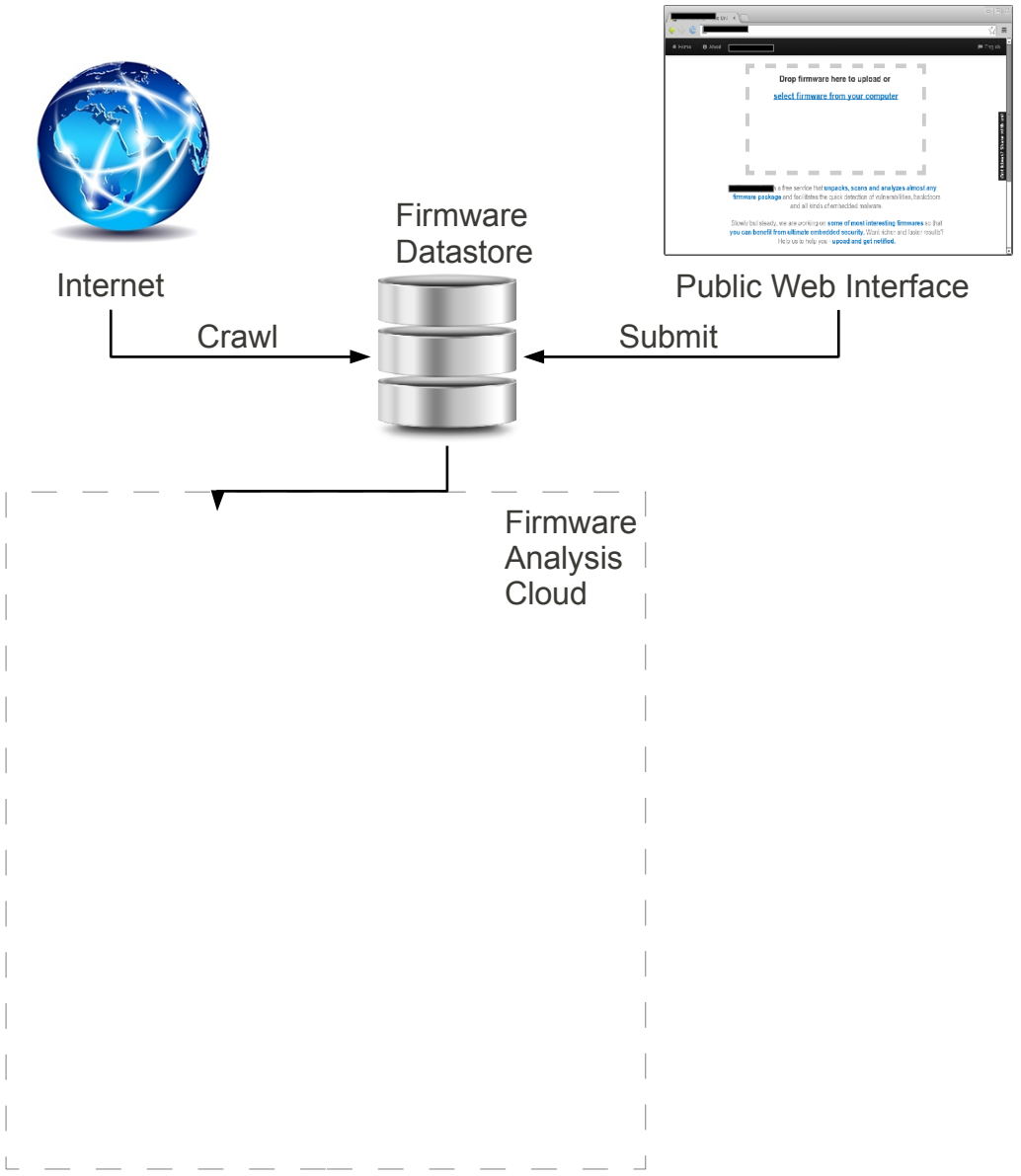
Architecture



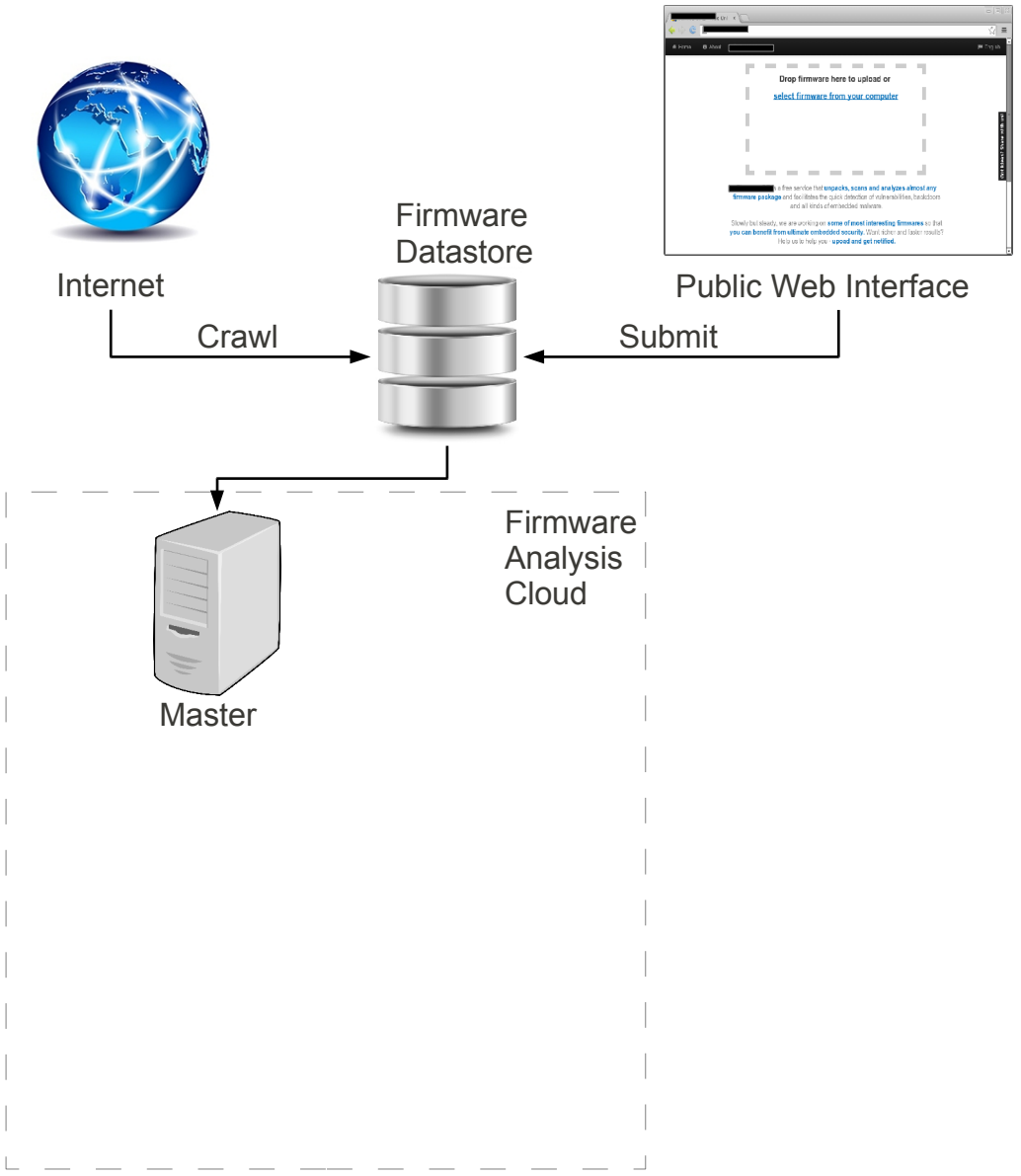
Architecture



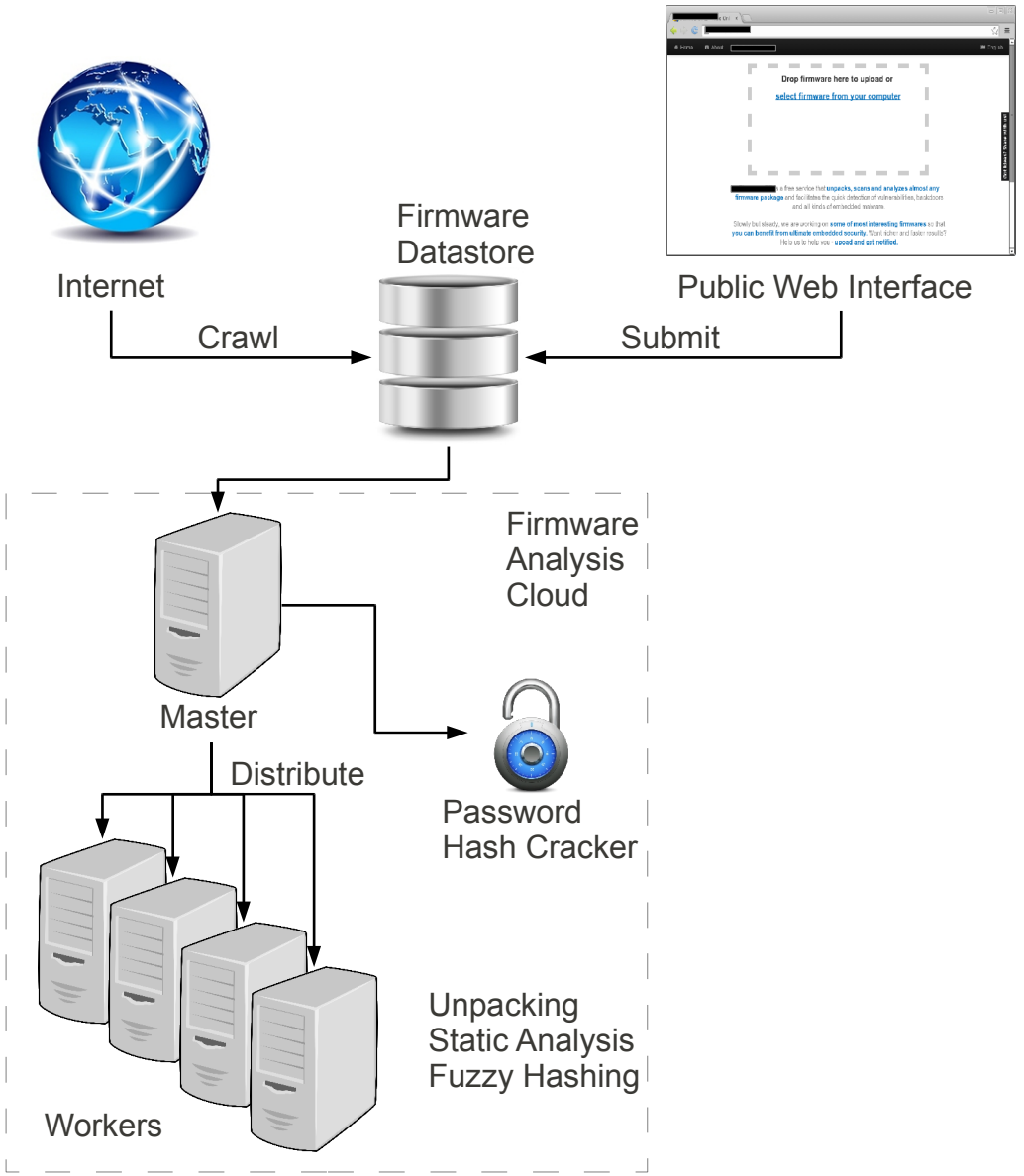
Architecture



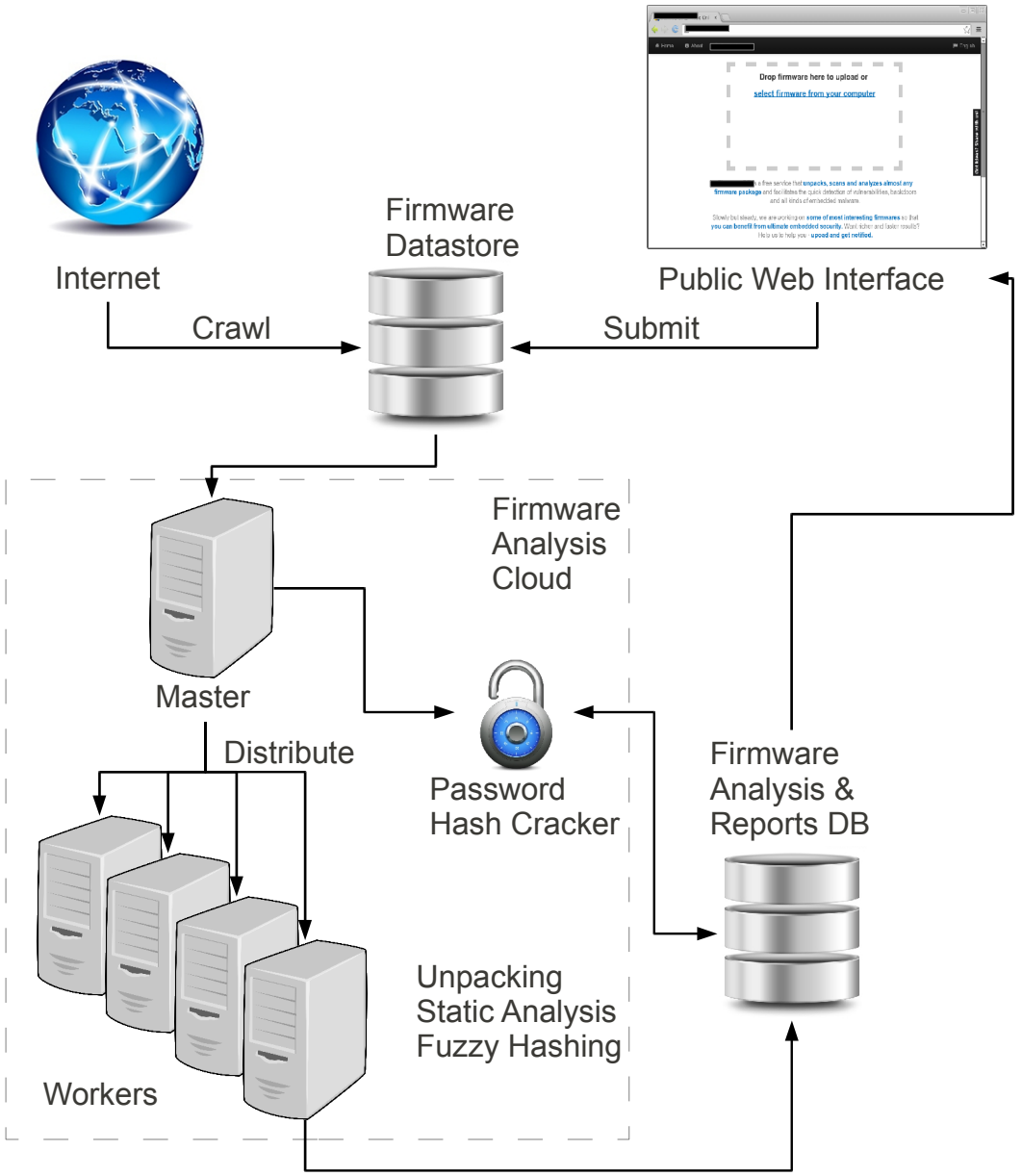
Architecture



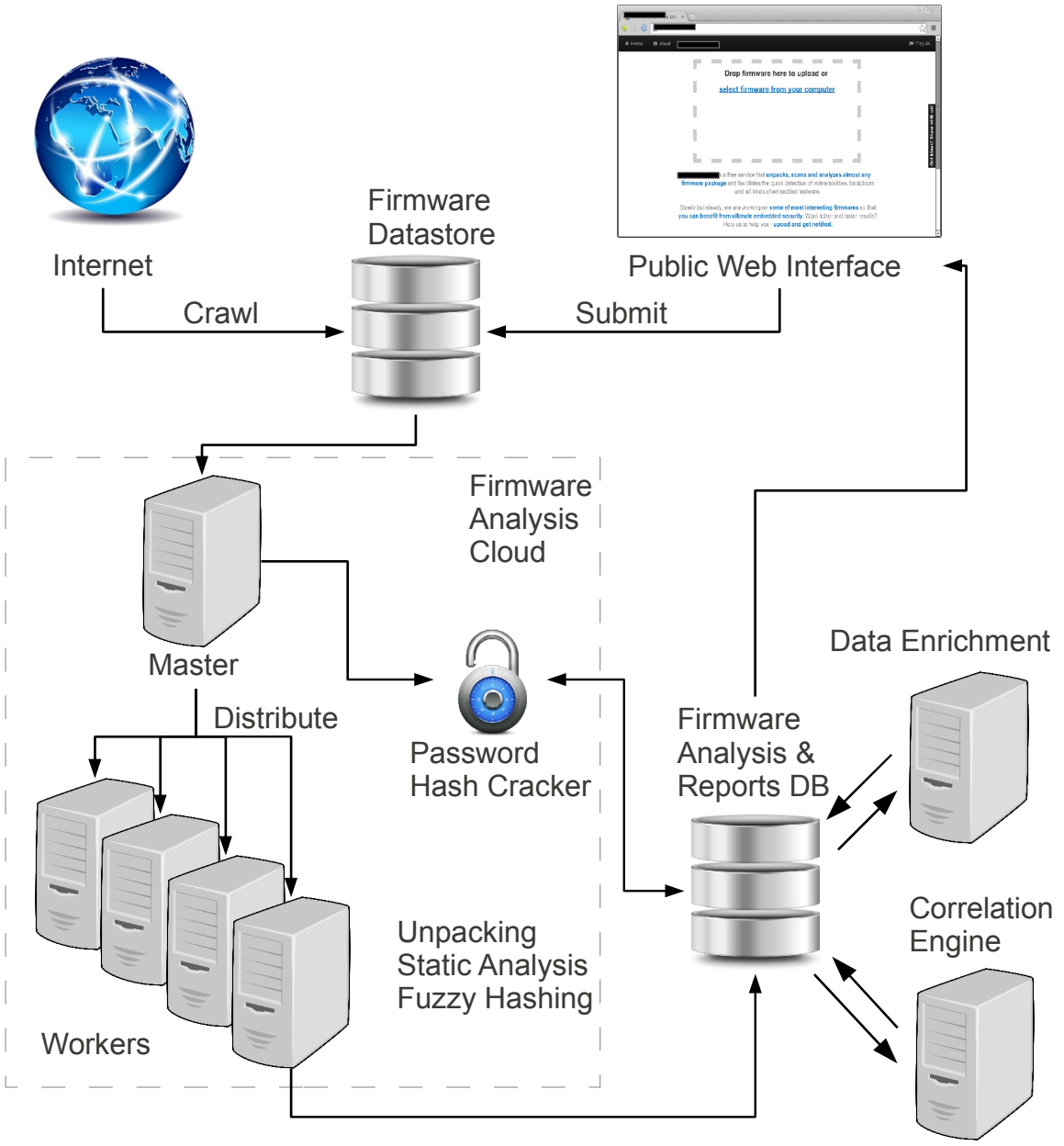
Architecture



Architecture



Architecture



EURECOM
Sophia Antipolis

Crawler

- 759 K collected files, 1.8 TB of disk space

Crawler

- 759 K collected files, 1.8 TB of disk space
- FTP-index engines

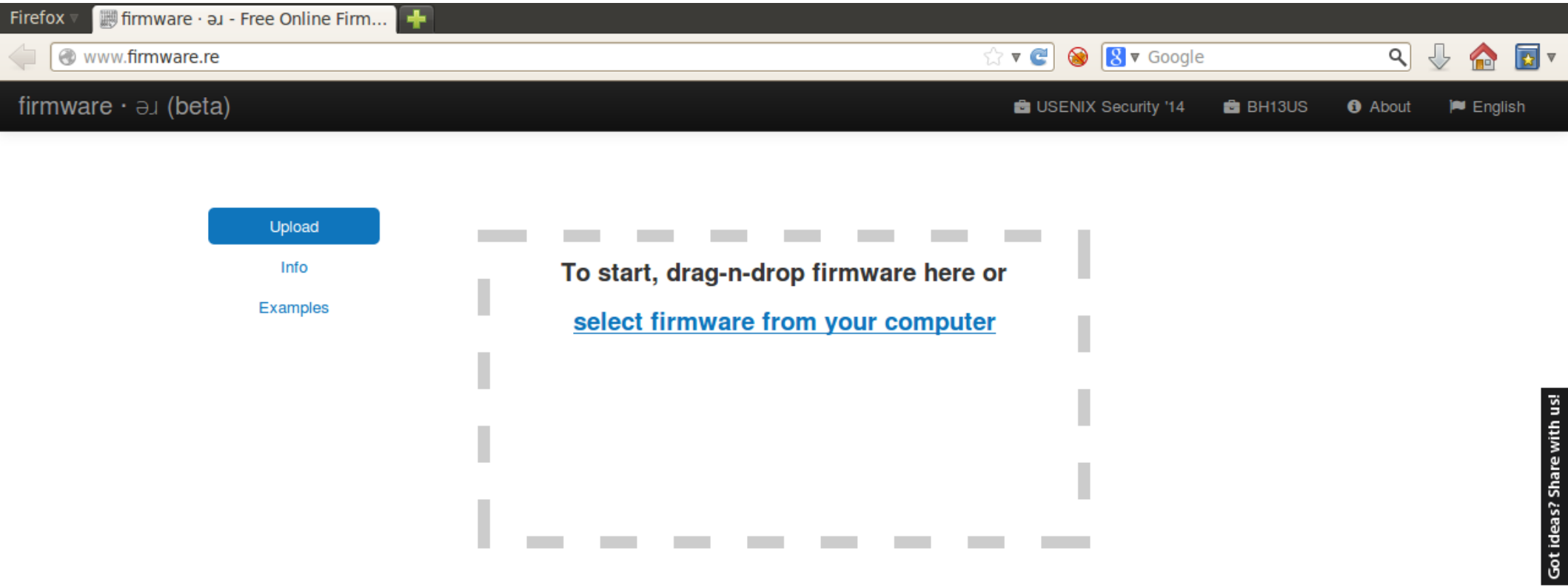
EURECOM
Sophia Antipolis

Crawler

- 759 K collected files, 1.8 TB of disk space
- FTP-index engines and GCSE

EURECOM
Sophia Antipolis

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



Unpacking

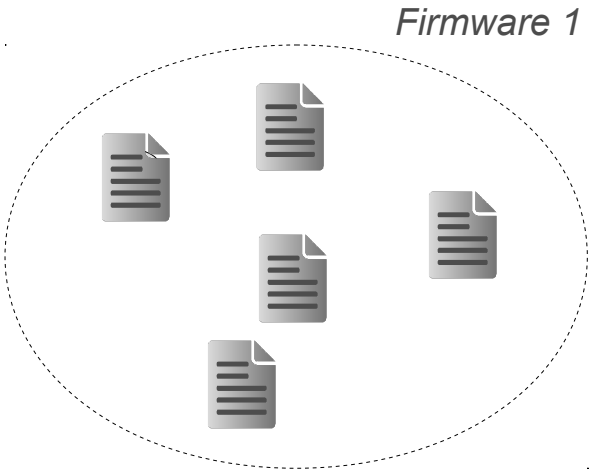
- 759 K total files collected
  - Filter non firmware
- 172 K filtered interesting files
  - Random selection
- 32 K analyzed
  - Successful unpack
- 26 K unpacked (fully or partially)
  - Unpacked files
- 1.7 M resulted files after unpacking

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)

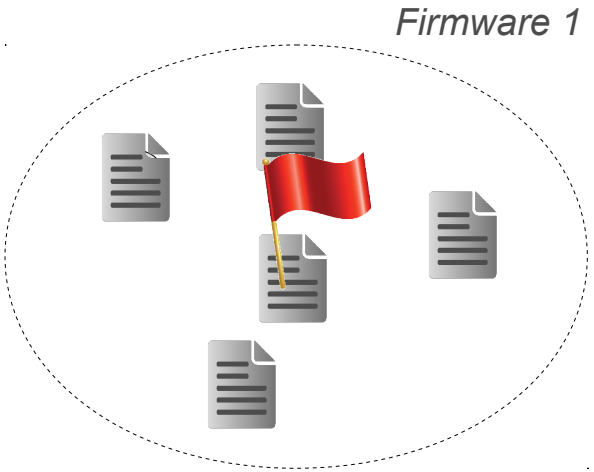
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)
  - E.g., Vulnerability Propagation



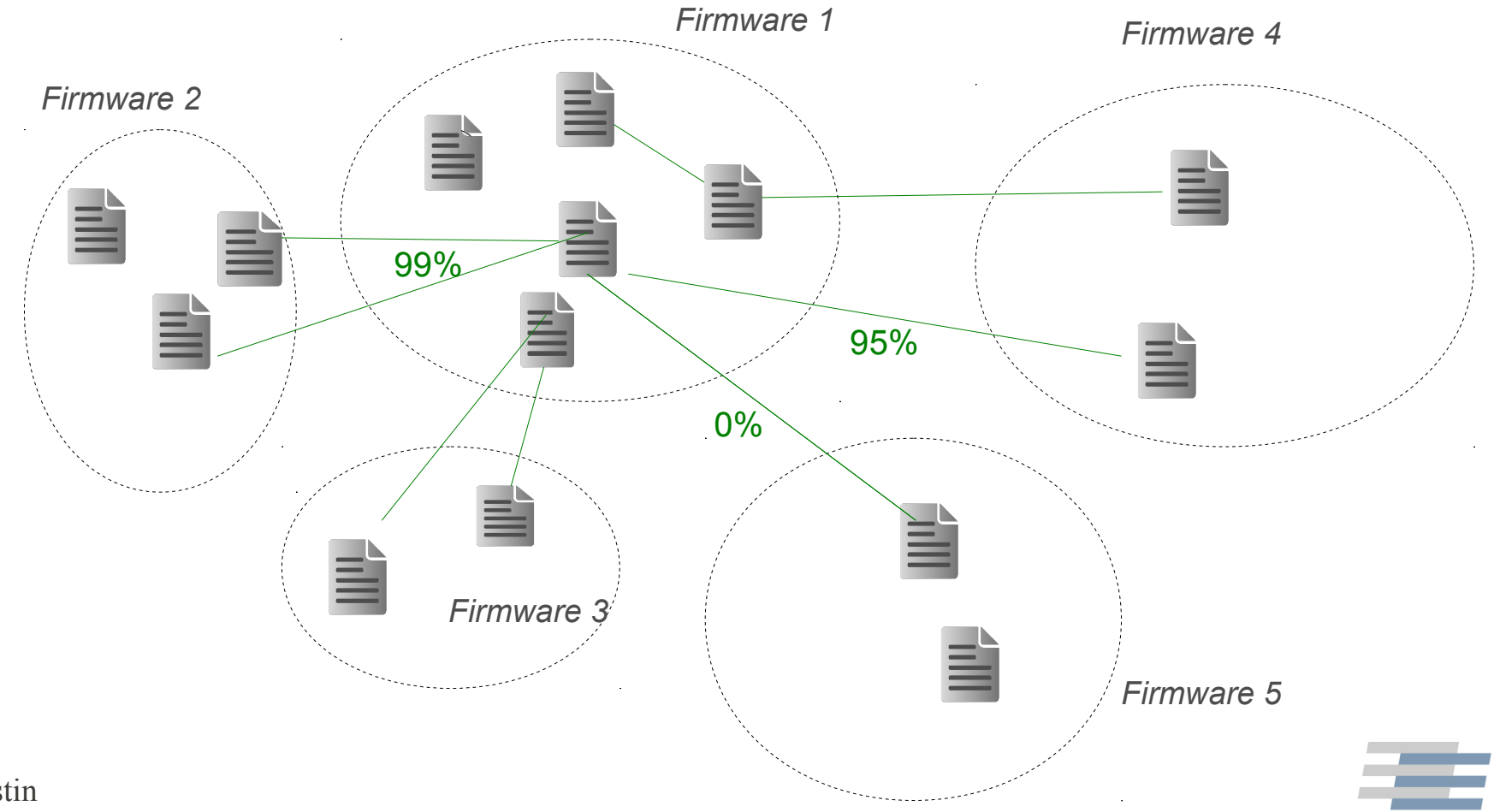
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)
  - E.g., Vulnerability Propagation



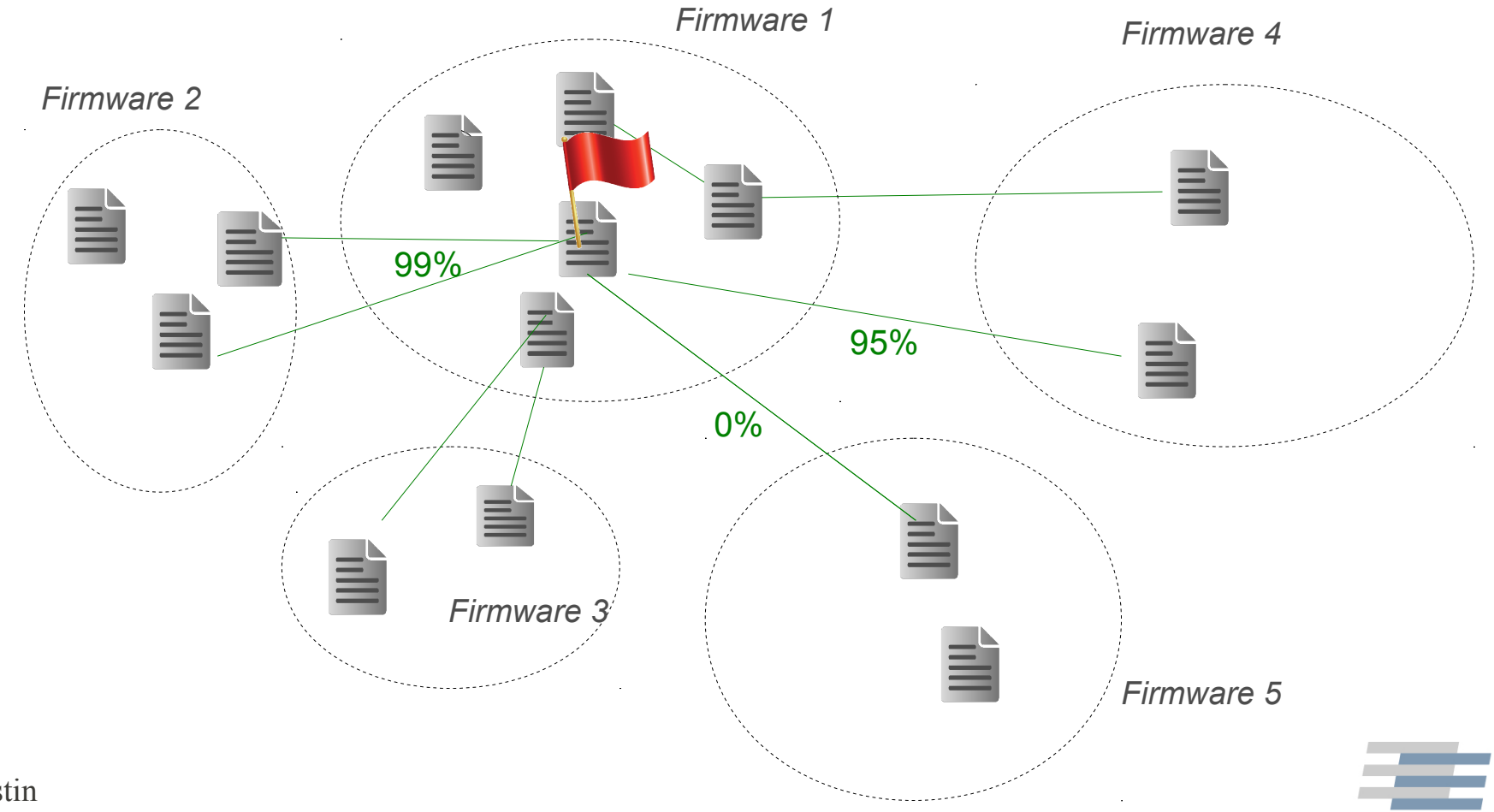
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)
  - E.g., Vulnerability Propagation



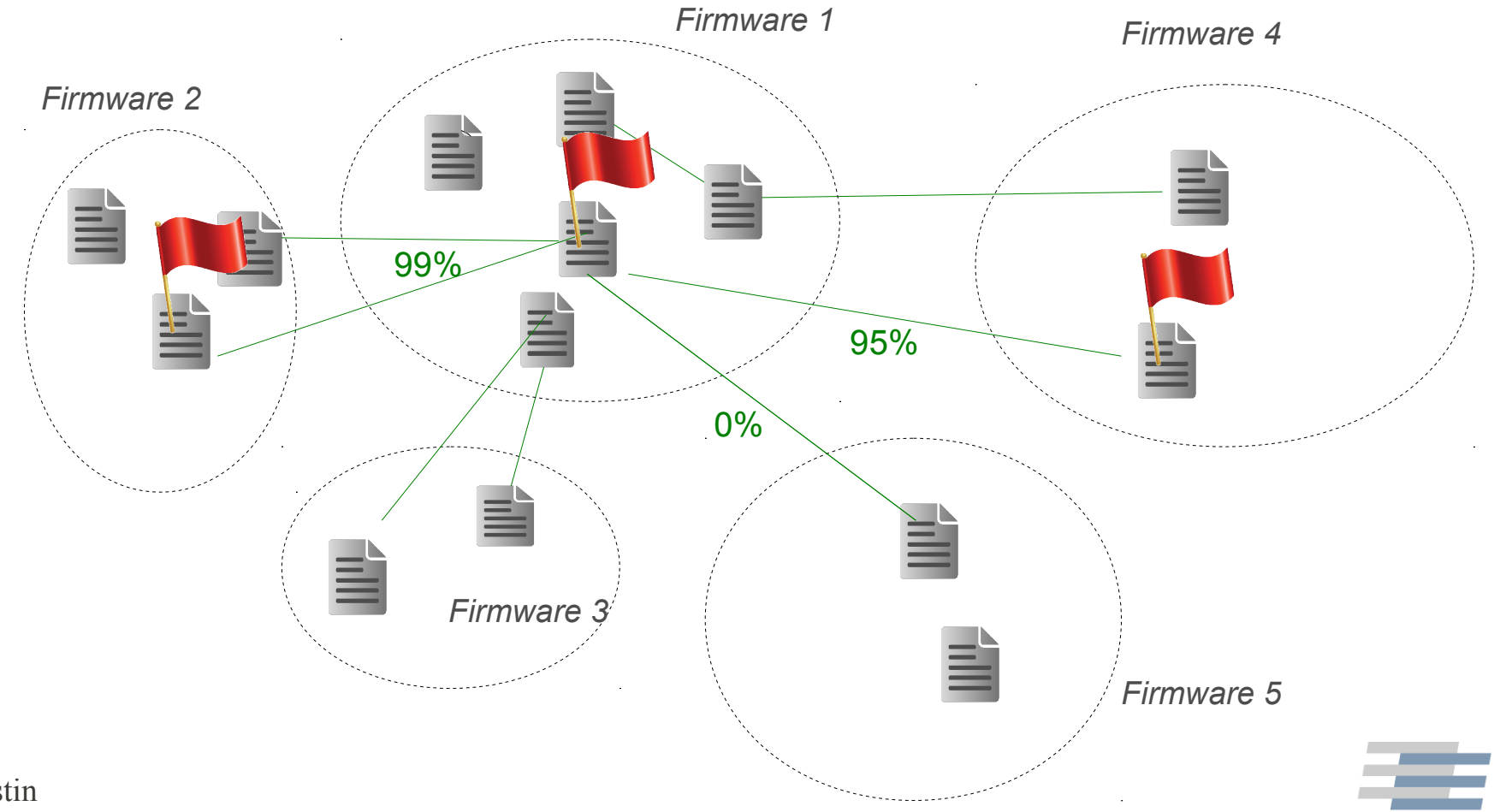
Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)
  - E.g., Vulnerability Propagation



Example: Correlation

- Correlation via fuzzy-hashes (ssdeep, sdhash)
  - E.g., Vulnerability Propagation



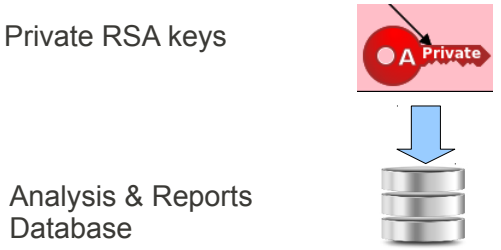
Example: RSA Keys

- SSL keys correlation + vulnerability propagation

Private RSA keys

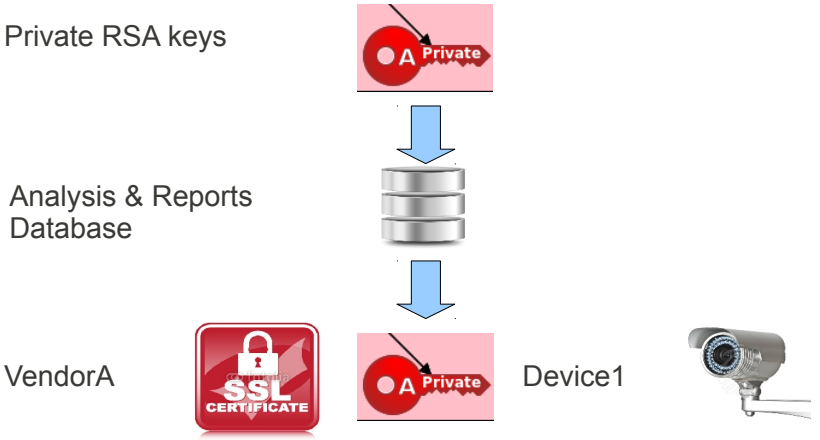
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



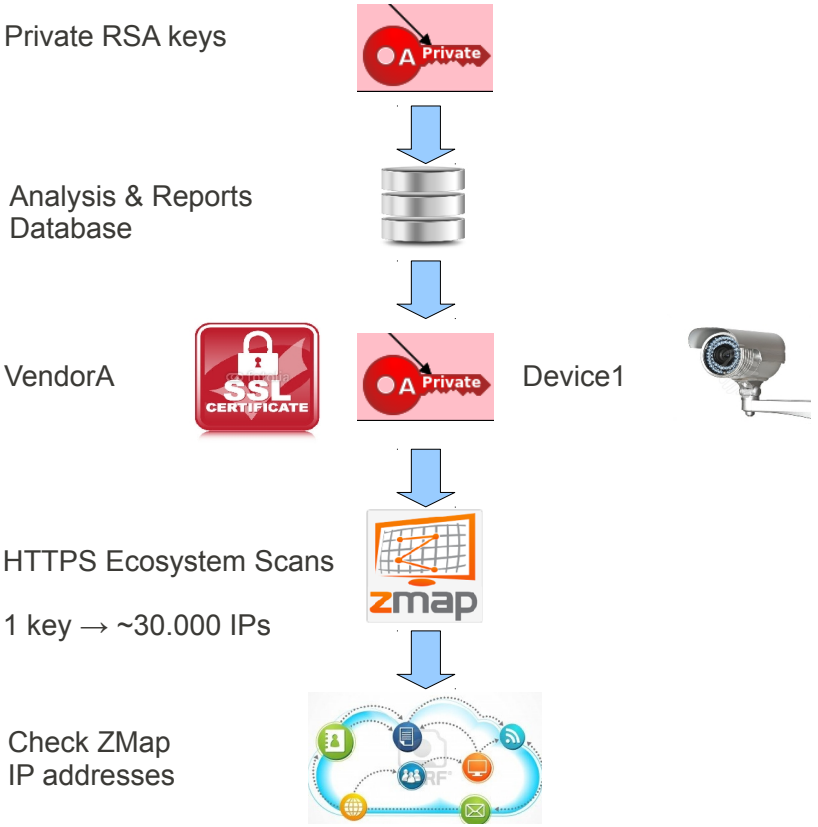
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



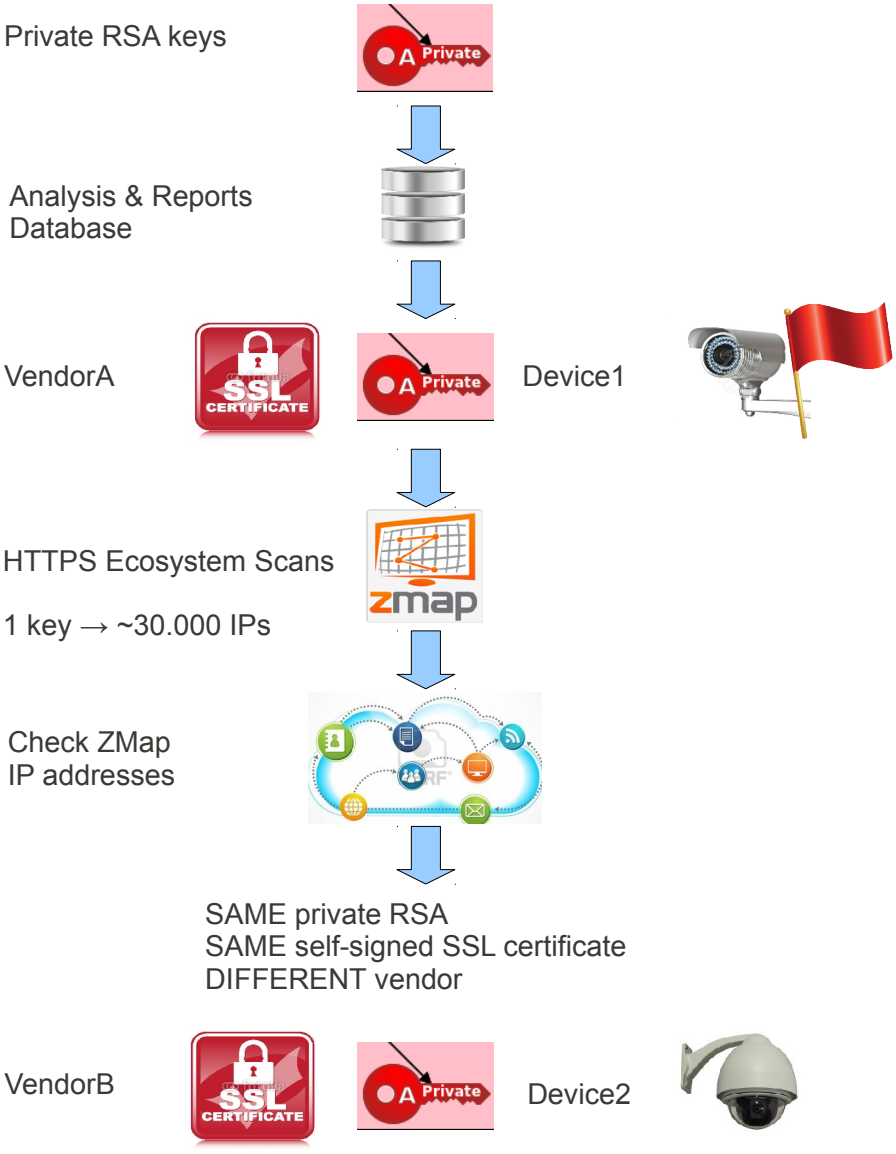
Example: RSA Keys

- SSL keys correlation + vulnerability propagation



Example: RSA Keys

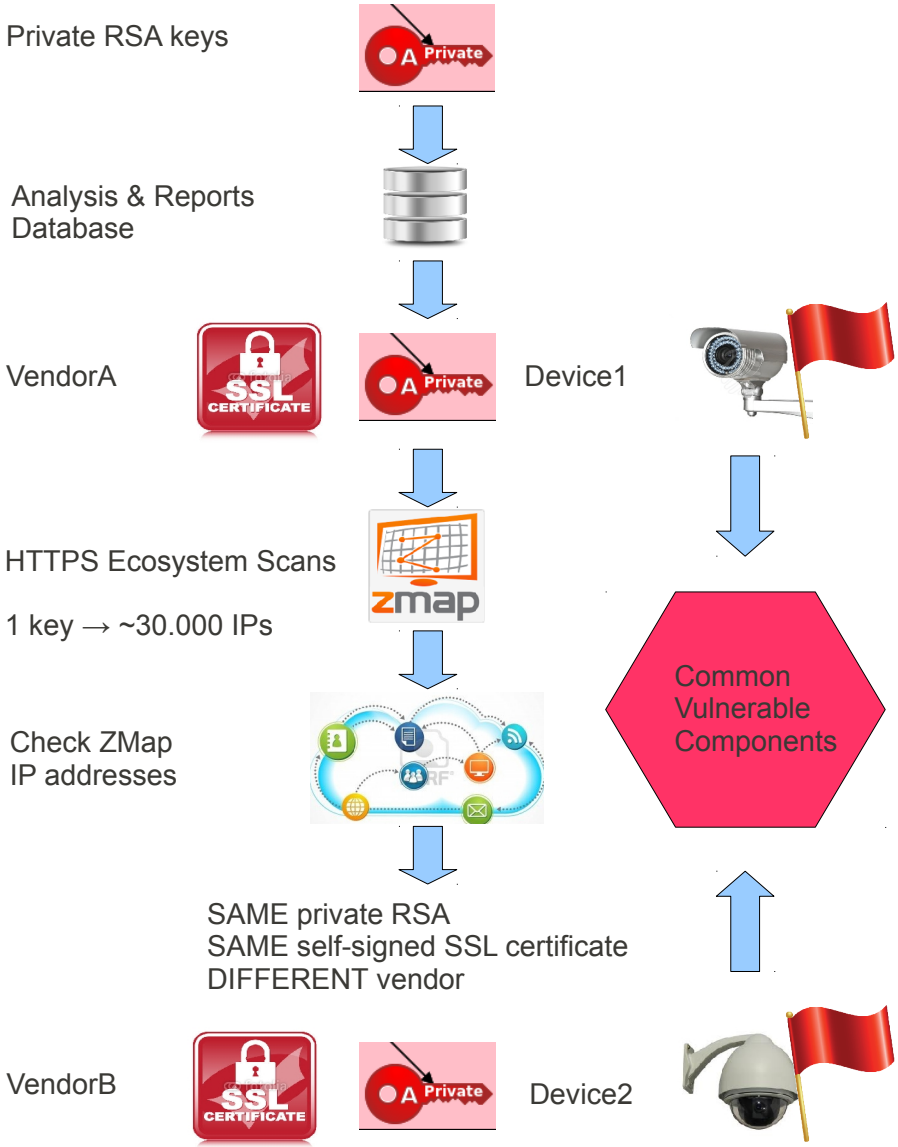
- SSL keys correlation + vulnerability propagation



EURECOM
Sophia Antipolis

Example: RSA Keys

- SSL keys correlation + vulnerability propagation



EURECOM
Sophia Antipolis

Results: Summary

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

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 relatioship
  - Shows how vulnerabilities reappear across different products
  - Could allow seeing how firmwares evolve/get fixed

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

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