IoT Sandbox – To Analysis IoT malware Zollard

Kai-Chi Chang
Department of Computer Science
National Chengchi University
National Center for Cyber Security Technology
NO.64,Sec.2,ZhiNan Rd.,Wenshan District,Taipei City 11605,Taiwan
nesoking@nccst.nat.gov.tw

Raylin Tso
Department of Computer Science
National Chengchi University
NO.64,Sec.2,ZhiNan Rd.,Wenshan District,Taipei City 11605,Taiwan
raylin@cs.nccu.edu.tw

Min-Chun Tsai
National Center for Cyber Security Technology
No.116, Fuyang St., Daan Dist., Taipei City 10676, Taiwan
jamestsai@nccst.nat.gov.tw

ABSTRACT
As we know, we are already facing IoT threat and under IoT attacks. However, there are only a few discussions on, how to analyze this kind of cyber threat and malwares. In this paper, we propose IoT sandbox which can support different type of CPU architecture. It can be used to analyze IoT malwares, collect network packets, identify spread method and record malwares behaviors. To make sure our IoT sandbox can be functional, we implement it and use the Zollard botnet for experiment. According to our experimental data, we found that at least 71,148 IP have been compromised. Some of them are IoT devices (DVR, Web Camera, Router WiFi Disk, Set-top box) and others are ICS devices (Heat pump and ICS data acquisition server). Based on our IoT sandbox technology, we can discover an IoT malware in an early stage. This could help IT manager or security experts to analysis and determine IDS rules. We hope this research can prevent IoT threat and enhance IoT Security in the near future.

Keywords;

IoT; Sandbox; Malware

1. INTRODUCTION
IoT (Internet of Things), has come into limelight in recent years. IoT is collective nouns, made up of sensing technology, wireless communication technology, cloud computing and massive data analysis technology. In formal, IoT means physical devices connect to Internet base on traditional telecommunications with address, moreover could search and communicate each other. Gates、Myhrvold and Rinearson propose the concept it in 1995 and ITU (International Telecommunication Union) also published in 2005[1]. Cisco and Morgan Stanley say, globe IoT devices would more than 50 billion [2]. Until now, IoT devices are more than the global population. Then we already live in IoT generation. IoT brings convenience to life and also hidden crisis. People use DVR (Digital Video Recorder) and IP camera to protect valuables surrounding people or things. Use Wi-Fi Disk and Set-top box to sundry leisure life. However, most of them are easier to hack.

ICC '17, March 22 2017, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/3018896.3018898

John Matherly published a IoT search engine in 2009 DEFCON conference, it's name is Shodan [3]. Shodan can be used to search IoT devices and find out vulnerable devices. Dan Tentler showed how ICS and IoT devices base on Shodan in 2012 DEFCON conference. Shekyan and Harutyunyan also demoed, how to hack DVR and door lock in 2012 Hack in a Box conference [4].

According to the above description, we know that to hack IoT devices is not difficult, at least from the perspective of a hack. If hackers hack your devices successful then what could be happen? It means, hackers can extract your image from DVR or IP camera. Hackers also can infect your DSL Router and set up malicious DNS (Domain Name server). It is also possible that hackers steal your personal information or financial information via malicious phishing website. Consequently, it caused a lot of security issues such as privacy issue, data leakage, public safety and etc..

In addition, hackers also could write an IoT malware to compromise devices, and used to lunch cyber-attack such as DDoS attack, spread spam mail and etc.. DDoS attack is well known as a cyber weapon, it used to denial service. Hacker also can spread mail base on IoT devices. Internet-connected fridge was discovered as a part of a botnet sending over 750,000 spam e-mails [5]. So how to analyze IoT malware is getting important. If we want to analyze malware efficiently, then we need to build a sandbox. A sandbox can be divided into two categories, physical sandbox and virtual sandbox.

1.1 Physical sandbox
To run the malware in physical device without virtual machine. The advantage is, we do not need to worry about anti-VM technology. As we know, most of the exquisite malwares use anti-VM techniques to thwart attempts at analysis. So most malware can run in physical sandbox. The drawback is its inefficiency, if you want to recover OS.

1.2 Virtual sandbox
To run the malware in virtual machine. The advantage is, it can recover OS quickly and deploy sandbox conveniently. Most of the existing malwares can be analyzed by a virtual sandbox. The drawback is that it cannot analysis a malware with anti-VM module.

With the fact that most of the existing malwares can be detected by a virtual sandbox, and our work is the first attempt to collect and analysis IoT malwares working on different CPU architectures, in this paper, we focus on IoT sandbox in the type of virtual sandbox.

In this paper, we propose IoT sandbox with different type CPU architecture and investigate analysis process that include host behavior and network behavior. Also use a real case to prove it.

2. RELATED WORK

Since years, it is known that many Internet of Things (IoT) devices are vulnerable to simple intrusion attempts, ˇCeleda et. al detect Chuck Norris Botnet since Oct. 2009 to Feb. 2010. They found that most IoT malware spread base on telnet protocol and using weak or even default passwords [6]. But there is no description about how to analyze IoT malware that have different CPU architecture.

Yin Minn Pa Pa et. al found there are a lot of telent connection since 2014. So they develop IoT pot base on telnet protocol to capture IoT malwares [7]. Then also develop IoTBOX to analyze IoT malwares with multiple CPU architecture. But they focus on network behavior. There is no descriptive analysis process or how malware host behavior is recorded?

Asmitha K A and Vinod P collect malwares behavior base on Strace command and analyze it base on machine learning approach [8]. But there is no description about how to analyze or record network behavior.

3. IOT SANDBOX

According to the above description. We are facing the flood of IoT malwares and Botnet. On the other hand, in order for a hacker to compromise IoT devices, he has to create a lot of malwares with different kind of CPU architectures. IoT malwares cannot be triggered on tradional sandboxes. What is the sticking point? The answer is limited CPU architecture. Most of traditional sandbox used Intel CPU.

So, we propose IoT sandbox architecture. It has multiple CPU architecture. Until now, it can support 9 kinds of CPU architectures. Including arm (el,hf), mips, mipsel, powerpc, sparc x86, x64 and sh4. It used to analyze malwares and collect malwares behavior. After that, we can extract intelligence from those information. And also do some preventions, for example generate IoT malwares snort rules to protect our network. The detail architecture is described in Figure 1. There are five element and four process in our IoT sandbox. It is implemented in a Vmware [9] virtual machine with ubuntu 14.04 [10].

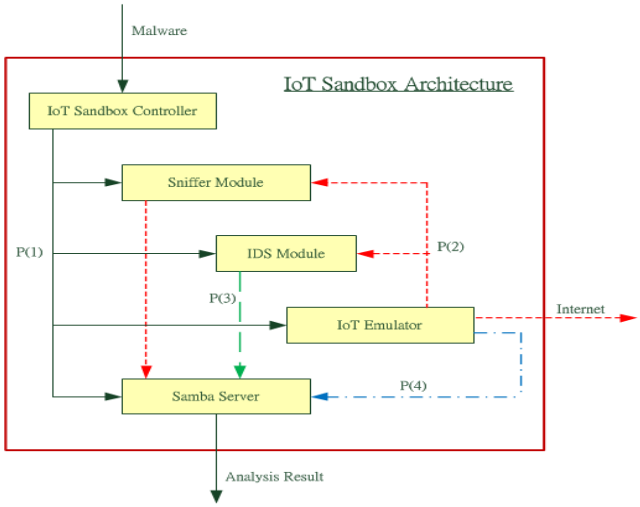

Figure 1. IoT Sandbox Architecture

3.1 Element

There are five elements in this IoT sandbox, IoT Sandbox Controller, Sniffer module, IDS module, IoT Emulator and Samba server.

IoT Sandbox Controller: it is used to start up each module and to copy a malware to the Samba server. After five minutes, it will shut down each module sequentially.

Sniffer module and IDS module: it is used to record and analyze packets. In this paper, we used two Snort [11] process. One Snort process is sniffer mode and the other is IDS (Intrusion Detection System) [12] mode.

Samba Server, it is used to connect IoT emulator and other modules: IoT Emulator can find a malware on it. And also could save malware behavior on it.

IoT Emulator: our IoT Emulator is based on Qeum [13]. It is a famous virtual machine and can support 26 CPU architectures. It can also support many kind of OS. In order to run IoT emulator smoothly. We need some specific setting in image. For example, how to mount samba server, copy malwares, track malware behavior automated? In our design, we chose wheezy standard image as OS for system stable reason, except sparc and sh4. Because we can't find wheezy standard in official website..

3.2 Process

There are four process, they are Initialize Process (P1), Network Base Analysis Process (P2), Exploit & Spread Analysis Process (P3) and Host Base Analysis Process (P4). The process are described in Figure 1.

Initialize Process (P1): IoT Sandbox Controller will start each module, and copy a malware to Samba Server. According to that, IoT Emulator can get a malware from Samba server. Then, can analyze it and save malware behaviors in Samba server.

Network Base Analysis Process (P2): sniffer module will sniff packets and record network behavior. It used to observe communication which between malware and C&C (command and control) [14] server. It also is a necessary information for IDS rule

Exploit & Spread Analysis Process (P3): IDS module use some exploit signatures to detect spread method and exploit. This information is useful to generate IDS rule.

Host Base Analysis Process (P4): IoT Emulator will get a malware from Samba server and rut it with Strace [15] command. And also save malware behaviors on Samba server. Strace can traces the system calls along with its arguments and return values. Through it, can record the malware behavior such as which files have been access, read or write by malware.

Researchers could put IoT malwares to IoT sandbox. After all of process, we could have three kind of analysis result. First one is network communication packets between malware and C&C server. Researchers can analyze it and except to useful information. Second one is IDS detection rule, which has some record about exploit and spread method. Through it, researchers could find out prevent method. Al last is malware behaviors, which has access, read, write, and etc. information in it. Then researchers could know, what is IoT malware behavior? Which IP address is C&C server? What command execute by malware?

4. ANALYSIS RESULT

As we know, IoT threat is getting important. Is it a new attack? I do not thinks so. According to our Honeynet data, we are under attack since 2013 with high frequency more than 83,318 times. We extracted 217 malwares with different MD5. In order to understanding malware behavior. We sand malware Zollard to IoT Sandbox. After analysis then we know, it spread malware with two method: RCE (Remote Code Execution) and dictionary attack.

4.1 RCE (Remote code execution) Attack

In this method, it use four different exploit such CVE-2012-1823 [16], CVE-2012-2311 [17], CVE-2012-2335 [18] and CVE-2012-2336 [19]. This kind of exploit which allows remote attackers to bypass a protection mechanism in PHP 5.3.12 and 5.4.2 and execute arbitrary code by leveraging improper interaction. The exploit code could find on Figure 2. Researcher can found information from IDS module.

Figure 2. One of spread exploit - CVE-2012-1823

As you can see this exploit will turn on *allow_url_include* and *suhosin.simulation*. Then turn off *safe_mode*. According to that, malware can include *myshellexec* function in POST. After include, it could execute command. In *myshellexec* function, we could fond exec. It used to execute an external program. In other words, malware can execute system command base on this function.

In execute command phase, malware will delete old version malware (*rm -rf /tmp/armeabi*), download six malwares with different CPU architecture in tmp folder (*wget -P /tmp http://X.X.55.85:58455/armeabi*), modify malware execute permissions (*chmod +x /tmp/armeabi*) and run malware (*myshellexec("/tmp/armeabi;/tmp/arm;/tmp/ppc;/tmp/mips;/tmp/mipsel;/tmp/x86 ")*;). Why malware did that? The possible reason is, it don't know what it compromise. So, it download six malware and run them at the same time. According to our research, sig and node is a kind of configuration file. Malware will check file exit or not before it spread. Then can found those information on Figure 3.

Figure 3. Myshellexec function and execute command phase.

4.2 Dictionary attack

The malware Zollard launch dictionary attack base on telnet service. If it success then will login to devices and used command *echo –n –e* spread malware. According to packets from sinffer module and our observed, it used 14 username and password. Some of password are DVR, Router and IP camera default password with Bold. Others are weak password. Table 1 show 14 username and password.

Table 1. Username and password list

Username	Password
root	[blank], **dreambox**, root, **vizxv**, admin, **stemroot**
sysadmin	**Superuser**
admin	[blank], admin, 1234, 12345, 1111,**smcadmin**
mysql	123456

4.3 Malware behavior

Those 217 malwares are ELF (Executable and Linkable Format) format. So we need to use IoT sandbox to analyze it with Debian [20] OS. Researchers can found malware behavior on Samba server. In Zollard case, if it spread success. Then will use iptables [21] to drop packets which destination port 23 and port 32764. (*iptables -D INPUT -p tcp --dport 23 -j DROP*. *iptables -D INPUT -p tcp --dport 32764 -j DROP*). The possible reason is, it did not want other people to login telnet server. Why it drop packets which destination port 32764. The possible reason is, port

32764 is a backdoor in many kind of router [22]. Zollard also listen port 58455, it used to received bot's information and spread malware.

According to our malwares behavior data from strace, then we found Zollard will scan telnet service and try to login server. And also try to link other version malware or telnet scan result in */var/run*. Show on Figure 4.

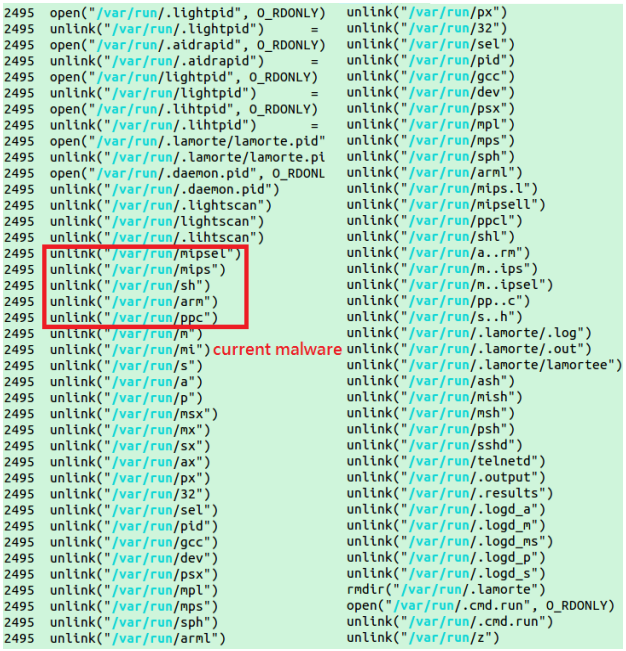

Figure 4. Zollard link *"/var/run/"* folder

5. VICTIMIZATION SCOPE

In this case, researchers used IoT sandbox to analyze Zollard malwares. After that to generate snort rule to detect this kind of malwares. And cooperate with G-ISAC (Government Information Sharing and Analysis Center) [23] member to deploy it. Since April 4th to 19th in 2016. We found 71,148 IP have been compromise. In order to grasp victimization scope, we try to connect bake to those devices. And to analyze banner. There are 5,599 IP have banner information, after analyze result as below. From Table 2. , DVR, Web Camera and Router take Top 3 place in our data. And found some unusual devices such as WiFi Disk, NAS (Network Attached Storage) [24], IP Phone and Printer. And found some of victims has diffusing ability, we call Spread Node. You might have question, the same malware why that have different behavior. The possible reason is, the victims don't have public IP address but it has been compromised. Why malwares can do that, user might use Port Forwarding on DSL router.

Table 2. IoT victims list

Devices Type	Number of Victim	Number of Spread Node
DVR (Digital Video Recorder)	4,599	1,289
Web Camera	895	345
Router	629	197
Wi-Fi Disk	46	24
Set-top box	40	5
Bandwidth Speed Test Server	6	0
Devices Type	**Number of Victim**	**Number of Spread Node**
NAS (Network Attached Storage)	5	0
Firewall	3	0
IP Phone	2	0
Printer	3	0
IP Camera	1	0
NVR (Network Video Recorder)	1	0
VoIP Phone Systems	1	0
LAMP Stack Server	1	0
Unknow Devices	3,617	1,088

Besides IoT, Zollard also compromised ICS (Industrial Control Systems) [25] devices. Such as Heat pump, ICS Data Acquisition Server. Why it can do? The possible reason is, it can launch RCE and dictionary attack. Even Zollard don't have ICS devices default password.

From Table 3. As we can see, the victims include Heat pump, Multichannel Energy Meter, Network Socket and etc.. Some of victim are crossing Internet and LAN (Local Area Network). Few of them are Spread Node. It means, it has ability to attack other ICS devices in LAN environment.

Table 3. ICS victims list

Devices Type	Number of Victim	Number of Spread Node
Heat pump	23	4
Multichannel Energy Meter	2	0
Network Socket	1	1
Telecommunications equipment	1	1
WIMAX CPE device	1	0
UTM (Unified Threat Management)	1	0
ICS Data Acquisition Server	1	1

6. CONCLUSION & FUTURE WORK

According to above-mentioned, IoT Sandbox could analyze IoT malwares, collect network packets, identify spread method and record malwares behavior. It is already used in real analysis case. The analysis result is significant achievements. After analysis, how to generate useful snort rule is also important.

In future, we will keep research this kind of threat. To make sure IoT sandbox would be getting better and better. Then most of IoT threat spared via Ethernet. So, if we grasp signatures, it possible to prevent it via NIDS (Network Intrusion Detection System) [26]. How to design IoT - NIDS base on anomaly detection technology? That is our future work. In this paper, we connect bake to victim, used to collect devices banner. It also need to enhance the

classification efficiency. So need to design classification algorithm on it.

Finally, hope through IoT sandbox to shorten time to analyze IoT malwares. And generate IDS rule as soon as possible. Then could prevent IoT threat and enhance IoT security base on IoT sandbox technology.

7. ACKNOWLEDGMENTS

Thanks for NCCST (National Center for Cyber Security Technology) [27] and TWNCERT (Taiwan National Computer Emergency Response Team) [28] to provide Honeynet data, malwares and G-ISAC platform. This research was supported by the Ministry of Science of Technology, Taiwan, R.O.C., under Grant MOST 105-2221-E-004-001-MY3.

8. REFERENCES

[1] International Telecommunication Union, ITU Internet Reports 2005: The Internet of Things (2005)

[2] How Big The Internet Of Things Could Become, http://readwrite.com/2013/09/30/how-big-the-internet-of-things-could-become#feed=%2Finfrastructure&awesm=~opF6zyqLuIAM4L

[3] Shodan, https://www.shodan.io/

[4] Shekyan and Harutyunyan hack IOT devices in 2012 hack in box conference, http://www.forbes.com/sites/kashmirhill/2014/05/27/article-may-scare-you-away-from-internet-of-things/#3219002a23dd

[5] Hackers Use Refrigerator, Other Devices to Send 750,000 Spam Emails, http://www.dailytech.com/Hackers+Use+Refrigerator+Oter+Devices+to+Send+750000+Spam+Emails+/article34161.htm

[6] Pavel ˇCeleda, Radek Krejˇc´I, Jan Vykopal and Martin Draˇsar, "*Embedded Malware – An Analysis of the Chuck Norris Botnet*", Computer Network Defense (EC2ND), 2010 European Conference, pp 3-10

[7] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama and Christian Rossow, "*IoTPOT: Analysing the Rise of IoT Compromises" 2015*

[8] Asmitha K A and Vinod P "*A Machine Learn ing Approach for Linux Malware Detection*" Issues and Challenges in Intelligent Computing Techniques (ICICT) 2014, pp 825-830

[9] Vmware, http://www.vmware.com/

[10] Ubuntu, http://www.ubuntu.com/

[11] Snort, https://www.snort.org/

[12] Intrusion Detection System, https://en.wikipedia.org/wiki/Intrusion_detection_system

[13] Qeum, http://wiki.qemu.org/Main_Page

[14] C&C Server, http://whatis.techtarget.com/definition/command-and-control-server-CC-server

[15] Strace tool, http://sourceforge.net/projects/strace/

[16] CVE-2012-1823, https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2012-1823

[17] CVE-2012-2311, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2311

[18] CVE-2012-2335, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-233

[19] CVE-2012-2336, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2336

[20] Debian, https://www.debian.org/

[21] Iptables, https://en.wikipedia.org/wiki/Iptables

[22] Router backdoor, https://wikidevi.com/wiki/TCP-32764

[23] G-ISAC (Government Information Sharing and Analysis Center), http://www.nccst.nat.gov.tw/GISAC?lang=en

[24] Network Attached Storage, https://en.wikipedia.org/wiki/Network-attached_storage

[25] Industrial Control Systems, https://en.wikipedia.org/wiki/Industrial_control_system

[26] Network Intrusion Detection System, https://www.sans.org/security-resources/idfaq/what-is-intrusion-detection/1/1

[27] NCCST (National Center for Cyber Security Technology), http://www.nccst.nat.gov.tw/

[28] TWNCERT (Taiwan National Computer Emergency Response Team), http://www.twncert.org.tw/

9. APPENDIX

Note that all samples in the table and corresponding honeypot traffic.

Table 4. IoT malware samples

Filename	MD5	Filename	MD5
x86_21	a481164f8487077784b30104cea22a6f	x86_20	8d6b6d575e7d40f772c1155f8c088f80
x86_22	e1bee1c91a64caaae4c476aada8aed9d	x86_6	994e3a65c22a80b56a716b525471e34f
x86_49	fbaf7e745fce81dd20c2472308c20a8a	arm7	ec41fafc386a4afd10b8cc5df0a4813a
x86_38	fd06f2d3f27d48561f56b7e7be2d7378	arm9	7fa81602312d2118dda03851651a3874
x86_3	cf5a8c168b30bcb7860c710bf180749a	arm29	7c2c53922efa8cd3eee42eea51bfd622
x86_66	640e86ce5985bd37e40ced50c91f442b	arm34	a465d5f7b9d898eb4bb7f16087c8cc2e
x86_57	c99bc6bba58a58e349ab4800f2be3e4d	arm24	f85a0413b96a896c35aa0963065d6a08
x86_9	a4fba2b09166475b3b5c29b49f15acdd	arm15	eb2c7c74b21291504aa3b5d6cb6666a9
x86_12	b003af27251d78ca340398929e094dad	arm	e6f4413cfc53559ed35e9fd04adfddb9
x86_43	6195f7588fe4207f148b67d34bff1df6	arm22	146e936d1616ff12829430d168e96992
x86_4	d8623f8d7faba750dd4696f726fd38c4	arm43	3ef232d535f00892710bf65c907611ab
x86_19	a9b889508485419124 97b8d03de6ca4c	arm5	ce0492977ae4ef0b137435a3c3b7d3f2
x86_5	0e60bac86972e2cfc328332ce37a59af	arm17	b1ff89f130822aa40be64b555af31e22
x86_10	0ee0361fefb5bb689299de60f8ba0187	arm26	bfe02a05459911afeb41cabca3bf6ad4
x86_63	7a6dd7d5b39663d3dcae85004646d827	arm23	2b6aefc9f341aeb344f5fde7846e3255
x86_2	4a72434154542cb879d62cc7688d5629	arm46	0e9cc8a3db209e964ff22d87223191d5
x86_46	0cbd2b277a9a5df5b8a73ad5b9dcacf3	arm25	2f661d7445df657ab77b6502c73cbe90
x86_40	5ea14a8607a07719146a2668f1417eca	arm16	678d75bf183a7ef8e0f01659f1e6ddf5
x86_59	1709e2d7087e2902be990daa55df0c1e	arm3	d47c250f4ef16b408911d957842a9d04
x86_60	9cd1183bb63d624f0173ea8f5806492a	arm31	2d21644350f18c98346c7ec940b2a103
x86_36	2229824a85a944f2b74a1e71ff118dae	arm4	8e4e18e480c4e1e4051ef2463e26c0f0
x86_28	9301c588ec83848375308d8df8c6d686	arm6	34430c246b8740ffa208b38a0077160d
x86_39	e3dd272dfc01708bb6b879093715bf9d	arm38	b36f762d262c733b26feea032e67717c
x86_7	d686ec4b59b7f25f84671b2053097479	arm12	0535d8d199eeaa5acc55873f7ee7d74f
x86_23	1315d76421bd5531d4b8bcd823cbc939	arm19	b20dd5176ec5f72356c52358c11a4ed4
x86_31	0ba6516c763628fa502d26dbf38ae036	arm42	9fa076b65765133f19f214a4b20fee08
x86_24	cecc05c476ffcb383d75c4d620239f35	arm44	c592de398582228f4f25a86b5a4b0124
x86_45	bd8637883d6cac287dcc04a4dc032d8d	arm40	2553c92a069db26e11a0da58dfe75cea
x86_67	5ae4348ca98e380909f3185fc989be67	arm21	a12f3047d7671ca2e119c3ee6c685370
x86_27	78b996cafcff49be31d199c59316a121	arm10	fc6edc0ef924d7c46cff2c6ed51079ff
x86_64	dac21f56af6f3afd32e9e1434420d0f6	arm2	1bd79116bb515e7d523f26aafb065963
x86_68	2fa8967a7bde359af2de4ef51c73dd50	arm8	9330d17f8a4c582e2522e564c679b69c
x86_32	fd3ee359fb03a8304f1d548032591fa8	arm13	d890b6e63e556b1d12de9014824d009b
x86_50	c6311e9c014d91190d46cbbbf9fcac59	arm33	83f18d7877ca4659e1f2bc24841368f5

Filename	MD5	Filename	MD5
x86_51	f8e35732287834d079ac6072862f5187	arm32	de2065967ee36139dcff2ce720887c76
arm20	7c50739d369962d3abb577495001d949	mips4	7ef001c2c177ac0e7bf0093177696f62
arm27	c411bfb3a920ebc4c3b710fc9d6263ad	mips60	71a62d10afcc2c06e4458b7085766b2b
ppc4	63b1536ce365da8f2c5a1c4c79d5c60b	mips79	1f88b5144a43f4f95512b7d4d46949e6
ppc56	7ebffd2f94fbede8208506da87042141	mips78	8d73dede6516a236917a48eaeb58490b
ppc55	fd119e53805368320ec641266c20706b	mips29	75987699c6303fb94ab0d50fec9f891a
ppc6	8586e4d7756d6c6ea725d36ace393ba5	mips21	ec2b2256e81e0a9a6776578aa5e600c4
ppc65	07b9e2ed060dca8158c909ba0e93fa54	mips25	fc907f3028c9ce81872b2c532f51eff8
ppc12	0bfa4c20d1aaa467597ec040f0c48b2c	mips40	03a55348b1e815a35136f219ca08a7b3
ppc79	7b4da9fb13a07d8ebf7facd1b6feaeda	mips59	55d459ab7cc6df0a70fedb4f43e2977a
ppc59	1d776d11b2b95f82e44e910ebb617ecc	mips30	060b06ea9ea64dac434ec6f52d90b334
ppc16	fba9de721103ed99ede31f4174637281	mips57	e0ccdd62353d608deee1a9f23c8207e4
ppc5	9ce33117ff2f60bb72056def9738192b	mips17	99a549f7433f49bd151569ecc0828438
ppc21	40349d6cf51b260f33a9a46de2402967	mips39	0ac70024d8f00db6287234a636860acf
ppc66	7d1658fb7424fae180404e4421f6ecad	mips33	5f998c85c93177a3d18d18336756b487
ppc58	9ebd5d9f4ff389742b75198b23ba6222	mips82	a54736e6ee4dd207187d22feff69e482
ppc13	fc71e8e02700bade55e0733ff6ce8a2d	mips85	07d8b3a9d1644d36a6072df7c28dde0d
ppc10	2f46eeeb1a86b38cadc831b1d04ab009	mips62	e0bd8a887d66d32be4482d72c5b57cda
ppc69	cbedcf6af50b4d8d7f2461f0219cb3ee	mips75	bc7230fefef2f6ac7d7da2f33c5dfe82
ppc70	f8e9950be70fa8d83a6609d4c378a9e8	mips2	78f5eb3423b14c8a9757562c3e7446ab
ppc78	4130b4bdad1ec0c8791f3eae441d53bf	mips20	27fb647316f7041aaa94de1ad5192f29
ppc77	24f99d74c02b9ce8682f70528fc6b6f0	mips19	06d57124f217eb9ae3f621dcc1598cd3
ppc7	01ad371d727a5aede23a6afd803f5abe	mips73	f2c910bdc0ab73c9c013b2c0b2a8b1eb
ppc8	4538d1320a893fd93b558823d8f24e67	mips34	7a1bc34bea8011e91ebece2f2565aa03
ppc72	f90b12706f9379698c84a1662e802e46	mips49	54142a378a5bbe68c1fb3f75d2cb7cbd
ppc73	c781be6796d9c3fc541715264003690a	mips71	ad4201d1e12b2f42a6b79d61adee2c29
ppc23	a70b1efb9a7cd5f467055efe096c671a	mips47	b1c9be5245905af3030004a76f44f4b3
ppc57	8314bf19e7f92f5ea3003db02636d45e	mips23	56b19d5b17f1505a3f3f5770ad006554
ppc	3daf8916ff721643be7c2a1a8a01bf1d	mips26	7bcde1f5eaf2ca43877388e735e85909
ppc9	697230f1a96e20f994cf29c80d8f3df6	mips58	bc39775a335bf527eca71664ec1ebee7
ppc24	ac78fd5f621c3f380c2e9d5112327e42	mips61	43d400704653c83006638290d8787141
ppc67	1c68919a9000251c87f2ae9063712b10	mips12	e3da337fdc7bc1b67dc94a93af4728aa
ppc76	26a37f8b5d4baedb261e0f46d35e29b8	mips42	1d69f8059ef3efe651deb41ecdd21ff3
ppc75	5a11bf8613ebc1c47c853ca446a0f3a3	mips68	dae27e785102d714114f400978808f25
ppc62	3e0a0fd8913d6fba12f56963106c644e	mips87	de6efa295ff2695aaf60035e49fad435
ppc64	3dc0940870f19ffdee73f2ff2a961a9d	mips74	c47dda6e0040c478ab7e051aa1552f7c
ppc11	4179a8d0b2f58443ac3789259bb58f74	mips70	fc0c0594d0f8a24eac4d6c9e1f2adf50
ppc17	585a6670ad7aa8e7721ea6e00c025e55	mips5	e4eb21f5a5464c6ab7f2ae6ae91597ab

Filename	MD5	Filename	MD5
ppc60	85578174d4ce22d137c59801ae8ebe39	armeabi_076	8e24973b56c219d16ed822a565032ccf
mips	0920d0ae9a2e026c9b6f6b7d8cef23ba	mips72	c865bcaa9b5c1e4a16fb397e2eefe1a1
mips66	a4adacec3cc64e903d48c7606776bdfd	mipsel_001	76172fd6cedc27dbbd15c989683cc051
mips43	711ef0ca73834c8d59ac2e495151b996	mipsel_006	ed6e70bbe9aa8e7b26cdd7c5611e9269
mips81	c5ec67ddb26bbf5ffe150010a5dd8e70	mipsel_007	b5904c225b22a1d67f4007268f3b90e6
mips24	88c389c1fef7784e54f031eba0269c38	mipsel_008	f2b870e2b59b1c5a90f39186646650df
mips10	be68d92117335424c39cad8b9ddfa1c7	mipsel_013	38047cb8cd4f7d46a9ab16abfcb369e0
armeabi_001	2124502f4207d8a5a88b4669df7c9686	mipsel_017	77e8cac1ecfc5a682c5649040760875d
armeabi_002	63d2ce014157a28d9b09c21283952511	mipsel_018	d13ec27f49ec68c1219b8200b4854ae0
armeabi_005	4ca59b25bb629b4561e9dea51ace3d16	mipsel_023	09fb0cde95a72f434e3359a8737dd5c8
armeabi_006	dfeb77cb0ba28ac3ba4be55d7bc91fad	mipsel_024	3047f954888d8a479b12dfaeb4edce8d
armeabi_008	e2310b0e6910d666c4d0d3d26988740c	mipsel_025	8259871c0df304503256ba799f7c30f5
armeabi_014	d31bfb47545179ee6b63bad457c9d0f6	mipsel_026	fdf36ca3aedb5e4b083d6acd1998f482
armeabi_020	80d2cb3fe95930747329f975ad805173	mipsel_029	edc95d2a7c0bad4b278ce3842a7e231c
armeabi_024	372d100d670fa3de77f816f60167ad3c	mipsel_030	4578bcc1e84250b4cb68ef9d4bd43c7c
armeabi_026	d359268a4bac96ed00dcdb271e96baea	mipsel_032	18928cc1f6684e975be790e5cafa3c37
armeabi_027	5d6e86a128d009dfb5da475ae7cf9c5c	mipsel_034	51e5648bee24384d46439887702b103b
armeabi_029	97c440b69bc65a4995e6119ea9e491db	mipsel_036	7d0fa0197e693e8701137b3183237e64
armeabi_030	b222961045e96c18c34285cf76cbad4d	mipsel_038	cd47ef42ce4618396534935d2fea35b5
armeabi_031	ed80f886119c0ac7e125b5495ac10a04	mipsel_041	245ee077046f8a999499bc5628c948b9
armeabi_032	e0dab5c9730cf66468aacc93940f74c6	mipsel_044	1e2d0c01e161ce8863e8f564ef349f84
armeabi_033	7ba9d99d12511017ac0189dccae77ebb	mipsel_047	41d679d09d279cc100d8dbce560e1b3b
armeabi_034	c94ce5ebf3563e9834b2af00ff945c9f	mipsel_048	e4495654fac3086cf669418940ec3731
armeabi_040	0e1ed98ae02d7a8aef5be50efb0d9eaa	mipsel_050	39fa1ce28f71f739b8c578319a5a80f0
armeabi_042	1eca2ecba9f206aaf59ef21e504e6a2f	mipsel_051	9811a2780687bbbecb7c65564fee6f5a
armeabi_045	8f8ddea8754181980823270da778c36b	mipsel_052	4089b30ddc8c609fc80aca793cf57ac5
armeabi_047	082ce4186c898971e668264c99e73f5b	mipsel_053	cd9a859ca3ba6d5af408abe9fd62b96e
armeabi_056	4388ef311dcd2bd1994035f5a79bc56a	mipsel_055	dd94f71f0357d6b31c23c6c6c3a9ce33
armeabi_058	6efdd15746d0161cc2c479756af5daad	mipsel_056	303a40bb761f0cb113a9df4d9f15fe15
armeabi_059	cf77ada5f9109f27acee33658f03a652	mipsel_057	273de72d63f51d055c82052df6aeabbc
armeabi_062	e191fc8dd5ff020fdab0de9a3f302800	mipsel_058	a55630a3cbf6bbdf5e47bca76574a3e2
armeabi_064	1e42e47998f27bef131d013cec5ad0fe	mipsel_059	c4a6bb610e6a76b3ba5b2cea2c2dbb63
armeabi_065	e251ed13bb18c9580acd4703e5a1ca0f	mipsel_060	ca0bbbc91f45745bafe55f60428f8466
armeabi_066	6d9bcdc9fc91abbc7122c62b8f62f5e8	mipsel_061	a4064990a60ffccb7428b2d10afd68a2
armeabi_067	bd8ce49dac1a2be16da4b6670c561ade	mipsel_062	1331b141c70223158b51b92f1a639b83
armeabi_069	4fb5e4fcefb286b05c503ec980f39fe2	mipsel_065	b58c9e0df2c23d77d1ce9c1df87b1018
armeabi_075	2b360b5eac3bf6d2e5f5d6d4f6784f4e		