THE IMPACT OF TORII ON ARM ARCHITECTURE

Milorad Murić[1]; Žarko Bogićević[2];

[1] High school of vocational studies, Užice, Serbia, muricmilorad@gmail.com

[2] Technical school, Užice, Serbia, zarko1993@hotmail.com

***Abstract*:** *Already at the end of 2016, there are suspicions of the active operation of various botnet malware, of which the most famous are Mirai and Qbot. At a time when millions of different IoT devices are in use, the operation of such malware can be extremely dangerous. In the light of such developments, a new malware called Torii, which was detected at the end of September 2018, emerged. This brand new strain of malware strives to remain deeply hidden, acting persistently from the background and still does not take steps that are characteristic of most botnets . What's clear is that this malware can attack a wide range of different hardware-targeted architectures, such as MIPS, ARM, x86, x64, PowerPC, SuperH and others. The paper will explain how this malware works and the consequences it can cause, as well as the way we can protect itself and devices.*

Keywords : *Malware, Botnet,Torii, ARM*

1. INTRODUCTION

The number of IoT (Internet of Things)[1] devices as of Q1 2018 is estimated around 7 billion devices. Companies that provide IOT software and cloud services are exceeding revenue expectations. Amazon AWS and Microsoft Azure grew 93% with the IOT portion contributing significantly. The number estimated does not include smartphones, tablets, laptops etc.

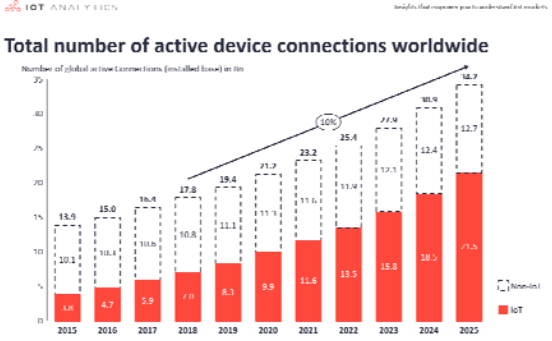

Figure 1: Number of IoT devices and growth estimate

The sudden rise in IoT is driven by smart home devices and industry purpose devices. In terms of connectivity the devices vary, but most of them are WIFI based with a very small percentage that are wired or cellular.

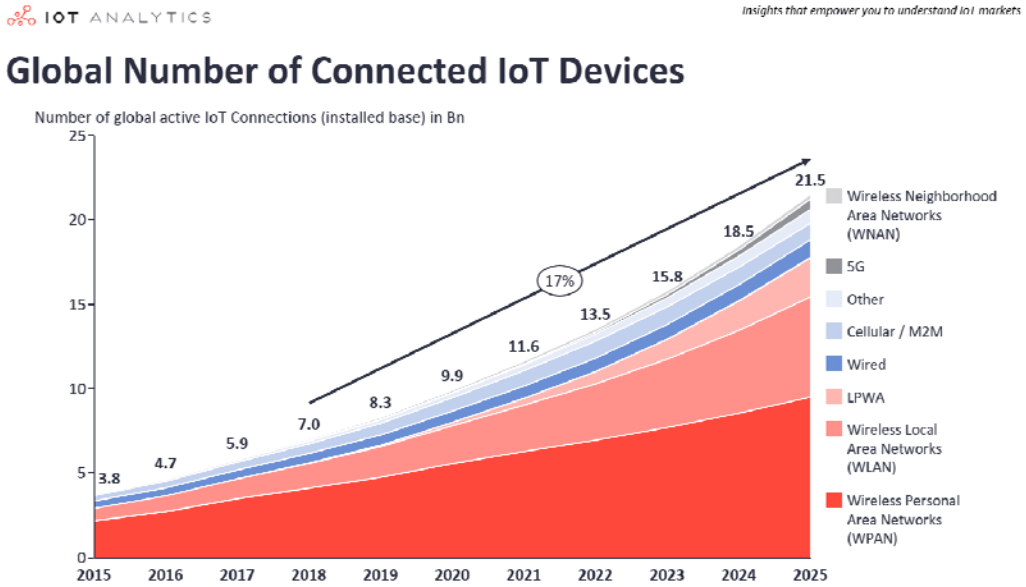

Figure 2: Most common connection types for IoT

A very small portion of IoT devices receives updates from their manufacturers. These updates come in a form of software or firmware. Due to the fact that the most devices never get updated their software is exposed to all kind of exploits. Torii attacks devices based on x86, x64, PowerPC, MIPS, ARM and several other architectures. Most of Torii's functions are unknown for now end due to this fact it is very difficult to protect IoT devices from this malware.
The global smartphone market is constantly growing and new manufacturers are gaining market share. Most of these manufacturers are from the Asian market, as seen in the table below.

Table 1. Global smartphone shipments in Q3 2017/2018[2]

Company	3Q18 Shipment Volumes	3Q18 Market Share	3Q17 Shipment Volumes	3Q17 Market Share	3Q18/3Q17 Change
Samsung	72.2	20.3%	83.3	22.1%	-13.4%
Huawei	52.0	14.6%	39.1	10.4%	32.9%
Apple	46.9	13.2%	46.7	12.4%	0.5%
Xiaomi	34.3	9.7%	28.3	7.5%	21.2%
OPPO	29.9	8.4%	30.6	8.1%	-2.1%
Others	119.9	33.8%	149.8	39.6%	-19.9%
Total	**355.2**	**100.0%**	**377.8**	**100.0%**	**-6.0%**

2. Meaning And Anatomy Of The Torii

Given this extremely basic introduction to the Shinto religion, we can finally explore the meaning of the Torii. The Torii is, in fact, a gateway, that signals the transition from the profane to the sacred, as it is usually located at the entrance to Shinto shrines, though it isn't rare to find them even at the entrance of Buddhist temples. As a matter of fact, the first documentation of the Torii dates back to the mid-Heian period, in 922, when Buddhism had already been introduced in Japan. Because of this, and the existence of similar structures in the rest of Asia, typically associated with Buddhist sites, it is quite hard to find a clear-cut origin of the Torii, there are many theories, none of which seems to satisfy the question of its origin. However, it is a matter of fact that today, the Torii, though present in Buddhist sites as well, is more closely associated to Shinto, for instance, the Shinto shrines are signaled on maps with Torii icons.
Now that we have a good grasp of why the Torii is found in some places rather than other, let's see what makes up a Torii. The structure of these gates is made up of several elements, depending on its style (shinmei or myojin). Certainly, the most important elements of the Torii are the pillars (柱 hashira), the kasagi (笠木), the lintel placed on the two pillars, and the nuki (貫) a tie-beam that keeps the structure together. These are the essential elements, present in all the known variants of the Torii

styles, except for the Shime Torii, which is only built by two pillars and a shimenawa (rice cord) tied to the extremities of the pillars. Other elements usually present in the Torii are: the shimaki (島木), a second horizontal lintel placed under the kasagi, the daiwa (大輪), a decorative ring place around the top of the pillars, the kusabi (楔), two mostly ornamentl wedges holding the nuki in place and the gakuzuka (額束) a supporting strut connecting the nuki to the kasagi or shimaki. The gakuzuka can sometimes be covered by a tablet with the name of the shrine[3].

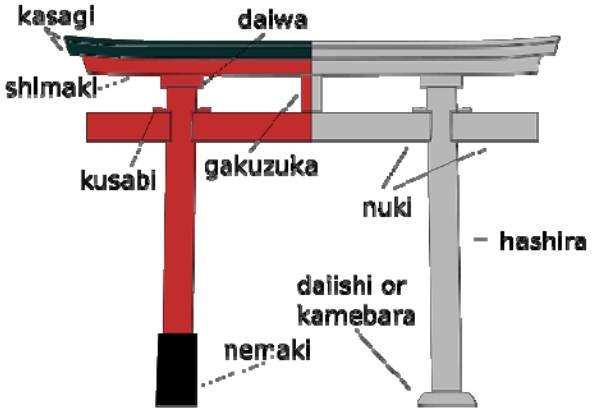

Figure 3: Antomy of Torii

3. Overview Of The Arm Architecture

The valid architecture variants are as follows (variant in brackets for legacy reasons only):ARMv4, ARMv4T, ARMv5T, (ARMv5TExP), ARMv5TE, ARMv5TEJ, and ARMv6.The following architecture variants are now OBSOLETE: ARMv1, ARMv2, ARMv2a, ARMv3, ARMv3G, ARMv3M, ARMv4xM, ARMv4TxM, ARMv5,ARMv5xM, and ARMv5TxM.[4].
The Thumb instruction set is a re-encoded subset of the ARM instruction set. Thumb instructions execute in their own processor state, with the architecture definin the mechanisms required to transition between ARM and Thumb states. The key difference is that Thumb instructions are half the size of ARM instructions (16 bits compared with 32 bits). Greater code density can usually be achieved by using the Thumb instruction set in preference to the ARM instruction set. However, the Thumb instruction set does have some limitations:

- Thumb code usually uses more instructions for a given task, making ARM code best for maximizing performance of time-critical code.
- ARM state and some associated ARM instructions are required for exception handling.

The Thumb instruction set is always used in conjunction with a version of the ARM instruction set.

New features in Version 5T

This version extended architecture version 4T as follows:

- Improved efficiency of ARM/Thumb interworking;
- Count leading zeros (CLZ, ARM only) and software breakpoint (BKPT, ARM and Thumb) instructions added;
- Additional options for coprocessor designers (coprocessor support is ARM only);
- Tighter definition of flag setting on multiplies (ARM and Thumb);
- Introduction of the E variant, adding ARM instructions which enhance performance of an ARM processor on typical digital signal processing (DSP) algorithms;
- Introduction of the J variant, adding the BXJ instruction and the other provisions required to support the Jazelle® architecture extension.

New features in Version 6

The following ARM instructions are added:

- CPS, SRS and RFE instructions for improved exception handling;
- REV, REV16 and REVSH byte reversal instructions;
- SETEND for a revised endian (memory) model
- LDREX and STREX exclusive access instructions;
- SXTB, SXTH, UXTB, UXTH byte/halfword extend instructions;
- A set of Single Instruction Multiple Data (SIM) media instructions;
- Additional forms of multiply instructions with accumulation into a 64-bit result.

The following Thumb instructions are added - CPS, CPY (a form of MOV), REV, REV16, REVSH, SETEND, SXTB, SXTH, UXTB, UXTH.

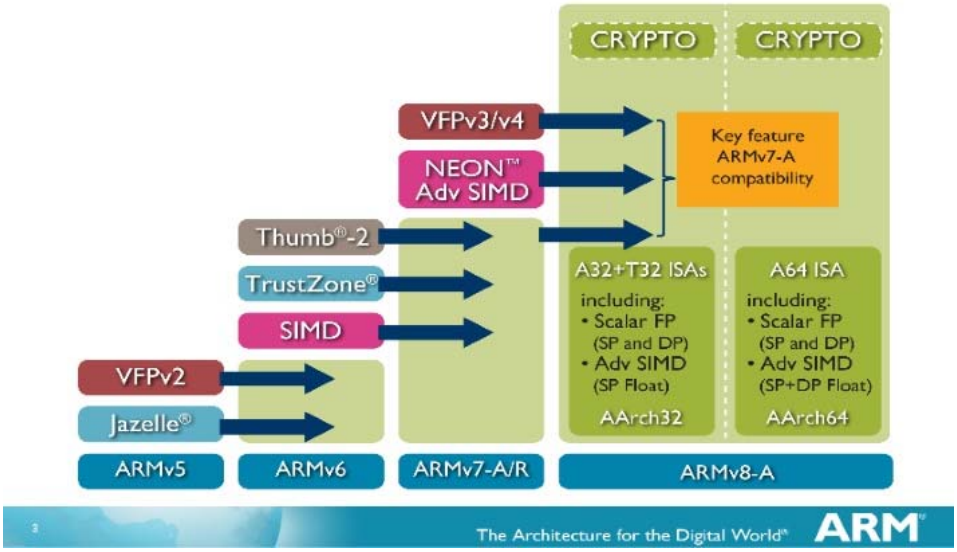

Figure 4: ARM architecture[5]

4. Introduction To The Arm Cortex A53

Cortex-A53 (formerly Apollo) is an ultra-high efficiency microarchitecture designed by ARM Holdings as a successor to the Cortex-A7 microarchitecture, which previously dominated the market in terms of devices using it. The Cortex-A53 implements the ARMv8 ISA, which is typically found in entry-level smartphones, tablets and embedded devices. It should be also noted that this architecture can be found in mid to high end devices as a part of the big.LITTLE core architecture which uses these cores in combination with A57 and A72 cores in order to combine high performance and greater energy efficiency.
It should be noted that ARM designs microarchitectures and licenses it to other manufacturers and semiconductor companies so that they may implement it in their own chips. That is why the same core could have slight modifications to soma parts, for example cache, in different manufacturer devices, for example Samsung's Exynos CPU. ARM is an interesting company in terms of CPU and ISA development. They license their CPU designs and Instruction Set Architectures to other companies. In financial terms this allows profit from their CPU designs and license royalties. Some of their designs are known well ahead of their production for the consumer market. For example, the previously mentioned core designs were ready almost two years before they hit the market for consumers.
The A53 is a part of ARM's ARMv8 processor lineup which is meant to replace the A7 CPU. The A7 CPU was meant for the budget segment and was widely used due to its good performance and excellent power savings while maintaining a low price point. For the A53, ARM decided to focus on further improving the performance, and switching from a x86 architecture to the x64.

Table 2. Comparing ARMv7 and ARMv8 architecture

ARM CPU Core Comparison		
	Cortex – A7	**Cortex – A53**
ARM ISA	**ARM v7 (32 bit)**	**ARM v8 (32 – 64 bit)**
Issue Width	**2 micro - ops**	**2 micro - ops**
Pipeline Lenght	**8**	**8**
Integer Add	**2**	**2**
Integer Mul	**1**	**1**
Load / Store Units	**1**	**1**
Branch Units	**1**	**1**
FP / NEON ALUs	**1 x 64 bit**	**1 x 64 bit**
L1 Cashe	**8B – 64B IS + 8B – 64B DS**	**8B – 64B IS + 8B – 64B DS**
L2 Cashe	**128 KB – 1 MB (Optional)**	**128 KB – 2 MB (Optional)**

In terms of power savings, the A53 has an optional ability to switch between power states which allows for greater power savings. The A53 matches the A9 performance on the same clock speeds, however, the A53 offer much higher clock rates.
Looking at further improvements, ARM has made progress on memory latency as well, compared to the previous architecture.

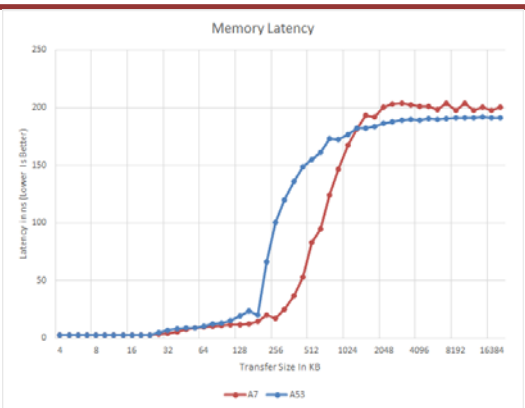

Figure 5: Memory latency between architectures

As well as memory bandwidth.

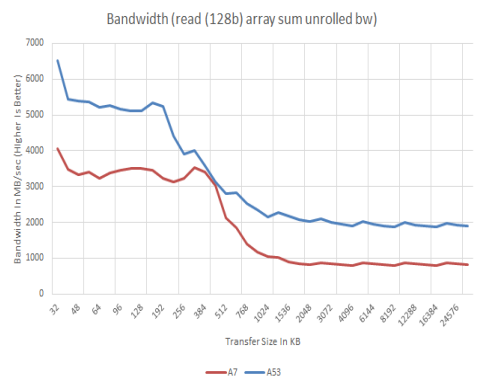

Figure 6: Memory bandwidth between generations

5. Torii's Predecessors

Before Torii was launched there were several of its predecessors. Torii vastly improves on the predecessor's effectiveness and purpose[6].
Persirai, an Internet Of Things (IOT) botnet was discovered some time in 2017. At the time it was targeting over 1000 Internet Protocol (IP) camera models on various Original Equipment Manufacturer (OEM) products. Persirai was developed on Mirai which was used to cause a lot of incidents in 2016 that included Distributed Denial of Service (DDoS) attacks that compromised IOT devices such as Digital Video Recorders (DVRs), CCTV cameras and other similar devices. Around 120 000 IP cameras were vulnerable to Persirai. While many of these device owners were unaware that their devices were infected.

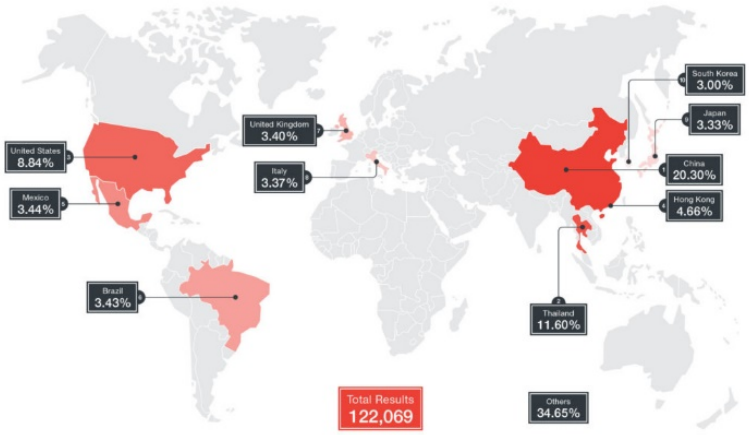

Figure 7: The number of vaulnerable IP cameras as of April 26, 2017 (derived from Shodan data)

This made it easier for the perpetrators to gain access to IP camera web interface which is, by default, located on TCP Port 81. Mirai, the original virus, took advantage of open Telnet ports by scanning the Internet for unsecure devices with default username and password. Due to this vulnerability, in 2016, most of the U.S. east coast was left without internet access as the virus had accumulated a huge botnet of these devices. Before the culprits were captured, they released the botnet code online, in order to mask their tracks, however, they were arrested and the code was publicly available for anyone to use it or further develop it. Mirai stores itself in the device memory and detects and deletes other possible malware that is on the device. However, rebooting the device will purge the infection, but the devices are usually re-infected swiftly[7].

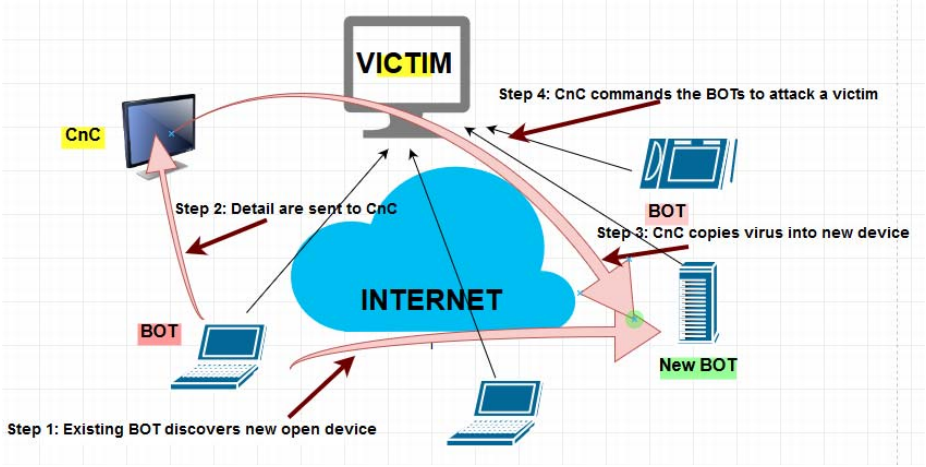

Figure 8: How Mirai works

6. Torii Analysis

Torii is new, more sophisticated strain of malware, and his target is wide range of IoT devices. Torii has spread over unsecured or poorly secured ports for telnet services. Torii search for targeted devices with bad username and password and attempt to execute initial shell script. The script tries to determine the exactly architecture of the targeted device and once when operation is complete followed by an attempt to download the appropriate payload for the devices (binary files in the Executable and Linkable Format - ELF, formerly named Extensible Linking Format) format.
The main functionality of these payloads is attempt to install an inner ELF with the first ELF file. This is the second executable step when malware uses at least six methods to permanently stay on the device and continue the ELF execution. After that, the inner ELF is executed to deliver the second stage payload, a fully-fledged bot capable of executing commands from its master command and control (CnC) server. The command-and-control server with which the observed samples of Torii have been communicating is located in Arizona.
Torii steal data from IoT device and allow attackers to execute code remotely and run any command on the infected devices. This malware is capable of fetching and executing wide range of commands using multiple layers of encryption and allows attackers to execute any code or deliver any payload to an infected device.
So far, Torii is not being used to assemble DDoS botnets or to drop cryptomining tools. With modular, multistage architecture, Torii attacks devices based on x86, x64, PowerPC, MIPS, ARM and several other architectures.

7. Countermeasures

In general purpose, there are steps that are recommended end users should take.
Set up the router and block vulnerable ports especially for telnet:

- 20/21 - File Transfer Protocol (FTP) control is handled on TCP port 21 and its data transfer can use TCP port 20 as well as dynamic ports depending on the specific configuration.
- 22 - Secure Shell (SSH) is the primary method used to manage network devices securely at the command level and it is typically used as a secure alternative to Telnet which does not support secure connections.
- 23 – Telnet - Caution should be used when connecting to a device using Telnet over a public network as the login credentials will be transmitted in the clear.
- 25 - Simple Mail Transfer Protocol (SMTP) is used for two primary functions, it is used to transfer mail (email) from source to destination between mail servers and it is used by end users to send email to a mail system.
- 53 - The Domain Name System (DNS) is used widely on the public internet and on private networks to translate domain names into IP addresses, typically for network routing.

- 67/68 - Dynamic Host Configuration Protocol (DHCP) is used on networks that do not use static IP address assignment (almost all of them).
- 80 - Hypertext Transfer Protocol (HTTP) is the main protocol that is used by web browsers and is thus used by any client that uses files located on these servers.
- 110 - Post Office Protocol (POP) version 3 is one of the two main protocols used to retrieve mail from a server.
- 123 - Network Time Protocol (NTP)is used to synchronize the devices on the Internet.
- 143 - Internet Message Access Protocol (IMAP) is the second of the main protocols used to retrieve mail from a server.
- 161/162 - Simple Network Management Protocol (SNMP) is used by network administrators as a method of network management.
- 443 - Hypertext Transfer Protocol over SSL/TLS (HTTPS) is used in conjunction with HTTP to provide the same services but doing it using a secure connection which is provided by either SSL or TLS.

If suspicion of IOT devices we have to monitor network flow by software or turn on router traffic logging and review those logs. A great software and a recommendation for monitoring network traffic is WireShark. Compared to computer software that logs network traffic, router network logging is guaranteed to log all packets that travel through the router and does not require an active computer that logs, as it has an integrated memory logging function. All the user has to do is download those logs and manually review them for suspicious traffic.

Date	Time	Dir		Remote IP Addr	Remote Name / Message	R Port	Local IP Addr	L Port
/07/26	15:13:46.89	O	tcp	209.123.109.176	i.dslr.net	80	192.168.1.117	55071
/07/26	15:13:46.89	O	tcp	209.123.109.176	i.dslr.net	80	192.168.1.117	55063
/07/26	15:13:46.89	O	tcp	209.123.109.176	i.dslr.net	80	192.168.1.117	55070
/07/26	15:13:45.61	O	tcp	124.169.208.46	124-169-208-46.dyn.iinet.net.au	48057	192.168.1.117	55304
/07/26	15:13:44.93	O	tcp	174.129.203.189	mail.reddit.com	80	192.168.1.117	55302
/07/26	15:13:44.60	M			dnsmasq-dhcp[562]: dhcpack(br0) 192.16			
/07/26	15:13:44.60	M			dnsmasq-dhcp[562]: dhcpinform(br0) 192			
/07/26	15:13:41.10	O	udp	186.45.208.134	186-45-208-134.dynamic.tstt.net.tt	18360	192.168.1.117	29011
/07/26	15:13:40.78	O	tcp	174.129.203.189	mail.reddit.com	80	192.168.1.117	55302
/07/26	15:13:40.78	O	tcp	174.129.203.189	mail.reddit.com	80	192.168.1.117	55294
/07/26	15:13:40.78	O	tcp	68.195.219.34	ool-44c3db22.static.optonline.net	57458	192.168.1.117	55303
/07/26	15:13:35.90	O	tcp	173.231.140.218		80	192.168.1.117	55297
/07/26	15:13:35.43	O	tcp	174.129.203.189	mail.reddit.com	80	192.168.1.117	55294
/07/26	15:13:34.12	O	tcp	199.47.216.173	v-client-2a.sjc.dropbox.com	443	192.168.1.117	55291
/07/26	15:13:34.12	O	tcp	75.101.142.23	ec2-75-101-142-23.compute-1.amazonav	80	192.168.1.117	55290
/07/26	15:13:33.94	O	tcp	66.220.147.14	api-10-04-snc4.facebook.com	443	192.168.1.117	55289
/07/26	15:13:33.94	O	tcp	50.19.118.145	ec2-50-19-118-145.compute-1.amazonav	80	192.168.1.117	55288
15:13	IN:	9 / min		105 / ten min	125 / hr	OUT:	50 / min	356 / ten min	411 / hr

Figure 9 : Router traffic log example

Some manufacturers issue updates even after the devices EoL (End of Life) has been reached, these updates come in a form of a software security update. It is important to periodically check for these updates as well as choosing the right IoT devices with prolonged software support.

If a malware has been found after reviewing the network logs it is highly recommended to reboot the device and check again. As previously described, it might be a malware that does not have persistency and will be cleaned after a device reboot. If the malware persists, restore the device to factory settings and reboot, however, before connecting the device to the Internet again, make sure to change the default username and password, otherwise the device will be vulnerable again.

It is highly recommended to create full restore/recovery points and backups on the computers. System updates and virus definition updates are also highly recommended.

Setting up a firewall on the router itself protects any unwanted traffic from coming through the router, independent of the type of device connected to it. Some devices posses built in firewalls and they may be set up independently of the router firewall.

For ARM specific devices, these extra steps could prove crucial:

Most ARM devices (tablets and smartphones) come with manufacturer supported software for syncing and backup. This software can be used for creating full or partial backup of the system software and user data on the user's devices. It is considered good practice to create scheduled device backups (daily, weekly or monthly).

If the device is Android based, there is to possibility to use software that creates full image backups of the device software and data[8]. Commonly, TWRP (Team Win Recovery Project)[9] is used, but there is other software available as well, depending on the device support list.

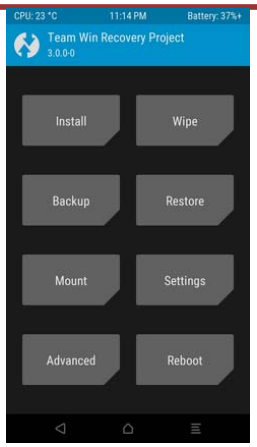

Figure 10 : TWRP software interface

Install mobile version of antivirus software and keep it updated.
Only install software from official and certified app stores (Google Play, Apple Store, Microsoft Store etc.)
Monitoring the data flow on devices (Wi-Fi and DATA) will reveal if there is any extra data being used. Some malware knows how to conceal itself and only use the device when the screen is powered off, which indicates that the user is not using the device. Pay attention to enormous amounts od DATA usage, as it could lead to a very high provider bill.
Regularly update the Android system and security patch levels (independent of the OS version) and others OS based on ARM architecture.

8. CONCLUSION

In a digital world that is constantly growing in terms of IoT and other personal smart devices, the amount of personal data stored and data transferred over a network, a malware that is capable of attacking almost all architectures used by those devices, especially ARM, x86 and x64, is incredibly dangerous. In order to protect ourselves and our personal data, certain countermeasures must be taken.
The purpose of this paper is to introduce the audience to the type of malware that is on the constant rise in order to create awareness so that they may better protect themselves. The countermeasures listed and explained in this paper are always welcome and recommended, by implementing them users can be protected against other existing and future malware and these steps are considered good practice. Once these countermeasures are in place, even infected devices will be contained and the malware won't be able to complete its purpose.

LITERATURE

[1] https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

[2] https://www.idc.com/getdoc.jsp?containerId=prUS44425818

[3] https://hubjapan.io/articles/the-torii-and-its-meaning-in-the-shinto-religion

[4] http://infocenter.arm.com/help/topic/com.arm.doc.ddi0500j/DDI0500J_cortex_a53_trm.pdf

[5] https://www.androidauthority.com/arms-64-bit-architecture-is-good-for-developers-407346/

[6] https://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/

[7] https://www.csoonline.com/article/3258748/security/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html

[8] https://www.makeuseof.com/tag/whats-custom-recovery-exploring-cwm-twrp-friends/

[9] https://www.addictivetips.com/android/what-is-twrp-how-to-install-use-it-on-android-devices-guide/