10 Years of IoT Malware: a Feature-Based Taxonomy

Benjamin Vignau*, Raphaël Khoury*, Sylvain Hallé*
*Laboratoire d'informatique formelle
Département d'informatique et de mahtématique
Université du Québec à Chicoutimi, Canada
Email: {benjamin.vignau1, rkhoury}@uqac.ca, shalle@acm.org

***Abstract*—Over the past decade, there has been a rapidly growing interest in IoT-connected devices. But as is usually the case with computer systems and networks, malicious individuals soon noticed that these objects could be exploited for criminal purposes. The problem is particularly salient since the firmware used in many Internet connected devices were developed without taking into consideration the expertise and best security practices gained over the past several years by programmers in other areas. Multiple attacks on IoT devices took place therefore over the last decade, culminating with the largest ever recorded DDoS attack, the Mirai botnet, which took advantage of the weaknesses in the security of the IoT. In this survey, we seek to shed light on the evolution of the IoT malware. We compare the characteristic features of 16 of the most widespread IoT malware programs of the last decade and propose a novel methodology for classifying malware based on its behavioral features. Our study also highlights the common practice of feature reuse across multiple malware programs.**

***Index Terms*—IoT security, evolution of IoT malware, VPNFilter, IoTReaper, BrickerBot**

I. Introduction

Since their apparition at the beginning of the century, connected devices have shown a tremendous and continuous growth. Every year, new types of connected devices reach the market. Taken together, they form the Internet of Things (IoT), defined by Bertino et al. [1] as the aggregation of sensors, actuators and services deployed by different organizations and individuals to support a variety of applications. It is predicted that by 2020, the number of IoT devices could reach 50 billions [2].

They are omnipresent in multiple fields including home automation [3], medicine [4], agriculture [5] or in smarts cities [6]. As with any innovation, the growth of IoT devices also raises several challenges. In particular, due to poor security design, several attacks against IoT devices have taken place in recent years. This is due in part to the fact that the development of IoT firmware does not always draw upon the security expertise accrued over the years by code developers working on other platforms. This in turns makes IoT devices tempting targets for malware developers.

In this survey, we analyze some of the most damaging IoT malware and paint a features-based evolution of these attacks. By malware, we describe malicious programs that infect numerous devices rather targeting a single organization. We focus specifically on 16 botnets, published between 2008 and 2018, that caused considerable damage to the IoT ecosystem. Most of these botnets were used to launch DDoS attacks, but we do not limit ourselves to this class of attack and instead cast a wide net over all IoT malware. Other objectives of IoT malware include industrial spying and crypto currency mining, amongst other things.

Our analysis focuses on the features and capabilities of each malware and highlights the relationships of feature borrowing that can be observed between malware. To make this relationship more salient, we propose two new visuals representations of the evolution of IoT malware over time. These representations make explicit the ways in which malware developers borrow from each other. Furthermore, drawing upon an analysis of the spread of features that we observe, we make recommendations that will help better secure the IoT ecosystem against malware.

By feature, we mean a distinctive functionality or capacity implemented by malware to perform a specific task. Notably, this includes the malicious behavior implemented by a malware, the means used to achieve its goal, or to obfuscate its presence as well as mechanisms used to improve its efficiency. For example, if a malware is able to perform DDoS attack using UDP flood, we state that it has the UDP flood feature (feature number 1.4), regardless how it implements this attack. [1]

The remainder of this paper is organized as follows: in Section II we list and categorize the papers used in this survey. In Section III, we described the features used to classify IoT malware. Section V provides an analysis of the evolution of IoT malware and of the pattern of shared features we observe. Consequent recommendations on securing the IoT ecosystem are given in Section VI. Concluding remarks are given in Section VII.

II. Related Works

We begin by introducing and classifying some of the papers that we relied upon in this survey. We selected 16 IoT malware, focusing on those malware with the most impact and for which the most data available. Our sample set dates back to 2008 and while we focus exclusively on IoT malware, we do not limit ourselves to malware that performs DDoS attacks.

[1] All our data, high resolution version of Figures 1, 2, 3 and our algorithms are available at https://github.com/bvignau/Softawre-Phylogenic-classification.

Malware	Technical Details	General Information	Source Code
Linux.Hydra (1)	[7], [8]	[8]–[11]	[12]
Psyb0t (2)	[7], [13], [14] [8]	[8]–[11], [15]	not available
Chuck Norris (3)	[7], [8], [15]	[8], [9], [11], [15]	not available
Tsunami/Kaiten (4)	[7], [8]	[8], [9]	[16]
Aidra (5)	[8], [9]	[9]	[12]
Carna (6)	[17]	[9], [10]	non available
Linux.Darlloz (7)	[9], [18], [19]	[9], [10], [18], [20]	not available
Linux.wifatch (8)	[21]	[11], [21]	[22]
Bashlite (9) [12]	[8], [23]	[9]–[11], [18]	
Remaiten (10)	[8], [18], [24]	[9]–[11], [18]	not available
Hajime (11)	[18], [25]	[9], [11], [18], [25]	not available
Mirai (12)	[8], [26]–[28]	[8]–[11], [18], [23], [26]–[28]	[12]
Amnesia (13)	[29]	[9], [18], [29], [30]	not available
BrickerBot (14)	[18], [31]	[11], [26], [32]	not available
IoTReaper (15)	[33]–[35]	[36], [37]	not available
VPNFilter (16)	[38]–[40]	[41], [42]	not available

TABLE I
CLASSIFICATION OF THE REFERENCES USED IN THIS PAPER.

For each malware, we sought papers (including academic research and popular press articles) that contained general information on the malware, such as the number of victims, targeted devices, discovery date, impacts of the attack on society etc. We also searched for papers with more technical details such as the type of the attack (DDoS, crypto currency mining etc.), the infection process, source code analysis or behavior analysis. In some cases the source code was available allowing us to analyze the structure of the botnet directly. The papers used in this study are summarized and categorized in Table I.

Every malware studied in this paper is the topic of at least two academic papers or technical reports. Moreover, we excluded any malware for which there was insufficient publicly available information to ascertain whether or not the malware implements each the features considered in this study. Our data set may not be complete, but it does contain the most widespread malware, and those that produced the largest and most damaging botnets. In this respect, we believe that our data set contains those malware that are most likely to influence future malware developers.

III. Malware Features

Our classification of IoT malware is derived from the features present in each malware. As a consequence, we build upon a more specialized taxonomy DDoS malware proposed by De Donno et al. [8]. Their taxonomy is based on the analysis of multiples features that describe DDoS attacks. It is not limited to IoT malware and thus includes malware that runs on other platforms, but only categorizes malware that performs DDoS. We extend their taxonomy with the inclusion several new features, such as crypto mining, infection process etc. thus giving us a greater insights on the evolution of malware.

A. Features List

The features are described, numbered and grouped into families below. The Table II maps each feature to a list of malware that implement it.

1) Denial of Service Capabilities: Several botnets aim to perform DDoS attacks. In this respect, Donno et al. [8] differentiate between different strategies employed to perform a DDoS attack, each of which we consider a different feature. This includes different types of flood attacks (features 1.1-1.6), the TCP XMAS (1.7) , DNS Waterboarding (1.8) and the DNS amplification (1.9).

In our study we add Physical DoS (1.10) feature, which consists in physically destroying object. This can be achieved by wiping the firmware of the device, or by overusing it, thus overheating of the object. The Firewall DoS (1.11) consists in disconnecting the victim by adding rules to his firewall. For example, adding a rule that drop all input and output packets will make it impossible for the victim to connect to the network.

2) Data Stealing: Aside from DDoS attacks, a common purpose of malware is to steal data from the target device. For example, a DNS Spoof (2.1) consists in changing the authoritative name server associated with the victim, thus redirecting some of his web traffic to a malicious server. If a user is redirected to a fake website, it becomes possible to steal his credential or to infect him with other malware. This type of attack is performed by the Chuck Norris Botnet [15]. It responds to a request for facebook.com, providing a fake website and asks the users to login in order to steal his credentials. Data exfiltration (2.2) consist in stealing data collected by IoT-connected devices and transmitting them to the attacker in a convert manner.

3) Endpoint Exploit: This feature is used by malware to infect other devices directly connected to any IoT device it has already infected (3.1). This can be done by exploiting known vulnerability in other devices, or by injecting malicious code in network communication (if the infected device is a router). For instance, through an analysis of network communications, the VPNFilter bot is able to detect if a Windows executable file is in the process of being exchanged. Researchers believe that this malware could inject a binary payload in an executable file and leverage on-the-fly patching of Windows executable to exploit endpoints device [40]. This malware also performed a Man-in-the-Middle attack (3.2) [43] in order to spoof communications between IoT devices.

4) Industrial Spying: Malware that targets industrial plants use several different strategies to spy on and control their targets. In our sample, these features are mainly used by the VPNFilter botnet [38]–[40] and include a SCADA monitoring feature (4.1). SCADA systems are used in industrial plants to acquire data from mechanical parts such as turbines and to control them. SCADA systems are often attacked to slow down industrial activities [44]. Moreover, VPNFilter was able to map the local network of the infected device (4.2). Another innovative feature of this malware is its ability to create a reverse TCP-VPN to allow a remote attacker to access the internal networks (4.3). Finally, malware that targets industrial plants also need to obfuscate malicious traffic to avoid detection while exfiltrating data or when the attackers use the VPN to spy on and control local network devices (4.4).

5) Exploit: In the course of our study, we noticed that malware use several different strategies to infect new devices. The most common is the simple dictionary attack (5.1), in which the botnet attempts to brute force the credentials of the victim. Multiple botnets use this technique, only varying the credentials dictionary. The second method used is the balanced dictionary (5.2), introduced by the Mirai botnet [28]. Here, the botnet selects a random subset of credentials from the dictionary and only attempts those credentials. Each credential has a different probability of being picked; chosen by the attacker based on the most frequently used default password for the device in question. This latter technique is far more effective than the dictionary attack. Another strategy is to exploit vulnerable devices is to make use of an unpatched CVE. At first, each botnet relied upon a single CVE (5.3), but the past two years have seen the emergence of botnets that rely upon multiple CVE thus infecting more devices (5.4).

6) Target Architecture: IoT devices can run on any one of several possible architectures. Early botnets mostly targeted MIPS and MIPSEL devices (6.1). Then, starting in 2012, all other architectures commonly used, such as ARM (6.2) or x86/64 (6.3) became targeted, due to the adoption of cross compilation capabilities. Recently, some botnets began to use scripting languages such as BASH to reach multiple IoT devices without the need to create multiples binary (6.4). On the other hand, some malware is designed to attack and exploit IoT devices from one specific constructor, taking advantage of an unpatched vulnerability (6.5).

7) Scanning methods: Botnets employ several different methods to scan the web in order to detect new targets and infect them. The earliest one, used by Hydra, consisted in scanning a pre-programmed list of IP addresses (7.1). Attackers later automated this process by incorporating code that generates a hit list of potential targets from a given subnet in the bot's code (7.2). In 2012, botnets began to use random IP scan, which are stealthier and easier to implement (7.3). Most botnets scan the Telnet protocol with ordinary probes. Mirai introduced a new and far more rapid scanning method: it performs a stateless scan and does not wait for a timeout before moving on to a new IP [28] (7.4).

8) Botnet architecture: Following De Donno et al. [8], we categorize botnets' architectures in three categories. The first one is the centralized form, in which a Command & Control (C2) server uses IRC to communicate with the bots (8.1). The second one is the P2P architecture, where there is no centralized Command & Control (C2) server (8.2). The P2P protocols used by this botnet are often derived from the BitTorrent or uTorrent protocol. The final category includes botnets where a centralized architecture uses a custom communication protocol (8.3).

9) Anti-detection Features: To better avoid detection several malware kill legitimates processes (9.1) and adopt the name of those processes as their own. For example, the Hajime botnet masquerades itself as the Telnet process [25]. Moreover several botnets also delete their own binary files (9.2).

10) Efficiency Enhancing Features: Several malware include other novel and interesting features that improve their efficiency. The earliest one was port closing (10.1). Botnets close ports in the interval 22 to 80 to avoid new infections and thus monopolize the device's entire computing power. Later, some botnets began to delete other older botnets already present on the device (10.2).

Because all malware must reside in the RAM, some botnets such as Mirai, developed anti-reboot features. Such features kill the watchdog process in order to avoid an auto-reboot of the infected device (10.3). Later, the VPNFilter malware successfully alters the firmware of infected device, allowing the bot to persist even after a reboot (10.4).

11) DGA Algorithm: When a centralized botnet becomes too prevalent, law enforcement will usually try to shut down the C2 server. In most botnets, the IP address of the C2 server is hardcoded in the binary. Consequently, the server can be shutdown as soon as security researchers obtain a copy the binary code. To make the task of law enforcement more complex, some malware include a Domain Generation Algorithm [45] (11). Each day, they generate up to several hundred pseudo-random domain names, only one of which is actually valid. The attacker chooses any single one of them as a rendezvous point to control his botnet. The bot will have to attempt communication using each domain name to communicate with the C2.

12) Code Modularity: Some malware integrate update and code modularity features (12). For example, VPNFilter attempts to download new modules each day, including scripts that it will then execute. Consequently, the malware can perform virtually any kind of attack.

13) Victim Scan: Starting in 2012, malware began to include cross compiled binaries in order to infect multiple IoT binaries at the same time. The earliest malware to include this feature simply downloaded every variant of the binary on the device and attempted to execute each of them. Later variants, such as Remaiten or Mirai, scan the host to determine its architecture and only send the correct binary (13). Malware also differ in how they detect potential victims, with some variants using TCP scan and other using vulnerability scans.

14) Virtualization Evasion: One important feature used by the Amnesia botnet is the ability to detect virtualized environment. If operating in such an environment, Amnesia will delete itself as well as the entire file system of the target (14). This feature was first used by more advanced malware, such as those that target Windows PC and Smart phones and makes it harder for security professional to study the malware's code.

15) Crypto Mining: Crypto mining capabilities (15) allow malware to hijack the computing power of infected devices in order mine crypto currencies. This is an emerging threat, which originates in malware targeting PC and was first observed in IoT malware in 2014 with the Linux.Darlloz malware.

IV. Methodology

Having identified the features that distinguish malware, we now endeavor to draw a phylogenic classification of malware, highlighting how each malware may have influenced its successors. To this end, we propose two new graphic representations of the relationships between malware, each of which highlights a different aspect of the patterns of feature borrowing between malware.

A. Phylogenic Graph

The first representation that we introduce is the *Phylogenic graph*, which captures the number of features common between different malware. In this graph, each malware is present as a vertex, whose radius is proportional to the number of features from Section III exhibited by this malware. An arc links two vertices, from the oldest to the most recent, (1) if they share at least one feature and (2) if the release date of the former precedes that of the latter by at least six months. The latter condition is added because, given the expected time needed to develop new malware, it is prudent to assume that malware released in an interval of less than six months have been developed independently and any share features are likely coincidental. Each arc is weighted with the number of common features shared by the vertices it connects.

Finally, if multiple paths link two vertices, we only preserve the arcs with the highest weight. This prunes the graph of connections that can be considered redundant, as they refer to the same features being borrowed from the same malware. For example, Chuck Norris and Psyb0t both borrow four features from Hydra. Chuck Norris additionally borrows six features from Psyb0t: the four it borrowed directly from Hydra, plus two others it borrowed from Psyb0t. As a consequence, a direct connection between Chuck Norris and Hydra can be deemed redundant and deleted from the graph.

B. Feature Propagation Multigraph

The phylogenic graph provides a striking visual representation of the relationship between malware. It does however, obfuscate some important details, such as exactly which features are shared by any two bots. Furthermore, some features are shared by multiple bots, which leads to the presence of several uninformative arcs. We thus propose an alternative representation, the *Feature propagation multigraph* (FPM), which illustrates how individual features spread in the malware pool.

The feature propagation multigraph identifies each feature with a distinct color. A vertex of that color identifies the first malware to exhibit this feature and an edge of the same color links it to every other malware that share this specific feature. The feature propagation multigraph provides a more detailed

Feature	malware	feature	malware
Syn Flood (1.1)	1, 2, 3, 4, 5, 9, 10, 12, 13, 15	UDP Flood (1.2)	2, 3, 4, 9, 10, 12, 13, 15
ICMP Flood (1.3)	2	ACK Flood (1.4)	3, 4, 5, 9, 10, 12, 13, 15
Push flood (1.5)	4, 10, 12, 13, 15	HTTP Flood (1.6)	4, 9, 10, 12, 13, 15
TCP XMAS (1.7)	4, 10	DNS Waterbording (1.8)	12, 13, 15
DNS Amplification (1.9)	12, 13, 15	Physical DoS or Permanent DoS (1.10)	14, 16
Firewall DoS (1.11)	16	DNS Spoof (2.1)	3, 4
Data exfiltration (2.2)	16	End point Exploit (3.1)	16
Man In The Middle attack (3.2)	16	SCADA Monitoring (4.1)	16
Local network mapping (4.2)	16	Reverse TCP VPN (4.3)	16
Malicious traffic obfuscation (4.4)	16	dictionary password (5.1)	1, 2, 3, 4, 5, 7, 8, 9, 10, 11
Balanced dictionary password (5.2)	12, 14	CVE Exploit (5.3)	7, 9, 13
Multiple CVE (5.4)	15, 16	MIPS (6.1)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16
ARM (6.2)	5, 6, 7, 8, 9, 10, 11, 12, 16	x86/64 (6.3)	5, 6, 7, 8, 11, 12, 16
Script (6.4)	14, 16	targeted manufacturer (6.5)	13, 15
Harcoded hit list (7.1)	1	Network class automated hit list (7.2)	2, 3, 4
Random (7.3)	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	Stateless (7.4)	12, 14, 16
Centralized (IRC CC) (8.1)	1, 2, 3, 4, 5, 7, 10, 13	P2P (Decentralized) (8.2)	8, 11
dedicated protocol (Centralized CC) (8.3)	6, 9, 12, 14, 15, 16	process masquerade (9.1)	10, 11, 12, 15
binary removal (9.2)	11, 12, 15	ports closing (10.1)	4, 5, 7, 8, 9, 11, 12, 13, 15
other botnet removal (10.2)	5, 7, 8, 11, 12, 13, 15	Anti reboot (10.3)	11, 12, 15, 16
Persistence (10.4)	16	DGA Algorithm (11)	12, 16
code's modularity or update system (12)	11, 15, 16	victim architecture's scan (13)	10, 11, 12, 15, 16
virutalisation evasion (14)	13	CryptoMining (15)	7

TABLE II
Malware's features.

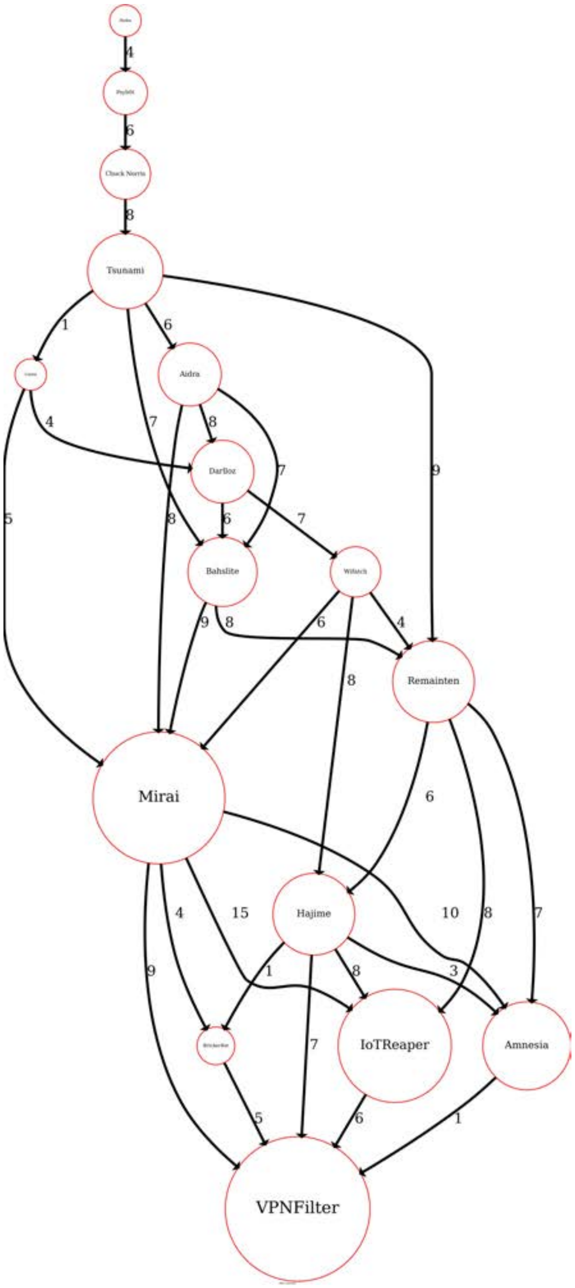

Fig. 1. Phylogenic graph.

view of the way in which malware developers introduce new features and how they are used by other developers. It also highlights which features are the most commonly used.

Out of a concern for legibility, we split the features in two groups and produced two Feature propagation multigraphs. Figure 2 shows the spread of features related to the type of attack and Figure 3 shows the spread of features related to the target architecture, efficiency and exploit method. Features related to botnet architectures are omitted here.

V. Analysis of the Evolution of IoT Malware

As can be seen from an inspection of both graphs, malware has evolved considerably in just a few years. In general, it tends to became more and more complex and thus able to infect more devices and launch larger attacks. Using the phylogeny graph, we can detect a close proximity between certain malware that have been confirmed through code analysis. For example, Chuck Norris is very closely related to Psybot and a code analysis [15] strongly suggests that the same authors wrote both.

Another interesting point is that some malware are very closely in our graph, while their goals are largely different. For example, Aidra and Wifatch share multiple features, but the former aims to create a DDoS botnet, while the latter aims to remove malicious botnet and warn the users. The similarities are due to the fact that both target the same architectures and to the manner in which they exploit IoT devices.

As an another example, Carna and Aidra also share multiple features. In this case, the two malware seems to have been developed simultaneously but independently, by two different groups of authors, a fact that is visible in Figure 3. The majority of botnets developed in subsequent years reused these features and indeed these features are the most commonly used in our sample, alongside with the MIPS feature. Finally, we can also see that Mirai and VPNFilter are the two botnets that use the widest number of features. Moreover, we can see that Mirai directly influenced every subsequently created botnet. This is due to the numerous features that it introduced and by the release of its source code.

Figure 2 highlights the consistent evolution of attacks features used by malware developers. The majority of malware under consideration in this paper aims to build DDoS capabilities. Such capabilities are easily transmitted across different malware. However, we can see that the “ICMP Flood” feature is not propagated to other malware. This is certainly due to the poor effectiveness of this technique. Nowadays, most firewalls disable ping response and it is easy for ISPs to detect and mitigate this kind of attack [46]. Just like biological evolution, some features are selected according to their effectiveness at their intended task and passed on to subsequently developed malware. Mirai, one of the most successful botnet ever recorded, is an unavoidable reference for any aspiring botnet developer. Another interesting case is that of VPNFilter. It introduces several new features that enable it to launch severely damaging attacks against industrial plants and companies. It can infect and spy on any local network from an infected device. Such features are far more difficult to implement than the DDoS capabilities present in most other botnets. This attack is believed to have originated with Russia [41] and it likely that these features will be used in the future by other like-minded nation-states.

Figure 3 also shows that comparatively little evolution occurred in selection of the architecture targeted by malware. This is due to the introduction cross-compiled binary by the Carna and Aidra botnets. The majority of subsequently developed malware employed this strategy in order to spread

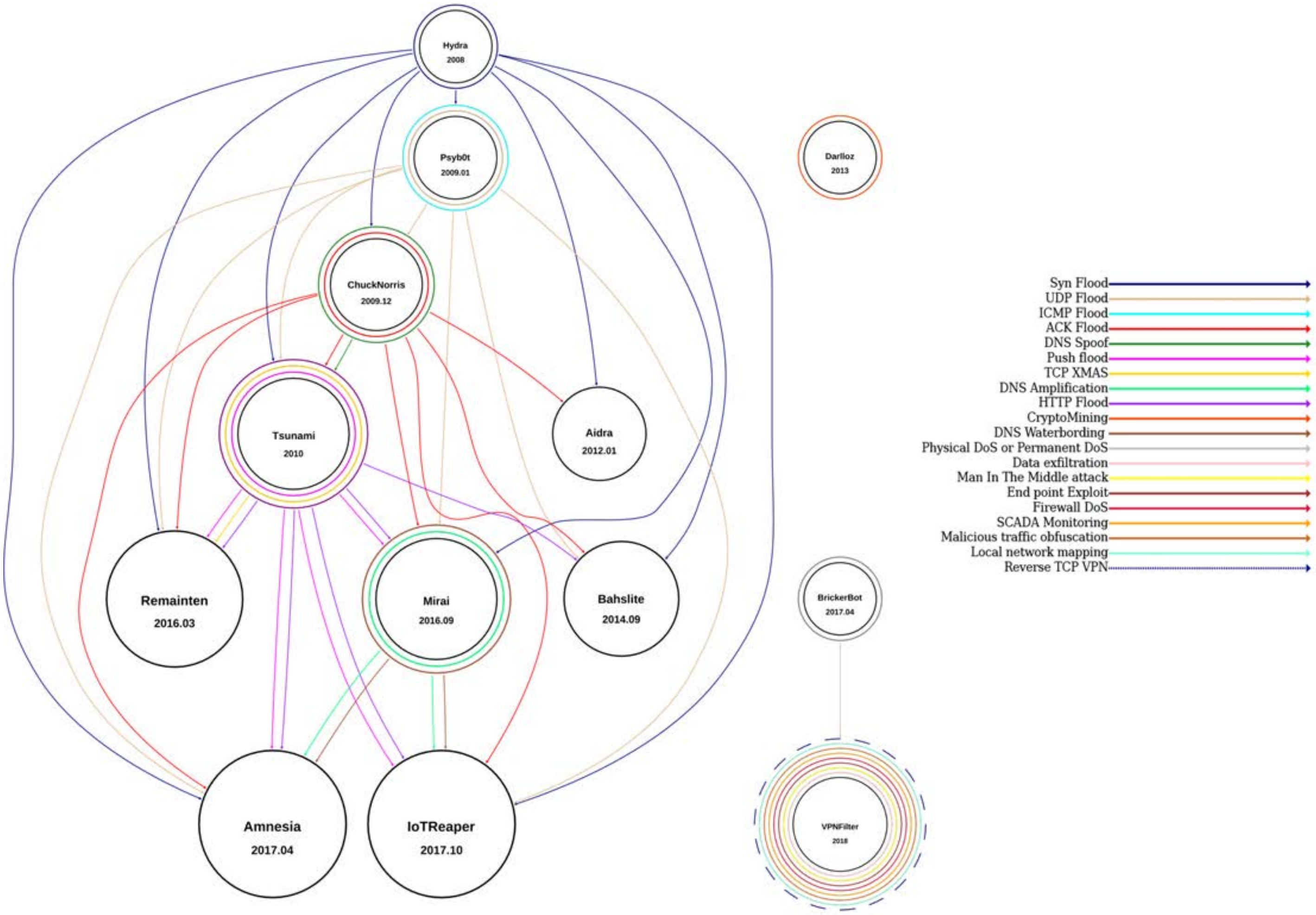

Fig. 2. Feature propagation multigraph: attack type features.

the infection more widely. The only ones who do not are bots designed to target specific devices such as Amnesia. This malware used a CVE exploit in DVR recorders from a Chinese manufacturer and, therefore, did not require cross-compilation to reach different architectures. The “script” architectures are only used by the BrickerBot, which used bash commands to destroy IoT devices. This feature is very effective at rapidly creating a small botnet. It is also very effective at executing security-critical commands on the infected device, without having to code them. It is an interesting feature, but is used by only one malware in our sample, largely because in using only shell command, the attacker reduces the number potential victims considerably.

We observe the same behavior for every feature family: new features are rapidly adopted by subsequent malware if they are both useful and easy to implement. Moreover, a simple feature can evolve. For example, the dictionary attack evolved with each malware that implemented it. In particular, the number of credentials varied, with malware either adding or removing credentials from the dictionary. After Mirai’s source code was released, more than one thousand different variants appeared. Researchers analyzed the evolution of the dictionary size and use of credentials [28]. Notably, adding more credentials in the dictionary often did not result in a malware that spread more rapidly. Moreover, some clusters that relied upon the original, unedited credential dictionary were able to infect more devices than later, updated variants.

Finally, in Figure 3, we observe that the classic dictionary attack is the attack mechanisms most widely used by malware in our sample set. However, since Mirai, this infection method has largely been supplanted by the use of balanced dictionaries and of multiple CVE. These two techniques are far more effective than the simple dictionary attack. The use of unpatched CVE, in particular, is by far the fastest and most effective way to gain control of a device, but naturally necessitates more competence to be implemented than a simple dictionary attack.

The same phenomenon is observable when looking at the architectures of P2P botnets. Our sample set contains two such botnets Wifatch and Hajime. Wifatch is a “vigilant malware”, which warns users that they have been infected and

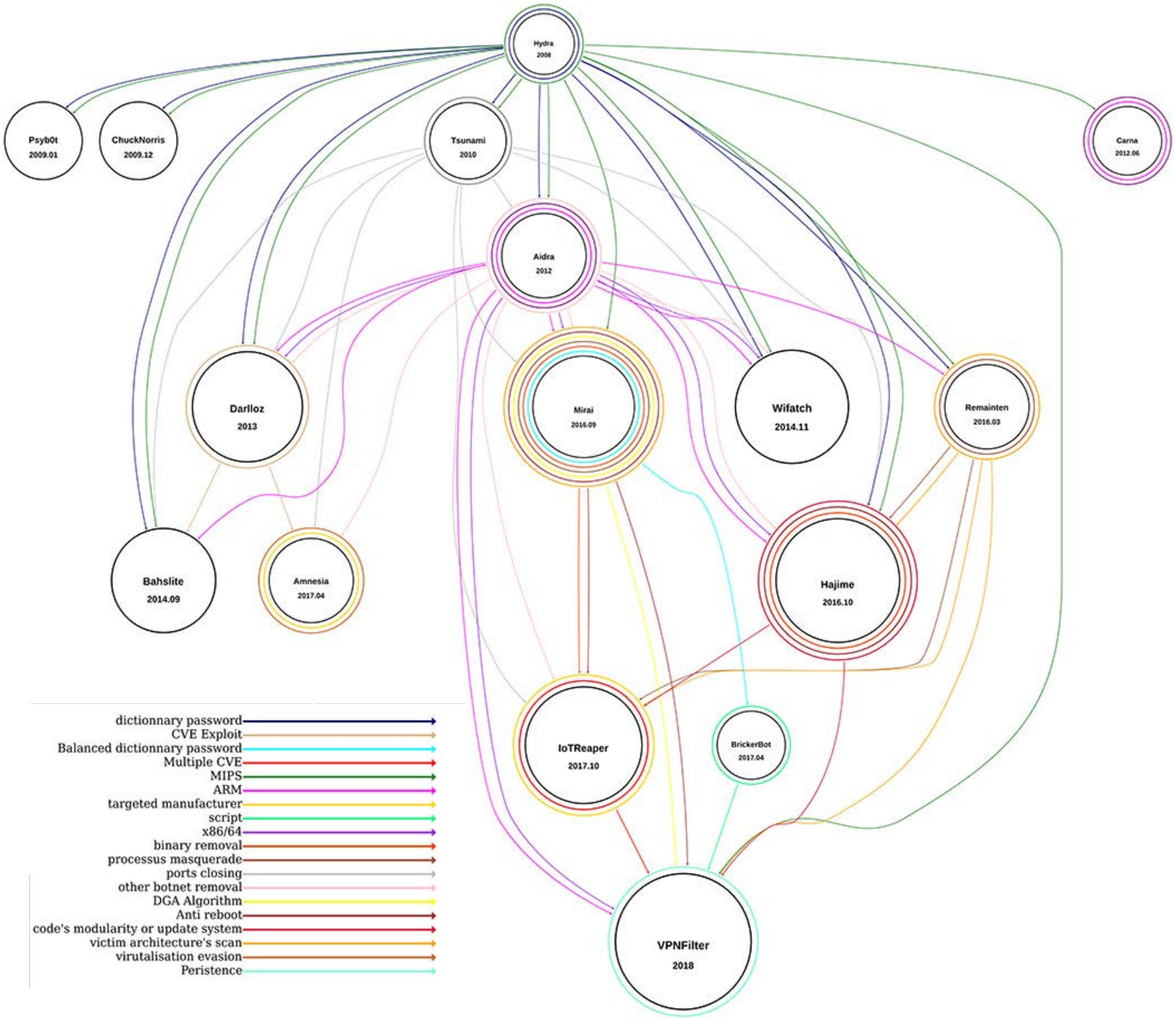

Fig. 3. Feature propagation multigraph: target architecture, efficiency and exploit method features.

removes other botnets from vulnerable devices. Hajime installs a backdoor, but so far it has not been observed performing any other malicious activity. The two botnet used different P2P protocol to communicate. P2P botnets are both much harder to develop than centralized botnets but also much harder for an adversary to take down. Furthermore, researchers have created a P2P mechanism based on the Bitcoin networks, creating a very robust botnet [47]. As a consequence, we predict that Centralized botnets will come to disappear and will eventually be replaced with P2P botnets.

VI. Discussion and recommendation

As we have shown in this paper, IoT malware has gradually become much more complex in the span of just a few years. Moreover, multiple features, such as virtualization evasion, were originally used in malware that targets PC and smart phones; and only later included in IoT malware, starting with the Amnesia botnet. This development hints that in the next few years, malware will likely evolve to target all numerical platforms. Such a fusion is already underway with the VPNFilter attack, which can target and exploit endpoints devices from infected IoT devices.

Our analysis reveals that dictionary attacks on Telnet and SSH protocol are the most widely used and among the most effective strategies to infect IoT devices. This kind of attacks is present since the first IoT botnet in 2008 and is still commonly used today. A simple and effective strategy to frustrate this attack is to assign each user a strong, random default password. Ideally, manufacturers could be compelled to do so. Moreover, all unused ports should be closed by default and unused protocols should be removed from the device's firmware, thus reducing the attack surface.

However, we found that the use of unpatched CVE to exploit IoT device has grown in recent years and can be expected to grow further, alongside with a growth in cross-platform infections. This type of exploit cannot be stopped with strong passwords alone and is far more effective than a dictionary attack, but also much more difficult to implement. Moreover, this feature allows attackers to use an "out of band" infection vector. If he employs such a strategy, the attack cannot be detected through conventional means such as network telescope or honey pots.

For example, this kind of attack might use a method similar to the one created by Ronen et al. [48] to infect the ZigBee light bulb. Their attack consists in replacing the firmware of the light bulb with an alternate, malicious firmware, using the Over The Air (OTA) update system. A ZigBee chip mounted on a drone triggers the first infection. Subsequently, a single infected light bulb can infect every other light bulb within a few hundred meters. Researchers used this worm to transmit "SOS" in Morse code with light flashes, but it is easy imagine more damaging uses of IoT light bulbs, such as covert data exfiltration. Attackers could also use the infected light bulb to infect other devices in the local network.

It is also highly probably that in the near future, IoT botnets will evolve the capacity to perform "ransomware attacks", since this type of attack has become very prevalent in the world of PC malware [49]. In other words, botnets will hinder the user's experience by shutting down IoT devices and demand ransom in exchange for releasing control of the infected device. This type of attack can take the form of a reduction attack or a misuse attack. Such attacks are more damaging than DDoS attacks and can be much more lucrative.

To reduce the use of CVE, each device should have a reliable update system, using public key signatures, or physical update. The use of weak methods to update firmware must be prohibited. For example Philips used a symmetric key to sign their updates and researchers discovered a vulnerability that allows the extraction the key and the creation of a malicious update [48] from a retro-engineering of the firmware.

Another part of the reason why IoT-connected devices are so vulnerable to attack is that the firmware used in several Internet-connected devices was developed without taking into account the expertise and best security practices accrued over the last several years by programmers working on other, more frequently targeted platforms. Notably, security considerations must be included from the onset of the development process. For examples an IP Camera does not require to have a bash and a full Linux OS. In addition, some the use of certain critical features must be prohibited, such as the possibility to wipe the memory of a device. Indeed, the BrickerBot malware executes shell commands on the infected devices that allows it to delete the firmware. Some of these commands write random bytes on the storage device while others change the firewall rules in order to reset all TCP connections. There is no reason to include such commands in an IoT-connected device.

Another step that could be taken to further secure IoT systems is to analyze their normal behavior and develop an anomaly detection tool that detects deviation from the expected behavior and reacts accordingly. While a general definition of normal behavior has so far eluded security professionals, the narrowly defined behavior of IoT devices could make such a tool highly effective in securing such devices.

Of course, other well-established security practices such as regular password changes, strong password policies and the implementation of a good firewall are an essential part of the solution. Moreover, it would be interesting to separate IoT devices networks and PC/smart phones networks and use gateway to analyze and filtrate data flows.

VII. Conclusion

IoT malware have infected millions of devices around the world in the last few years. Our study traces a taxonomy of such malware, highlighting how features are shared across multiple malware and how one malware influences successive ones. We also argue that in the near future, IoT malware will slowly merge with malware targeting other platforms to create large-scale worms and will slowly include ransomware and crypto mining features. We believe that this survey will be useful to the scientific community, security experts and constructors and will help create tools that improve the security of IoT networks.

References

[1] E. Bertino, K.-K. R. Choo, D. Georgakopolous, and S. Nepal, "Internet of things (IoT): Smart and secure service delivery," *ACM Trans. Internet Technol.*, vol. 16, pp. 22:1–22:7, Dec. 2016.

[2] G. Davis, "2020: Life with 50 billion connected devices," in *2018 IEEE International Conference on Consumer Electronics*, pp. 1–1, Jan 2018.

[3] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.

[4] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT ehealth: Promises and challenges of IoT in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659 – 676, 2018.

[5] S. Hu, H. Wang, C. She, and J. Wang, "AgOnt: Ontology for agriculture internet of things," in *Computer and Computing Technologies in Agriculture IV* (D. Li, Y. Liu, and Y. Chen, eds.), (Berlin, Heidelberg), pp. 131–137, Springer Berlin Heidelberg, 2011.

[6] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, pp. 22–32, Feb 2014.

[7] M. Janus, "Heads of the Hydra. malware for network devices," 2011.

[8] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "DDoS-capable IoT malwares: Comparative analysis and Mirai investigation," *Security and Communication Networks*, vol. 2018, pp. 1–30, 02 2018.

[9] B. Botticelli, "IoT honeypots: State of the art," 2017.

[10] K. Angrishi, "Turning internet of things(iot) into internet of vulnerabilities (iov) : Iot botnets," 02 2017.

[11] A. C. Zaddach and Jonas, "IoT malware comprehensive survey, analysis framework and case studies," in *Black Hat 2018*.

[12] ifding, "GitHub repo of open source IoT malware," 2017.

[13] L. Ďurfina, J. Křoustek, and P. Zemek, "Psybot malware: A step-by-step decompilation case study," in *2013 20th Working Conference on Reverse Engineering (WCRE)*, pp. 449–456, Oct 2013.

[14] DroneBL, "DrobeBL," 2009.

[15] P. Celeda, R. Krejci, J. Vykopal, and M. Drasar, "Embedded malware - an analysis of the Chuck Norris botnet," in *2010 European Conference on Computer Network Defense*, pp. 3–10, Oct 2010.

[16] unlnow, "Kaiten source code," 2015.

[17] C. Botnet, "Internet census 2012, port scanning /0 using insecure embedded devices," 2012.

[18] S. Manoharan, "Iot malware:an analysis of device hijacking," 10 2018.

[19] K. Hayashi, "Linux.darlloz," 2013.

[20] K. Hayashi, "IoT worm used to mine cryptocurrency," 2014.

[21] M. Ballano, "Is there an Internet-of-things vigilante out there?," 2015.

[22] T. W. team, “Wifatch gitlab,” 2016.

[23] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. Chaves, I. Cunha, D. Guedes, and W. Meira Jr, “The evolution of Bashlite and Mirai IoT botnets,” 06 2018.

[24] M. Malik and M.-E. M.Léveillé, “Meet Remaiten – a Linux bot on steroids targeting routers and potentially other IoT devices,” 2016.

[25] I. P. Sam Edwards, “Hajime: Analysis of a decentralized internet worm for iot devices,” 2016.

[26] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[27] Y. Ji, L. Yao, S. Liu, H. Yao, Q. Ye, and R. Wang, “The study on the botnet and its prevention policies in the internet of things,” in *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, pp. 837–842, May 2018.

[28] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, “Understanding the mirai botnet,” in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 1093–1110, USENIX Association, 2017.

[29] C. Xiao and U. . Cong Zheng, “New IoT/Linux malware targets dvrs, forms botnet,” 2017.

[30] J. Leyden, “’amnesia’ IoT botnet feasts on year-old unpatched vulnerability,” 2017.

[31] Radaware, “”BrickerBot” results in PDoS attack,” 2017.

[32] C. Cimpanu, “BrickerBot dev claims cyber-attack that affected over 60,000 Indian modems,” 2017.

[33] C. Point, “Iotroop botnet: The full investigation,” 2017.

[34] T. M. System, “New rapidly-growing iot botnet - reaper,” 2018.

[35] yegenshen, “IoT_reaper: A rappid spreading new iot botnet,” 10 2017.

[36] F. SE, “Reaper: The next evolution of IoT botnets,” November 16, 2017.

[37] B. Krebs, “Fear the reaper, or reaper madness ?,” 2017.

[38] T. U. William Largent, “New VPNFilter malware targets at least 500k networking devices worldwide,” 2018.

[39] T. U. William Largent, “VPNFilter update - VPNFilter exploits endpoints, targets new devices,” 2018.

[40] T. U. Edmund Brumaghin, “VPNFilter iii: More tools for the Swiss army knife of malware,” 2018.

[41] C. Cimpanu, “FBI takes control of APT28’s VPNFilter botnet.”

[42] C. Cimpanu, “Ukraine says it stopped a VPNFilter attack on a chlorine distillation station,” 12/07/2018 2018.

[43] M. Conti, N. Dragoni, and V. Lesyk, “A survey of man in the middle attacks,” *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2027–2051, thirdquarter 2016.

[44] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, pp. 380–388, Oct 2011.

[45] A. K. Sood and S. Zeadally, “A taxonomy of domain-generation algorithms,” *IEEE Security Privacy*, vol. 14, pp. 46–53, July 2016.

[46] CloudFlare, “Ping (icmp) flood DDoS attack.”

[47] S. T. Ali, P. McCorry, P. H.-J. Lee, and F. Hao, “ZombieCoin 2.0: managing next-generation botnets using bitcoin,” *International Journal of Information Security*, vol. 17, pp. 411–422, Aug 2018.

[48] E. Ronen, A. Shamir, A. Weingarten, and C. O’Flynn, “Iot goes nuclear: Creating a zigbee chain reaction,” in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 195–212, May 2017.

[49] I. Yaqoob, E. Ahmed, M. H. u. Rehman, A. I. A. Ahmed, M. A. Algaradi, M. Imran, and M. Guizani, “The rise of ransomware and emerging security challenges in the Internet of things,” *Comput. Netw.*, vol. 129, pp. 444–458, Dec. 2017.