Proceedings of the Federated Conference on Computer Science and Information Systems pp. 807–816 DOI: 10.15439/2017F288 ISSN 2300-5963 ACSIS, Vol. 11

Analysis of DDoS-Capable IoT Malwares

Michele De Donno*, Nicola Dragoni*†, Alberto Giaretta† and Angelo Spognardi*‡
*DTU Compute, Technical University of Denmark, Denmark
Email: micheledEdonno@gmail.com, {ndra, angsp}@dtu.dk
†Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden
Email: alberto.giaretta@oru.se
‡Computer Science Department, Sapienza University of Rome, Italy

***Abstract*—The Internet of Things (IoT) revolution promises to make our lives easier by providing cheap and always connected smart embedded devices, which can interact on the Internet and create added values for human needs. But all that glitters is not gold. Indeed, the other side of the coin is that, from a security perspective, this IoT revolution represents a potential disaster. This plethora of IoT devices that flooded the market were very badly protected, thus an easy prey for several families of malwares that can enslave and incorporate them in very large botnets. This, eventually, brought back to the top Distributed Denial of Service (DDoS) attacks, making them more powerful and easier to achieve than ever. This paper aims at provide an up-to-date picture of DDoS attacks in the specific subject of the IoT, studying how these attacks work and considering the most common families in the IoT context, in terms of their nature and evolution through the years. It also explores the additional offensive capabilities that this arsenal of IoT malwares has available, to mine the security of Internet users and systems. We think that this up-to-date picture will be a valuable reference to the scientific community in order to take a first crucial step to tackle this urgent security issue.**

I. Introduction

THE Internet of Things (IoT) is rapidly and unavoidably changing our society, affecting the way we live and work. The IoT mission is to enable everyday objects to communicate with each other through the Internet, resulting in a figurative tsunami of connectivity. From a business perspective, IoT is all about excitement. Firms are rushing the development of their IoT products in order to commercialise them as soon as possible, and stay on the crest of the wave. IoT predictions by several consultancy firms (like Bain, McKinsey, General Eletric, to mention only a few) clearly show that the IoT market will become massive in the coming 10 years. For instance, IHS forecasts that the IoT market will grow from a base of 15.4 billion devices in 2015 to 30.7 billion devices in 2020 and 75.4 billion in 2025[1].

From a security perspective, all this excitement goes to the detriment of the IoT devices security, causing a potential disaster. Indeed, security still represents the most overlooked characteristic when quickness is considered of paramount importance for business. Moreover, the massive distribution of such connected devices to the "average security-unsavvy user", evokes IoT acronyms like the not-so-funny "*Internet of Troubles*"[2]. More connected and non-secure (or unsecured) devices entails more attack vectors and more possibilities for hackers to target us, access our sensible data and control our devices. Talking about security and IoT devices, the 2016 is still remembered as the year of *Mirai*, namely a powerful malware that managed to infect hundreds of thousands of connected devices all over the world through a dictionary attack (composed of just 50 entries), relying upon the fact that these devices use default login credentials and that most of the users never change those credentials. On October 21th 2016, this massive botnet (network of infected devices) was used to struck what is currently considered the largest Distributed Denial of Service (DDoS) attack ever seen, reaching a magnitude of about 1.2 Terabits per second.

Contribution of the Paper. The security disaster in this IoT tsunami of connectivity has made DDoS attacks more and more popular among the cyber-criminal community. DDoS attacks have rapidly evolved in the last few years, becoming more complex and especially more powerful and effective, as Mirai showed. Besides, to the best of our knowledge, the last research work discussing a taxonomy of DDoS attacks has been conducted in the early 2008 [1], long before the IoT outburst. Therefore, this paper aims at studying DDoS attacks with focus on the IoT context. In particular, the contribution of our analysis is twofold:

1) We start from an up-to-date comprehensive taxonomy of DDoS attacks based on previous scientific literature and the latest performed attacks, and we place the emphasis on IoT devices. The taxonomy is obtained by combining several surveys in the literature [1]–[13] and by refining the taxonomy previously proposed in [14].
2) Using the new DDoS taxonomy as foundation of our study, we provide a detailed analysis of all the DDoS capable IoT malwares since 2008. The analysis clearly shows the evolution of these malwares through the years, as well as the increasing number of new malware families per year.

The overall aim of the paper is to provide a first comprehensive reference to the security community, in order to understand the latest DDoS attacks targeting the IoT domain.

Outline of the Paper. Section II introduces DDoS attacks, focusing on the key characteristics that make them possible and so powerful. Sections III and IV present the proposed taxonomy of DDoS attacks and the analysis of DDoS-capable

[1] https://www.ihs.com/Info/0416/internet-of-things.html [May 10th, 2017].
[2] https://security-online.net/iot-like-internet-troubles [May 10th, 2017].

IEEE Catalog Number: CFP1785N-ART ©2017, PTI

IoT malwares, respectively. Section V analyses the collected data and draws some remarkable observations. Finally, Section VI sums up the contribution of the paper.

II. How DDoS Attacks are Possible?

What makes DDoS attacks possible and extremely powerful is the intrinsic nature of Internet itself, designed with the aim of functionality, rather than security. While being utterly effective, the Internet is inherently vulnerable to several security issues that can be used to perpetrate a DDoS attack [3], [5]:

- *Internet security is extremely interdependent* – It does not matter how well secured the victim system may be, its vulnerability to DDoS attacks depends on the security of the rest of the global Internet;
- *Internet entities have limited resources* – Each Internet entity (such as hosts, networks, services, etc.) has limited resources that can be saturated by a given number of users;
- *Many is better than a few* – Coordinated and concurrent distributed attacks will always be effective, if the resources of the attacker are greater than the resources of the victim;
- *Intelligence and resources are not collocated* – Most of the intelligence, needed to guarantee services, is located in end hosts. Nevertheless, the requirement of large throughput brought to design high bandwidth pathways in the intermediate network. As a result, attackers can exploit the abundant resources of the intermediate network in order to deliver a great number of malicious messages to the victim;
- *Accountability is not enforced* – In IP packets, the source address field is assumed to carry the IP address of the host that creates the packet. However, this is an assumption which is not validated or enforced at all, therefore there is the opportunity to perpetrate an *IP source address spoofing*[3] attack. This attack provides attackers a powerful mechanisms to avoid responsibility for their actions;
- *Control is distributed* – Internet management is distributed and each network can work with local policies defined by its administrators. Consequently, there is no way to deploy a global security mechanism or policy and it is often impossible to investigate cross-network traffic behaviour due to privacy issues.

Notably, a DDoS attack needs to go through the following phases in order to be struck [3], [5]:

1) *Recruitment*. The attacker scans for vulnerable machines (called *agents*), aiming to use them later in the DDoS attack against the real victim. In the past this process was performed manually but nowadays several scanning tools can be used to do this automatically;
2) *Exploitation & Infection*. The agent machines are exploited using the discovered vulnerabilities and the malicious code is injected. This phase has also been automated and nowadays several self-propagating tools can be used for further recruiting new agents;
3) *Communication*. The attacker uses the handlers or the IRC channel (depending on the botnet architecture, refer to subsection III-A for further details) to identify which agents are up and running, when to schedule the attacks or when to upgrade the agents;
4) *Attack*. The attacker commands the onset of the attack and the agent machines start to send malicious packets. Attack parameters (such as victim, duration, malicious packets properties, etc.) are tuned in this phase. Although IP spoofing is not always required for a successful DDoS attack, attackers usually opt for an additional anonymity layer, hiding the identity of agent machines during the attack.

III. DDoS Attacks Classification

DDoS attacks can be classified in many ways (Fig. 1). In this section, we succinctly report a complete taxonomy, obtained by combining several surveys in the literature [1]–[13].

A. Architecture Model

The architecture of a DDoS attack considers how the involved actors interact. There are basically four types of network architectures that can be used to perpetrate a DDoS attack [1], [9]: *Agent-Handler Model*, *Reflector Model*, *IRC-Based Model*, *Web-Based Model*.

1) Agent-Handler Model: This model (Fig. 2a) is composed by *clients*, *handlers* (or masters) and *agents* (or daemons or secondary victims) [2]. Clients are used by the attacker to communicate with the handlers, which are software packages located somewhere in the Internet, that infect network resources and rely information from the clients to the agents. The agent is a block of code that runs on a compromised system and performs the attack against the final victim. The term agent is used to refer both to the compromised machine and to the running code. According to the configuration of the network architecture, the set of agents (referred as a *botnet*) can equally interact with a single handler or multiple handlers.

2) Reflector Model: This model (Fig. 2b) is similar to the Agent-Handler one, but exhibits an additional set of uninfected machines, called *reflectors*. The reflectors are induced by the handlers to send a stream of packets against the victim. Often, the handlers spoof the victim IP address, in order to solicit the reflectors to send the replies to the victim. This leads to the production of a large amount of network traffic addressed to the target host [1]. The reflectors are often used as amplifiers by sending the stream of packets to the broadcast address[4] of the reflector network and triggering reply packets from each host within their LAN. A Reflector can be any host in the Internet able to respond to IP requests (e.g., a web server that responds to TCP SYN requests) because the attacker does not

[3] *IP source address spoofing* is a cyber-attack which consists in creating an IP packet with a false source IP address, hiding the identity of the real sender or even impersonating another Internet entity.

[4] *Broadcast IP address feature*: when a sending system specifies a broadcast IP address as the destination address, the routers replicate the packet and send it to all the IP addresses within the broadcast address range [2].

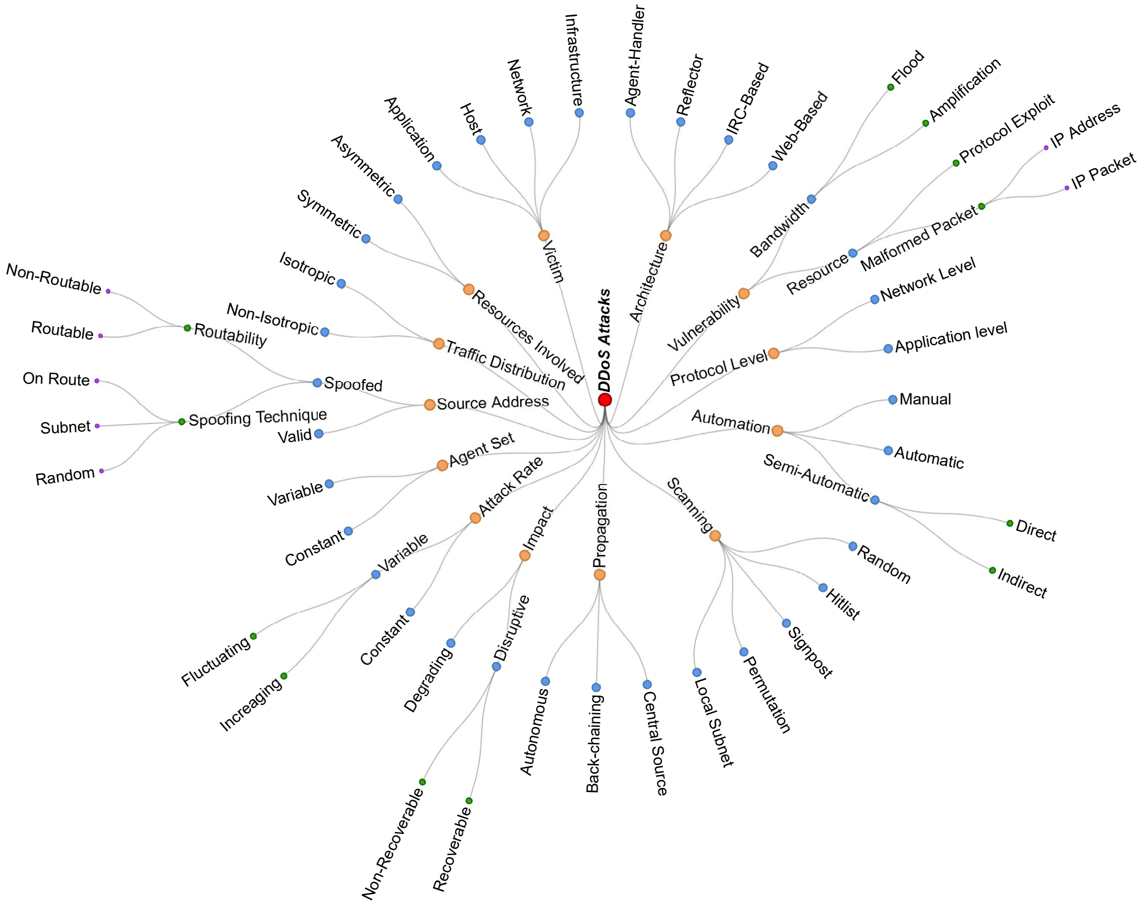


Fig. 1. DDoS Attacks Taxonomy

need to infect it. DDoS attacks that use this model are also known as *Distributed Reflection Denial of Service* (DRDoS) attacks and they are harder to trace back than the ones based on the Agent-Handler Model [4], [5], [15], [16].

3) Internet Relay Chat-Based Model: This model (Fig. 2c) is similar to the Agent-Handler one, with the only difference that the client connects to the agents relying on an *IRC*-based communication channel, instead of the handlers. *Internet Relay Chat* (*IRC*) is a client/server textual protocol, used to implement a multi-user and multi-channel chat system.

4) Web-Based Model: This model is similar to the IRC-Based one, but here the communication is HTTP/HTTPS based. Moreover, the majority of the agents are fully configured and controlled through complex PHP scripts and encrypted communications, while a number of agents is used only to report statistics to a controlling Web site [9].

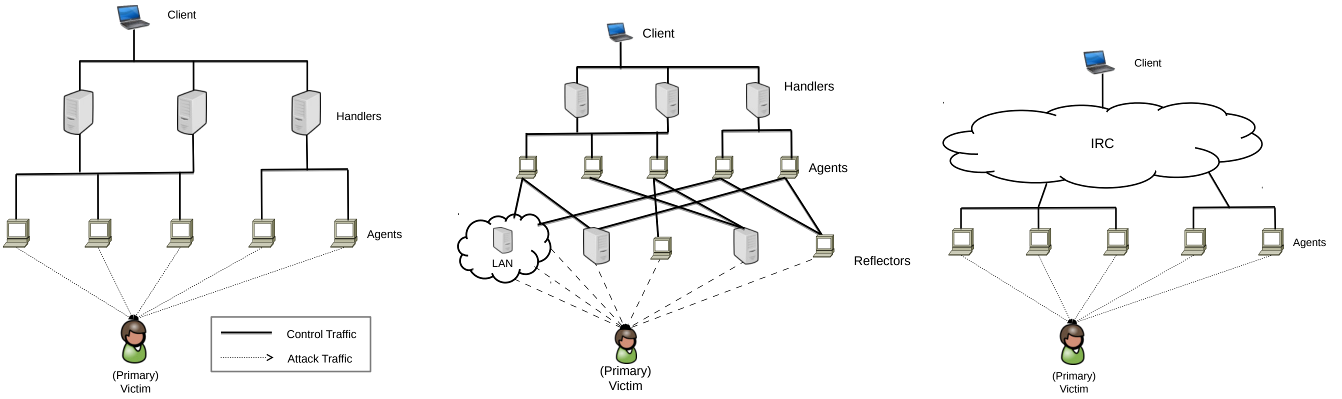
B. Exploited Vulnerability

DDoS attacks can exploit different vulnerabilities to jeopardize their victims. Based on the strategy that is used to deny services, it is possible to classify them in two different categories [1]–[4], [7], [10], [13]: *Bandwidth Depletion* (or *Brute-Force*) and *Resource Depletion*.

1) Bandwidth Depletion (or Brute-Force): In this type of attacks, a great amount of apparently legitimate packets are sent to the victim, in order to clog up its communication resources (e.g., network bandwidth) and also its computational ones (e.g., CPU time, memory, etc.) preventing them to be reached by legitimate traffic. These attacks can be further divided into *Flood* and *Amplification* attacks [1], [2], [5], [6], [10], [13]. In Flood attacks, the botnet directly sends a large volume of IP traffic to the victim machine to congest its network resources and prevent access by legitimate users, while in Amplification attacks the agents use intermediaries reflectors (Section III-A), exploiting the *broadcast IP address feature* with the spoofed address of the victim.

Flood attacks are the most used ones because they are easy to achieve, yet very effective; well-known examples are SYN Flood and UDP Flood attacks. On the other hand, DNS Amplification is a highly popular type of Amplification attack: based on the principle that tiny DNS requests generate much bigger reply packets, a whole botnet can impersonate the target, spoofing its IP address, and send a high number of requests in its stead. As expected, the target will be hit by a massive quantity of replies and experience a DoS event.

Another emerging DDoS attack, exhibited recently by Mirai,



(a) Agent-Handler Model (b) Reflector Model (c) IRC-Based Model

Fig. 2. Architecture Models examples

is the so-called Valve Source Engine (VSE) Flood, which is a particular type of UDP Amplification attack that targets gaming servers by sending them specific requests (TSource Engine Query) from many different devices.

2) Resource Depletion: These attacks aim to preventing the victim to process legitimate requests, by exhausting its resources, and can be further characterized in *Protocol Exploit* and *Malformed Packet* attacks [1], [2], [5], [6], [10], [13]. In Protocol Exploit attacks, an implementation bug of a protocol or a specific feature installed on the victim are exploited in order to consume its resources, whereas in Malformed Packet attacks incorrectly formed IP packets are sent from the agents to the target (e.g., putting the same IP address into both source and destination fields).

An interesting example of Malformed Packet attack is the so-called TCP XMAS. This type of attack consists into manipulating some packets by turning on all the flags (especially URG, PUSH and FIN flags). It is very unusual and totally unexpected that this combination of flags appears into a standard packet, and a lot of time and effort is required, in order to process it, which can eventually crash the target system.

C. Protocol Level

DDoS attacks can be distinguished according to the TCP/IP layer of the protocol used during the attack [9], [17]: *Network Level* and *Application Level*. In Network Level DDoS attacks, either Network or Transport layer protocols are used to carry out the attack, while in Application Level DDoS attacks the victim resources (e.g., CPU, memory, disk/database, etc.) are exhausted targeting Application layer protocols. Clear examples of Network Level attacks are SYN Flood, UDP Flood and TCP Flood attacks, whereas HTTP Flood, DNS Query Flood and DNS Amplification attacks belong to Application Level group of attacks.

An interesting example of an Application Level attack is the DNS Water Torture, which is a DDoS attack that targets specifically Authoritative DNS servers, which are indirectly disrupted by sending a huge quantity of random queries to Open Resolvers, queries that are forwarded to Cache DNS servers and, finally, to the Authoritative DNS servers. Even though the intended target is the latter, as a side-effect also Cache DNS servers face huge slow-downs in their operations.

D. Degree of Automation

Based on the Degree of Automation, DDoS attacks can be classified into three different categories [1], [3], [5]: *Manual*, *Semi-automatic* and *Automatic*.

1) Manual: In Manual DDoS attacks, the attacker individually scans remote devices looking for any vulnerability. Once a vulnerability is found, the attacker manually breaks into the machine, installs attack code and then commands the onset of the attack. Only the early DDoS attacks belong to this category because today all the attack phases are automated.

2) Semi-automatic: In Semi-automatic DDoS attacks the recruitment and exploitation & infection of the agents are automated. The only phases which are still manually performed by the attacker are the communication phase (when the attacker instructs the botnet with type, start time, duration and victim of the attack) and the attack phase [18]. Based on the *Communication Mechanism* used between attackers and handlers (see Section III-A), Semi-automatic DDoS attacks can be done by *Direct Communication* (if based on the Agent-Handler Model) or by *Indirect Communication* (if based on the IRC-Based Model.

3) Automatic: In these attacks, all the phases are automated (recruitment, exploitation & infection, attack), thus there is no need for communication between attacker and botnet. The start time, type, duration and victim of the attack are preprogrammed in the attack code. This category is the one which offers the minimal exposure for the attacker, since he is only involved in issuing the command that starts the attack.

In both Automatic and Semi-automatic attacks, the recruitment of agent machines is achieved through automatic scanning strategies (Subsection III-E) and propagation techniques (Subsection III-F). Notably, some DDoS attacks can use a mixed approach: for instance, the recruitment and the attack could be automated while the exploitation & infection and the communication could be performed manually.

E. Scanning Strategy

During the recruitment phase, the attacker finds as many vulnerable machines as possible with a network scanning. Based on the scanning strategy, it is possible to classify DDoS attacks into five classes [1], [3]: *Random Scanning*, *Hitlist Scanning*, *Signpost* (or *Topological*) *Scanning*, *Permutation Scanning*, *Local Subnet Scanning*.

1) Random Scanning: With this scanning strategy, each compromised host uses a different seed to probe random addresses in the IP address space. As an example, Mirai utilizes a pure Random Scanning approach, randomly looking for any kind of IoT equipped with default login credentials.

2) Hitlist Scanning: With this scanning strategy, the scanning machine has an external list of possible victims to probe. Once the attacker detects and infects a new vulnerable machine, it forwards a portion of the initial hitlist, in order to have a high propagation speed and no collisions during the scanning.

3) Signpost Scanning: In DDoS attacks with Signpost Scanning, some pieces of information on the compromised machines are used to find new targets. As an example, e-mail worms could exploit information from address books of infected machines, a Web-server based worm could spread by infecting each vulnerable client that access to the server Web page, and so on.

4) Permutation Scanning: With this strategy, there is first a brief Hitlist Scanning from which a small initial population of agents is added to the botnet. Subsequently, all the compromised hosts share a common pseudo-random permutation of the IP address space and each IP address is mapped to an index in this permutation. A machine infected during the initial phase begins scanning through the permutation by using the index computed from its IP address as a starting point. Whenever it finds a machine that has already been infected, it chooses a new random starting point.

5) Local Subnet Scanning: The Local Subnet Scanning can be added to each of the previously described strategies, to include a scan for targets located on the same subnet of the compromised host. This technique allows a single copy of the scanning program to compromise many vulnerable machines behind a firewall.

F. Propagation Mechanism

After the recruitment and the exploitation, the agent machine is infected with the attack code and, based on the mechanism chosen in this phase, it is possible to classify DDoS attacks into three different categories [1], [3]: *Central Source Propagation*, *Back-chaining Propagation* and *Autonomous Propagation*.

1) Central Source Propagation: With this propagation approach, the attack code is stored on a central server (or a set of servers) and downloaded through a file transfer mechanism (e.g. wget or tftp) as soon as a new agent is compromised.

2) Back-chaining Propagation: Back-chaining enables the machine that exploited the system to also inoculate the attack code The infected machine then becomes the source of the next propagation step. This propagation mechanism is more durable then the Central Source one because it does not have a single point of failure.

3) Autonomous Propagation: With this approach there are no extra files downloaded, but the attack instructions are directly injected into the target host during the same exploit phase, reducing the possibility that the attack is discovered [18].

G. Impact on the Victim

Depending on the impact that DDoS attacks have on the victim, it is possible to classify them into two different categories [3], [5]: *Disruptive* and *Degrading*.

1) Disruptive: This type of attacks try to completely deny the victim services to its legitimate users. Nowadays, the majority of attacks belong to this class. Based on the *Possibility of Dynamic Recovery* during or after a disruptive DDoS attack, it is possible to further divide them in *Dynamically Recoverable*, when a victim can automatically restore its services as soon as the attack stops, and *Dynamically Non-Recoverable*, when the victim needs human intervention, such as a reboot or even a reconfiguration [3].

2) Degrading: This type of attacks aim at consuming some portion of the victim resources without causing a total service disruption, in order to remain undetected for an extended amount of time. Nevertheless, the damage inflicted to the victim could be huge: as an example, an attack that affects 30% of the victim resources could lead to a DoS for some customers during high load periods and the average performance of the service would be worse than expected.

H. Attack Rate

The DDoS attack requires each agent to send a stream of packets to the victim. The Attack Rate generated by the botnet makes possible to classify DDoS attacks into two different categories [1], [3]–[6], [19]: *Constant Rate*, *Variable Rate*.

1) Constant Rate: The botnet produces attack packets at a fixed rate, usually at the highest rate possible. The output burst is so powerful that the target resources are filled up very quickly, hence the effects of the attack are quite instant on the victim.

2) Variable Rate: The attack rate of agent machines varies, in order to avoid or delay the detection. According to the *Rate Change Mechanism*, variable rate DDoS attacks can be further divided [19] into *Increasing rate*, where the attack rate is gradually and constantly increased through time, and *Fluctuating rate* where the attack is sporadically relaxed, in order to reduce detection chances [1], [3], [5].

I. Persistence of Agent Set

This classification is based on the set of agents active at any time of a DDoS attack. Based on the persistence of the botnet, it is possible to distinguish two different categories [3]: *Constant Agent Set* and *Variable Agent Set*.

1) Constant Agent Set: All agents into the botnet act in the same way, taken into consideration resource constraints: they all receive the same set of commands and they are all engaged simultaneously during the attack.

2) Variable Agent Set: The available agents are divided into several groups and the attacker engages only one group of agents at a given time. An agent could belong to more than one group and each group could be engaged again after a period of inactivity. As a matter of fact, this entails that the botnet is internally partitioned.

J. Source Address Validity

Source address spoofing plays a critical role in most of DDoS attacks, because it hinders the prosecution of the attacker. Based on the Source Address Validity, it is possible to classify DDoS attacks into [3]: *Spoofed Source Address* and *Valid Source Address*.

1) Spoofed Source Address: This is the most common type of DDoS attack, where source addresses are spoofed without any kind of constraint. Moreover, the *spoofing technique*, that defines how the attacker chooses the spoofed source address, makes possible to further divide this DDoS attacks [3] in:

- *Random Spoofed Source Address*, in which source addresses are completely random 32-bit numbers [20], [21];
- *Subnet Spoofed Source Address*, in which source addresses are chosen within the agent machine subnet;
- *On Route Spoofed Source Address*, in which the address is picked from a machine which is on the route (or in a subnet) between the agent machine and the victim.

Based on the *Address Routability*, spoofed source address DDoS attacks can be further divided in *Routable Source Address* attacks, which spoof routable source addresses by taking over the IP address of another machine, and *Non-Routable Source Address* that spoof non-routable source addresses, which could belong to a reserved set of addresses (such as private IP addresses) or be part of an assigned but unused address space of a network.

2) Valid Source Address: These type of attacks usually require interactive exchanges between botnet and victim, hence a valid source address is needed.

K. Attack Traffic Distribution

The locations used as source of attack packets can be utilized to classify DDoS attacks into two *Attack Traffic Distribution* categories [4], [12]: *Isotropic* and *Non-isotropic*.

1) Isotropic: In Isotropic DDoS attacks, the attacker tries to distribute as much as possible uniformly the origin of its malicious packets.

2) Non-isotropic: In Non-isotropic DDoS attacks, the traffic origin is more aggregated in specific parts of the Internet than in others. It means that the victim receives malicious packets from one or more directions which are partially or totally aggregated and not uniformly distributed in the whole Internet.

L. Resources Involved

Based on the amount of Resources Involved in a DDoS attack, it is possible to classify it into two categories [22]: *Symmetric* and *Asymmetric*.

1) Symmetric: In this case, the resources involved are of the same type and scale as those denied to the victim. For instance, in a Network Flooding Attack the attacker uses the same amount of network bandwidth that the victim is deprived of.

2) Asymmetric: In this case, the resources required by the attacker are different from the resources neglected to the victim, in terms of type and scale (e.g., DNS Amplification Attack).

M. Victim Type

DDoS attacks can be classified according to the Victim Type into four classes [3]: *Application*, *Host*, *Network* and *Infrastructure*.

1) Application: In attacks of this class, one or more features of a specific application on the victim host are targeted, with the aim of preventing legitimate clients to use the application and possibly clogging up host resources.

2) Host: In this class of attacks, the victim machine is completely knocked out by disabling or overloading its communication mechanisms (e.g., network interface or network link). A peculiarity of this type of attacks is that all attack packets have the destination address of the target host.

3) Network: In this case, the incoming bandwidth of a target network is consumed with attack packets whose destination address can be taken from its network address space.

4) Infrastructure: In attacks of this class, the target is any distributed service that is extremely relevant for either the global Internet or a sub-network operations. The peculiarity of these attacks is the simultaneity by which multiple instances of the target service are attacked.

IV. IoT Malwares with DDoS Capabilities

Nowadays, one of the most popular way to deliver such DDoS attacks is to target IoT devices. The choice is easily explained by the high availability of such devices which, as if it was not enough, are poorly protected by manufacturers and poorly maintained by owners. Therefore, in order to understand what problems we are facing and possibly find a general solution, a thorough analysis of the present situation is absolutely mandatory. We want to stress out that this specific topic is inherently an extremely unstable one, with a considerable number of offspring malwares that borrow lines of code from deeply divergent families of malwares. Moreover, source codes have been disclosed only for a portion of the existing malwares and the largest part of these information comes from complex reverse engineering jobs which makes the whole situation even worse, if possible. In this section we focus only on the DDoS capable IoT malwares, which entails that we neglect on purpose some other IoT malwares that have different goals, such as cryptocurrencies mining.

TABLE I
IoT Malware DDoS Capabilities

Malware				DDoS	
Name	Year	Source Code	Agents CPU	Architecture Model	Feasible Attacks
Linux.Hydra	2008	Open Source	MIPS	IRC-Based	SYN Flood, UDP Flood
Psyb0t	2009	Reverse Eng.	MIPS	IRC-Based	SYN Flood, UDP Flood, ICMP Flood
Chuck Norris	2010	Reverse Eng.	MIPS	IRC-Based	SYN Flood, UDP Flood, ACK Flood
Tsunami, Kaiten	2010	Reverse Eng.	MIPS	IRC-Based	SYN Flood, UDP Flood, ACK-PUSH Flood, HTTP Layer 7 Flood, TCP XMAS
Aidra, LightAidra, Zendran	2012	Open Source	MIPS, MIPSEL, ARM, PPC, SuperH	IRC-Based	SYN Flood, ACK Flood
Spike, Dofloo, MrBlack, Wrkatk, Sotdas, AES.DdoS	2014	Reverse Eng.	MIPS, ARM	Agent-Handler	SYN Flood, UDP Flood, ICMP Flood, DNS Query Flood, HTTP Layer 7 Flood
BASHLITE, Lizkebab, Torlus, Gafgyt	2014	Open Source	MIPS, MIPSEL, ARM, PPC, SuperH, SPARC	Agent-Handler	SYN Flood, UDP Flood, ACK Flood
Elknot, BillGates Botnet	2015	Reverse Eng.	MIPS, ARM	Agent-Handler	SYN Flood, UDP Flood, ICMP Flood, DNS Query Flood, DNS Amplification, HTTP Layer 7 Flood, Other TCP Floods
XOR.DdoS	2015	Reverse Eng.	MIPS, ARM, PPC, SuperH	Agent-Handler	SYN Flood, ACK Flood, DNS Query Flood, DNS Amplification, Other TCP Floods
LUABOT	2016	Reverse Eng.	ARM	Agent-Handler	HTTP Layer 7 Flood
Remaiten, KTN-RM	2016	Reverse Eng.	ARM, MIPS, PPC, SuperH	IRC-Based	SYN Flood, UDP Flood, ACK Flood, HTTP Layer 7 Flood
NewAidra, Linux.IRCTelnet	2016	Reverse Eng.	MIPS, ARM, PPC	IRC-Based	SYN Flood, ACK Flood, ACK-PUSH Flood, TCP XMAS, Other TCP Floods
Mirai	2016	Open Source	MIPS, MIPSEL, ARM, PPC, SuperH, SPARC	Agent-Handler	SYN Flood, UDP Flood, ACK Flood, VSE Query Flood, DNS Water Torture, GRE IP Flood, GRE ETH Flood, HTTP Layer 7 Flood

A. *Linux.Hydra*

Progenitor of all the IoT malwares, Linux.Hydra appeared in 2008 as an open source project that specifically aimed to routing devices based on MIPS architecture. The exploitation phase relies on a dictionary attack or, in case that the target device is a D-Link router, on a specific and well-known authentication vulnerability [23]. Once that the device has been infected, it becomes part of an IRC-Based network able to perform only a basic SYN Flood attack. The malware documentation reports that this malware also enables the attacker to strike a UDP Flood attack, but online available sources do not exhibit such capability [24]. All in all, even if it is quite simple, this malware laid the groundwork for all the successive MIPS-aiming malwares.

B. *Psyb0t*

Pretty much similar to Linux.Hydra, this malware appeared on the wild in the early 2009. Compared to its predecessor, Psyb0t is able to perform also UDP and ICMP Flood attacks [23]. It targets the same MIPS architecture (therefore, essentially network appliances) and, even though a direct comparison cannot be performed since the sources have not been disclosed, the two malwares show so many common points that it is safe to assume that Psyb0t is a Linux.Hydra offspring.

C. *Chuck Norris*

As soon as the Psyb0t botnet was taken down by its creator, probably due to a growing interest towards his operations, another competitor came out in 2010. Called Chuck Norris, from a string found into the reverse engineered headers, this malware has a lot of common points with Psyb0t, at a point that it is probably its direct evolution [23]: the available attacks are the same, apart from the lacking of ICMP Flood which is replaced by the capability of carrying out an ACK Flood.

D. *Tsunami/Kaiten*

Last and strongest offspring of Linux.Hydra, Tsunami is a fusion of Kaiten-Tsunami DDoS tool and Chuck Norris. In particular, this malware shares with the latter many traits, such as the same encription key and some CNC IP addresses. Tsunami enables the botnet zombies to carry not only traditional SYN Flood, UDP Flood and ACK-PUSH Flood attacks, but also some more sophisticated ones like HTTP Layer 7 Flood and TCP XMAS attacks. Interestingly, in 2016 this malware was sneaked on purpose into the Linux Mint Official ISO [25], jeopardising a huge quantity of freshly installed OSes.

E. *Aidra/LightAidra/Zendran*

Born around 2012, these three malwares exhibit slight variations of the same source code, small enough to let us group them under the same family. Compared to the aforementioned families, the complexity of these malwares is higher: they are able to compile on a number of different architectures such as MIPS, ARM and PPC, even though the infection method relies upon a simple authentication guessing [26]. The resulting botnet architecture is, once again, IRC-based and the type of deliverable attacks is still restricted to basic attacks like SYN Flood and ACK Flood.

F. *Spike/Dofloo/MrBlack/Wrkatk/Sotdas/AES.DdoS*

After the Linux.Hydra family subsided, a new bunch of malwares appeared in different times around 2014 [27]. Many different malwares (such as Spike, Dofloo, etc.) belong to this family but they are so similar that it is hard to tell one from another. What is clear is that, conversely from all the previous families, the resulting botnet architecture is an Agent-Handler based one. Moreover, a mechanism of persistence has been developed by tampering with the */etc/rc.local* file, aiming to survive a device reboot. Another interesting characteristic is the so-called *SendInfo* thread that tries to derive the computing power of the infected host device [28], thus enabling the CNC server to tune the intensity of DDoS jobs that each bot should perform.

G. BASHLITE/Lizkebab/Torlus/Gafgyt

Another popular malware on the wild in 2014, BASHLITE shares similar characteristics with the Spike malware family. Particularly, the communication protocol is a lightweight version of IRC, but it has been so heavily modified that the resulting botnet architecture is totally non-dependant on IRC servers, therefore this botnet can be considered an Agent-Handler and not an IRC-Based one [29]. The variety of architectures vulnerable to this malware is impressive, as even SPARC devices can be infected. The DDoS attacks are basilar, nothing more than traditional SYN, UDP and ACK Flood attacks.

H. Elknot/BillGates Botnet

This 2015 malware has been mostly used by the chinese DDoS'ers, to such a point that the whole family has been dubbed China ELF [30]. Developed to target for the most part SOHO devices, the vulnerable architectures are MIPS and ARM; the possible DDoS attacks are quite a number, included HTTP Layer 7 Flood and some other TCP Flood attacks. Considering that all the available information are derived from reverse engineering techniques and, in addition, copious mutations of this malware has been created, in this case it is particularly hard to sketch out detailed characteristics.

I. XOR.DDoS

In 2015, during the tide wave of malwares that exploited the Shellshock vulnerability, XOR.DDoS started to silently infect many IoT devices all around the world , even though it did not rely upon the aforementioned vulnerability [31]. Probably another product of the chinese DDoS community, this malware is capable of various attacks like SYN Flood, UDP Flood, DNS Flood and more complex TCP Flood ones. As reported by Akamai [32], in October 2015 this botnet alone has been able to hit one of their customers with a DNS Flood of 30 million queries per second, combined with a SYN Flood attack of 140 Gbps.

J. LUABOT

Spotted in 2016, LUABOT is the first ever malware written in LUA programming language. In particular, the DDoS instruction script is detached from the main routines and this modular characteristic, highly simplified by the choice of LUA, in the first stages prevented researchers from understanding its real purpose [33]. So far, the only payload file that has been identified suggests an HTTP Layer 7 Flood attack, but we don't exclude that some other kind of payload scripts are available for this malware to be run. Much more interestingly, this malware includes a V7 embedded JavaScript engine to bypass DDoS protections offered by some enterprises, such as Cloudfare and Sucuri [34].

K. Remaiten/KTN-RM

Appeared in 2016 alongside the much more famous Mirai, Remaiten merges the main characteristics of two different malwares, namely Tsunami and BASHLITE. In particular, the DDoS attacks are mostly derived from the former malware, whereas the telnet scanning capabilities are borrowed by the latter one [35]; unlike BASHLITE, Remaiten botnet architecture is IRC-Based. Most of the embedded architectures are vulnerable to Remaiten, which is unsurprising, since that nowadays it is a common characteristic for all the IoT malwares to be able to compile on different architectures.

L. NewAidra/Linux.IRCTelnet

NewAidra, also known as Linux.IRCTelnet, is somehow a nasty combination between Aidra root code, Kaiten IRC-based protocol, BASHLITE scanning/injection and Mirai dictionary attack [36]. All the embedded devices based on standard architectures can be infected by this malware and the variety of attacks is large: starting from the standard attacks, the attacker can also choose TCP XMAS and TCP Flood attacks (as an example, URG Flood attack). At the present moment, NewAidra is the strongest Mirai competitor in its worldwide IoT infection crusade.

M. Mirai

Mirai is one of the most predominant malware of the last years. It has been used to perpetrate some of the largest DDoS attacks ever known, included the abuse of the French internet service and hosting provider OVH on 22nd September 2016 [37], [38], the attack to KrebsOnSecurity blog on 30th September 2016 [37], [39], and the takedown of Dyn DNS services on 21st October 2016 [37], [40], [41].

The Mirai worm is designed to infect and control IoT devices (such as home routers, DVRs, CCTV cameras, etc., mainly manufactured by XiongMai Technology) using a dictionary attack based on 62 entries. Once exploited, the devices are reported to a control server in order to be used as part of a large-scale botnet [42]. Afterwards, the botnet can be used to perpetrate several types of DDoS attacks exploiting a wide range of protocols (such as GRE, TCP, UDP, DNS and HTTP).

V. Discussion

By further analysing Table I we can highlight some interesting data. First of all, source codes have been disclosed only for few malwares and most of them have been analyzed through reverse engineering techniques, which entails that part of the available data, such as the relationship between the different families of malwares, is based on incomplete and limited information.

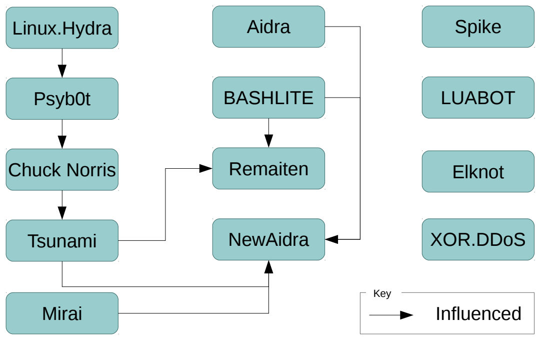


Fig. 3. IoT DDoS Capable Malwares - Correlations

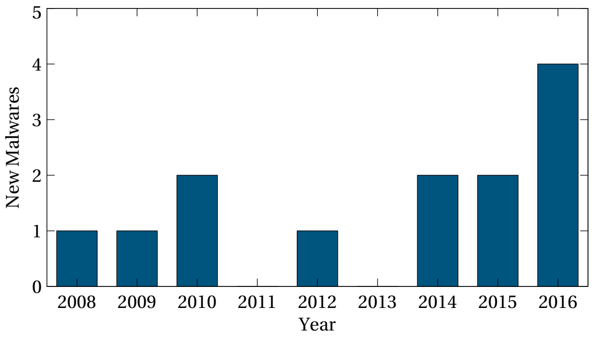


Fig. 4. IoT DDoS Capable Malwares – Year progression, as shown in Table I

Talking about relationships, Figure 3 shows how the different families are supposedly related to each other. Linux.Hydra was the first IoT DDoS capable malware and its source code evolved through the years into 3 different malwares. It seemed that Tsunami would have been Linux.Hydra very last evolution, but part of its code has also been used to develop chunks of Remaiten and even NewAidra, which is one of the most recently appeared malwares. Also, Figure 3 shows that the older malwares were mostly unrelated to each other, whereas in the last years we are witnessing a melting pot of characteristics borrowed from different families, which results into an increased complexity of detection and classification.

Nowadays we can clearly sense the growing in popularity of IoT malwares that exhibit DDoS capabilities. Figure 4 shows the yearly progression of such malwares, as reported in Table I, and clearly confirms this perception. As a matter of fact, it highlights that 4 new families were born in 2016 alone, which is troubling since the previous record was of only 2 new malwares per year (namely in 2010, 2014 and 2015) and before 2008 this category of malwares did not even exist.

Another thing that clearly stands out, is that the oldest malwares were designed to target specific devices that used MIPS processors, whereas the newest ones are able to target a much broader variety of devices and architectures, such as ARM and PPC.

Moreover, looking at the offensive capabilities we can easily see how the most recent malwares are able to hit the targets with much more attacks than the past. As an example, Linux.Hydra was only able to carry out SYN Flood attacks, but Mirai has been armed with refined attacks like GRE IP Flood, GRE ETH Flood and even the so-called DNS Water Torture. Furthermore, almost all the performable DDoS attacks are ascribable into the Flood attacks category, explainable with the enormous quantity of vulnerable IoT devices, which can be easily enslaved with such malwares. As a matter of fact, the Flood attacks require basic programming skills, few lines of code (which is relevant with embedded devices) and very little coordination between the bots.

Last thing, malicious coders take different approaches when it comes to choose the resulting malware botnet architecture. Some malwares build an IRC-based architecture and some others build an Agent-Handler one, therefore we currently cannot highlight a global favourite approach.

VI. Conclusion

The IoT earthquake shook the market and flooded it with a huge amount of poorly secured devices, that were turned by malicious attackers in a potential army, ready to be engaged in highly disruptive activities, mainly DDoS attacks.

Motivated by the increasing number of DDoS attacks that negatively characterize the IoT revolution and by the lack of adequate literature on these attacks in the IoT context, in this paper we have provided an analysis of IoT malwares exposing DDoS capabilities. As a matter of fact, to the best of our knowledge previous surveys about DDoS attacks are dated before the IoT revolution. The analysis is based on an up-to-date comprehensive taxonomy of DDoS attacks based on previous scientific literature and the latest performed attacks to IoT devices. We compared and analyzed the families of malware that characterized the recent years of the IoT-DDoS landscape. The aim of the analysis is to provide a first reference to the scientific community in order to understand all the latest types of DDoS attacks targeting the IoT domain. We believe this study represents a key step in order to raise the awareness of the research community and tackle this security emergency.

References

[1] A. Asosheh and N. Ramezani, “A comprehensive taxonomy of DDoS attacks and defense mechanism applying in a smart classification,” *WSEAS Transactions on Computers*, vol. 7, no. 4, pp. 281–290, 2008. [Online]. Available: https://goo.gl/K3lg7Z

[2] S. M. Specht and R. B. Lee, “Distributed Denial of Service: Taxonomies of attacks, tools, and countermeasures,” in *ISCA PDCS*, 2004, pp. 543–550. [Online]. Available: https://goo.gl/X4gpb7

[3] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, April 2004. [Online]. Available: http://dx.doi.org/10.1145/997150.997156

[4] B. Gupta, R. C. Joshi, and M. Misra, “Defending against Distributed Denial of Service attacks: issues and challenges,” *Information Security Journal: A Global Perspective*, vol. 18, no. 5, pp. 224–247, 2009. [Online]. Available: http://dx.doi.org/10.1080/19393550903317070

[5] C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art,” *Computer Networks*, vol. 44, no. 5, pp. 643–666, April 2004. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2003.10.003

[6] U. Tariq, M. Hong, and K.-s. Lhee, "A comprehensive categorization of DDoS attack and DDoS defense techniques," in *Advanced Data Mining and Applications: Second International Conference*. Springer Berlin Heidelberg, 2006, pp. 1025–1036. [Online]. Available: http://dx.doi.org/10.1007/11811305_112

[7] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying Denial of Service attacks," in *Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. ACM, 2003, pp. 99–110. [Online]. Available: http://dx.doi.org/10.1145/863955.863968

[8] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computing Surveys*, vol. 39, no. 1, p. 3, April 2007. [Online]. Available: http://dx.doi.org/10.1145/1216370.1216373

[9] E. Alomari, S. Manickam, B. Gupta, S. Karuppayah, and R. Alfaris, "Botnet-based Distributed Denial of Service (DDoS) attacks on web servers: classification and art," *arXiv preprint arXiv:1208.0403*, 2012. [Online]. Available: http://dx.doi.org/10.5120/7640-0724

[10] S. Specht and R. Lee, "Taxonomies of Distributed Denial of Service networks, attacks, tools and countermeasures," *Princeton University Technical Report CE-L2003-03*, 2003. [Online]. Available: https://goo.gl/xsZ3n0

[11] RioRey Inc. (2014) Taxonomy of DDoS Attacks. [Online]. Available: https://goo.gl/P2BDq4

[12] K. Kumar, R. C. Joshi, and K. Singh, "An integrated approach for defending against distributed denial-of-service (DDoS) attacks," *IRISS-2006*, pp. 1–6, 2006. [Online]. Available: https://goo.gl/hVfBcr

[13] G. Singn and M. Gupta, "Distributed Denial-of-Service," in *3rd International Conference on Recent Trends in Engineering Science and Management*, April 2016, pp. 1131–1139. [Online]. Available: https://goo.gl/lOvs9Q

[14] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "A Taxonomy of Distributed Denial of Service Attacks," in *Proceedings of the International Conference on Information Society (i-Society'17)*. IEEE, 2017.

[15] V. Paxson, "An analysis of using reflectors for Distributed Denial-of-Service attacks," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, pp. 38–47, July 2001. [Online]. Available: http://dx.doi.org/10.1145/505659.505664

[16] S. Gibson, "DRDoS : Description and analysis of a potent, increasingly prevalent, and worrisome internet attack," *Gibson Research Corporation*, 2002. [Online]. Available: https://goo.gl/zH26gj

[17] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against Distributed Denial of Service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013. [Online]. Available: http://dx.doi.org/10.1109/SURV.2013.031413.00127

[18] K. J. Houle and G. M. Weaver, "Trends in Denial of Service attack technology," CERT Coordination Center, Tech. Rep., 2001. [Online]. Available: https://goo.gl/Py3U0D

[19] X. Luo and R. K. C. Chang, "On a new class of Pulsing Denial-of-Service attacks and the defense," in *NDSS Symposium 2005*, February 2005. [Online]. Available: https://goo.gl/hmkSSF

[20] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for Distributed DoS attack prevention in power-law internets," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. ACM, August 2001, pp. 15–26. [Online]. Available: http://dx.doi.org/10.1145/964723.383061

[21] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "SAVE: Source Address Validity Enforcement protocol," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, June 2002, pp. 1557–1566. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2002.1019407

[22] A. Chen, A. Sriraman, T. Vaidya, Y. Zhang, A. Haeberlen, B. T. Loo, L. T. X. Phan, M. Sherr, C. Shields, and W. Zhou, "Dispersing Asymmetric DDoS Attacks with SplitStack," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, ser. HotNets '16. New York, NY, USA: ACM, 2016, pp. 197–203.

[23] M. Janus, "Heads of the Hydra. Malware for Network Devices," Securelist, 2011. [Online]. Available: https://securelist.com/analysis/publications/36396/heads-of-the-hydra-malware-for-network-devices/

[24] "Hydra irc bot, the 25 minute overview of the kit," Insecurety Research, 2012. [Online]. Available: http://insecurety.net/?p=90

[25] "Warning - linux mint website hacked and isos replaced with backdoored operating system," 2016. [Online]. Available: http://thehackernews.com/2016/02/linux-mint-hack.html

[26] "lightaidra 0x2012 (aidra)," Vierko.org, 2013. [Online]. Available: https://vierko.org/tech/lightaidra-0x2012/

[27] Akamai, "Spike ddos toolkit," Akamai Technologies, Tech. Rep., 2014. [Online]. Available: https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/spike-ddos-toolkit-threat-advisory.pdf

[28] M. J. Bohio, "Analyzing a Backdoor/Bot for the MIPS Platform," SANS Institute, Tech. Rep., 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/malicious/analyzing-backdoor-bot-mips-platform-35902

[29] "MMD-0052-2016 - Overview of "SkidDDoS" ELF++ IRC Botnet," MalwareMustDie! Blog, 2016. [Online]. Available: http://blog.malwaremustdie.org/2016/02/mmd-0052-2016-skidddos-elf-distribution.html

[30] "Linux/AES.DDoS: Router Malware Warning — Reversing an ARM arch ELF," MalwareMustDie! Blog, 2014. [Online]. Available: http://blog.malwaremustdie.org/2014/09/reversing-arm-architecture-elf-elknot.html

[31] "Linux/XOR.DDoS : Fuzzy reversing a new China ELF," MalwareMustDie! Blog, 2014. [Online]. Available: http://blog.malwaremustdie.org/2014/09/mmd-0028-2014-fuzzy-reversing-new-china.html

[32] Akamai, "Case Study: FastDNS Infrastructure battles Xor Botnet," Akamai Technologies, Tech. Rep., 2015. [Online]. Available: https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/fast-dns-xor-botnet-case-study.pdf

[33] "Linux/luabot - iot botnet as service," MalwareMustDie! Blog, 2016. [Online]. Available: http://blog.malwaremustdie.org/2016/09/mmd-0057-2016-new-elf-botnet-linuxluabot.html

[34] NSFOCUS DDoS Defense Research Lab and Threat Response Center (TRC), "2016 q3 report on ddos situation and trends," NSFOCUS, Tech. Rep., 2016. [Online]. Available: http://www.spectrami.com/wp-content/files-mf/1482155162NSFOCUSQ3DDoSThreatReportFINAL.PDF

[35] "Meet Remaiten – a Linux bot on steroids targeting routers and potentially other IoT devices," WeLiveSecurity, 2016. [Online]. Available: https://www.welivesecurity.com/2016/03/30/meet-remaiten-a-linux-bot-on-steroids-targeting-routers-and-potentially-other-iot-devices/

[36] "MMD-0059-2016 - Linux/IRCTelnet (new Aidra) - A DDoS botnet aims IoT w/ IPv6 ready," MalwareMustDie! Blog, 2016. [Online]. Available: http://blog.malwaremustdie.org/2016/10/mmd-0059-2016-linuxircteInet-new-ddos.html

[37] K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets," *arXiv preprint*, February 2017. [Online]. Available: https://arxiv.org/abs/1702.03681

[38] O. Klaba, "OVH suffers 1.1 Tbps DDoS attack," *SC Magazine UK*, September 2016. [Online]. Available: https://goo.gl/IUfDQI

[39] R. Millman, "KrebsOnSecurity hit with record DDoS," KrebsonSecurity Blog, September 2016. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[40] K. York, "Dyn statement on 10/21/2016 DDoS attack," Dyn Blog, October 2016. [Online]. Available: http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/

[41] S. Hilton, "Dyn analysis summary of friday october 21 attack," Dyn Blog, October 2016. [Online]. Available: http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[42] S. Mansfield-Devine, "DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare," *Network Security*, vol. 2016, no. 11, pp. 7–13, November 2016.