WARDOG: Awareness detection watchdog for Botnet infection on the host device

G. Hatzivasilis, O. Soultatos, P. Chatziadam, K. Fysarakis, I. Askoxylakis, S. Ioannidis, G. Alexandris, V. Katos, and G. Spanoudakis

Abstract—Botnets constitute nowadays one of the most dangerous security threats worldwide. High volumes of infected machines are controlled by a malicious entity and perform coordinated cyber-attacks. The problem will become even worse in the era of the Internet of Things (IoT) as the number of insecure devices is going to be exponentially increased. This paper presents WARDOG – an awareness and digital forensic system that informs the end-user of the botnet's infection, exposes the botnet infrastructure, and captures verifiable data that can be utilized in a court of law. The responsible authority gathers all information and automatically generates a unitary documentation for the case. The document contains undisputed forensic information, tracking all involved parties and their role in the attack. The deployed security mechanisms and the overall administration setting ensures non-repudiation of performed actions and enforces accountability. The provided properties are verified through theoretic analysis. In simulated environment, the effectiveness of the proposed solution, in mitigating the botnet operations, is also tested against real attack strategies that have been captured by the FORTHcert honeypots, overcoming state-of-the-art solutions. Moreover, a preliminary version is implemented in real computers and IoT devices, highlighting the low computational/communicational overheads of WARDOG in the field.

Index Terms—computer crime, forensic, intrusion detection, intrusion prevention, network security, security management.

1 Introduction

THe fight against botnets is ongoing for more than a decade now. According to Microsoft, around 1% of all machines that install updates automatically, are found infected with malware [1]. From one month to the next, most of these infected computers are unique which intuitively means that almost 1 in 10 users will experience an infection in the next year.

Several countermeasures have been proposed [2], [3], [4], involving Internet Service Providers (ISPs), industry and governmental organizations, and end-users. The infection rate has been reduced since the beginning of this war, and after 2009 it seems to be relative stable [5].

However, the financial cost and losses are still considered significant for the global economy [6]. Malwares have evolved from disruptive and highly visible at the early 2000s (ILOVEYOU, CODE RED, etc.), to stealthy code that resides undetectable in the victims machine as part of the criminal infrastructure (GAMEOVER ZEUS). Nowadays, cybercrime is organized in a Service Oriented Architecture (SoA) where botnet herders and malware authors can trade their assets in a market of attack-tools and integrated attack strategies [7], [8]. The criminal activity involves, among others, Distributed Denial-of-Service (DDoS) attacks, user credentials harvesting, financial fraud, spamming, hosting of phising sites, click fraud on advertising networks and so on [9].

Normally, the end-users do not bear the full cost of this botnet scourge, with ISPs undertaking the main mitigation efforts with the assistance of national initiatives (e.g. the London Action Plan (LAP) [10] that promotes anti-botnet and anti-spam policies). However, ISPs are not incentivized by the market to mitigate botnets and several ISPs try to avoid this additional cost [11].

Nevertheless, the end-user plays a significant role concerning the overall Internet security. This fact is becoming even more important with the evolution of the Internet of Things (IoT) where high volumes of personal and mobile devices must be protected against "botinization" [12], [13], [14], [15]. Marai botnet is an indicative case [16]. In a short period of six months, the malware had infected around 600,000 IoT devices, such as CCTV cameras with default passwords. Then on October 2016, the botnet performed massive DDoS attacks, overwhelming several high-profile targets and leaving much of the Internet inaccessible on the US east coast [17]. Thus, the more active involvement of the user should be considered as the next step towards a safer and sustainable Internet and the further reduction of the abovementioned infection rate [5].

This paper proposes the WARDOG; an end-user awareness system for botnet mitigation on the infected machine's side. Once a botnet attack is detected by a legitimate and trusted network entity, the involved infected machines are alerted.

As a practical example, we consider the detection of a

- *G. Hatzivasilis, P. Chatziadam, I. Askoxylakis, S. Ioannidis and G. Alexandris are with the Foundation for Research and Technology – Hellas (FORTH), Greece, GR 70013. E-mail: {hatzivas, panosc, asko, sotiris, alexandris}@ics.forth.gr. P. Chatziadam and I. Askoxylakis are also members of FORTHcert.*
- *O. Soultatos and G. Spanoudakis are with the City, University of London, UK, London EC1V 0HB. E-mail: {othonas.soultatos, G.E.Spanoudakis}@city.ac.uk.*
- *K. Fysarakis is with the Sphynx Technology Solutions AG, Switzerland, CH 6300. E-mail: innovation@sphynx.ch.*
- *V. Katos is with the Bournemouth University, UK, Dorset, BH12 5BB. E-mail: vkatos@bournemouth.ac.uk.*

******Please provide a complete mailing address for each author, as this is the address the 10 complimentary reprints of your paper will be sent***

*Please note that all acknowledgments should be placed at the end of the paper, before the bibliography (**note that corresponding authorship is not noted in affiliation box, but in acknowledgment section**).*

DDoS attack by a web server or honeypot [18]. At the current setting, the compromised equipment will continue serving the hacker's commands even after the release of the attack and the exposure of the infected infrastructure. With the WARDOG now in place, the entity that is under the attack can send alert messages back to the machines that transmit the malicious traffic (based on the IP address). WARDOG receives these messages at the device-end and acts as an intrusion mitigation mechanism. It will automatically verify the entity's claim based on locally logged information, block the malicious activity, and advise the user of the botnet's infection. Then, the notified users can authorize the collection of logging data by an anti-virus program. The traces from the various compromised machines are correlated in order to detect the handler-bots at the adjacent layers that forward the attacker's commands and remain hidden during the attack. Digital evidence is concentrated and automatically establishes legal documentation that can be used to prosecute the hackers. The proposed system's key advantages include:

1. **Compatibility:** No modifications are required on the Internet infrastructure. WARDOG is compatible with the routing infrastructure and runs upon intra-domain routing and tunneling mechanisms.
2. **Transparency:** The system is fully transparent to the end-user. It can protect legacy systems without any modifications to the client/server software applications.
3. **Scalability:** WARDOG can protect a high volume of users and services on the global scale with low impact on legitimate entities.
4. **Versatility:** A high variety of malicious activities can be effectively and efficiently mitigated.
5. **Economic incentive:** No further economic impact to the ISPs or the end-users.
6. **Forensic/Accountability:** The overall approach provides accountability and non-repudiation of performed actions (digital signatures, blockchaining, etc.). The forensic documentation is produced from the processed data in an automatic manner.
7. **Cross-border digital investigation:** As malware infection and botnet recruitment can spread all around the world, the proposed solution facilitates the collection of forensic data and the prosecution of the hackers despite the physical location of the victim or the involved machines.

WARDOG develops a cost effective mechanism that tackles the critical factors towards a sustainable information security and forensic computing framework. It has been properly designed to marshal important sustainability aspects such as computational costs, resource usage, scalability, and energy efficiency. The overall solution is able in providing a suitable degree of security and forensic capability, and accomplishes sustainable security tracking and detection, effective machine intelligence to cyber-attacks, efficient information sharing and digital cyber-crime investigation. Moreover, our proposal is suitable for IoT ecosystems and other modern networks, with the overall operation causing no additional costs to the ISPs.

The rest of the paper is organized as: Section 2 refers to background and related work, Section 3 outlines the WARDOG system operation, Section 4 presents the security aspects of the proposed system. The related theoretic analysis is detailed in Section 5 and the provided forensic evidence is presented in Section 6. Section 7 shows the simulation outcomes while a real preliminary version is described in Section 8. Section 9 discusses the overall results and compares them with relevant studies. Finally, Section 10 concludes and refers future work.

2 Background & Related Work

Today, several botnets have been neutralized and analyzed by security experts [19], [20]. The general botnet architecture was firstly revealed via the analysis of the Torbig [19], with other significant efforts including the exposure of Botters [8] and Conficker [20]. Surveys for botnet attacks, attacker tools, and mitigation techniques are detailed in [2], [4], [9], [21], and [22].

2.1 Botnet Infrastructure

A botnet [19], is typically defined as a network of infected end-hosts, called *bots*, which are controlled by one or more persons, known as *bot-master/s*. The botnet recruits vulnerable machines across the Internet utilizing several techniques that are exploited by various classes of malware (e.g. software flaws, social engineering, default system configurations, etc.). The infected machines establish a *Command and Control (C&C)* infrastructure among them, in order to receive instructions from the bot-master and coordinate malicious activities. The main C&C functionality [19], [20]:

1. Facilitates monitoring and recovery by the bot-master
2. Provides robust network connectivity
3. Limits the exposure of the botnet infrastructure that is visible by each distinct bot
4. Supports individual encryption and control traffic dispersion

Thus, the bot-master distributes commands to the bot armies via this C&C mechanism. Normally, the attacker establishes intermediate layers of bots, called *handlers*. Handlers forward the bot-master's commands to other bots that they control directly. The communication finally reaches to the end-bots that actually perform the attack. Thus, the individual's actual location and identity are concealed and the hacker is protected from the law authorities.

The communication channels can operate over various (logical) networks and utilize different communication means. Botnet management involves a series of systems and tools that typically install malicious code and control the victim via the Internet Relay Chat (IRC) [23]. Nonetheless, the hacker can alter the communication approach, with several botnets nowadays supporting more than one protocol in order to incommode their detection (e.g. [8], [20], [24]).

2.2 Attacks

Commonly, botnets are exploited for launching DDoS attacks on computer networks, applications, or the Web in general [4], [9]. The current trend is the performance of DDoS attacks at the application layer [4], [9], [25]. It remains among the most difficult issues to safeguard online, especially in the case of web servers.

The most common strategy includes *HTTP/S flooding* [9] that is originated from the bots to the targeted server. The attack presupposes a high volume of bots that can continuously exhaust the server's bandwidth and therefore prevent legitimate users from gaining access.

As the end-bots that perform the attack do not need to get any response back from the attacked server, they can send requests with spoofed IPs [26]. Each bot attacks the server with various fake IPs. The true IP address is kept hidden from the server and the deployed prevention mechanisms, like black-listed IPs from firewalls or other network monitoring tools, are overcome as the bot keeps changing addresses [27].

Moreover, the bot-master can further hide the end-bots via a layer of reflectors and attack the server indirectly [26]. *Reflectors* are non-compromised systems that exclusively send replies to a request. The bots make requests to the reflectors using as spoofed IP, the IP address of the attacked server. Thus, the reflectors answer back to the server, performing the actual attack.

Except from flooding, *Slowloris* constitutes a state-of-the-art variant of DDoS [25], [28]. The attacker establishes many connections to the targeted server and keeps them open with minimum effort for as long as possible. The attack can be effectively performed with less bots than in flooding. Moreover, the bots consume less resources and this fact increases the possibility of remaining unnoticed by the owner of the compromised machine.

The main difference between botnets and the typical malwares is the existence of the C&C. Thus, if we detect the location of the C&C, the botnet can be tracked and removed. This strategy exploits the possible weaknesses of the communication approaches that applied by the botnet. It is relatively easier to take down a centralized infrastructure. Therefore, as the detection mechanisms become more effective, hackers start moving towards Peer-to-Peer (P2P) and hybrid topologies [29], [30]. This comes with a cost of higher latency as the communication between the bot-master and the bots have to pass through several peers before reaching the end-host that will eventually perform the attack (i.e. HTTP flooding). On the bright side, it offers higher untraceability from the botnet's persecutors [29], [30].

Undoubtedly, botnets can be utilized for a variety of malicious activities [9]. This paper considers all types of malicious botnet activity, however for our purposes, we will concentrate on DDoS techniques and demonstrate the effectiveness of our proposal in mitigating HTTP flooding [9] and Slowloris [25] attacks (Sections 5-7).

2.3 Countermeasures

Three types of botnet countermeasures are identified [4]. The first type prevents the setup of the botnet, blocks the infection of secondary victims and detects/neutralizes the botnet's handlers. The second type deals with ongoing botnet attacks at runtime, including mechanisms that detect, prevent, or mitigate the malicious activity. The third type utilizes forensics technologies that analyze the botnet characteristics, after a launched attack.

The typical techniques for preventing systems from getting infected include anti-viruses/anti-malwares, firewalls, and patching [31]. Thus, malicious code is detected based on signatures, behavior and/or heuristic characteristics [32]. Then, it is quarantined for further analysis or permanent deletion. Fruitful information is also collected, resolving the attacker's tactics. The system's vulnerabilities are exposed and the legitimate software/hardware is updated accordingly. These mechanisms constitute an integral part of the overall defence. Except from protecting single machines or networks their functionality is now extended to the Cloud [33].

However, these techniques cannot always protect the legitimate assets. An anti-virus, for example, can only discovery malicious patterns that are already known. Thereafter, an attacker can examine the scanning capabilities of the protecting mechanism and apply a strategy to avoid detection (i.e. zero-days).

Thus, anomaly detection approaches are suggested [34]. The normal operation of the system is recorded by machine learning components (e.g. based on fingerprinting [35], fuzzy estimators [36], synergetic neural networks [37], or deep learning [38]). When a new type of attack is performed, the abnormal activity is tracked and mitigation policies are applied. So, DDoS attacks can be detected by network monitoring approaches that parse the traffic at runtime [36], [39]. Then, prevention mechanisms, like the Moving Target Detection (MTD) [40], can reduce the attack's side-effects. BotFlex [41] is a state-of-the-art community-driven solution for network monitoring. The raw data of the inspected networking operations, which have been performed by the underlying machines, are transformed in high-level events (e.g. port scan, download form site, or other transactions). An inference engine parses this information and tries to detect symptoms of malicious activities (formed as logic rules). One drawback is the high volume of data that must be processed. Thus, singular value decomposition from the Big Data filed are applicable here [42]. The high-order data dimensions are reduced, even for encrypted data [42], and the computational overhead is significantly reduced.

On the other hand, stealthy DDoS strikes where the attacker combines several different attacks instead of a single and easily identified pattern, can overcome anomaly detection and statistical analysis [43]. Moreover, the legitimate organization must devote sufficient effort in order to deploy and keep up-to-date the defence measures [11].

Apart from these main safeguards at the system level, Internet-wide mechanisms are also developed by the ISPs to marshal the networking activity without the active involvement of the end-users [44]. Although ISPs cannot take responsibility and lock down every customer's infected machine, they can at least ensure that they do not serve traffic that contains malicious packets. The main actions should include [45]:

1. **IP-spoofing:** The provider should not forward traffic with spoofed IP addresses and all packets that contain any RFC 1918 or reserved IP address in the source or destination should be immediately discarded.
2. **Filtering:** *Ingress filtering* should be performed for all the incoming packets to the ISP's network. For traffic that is coming from a customer's site, it should be verified that the NET_ID field in the source IP address matches the assigned NET_ID of this specific customer. *Egress filtering* should be also applied in order to examine the outgoing traffic to upstream and peer ISPs.
3. **Broadcast:** The IP directed broadcasts must be disabled.
4. **High-profile entities:** Careful attention should be paid for high-profile servers and customers.
5. **Dissemination:** The customers could be educated in order to increase the security awareness and protect themselves.

Ordinarily in botnets, the infected machines tend to connect malicious domains or Domain Name System (DNS) that are controlled by the bot-master in order to receive and respond to commands [19], [46]. If these communication patterns to the C&C are identified by the ISP (e.g. router-based TCP/UDP inspection [47], honeypots [48]), the interaction can be repealed (e.g. blocking malicious domains/IPs, routing and DNS blacklist) [46].

Nevertheless, relying on detecting bot communication is not considered viable in the long term [23]. The C&C interaction can be extremely flexible and polymorphic, utilizing encryption or other masking techniques [23].

Forensics are utilized throughout these procedures to gather juridical data. This mainly includes honeypots, computer and network forensics [49], [50]. Yet, the high volume of participating machines/users, the global coverence of bots, and, consequently, the various involved law authorities from different countries, pose great difficulties in the prosecution of the wily hacker [51].

WARDOG concentrates in the last two classes of protection mechanisms (prevention of ongoing attacks and forensics), while contributing in the detection and neutralization of the infected bots and handlers of the first line of defence (botnet's setup and secondary victims). To our knowledge, this is the first attempt that tackles these three aspects in a concrete manner. Once an ongoing attack is identified, the system stops the malicious activity in the host devices. Then, through crowdsourcing, the involved legitimate users can contribute in the collection of related data from their systems that are analyzed by security organizations (i.e. Computer Emergency Response Teams (CERTs) and anti-virus companies). After several iterations, the bot-master can be traced back. Forensic information is automatically gathered throughout this process, resulting in adequate digital evidence that substantiates the malicious activity.

3 WARDOG

WARDOG is an active botnet mitigation mechanism that is applied on the end-users' machines. It is considered a part of the mainstream security software. For example, the WARDOG functionality can be incorporated to a firewall or an anti-virus. Its goal is threefold: i) stop the attack, ii) inform the user that his/hers equipment is compromised, in order to perform a security upgrade and iii) provide adequate digital forensic/evidence that can potentially lead to the bot-master's identification, capture, and conviction.

3.1 Traffic Monitoring at Normal Device Operation

As aforementioned, botnet attacks do not always require to receive response traffic or acknowledgments from the target [26]. Thus, false IP addresses can be casted in order to hide the real bot/node source from the victim (target).

The WARDOG component detects IP-spoofing in the end-user device. The outgoing traffic is filtered. When a packet is sent with an IP address that has not been assigned to this machine (or to any Virtual Machine (VM) that runs in the same system), the incident is recorded, the traffic is blocked and the user is prompted to take further actions, (similarly with the case of receiving a WARDOG alert from an attacked entity as described in the next subsection). Such an ingress filtering method can significantly reduce the IP-spoofing and the indirect DDoS attacks via reflectors [52].

Moreover, to further constrain the bot's capabilities and mitigate Slowloris, WARDOG performs a failed connection (FC) mechanism [53] on the end-user's machine. FC tracks the TCP connections towards a unique IP address (e.g. packets with the TCP RST or TCP SYN flags). If the broken connections go beyond a threshold at a specific time-window, the new connection requests to this IP address are limited.

Except from monitoring, the system also logs the ongoing traffic. When an entity alleges that the machine participates in an attack (e.g. HTTP flooding), the stored information is utilized in order to verify the claim as it is described in the following subsections.

3.2 Under-Attack Functionality

When a network entity is under an attack, like a web server that is hit by a DDoS, it deploys intrusion detection mechanisms that discern the malicious traffic and collect related information regarding the hacker's strategy (e.g. [36], [35], [38], and [40]). However, at this point the victim can only utilize this data mainly for self-performing actions (e.g. discard packets from the suspicious sources) [40], [52].

3.2.1 First iteration – The bots' layer

With WARDOG now installed, as the attacked entity (AE) gathers evidence about the ongoing attack, it can inform, in real time, the directly involved end-user devices (bots) that are actively participating in the malicious effect. The entity provides digital evidence to each machine concerning its claims. For example, part of the logged HTTP traffic including the machine's IP address as a source and the entity's IP address as the destination (see Section 8).

At first, the WARDOG component authenticates the AE (see Section 4 – security mechanisms). Then, it evaluates the provided evidence by examining related log files that have been captured on the machine-end (i.e. network traffic logs). If the evidence is verified, WARDOG filters and blocks the outgoing traffic to the entity and prompts the user to take further actions.

The user can choose to erase the constraint and permit future transmissions, denoting that he/she is aware of the transactions and the communication is legitimate (accepting also the responsibility of this action). We expect that the high majority of the users will not unblock the inimical communication until they have fixed the security problem (i.e. anti-virus/anti-malware scan, operating system format, software/hardware upgrade).

Thus, the distributed attack will be automatically stopped once detected, and the bots will be neutralized (see Section 6 – simulation study). The AE will continue sending WARDOG alarms though, if the device keeps sending traffic during the ongoing attack. This will be repeated at most three times per case, to reserve AE's resources and prevent attackers from exploiting the alerting procedure for their benefit. Only a small amount of the overall bots that are directly administrated by the attackers are anticipated to remain active in the bots' network. Consequently, it is then feasible for ISPs and network forensics to detect this small amount of devices and provide adequate evidence in order to accuse their owners in the court of law (e.g. [2], [3], [4]). However, we consider that this will not be the usual case, as bots are mostly owned by legitimate users and not by the bot-masters.

3.2.2 Second iteration – The handlers' layer

After exposing the end-bots layer of the malicious infrastructure at the first phase, we move forward in detecting the bot handlers. The compromised machines that participated in the direct attack (end-bots) can further contribute in the forensic efforts.

Except from informing the user for the infection, WARDOG requests from the user to give his/hers explicit consent in order to transmit further information to a trusted cyber-security organization (i.e. the anti-virus company), which will analyze that attack-related data. We expect that a sufficient number of users will permit this interaction. Thus, log files are collected from various bots. The goal is to discover common communication patterns, investigate them further, and disclose the commands that where sent by the handlers. After excluding common traffic from legitimate services, the security experts concentrate in the malicious data in order to isolate the IP addresses of the handler machines. The process can be performed in an automated manner, similarly with the general network monitoring forensic approach [21], [50].

As with the first phase, the WARDOG component of the machine will receive an alarm from the anti-virus server (AVS) informing the user that he/she is part of the handlers' layer of the botnet. The digital evidence is a blockchain of the logs that were collected from the controlled bots. It contains traffic patterns with the machine's IP address as a source and each bot's IP address as the destination. The WARDOG verifies the claim, blocks communication to the bot IPs, and prompts the user. As with the first bot-neutralization phase, we consider that the legitimate users that own the handlers will also cooperate and perform the same actions.

3.2.3 Further iterations – The rest C&C infrastructure and the bot-master

The second phase is then iterated several times. The idea is to continue discovering the C&C traffic in the various nested or P2P botnet layers until we find the malicious equipment and reach as close as we can to the bot-master.

4 Security Protocols

Three security protocols are established in order to implement the above mentioned interaction between the various entities:

1. The AE detects an ongoing attack and informs the involved machines/end-bots.
2. The user allows the local WARDOG component of the affected machine to gather data and distribute them to the correlated AVS.
3. The AVS analyzes the collected local logs from the underlying machines and tracks their handlers. Then, AVS updates these machines regarding the infection. Protocol 2 can be repeated afterwards.

The combination of 1 and 2 is performed once for each end-bot, when the attack is launched. Then, protocols 3 and 2 are executed for several iterations as long as there exist contributing users on the various adjacent botnet layers.

4.1 AE to end-bot communication

At first, the end-bots start attacking the AE. The entity detects the malicious activity and records each suspicious IP (*sIP*). AE exports relevant *sub-logs* that contain the involved traffic patterns from each *sIP* and sends the evidence to the machines. Fig. 1 depicts the exchanged messages between the AE, the machine and the user.

AE initiates the WARDOG interaction by distributing its digital certificate. It contains information about the AE and its public key (AE_{PU}).

Then, the entity creates an incident ID (*IncID*) for this specific attack. The *IncID* is formed by the entities identifier (e.g. name, URL, IP address, etc.), the date, and a unique index that is generated randomly.

The evidence for the attack contains the *IncID*, the relevant *sIP* and *sub-log*, and a random nonce ($nonce_1$). The evidence's digest is signed with AE's private key (AE_{PR}). The result is sent to the corresponding bot.

The WARDOG component on each end-user machine begins with the verification of AE's certificate and the extraction of the AE_{PU}. Then, it validates the integrity of the transmitted evidence. If the message has not been replayed (based on $nonce_1$) or altered (digest check), WARDOG examines the *sub-log*'s events. If the same events have been also recorded in the local logs (*local-log*), the data is validated.

The communication with the AE is blocked automatically. The user is informed about the infection and can retain the restriction or unblock the interaction.

4.2 Infected machine to AVS communication

After the verification of the infection, the WARDOG component prompts the user to send the *local-log* to the AVS for further processing. If the user grants his/hers explicit permission, the machine establishes a secure channel with the AVS, based on SSL similarly with the update programming approaches that are commonly supported, where each software distribution comes with pinned asymmetric keys and the public key of the communicating server [54], [55], [56], [57], [58]. The result is the session key *SK* that encrypts the subsequent messages. Fig. 2 illustrates the communication protocol.

The machine sends a message (encrypted with *SK*) that contains the AE's message (including the *IncID* and evidence for the attack), the *local-log*, and a nonce ($nonce_1$). AVS decrypts the message with the *SK* and retrieves the data. It can (optionally) re-verify AE's evidence. An entry is created for the *IncID* in the local data base, if there is none yet, and the *local-log* is stored.

Henceforth, AVS can correlate the communicating patterns from several infected machines for this specific event and detect the handlers in the adjacent botnet layer. The outcome is structured as a set of blockchains for every identified handler. For each initial message *M* a chain is created that contains the trace from the AE to the specified machine. The related digital certificate for the AE is also included. The evidence is sent to the handler machine, as it is described in the next subsection.

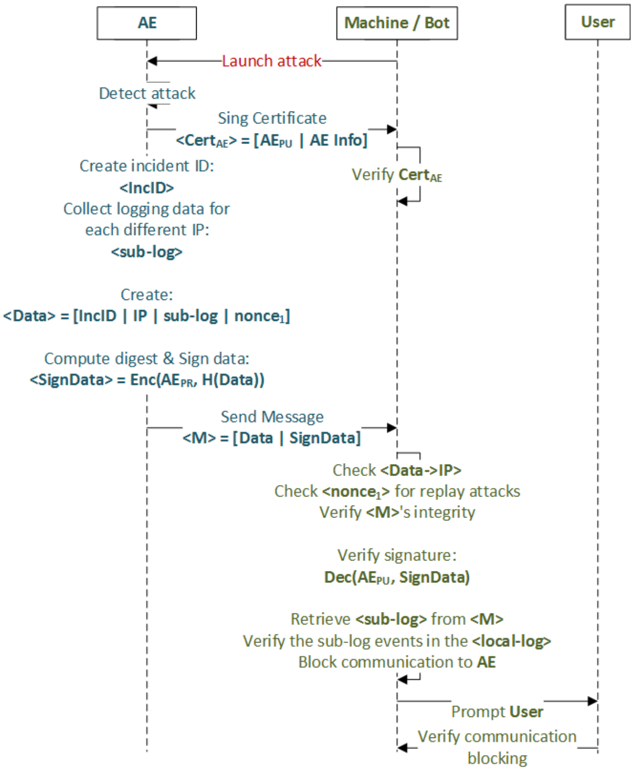

AE
Machine / Bot
User
Launch attack
Detect attack
Sing Certificate
<CertAE> = [AEPU | AE Info]
Verify CertAE
Create incident ID:
<IncID>
Collect logging data for each different IP:
<sub-log>
Create:
<Data> = [IncID | IP | sub-log | nonce1]
Compute digest & Sign data:
<SignData> = Enc(AEPR, H(Data))
Send Message
<M> = [Data | SignData]
Check <Data->IP>
Check <nonce1> for replay attacks
Verify <M>'s integrity
Verify signature:
Dec(AEPU, SignData)
Retrieve <sub-log> from <M>
Verify the sub-log events in the <local-log>
Block communication to AE
Prompt User
Verify communication blocking

Fig. 1. Communication between AE and end-bot.

4.3 AVS to handler communication

The AVS initiates the communication with the handler by establishing an SSL connection, as in the previous communication protocol. Then, AVS encrypts the evidence blockchain and transmits it to the machine.

The handler extracts the AE's public key and verifies the initial claims for the attack. For each message *M*, the machine can track back the logged activity. The final node of each chain includes the participation of the handler. This is contrasted with the *local-log*. If the malicious activity is verified, the user is informed accordingly as in the rest of the cases.

5 Theoretical Analysis

This section details the theoretical analysis of our proposal and its effectiveness in countering the attacker models that are detailed in the simulation study (Section 7).

5.1 Protocol Analysis

The theoretic security analysis of the communication links between the involved entities and the WARDOG component is modelled in the verification tool ProVerif [59] (the code is not included in this document due to the page limit). It is a widely-used automatic symbolic protocol verifier that proves the security properties of the examined protocol, like authentication, secrecy, and adversary equivalence aspects. The examined protocol is modelled in a process calculus and is automatically translated in Horn clauses [59]. The tool resolves these clauses and determines if the security properties hold or not. In case where all properties are validated, ProVerif returns "true". Otherwise, it outputs the properties that could not be satisfied.

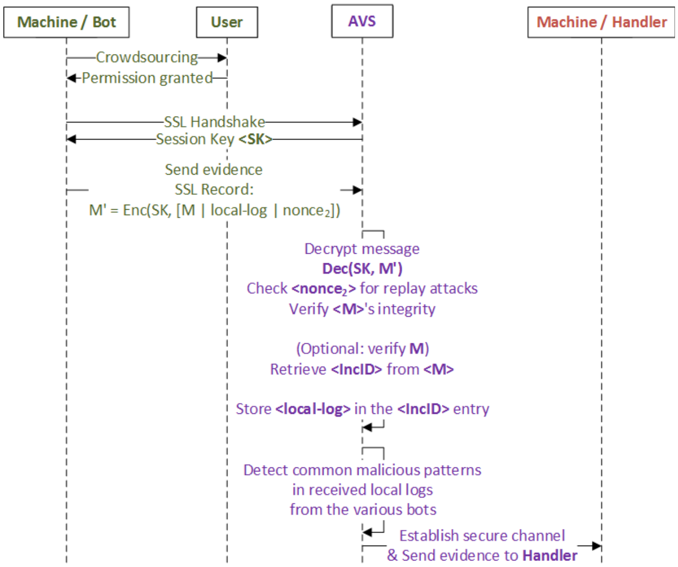

Machine / Bot
User
AVS
Machine / Handler
Crowdsourcing
Permission granted
SSL Handshake
Session Key <SK>
Send evidence
SSL Record:
M' = Enc(SK, [M | local-log | nonce2])
Decrypt message
Dec(SK, M')
Check <nonce2> for replay attacks
Verify <M>'s integrity
(Optional: verify M)
Retrieve <IncID> from <M>
Store <local-log> in the <IncID> entry
Detect common malicious patterns in received local logs from the various bots
Establish secure channel & Send evidence to Handler

Fig. 2. Communication between the infected machine and AVS.

5.1.1 Attack-notification by AE

The initial communication between the AE and the end-bot machines provides only one-way authentication without secrecy. No prior knowledge is required, the AE does not receive any response back and the exchanged data is already known to the attacker. This approach enables a lightweight and fast reaction of the entity that is under the botnet's attack and its computational/communicational capabilities are strained (see Section 8 – Implementation).

ProVerif validates that the interaction is safe, achieving the authentication of the sender, integrity, and immunity to replay attacks. Alternatively, a secure channel could be set up in order to accomplish confidentiality (as it is used for the rest interactions that are described below).

5.1.2 Interaction between AVS and the infected machines

The interaction between the anti-virus server and the end-user machines imposes two-way authentication. At first, both parties must authenticate each other. The process is similar with the *SSL handshake* phase [60]. The result also includes a session key that is randomly generated by the second participant and is securely transmitted to the first one that initiates the interaction (see subsections 4.2 and 4.3). Then, the two parties use this key to encrypt the exchanged data, as with the *SSL record* phase [60].

ProVerif evaluates the various protocol steps and validates that the overall setting provides, authentication, confidentiality, integrity, and immunity to replay attacks.

5.1.3 Blockchaining

Finally, the blockchaining enforces non-repudiation of the main performed actions. ProVerif validates the blocks' integrity, authenticity, and privacy, and consequently the security of the implemented chains [61]. The chaining approach offers the required authorship and accountability of each contributing participant, either for the attacked entity that informs about the ongoing incidents or the involved end-user machines that distribute their forensic data.

5.2 Botnet Mitigation

The theoretical security analysis for the WARDOG's effectiveness in mitigating the botnet's operations is performed in three steps. First, we prove that packet loss due to an attack is bounded (*Theorem 1*). Then, we argue that WARDOG's success is determined by the end-user's compliance with the AE's call to block the communication (*Lemma 1* and *2*). Finally, we evince that WARDOG can still mitigate the attack even without the full compliance of the end-users (*Theorem 2*).

We start by providing some definitions and then the theorems and the proofs.

Definition 1: Let pkt^+ be the total number of successfully transmitted packets.

Definition 2: Let pkt^- be the total number of lost packets.

Definition 3: Let *Tpkt* be the total number of transmitted packets, determined as the summation of pkt^+ and pkt^-.

Definition 4: Let ρ be the transmission success rate of the totally transmitted packets *Tpkt*.

5.2.1 Bounded packet loss due to the attack

Theorem 1: The ideal network exhibits $pkt^- - \rho \cdot pkt^+ \leq 0$. For up to an additive constant, ignoring a bounded number φ of packets lost, it holds that the number of lost packets is a ρ-fraction of the number of transmitted packets. Specifically, there exists an upper bound φ, as described in (1).

$$pkt^- - \rho \cdot pkt^+ \leq \varphi \qquad (1)$$

Proof: Assume that there are *N* nodes, *m* of which are malicious and *m<N*. Let *MIPs* be the set of IPs that are controlled by the malicious nodes (bots).

Let β be the number of served packets that exposes an IP as suspicious or malicious when a DDoS attack is detected. The number of convictions c_e is at least $\frac{pkt^+}{\beta/\rho}$, where e is a single *MIP*. Thus,

$$\sum_{e \epsilon MIPs} c_e - \frac{pkt^+}{\beta/\rho} < 0 \qquad (2)$$

Similarly, the number of rehabilitation operations r_e is at most pkt^-/β. Thus,

$$\frac{pkt^-}{\beta} - \sum_{e \epsilon MIPs} r_e < 0 \qquad (3)$$

Therefore,

$$\frac{pkt^-}{\beta} - \frac{pkt^+}{\beta/\rho} \leq \sum_{e \epsilon MIPs} (r_e - c_e) \qquad (4)$$

By combining (1) and (4), we derive:

$$pkt^- - \rho \cdot pkt^+ \leq \beta \sum_{e \epsilon MIPs} (r_e - c_e) \leq \beta \cdot MIPs \qquad (5)$$

Since $\beta = b \cdot m$, where b is the number of malicious packets that are served per window, (4) becomes:

$$pkt^- - \rho \cdot pkt^+ \leq b \cdot m \cdot MIPs \qquad (6)$$

Therefore, the amount of disruption an attacker can cause to the network is bounded. If there are no malicious nodes (6) describes the ideal case, where $pkt^- - \rho \cdot pkt^+ \leq 0$. ■

5.2.2 Correlation between the user's compliance and the disruptive capabilities of WARDOG

Lemma 1: The WARDOG proof alerting can decrease the attack rate.

Proof: Based on (6), it is derived that *b_up≥b_wd*, where *m_up* and *m_wd* are the number of malicious nodes per window with no protecting mechanism in place and with the WARDOG proof alerting, respectively. ■

Lemma 2: The mitigation rate of WARDOG is directly affected by the end-users compliance with the AE's request to block the malicious traffic.

Proof: The probability of blocking (p_{BL}) an end-bot from continuing the attack is described in (7):

$$0 \leq (p_{BL} = p_{US} \cdot p_{UC}) \leq 1 \qquad (7)$$

Where p_{US} is the probability of unspoofed IP addresses and, thus, the verification of AE's evidence, and p_{UC} is the probability of user's compliance with the AE's request. We assume that the ingress filtering mechanism of WARDOG will prevent spoofing at the device (p_{US}=*1*). Then, from (6) and *Lemma 1*, we can derive that the higher the user's compliance (p_{UC}), the lower the attacker's bounded effect.

As attacking nodes exceed the malicious threshold (mal_{thr}) during the congested period of the DDoS, they are detected and excluded ($\rho \cdot pkt^+ \geq mal_{thr} \rightarrow (m = m - 1)$). The attack rate is further decreased as *MIPs* is decreased. If all attackers are detected *MIPs* becomes *0*, resulting also the ideal case. ■

5.2.3 Attacker's effort and level of user compliance

Theorem 2: The WARDOG does not require the absolute compliance of end-users in order to effectively mitigate a DDoS attack.

Proof: Consider that pkt^{max} is the maximum volume of packets that can be sent be a node in *N*. The upper bound for the current traffic (*CT*) to the AE is at most:

$$CT = LT + MT \leq N \cdot pkt^{max} \qquad (8)$$

Where *CT* is the summation of the legitimate (*LT*) and malicious (*MT*) traffic, respectively.

The malicious effect starts as the *CT* exceeds the AE's bandwidth (*B*), as shown in (9):

$$CT = LT + MT > B \qquad (9)$$

Then, the attack is detected by the AE's IDS and the WARDOG informs the user's to stop this activity. The positive effect of the defence is described in (10), as the blocked malicious traffic (BMT):

$$BMT \leq pkt^{max} \cdot \sum_{e \epsilon MIPs} p_{BL,e} \qquad (10)$$

Henceforth, from (8) and (10), we derive:

$$CT = LT + MT - BMT \qquad (11)$$

With the full compliance of the end-users ($p_{BL,e}$=1 for all $e \in MIPs$), all detected bots stop the malicious traffic to the AE (*BMT=MT*). *CT* is now containing only legitimate traffic which is normally less than *B*. Thus, no packet loss is caused due to the attack.

Even with a lower level of compliance, the attack can still be countered. In order to initiate the malicious effect and make the AE not serving requests, the botnet must fill up the remaining bandwidth ($MT > B - LT$). Thus, there is a volume of bots that can remain active (not blocked by the end-user) while no side-effect is being noticed by the AE's legitimate users ($MT \leq B - LT$).

Nevertheless, even when this ratio is overcome by the botnet, legitimate requests can still be served. The service is degraded but not denied completely. In order to accomplish the absolute denial of the service, the malicious traffic volume must be significantly higher than the remaining traffic ($MT \gg B - LT$). Thus, there can be a tolerance threshold of uncompliant end-users (UC_{thr}) under which service is still provided to legitimate requests. ■

6 Forensics

The section describes the forensics features that are enabled with WARDOG. These are summarized as automatically generated and self-validated documentation, and enhanced cross-border digital investigation.

6.1 Forensics and Automatically Generated Documentation

The supported communication protocols validate the collected information and accomplish non-repudiation for the contributing data (i.e. through digital signatures and blockchaining). The overall security setting confirms the malicious activity and the accountability of a potentially identified attacker.

The AE creates the incident ID (*IncID*) representing the forensic identification for the specific case. The overall forensics evidence that is provided by the WARDOG includes:

1. The digital certificates (X.509) of each involved participant (CA, AVS, AE, end-user machines).
2. The blockchains that were collected by the AVS and entailed the initial evidence for the involvement in the bots' layer. Each chain contains the incident claim, made by the AE, and the related verification of a specific bot.
3. The blockchains that were constructed by the AVS and include the secondary evidence for the involvement in the rest botnet layers. Each chain contains the communication patterns between botnet nodes of adjacent layers, which have been verified by both parties (the bot and its handler).
4. The integration of the two abovementioned evidence sources for each contributing anti-virus organization to a unitary documentation by the law enforcement authorities or other collaborating organizations (i.e. CERTs).

The final document can act as an official record for this cyber-incident and utilized by the law authorities in case hackers or malicious equipment have been traced. Moreover, the document can be distributed between the involved entities (i.e. lawyers, victims, judges), with the full content being validated automatically (offline) through a recursive verifier.

6.2 Cross-Border Digital Investigation

It is almost the norm for botnets to extend their functionality across several countries [48], [49]. However, international cooperation in digital investigations remains a challenging task [51]. The collection of cross-border evidence raises many issues including data authorization, different legislation and investigation capabilities, and which authority has the command for the process.

It is common practice for hackers to attack the target that is deployed in one country through bots that lay in another country with which there is a conflict (i.e. attacking a web server in USA with bots that are located in China). With WARDOG, the security organizations can gather the appropriate information via crowdsourcing. The end-users authorize the distribution of timely and adequate data, assisting both in the mitigation of the threat and the prosecution of the liable persons [62]. To our knowledge, this is a unique feature of the proposed solution that is not handled properly by the current forensic solutions in such a systematic and automatic manner.

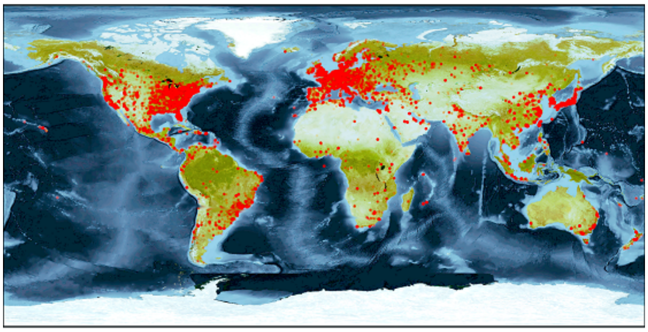


Fig. 3. Simulated DDoS attack with 50,000 end-bots. The red dots represent the infected IPs.

7 Simulation Study

The largest reported DDoS attack was recorded against GitHub in 2018 [63]. The attack launched from over a thousand of different autonomous systems across tens of thousands of unique endpoints. It exhausted the victim's memory resources that peaked at 1.35Tbps via 126.9 million of packets per second. The hacker exploited thousands of misconfigured Memcached servers, many of which are still vulnerable over the Internet and can be utilized again for more massive hits.

This section presents the simulation results for WARDOG and its effectiveness in mitigating the large scale malicious activity of a botnet. In order to simulate our proposal under a realistic attack environment, we further analyze the malicious traffic patterns (e.g. worm, SYN floods, etc.) that have been captured by a system of distributed honeypots that was run by the FORTHcert [48], [49], since 2014. We establish the various simulation aspects that are detailed below based on these observations and the investigated hacker strategies.

7.1 Attack Mitigation

The simulated botnet launches HTTP flooding (make a vast amount of requests to the victim) [9] attacks. The bots perform HTTP flooding on attacks on a central access-point (i.e. a server).

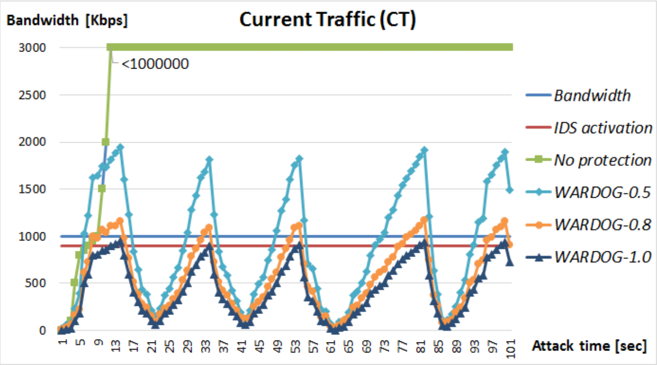


Fig. 4. The Current Traffic (CT) of the various simulated attack scenarios

TABLE 1
SIMULATION PARAMETERS

Parameter	Value
Simulator	BoNeSi, NS3
Visualization	PyGeolpMap, NSNAM
Bots (IP addresses)	50,000
Attack Type	HTTP Flooding
Packet payload	32 bytes
Packets per second	5000
Average bot's bandwidth (*MT*)	22 Kbps
Average botnet's bandwidth	1 Gbps
AE's max bandwidth (*B*)	1 Mbps
Service port	80
Average legitimate traffic (*LT*)	10 Kbps
User's compliance (p_{uc})	1, 0.9, 0.8, or 0.5

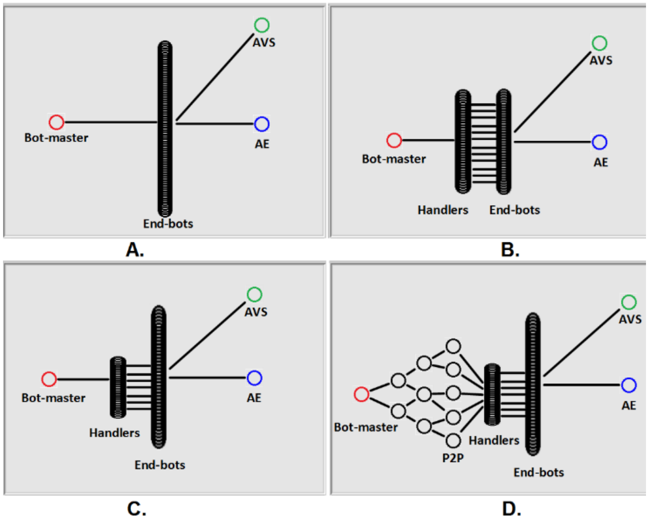


Fig. 5. The four main botnet topologies as modelled in NS3/NSNAM (A. Scenario 1, B. Scenario 2, C. Scenario 3, and D. Scenario 4).

The Botnet Simulator (BoNeSi) [64] is utilized for this study, which emulates a system of *50,000* bots. The simulator generates realistic traffic patterns for TCP and UDP flows, and enables the configuration of several networking aspects, like, the number of IP addresses, the total packets per second and the data volume that are sent to each target. Table 1 summarizes the specific simulation parameters that are applied for this study.

The PyGeolpMap [65] plots the involved IP addresses on a world map, as depicted in Fig. 3. The botnet is spread in almost every country with sufficient Internet infrastructure. The high majority of the infected machines is located in Europe and North America.

The bots perform a SYN flooding DDoS on the legitimate web server through port *80*. The bots send bursts of *1000* packets, with a packet generation frequency of *10*, *30*, or *60* packets per second. The average bots bandwidth is *22* Kbps, with the total botnet's attacking-power exceeding the *1* Gbps on average [66].

Based on the theoretical results, WARDOG's effectiveness is strongly affected by the user's cooperation (p_{uc}). We stress the system's responsiveness to the attack for p_{uc} = *1.0*, *0.8*, and *0.5* level of compliance to the mitigation effort.

Each experiment was performed 10 times and the average values were recorded. Fig. 4 illustrates the evaluation results. When the current traffic (*CT*) exceeds the 90% of the AE's bandwidth (*B=1* Mbps) the potential IDS detects the attacking IPs. Therefore, the threshold for the AE to start the communication with the machines/bots is overcome before the maximum bandwidth is filled up (alternative, the AE could deploy a redundant/alternative communication link for higher resilience). The WARDOG components are informed and block the malicious activity based on the p_{uc} level of each case (i.e. BoNeSi stops sending traffic with these IPs). As time progresses and more-and-more malicious IPs are blocked, the malicious traffic (*MT*) is increased with smaller rates.

As is evidence, for high collaboration degrees ($p_{uc} \geq 0.8$) the malicious activity is circumscribed and the attack ratio is retain below the AE's bandwidth. If the interplay with the end-users is moderate ($p_{uc} \approx 0.8$) the operational quality is degraded, but legitimate users can still be served by the AE. The attack can be successful for low volume of contributing users ($p_{uc} < 0.5$). Nevertheless, the gather forensic data could trace back part of the malicious infrastructure afterwards.

In this simulation study, we evaluate the effectiveness of WARDOG for different levels of user compliance, ranging from 50%-100%. As we mentioned before, we consider that even if the users do not want to participate in the crowdsourcing functionality, they will block the potential attack with high probability (to retain their own resources). Thus, we estimate that a compliance value of $0.8 < p_{uc} < 0.9$ will represent the normal state in real incidents. The overall discussion regarding the user's compliance also reflects the efficacy of the WARDOG approach in case of partial adoption level by the general audience.

7.2 Botnet's Exposure

Apart from being part of the first line of defence, the proposed system can track the nested botnet layers. We investigate the capabilities of WARDOG to dig out the bot-master in an emulated environment with the Network Simulator 3 (NS3) and the NS Network Animator (NSNAM) [67].

A botnet of *50,000* bots is modelled again. This time we consider that the user's compliance ratio is fixed at $p_{uc} = 0.9$ and we vary the botnet's topology. We track the trace originated from a single bot-master towards the end-bots. Four main scenarios are investigated:

1. **Scenario 1:** the bot's are controlled directly by the bot-master.
2. **Scenario 2:** a layer of handlers is intercepted with each handler commanding one end-bot (1-1 relation).
3. **Scenario 3:** each handler controls a small percentage of the overall end-bots (*10%-30%*).
4. **Scenario 4:** a P2P network of handlers is added between the bot-master and the handlers of the previous case.

Fig. 5 depicts the examined topologies.

We perform a single experiment for each of the two first cases, as the outcome was deterministic. In the third scenario, we execute *10* iterations where the control rate for each handler was assigned randomly (between 10%-30%). Finally, we investigate *3* P2P topologies with *10*

TABLE 2
END-USER DEVICES

Feature	PC	Smart phone	Embedded device
Model	Lenovo	Samsung Galaxy	BeagleBone
Operating system	64-bit Windows 8.1 Pro	32-bit Android 6.0.1 KNOX 2.6	32-bit Ubuntu Linux 3.16
Anti-virus	ClamWin	CyberGod	ClamAV
Cryptography	OpenSSL	Bouncy Castle	OpenSSL
Programming language	C++	Java	C++
CPU model	Intel Core i7	Krait 400	AM3359 ARM Cortex-A8
Frequency	2.1 GHz	2.5 GHz	500-720 MHz
Cores	6	4	1
RAM	8GB	16GB	256MB
Internet connectivity	Ethernet, WiFi	4G, WiFi	Ethernet, USB-WiFi

peers each for: i) 1-1 relations among the peers, ii) binary tree connection, and iii) links with higher granularity. For each setting, we perform *10* experiments for random control rates at the handler layer, similarly with the previous synthesis. The average results are reported below.

The direct control of the bots is the worst case scenario for the attacker (*scenario 1*). The derived evidence is strong as his/hers digital footprint is testified by all infected machines.

The best concealing strategy for the hacker can be achieved with a 1-1 analogy of handlers and end-bots (*scenario 2*). However, the adversary must pay a great cost of reducing the total attacking force in the half. WARDOG stops the attack at the end-bots but cannot spot the handlers at this point. Nevertheless, the collected forensic data can be kept by the AVS and reveal these unique handlers once there are utilized again in another incident. Moreover, the mitigation mechanisms that try to figure out the communication channels of the botnet are applicable here (e.g. [35], [38]). Such mechanisms analyze the collected data and discover the various interaction options of the botnet. This information could then be provided to the WARDOG components in order to expose the handler machines.

The two last tested scenarios would be the most common cases in real attacks. The handlers that command the end-bots are easily spotted even for smaller controlling ratios (lower than *10%*). This always leads to the botmaster's detection for the *scenario 3*, with quite strong evidence. In the final set, the P2P links are exposed as long as there is not a 1-1 relation among the peers and the attacker is also convicted (*scenarios 4.ii* and *4.iii*). Otherwise (*scenario 4.i*), the forensic procedure is similar as in *scenario 2*.

TABLE 3
RESOURCE CONSUMPTION

Parameter	Memory usage (KB)	Processing time (ms)
AE operation		
Attack alerting message	40.0 / - /-	10.2 / - /-
User device		
Local traffic logging and filtering	35.0 / 35.0 /36.0	34.0 / 40.0 /41.0
One-way authentication	7.0 / 8.0 / 8.6	0.3 / 0.2 / 2.8
Mutual authentication & session establishment	11.5 / 20.0 / 65.0	10.7 / 2.1 /94.7
Verification of bot participation	0.5 / 0.5 /0.5	0.2 / 0.6 /1.0
Verification of handler participation	1.5 / 1.5 / 1.5	0.5 / 1.5 /2.2
AVS contribution		
Discovery of handlers from received local logs	80.0 / - /-	20.3 / - /-

Each parameter describes the resource consumption for [PC] / [Smart phone] / [Embedded device].

8 IMPLEMENTATION AND PERFORMANCE EVALUATION

A testbed is evaluated on real machines. The research cloud platform GRNET Virtual MAchines (ViMA)[1] is utilized for the main server functionality. Two VMs are installed (Intel Core i7 at 2.1 GHz CPU, 8GB RAM, 64-bit OS Windows 8.1 Pro) that represent a simple web server as the AE and the set of three AVS services that collect the forensic data, respectively.

As mentioned in the introductory sections, the high volume of the IoT devices can be exploited for massive botnet attacks (e.g. [14], [15]). Novel security mechanisms have to be scalable and deal with the heterogeneity of the IoT ecosystem. Thus, we justify the feasibility of WARDOG by applying it in three different types of end-user devices (as identified in our previous work in the IoT domain): i) a laptop that controls a smart campus [68], ii) a smart phone that acts as the infotainment system of a smart vehicle in a Vehicular Ad hoc Network (VANET) setting [69], iii) and a BeagleBone embedded device that gathers environmental parameters in a Wireless Sensor Network (WSN) for precision agriculture [70]. The devices' features are detailed in Table 2. Each of these devices participates in an emulated botnet that performs a DDoS attack in the targeted web server.

As a case study, we develop the WARDOG operation in three different open source anti-virus, one for each one of the three deployed devices. For the laptop (power node with sufficient computational/communicational capabilities), we extend the functionality of ClamWin[2] – a variant for Windows of the most widely-used open source anti-

[1] GRNET ViMA: https://vima.grnet.gr/about/info/en/
[2] ClamWin: www.clamwin.com

virus for research purposes ClamAV[3]. For the smart phone (mobile personal device with moderate efficiency), we install the Android application of CyberGod[4]. For the BeagleBone (embedded device with constrained resources), we deploy the core ClamAV that runs on Linux. ClamWin/ClamAV are implemented in C++ and CyberGod is implemented in Java. For cryptographic operations (e.g. digital signatures, file digests, etc.), the ClamWin/ClamAV utilize the OpenSSL [60], while in the case of CyberGod we have to import the cryptographic library Bouncy Castle [71].

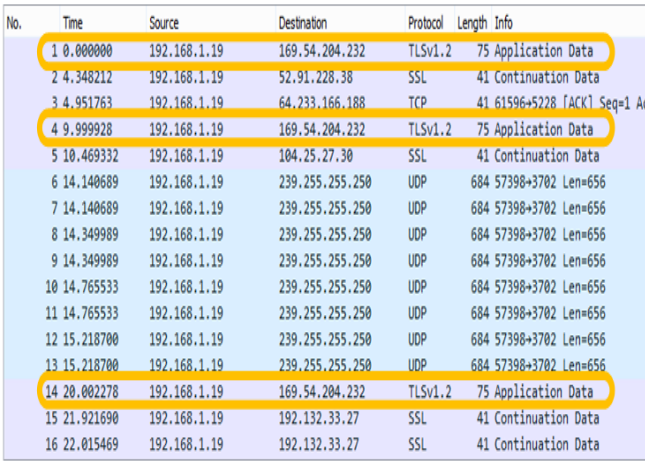
The system's network communication is binded in the port 9090 for each machine. Benchmarks were performed consisting of dummy traffic patterns that trigger several times the WARDOG functionality in each entity. Table 3 summarizes the average resource consumption in the three platforms (the transmission time has been excluded as it is strongly determined by the potential application environment).

The AE only sends the initial alerting messages. The cost is one message per bot, which corresponds to 40KB of RAM and 10.2 ms CPU time. The AE is supposed to have already in place logging mechanisms that would perform similarly with the reported WARDOG logging and filtering components.

At the client side, WARDOG captures the network traffic, creating a log file that is compatible with the widely-used PCAP format[5] (each PCAP entry in the log represents one transaction and requires around 40 bytes to be stored). Fig. 6 illustrates an example trace from a local log file for the WiFi traffic of the evaluated laptop. The trace highlights the transmission of data from the laptop to the AE. The AE provides as evidence its own version of the trace, capturing the laptop's interaction. The WARDOG installation at the device-end, correlates the two tracks and if the evidence is verified the communication is blocked. The local traffic logging and filtering parameter represents the runtime overhead of WARDOG during normal operation. As is evident, the additional effort is low (around 35-36KB RAM for 34-41 ms processing) and feasible even for embedded devices. When an attack is launched, the authentication and validation procedures are also efficient and adequate for operation in real-time settings.

The authentication is the same for the AVS (PC measurements). The main contribution of the AVS is the analysis of the received local logs from the infected machines and the induction of the handler IPs. The processing time is the most important aspect. The achieved delay is decent for our study as it enables the system to send evidence to the handler devices that can be verified in a factual period of time (before the machine changes the current IP address).

In the case of IoT devices with minimal or absent interface with the user, the device will block its participation in the attack and raise an alarm to its controller or gateway. These intermediate equipment should then inform the user afterwards. If the device's resources are constrained, the full WARDOG functionality could be also performed by these equipment (e.g. network logging).



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.19	169.54.204.232	TLSv1.2	75	Application Data
2	4.348212	192.168.1.19	52.91.228.38	SSL	41	Continuation Data
3	4.951763	192.168.1.19	64.233.166.188	TCP	41	61596→5228 [ACK] Seq=1 A
4	9.999928	192.168.1.19	169.54.204.232	TLSv1.2	75	Application Data
5	10.469332	192.168.1.19	104.25.27.30	SSL	41	Continuation Data
6	14.140689	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
7	14.140689	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
8	14.349989	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
9	14.349989	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
10	14.765533	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
11	14.765533	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
12	15.218700	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
13	15.218700	192.168.1.19	239.255.255.250	UDP	684	57398→3702 Len=656
14	20.002278	192.168.1.19	169.54.204.232	TLSv1.2	75	Application Data
15	21.921690	192.168.1.19	192.132.33.27	SSL	41	Continuation Data
16	22.015469	192.168.1.19	192.132.33.27	SSL	41	Continuation Data

Fig. 6. Example trace visualized with WireShark.

9 Discussion and Comparison with Other Systems

As aforementioned in the introductory sections, the current protection approaches target specific aspects of the overall defence against botnets. Their interplay is not always administrated and some vulnerabilities can lead to security breaches. WARDOG acts as an additional safeguard that can interoperate with these solutions and shield the legitimate operation. Moreover, the enhanced forensic capabilities permit the cross-border investigation and the conviction of the bot-masters.

The active involvement of the end-user's machine that is utilized by our system, is mainly proposed by offensive protection mechanisms. For example, the system in [72] proposes that in case of DDoS, the legitimate users should 'speak-up' and increase their request volume. Thus, the legitimate traffic could exceed the bounded malicious bandwidth, with the AE serving several requests (as it was also discussed in *Theorem 2* $\rightarrow MT \gg B - LT$).

However, the AE is still receiving a vast amount of traffic. Thus, an extension is proposed in order to perform the decision making closer to the edge network and permit a constrain number of the speak-up traffic to reach the target [73]. Yet, the countermeasure is effective only if a decent volume of legitimate users are interacting with the AE during the attack (otherwise the LT will not overcome the MT).

Nevertheless, the criticism against such offensive techniques is the fact that they can be exploited for malicious operations as well. An attacker can manipulate the underlying procedures in order to perform an attack. Consider an IoT setting where the user accesses Internet services via his/hers smart phone. The activation of [72], [73] would increase the requests that are made by the device, and consequently, the economic charge and the energy dissipation. WARDOG overcomes these problems, implementing secure and lightweight procedures that cannot be used for attacks.

Other defence approaches suggest the utilization of a pricing method, where the interacting entities will devote some effort in order to designate their legitimacy. The Completely Automated Public Turing test to tell Computers and Human Apart (CAPTCHA) is the indicated choice [74], [75]. Once a new request is made, the server demands from the clients to solve the CAPTCHA test. The

[3] T. Kojm, 'ClamAV': http://www.clamav.net/

[4] CyberGod: https://www.codeproject.com/Tips/1179918/CyberGod-An-Antivirus-in-Cplusplus-for-Windows-and

[5] Libpcap File Format: https://wiki.wireshark.org/Development/LibpcapFileFormat

TABLE 4
Comparison of Protection Mechanisms

Feature	**WARDOG**	Anti-virus /Anti-malware	ISP countermeasures	Forensic technologies	Offensive systems
Prevention of bot infection	**P**	Y	N	N	N
Block ongoing botnet attacks	**Y**	N	P	N	P
Forensics	**Y**	Y	P	Y	N
Cross-border investigation	**Y**	N	P	P	N
Botnet exposure	**Y**	N	P	P	N
Bot-master conviction	**P**	N	N	P	N

Notations for the offered compliance with the feature's main goal/functionality: yes (Y), no (N), or partial (P)

goal is discriminate if the request has been made by a human or a bot. Yet, designing secure CAPTCHAs that can be managed efficiently by the AE is not a trivial task [76], [77]. If the tests are lightweight, the creation/verification burden for the server is low, but the attacker can also resolve them easier. On the other hand, if the riddles are hard, they would result higher false negatives for the legitimate participants [77]. Also, these methods cause delays not only during the attack but in normal operation as well, and are annoying for most users.

Table 4 summarizes the qualitative comparison results. The WARDOG offers a unique functionality that tackles several important and practical security aspects which are not handled by the current solutions in such a solid and systematic way.

Additionally, the performance of WARDOG, enhanced/simple speak-up, and CAPTCHA mitigation approach is evaluated with qualitative metrics. We model the 4 systems in the simulation setting of Section 7 for mitigating the Scenario 1 botnet (*50,000* end-bots). We deploy the WARDOG-0.8 synthesis (80% of user compliance). For the enhanced/simple speak-up phase, we determine that each legitimate entity will reach its maximum bandwidth (22 Kbps). In the case of CAPTCHA, we consider a secure puzzle method that will not be overwhelmed by the attacker and impose an average 5 sec delay to the user, with no false attempts (this is the optimal option for the scheme, as in a real-world application there will be both false positives and negatives).

We evaluate the defence mechanisms based on three performance metrics. We estimate their mitigation effectiveness (percentage of unique legitimate requests that are processed by the AE) and make observations regarding the overall traffic that reaches the AE and connection establishment latency.

We test the systems for different volume of legitimate nodes which send traffic concurrently. Fig. 7, depicts the main evaluation results regarding the effectiveness of each system, as the average the delivery ratio (the higher, the better). An indicative web site that has an average of 150,000 visitors per month would expect 417 requests per hour, while a typical DDoS attack would conscript several thousands of bots [17]. The WARDOG can successfully stop the malicious traffic and serve all legitimate visits.

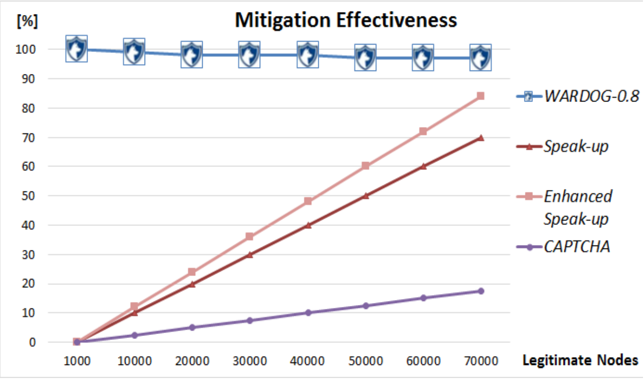


Fig. 7. The mitigation effectiveness of the 4 schemes.

However, this is not the case for the rest schemes. As the volume of the legitimate users is low, only a small number of requests reaches the target after several retries, even in the case of speak-up. The related mechanisms start to be effective only for a quite high volume of legitimate participants (which is not the actual case). Nevertheless, even under these assumptions, the speak-up variants enable the processing for only a small number of links. As the server receives too many requests, both from the bots and the increased number of users, the connection is further delayed and a successful communication takes more time. The speak-up variants increase significantly the traffic burden on the network for low profit. The performance of the CAPTCHA approach was even worse. As long as the legitimate links are fewer, it is harder to be queued in the server. The establishment of a connection takes also much time (as it would be expected) but the confidence on the request's legitimacy is higher. On the other hand, WARDOG does not impose any additional connection establishment delays. The additional traffic load (alerts to the compromise machines) is low and quite gainful as it instantly stops the malicious traffic. Thus, the connected users are served in a timely manner, while in the cases of the rest schemes a successful interaction takes several ms or even secs due to the overall bandwidth allocation.

WARDOG's effectiveness is far more advanced as it blocks the malicious activity instantly once an attack is exposed and permits the processing of user requests even when the volume of the legitimate entities is lower than the malicious ones. On the other side, the effectiveness of alternative schemes, like speak-up or CAPTCHA, is revealed only when the legitimate nodes are more than the compromised ones. However, this is not the ordinary case, as the botnets are composed from several thousands of bots that transmit traffic concurrently, while the users that consume services at the same time are much fewer. Nevertheless, even under an optimal setting for the alternative solutions, WARDOG is computational/communicational more efficient and does not impose any additional connection establishment overhead (i.e. like the CAPTHCA verification delay from the user).

10 Conclusion

The malicious botnet activity and the DDoS attacks are undoubtedly two of the most serious security issues across the Internet that challenge the growth rate and the public acceptance of online businesses and governmental services. In this paper, we propose a novel system, called

WARDOG, which works towards the exposure of the malicious infrastructure with the parallel collection of juridical evidence in order to accuse the liable wily hackers. The main functionality is performed by the infected machine that blocks the malicious activity, once an attack is detected. We prove that the proposed mechanism can mitigate DDoS attacks effectively and efficiently, even with not the absolute compliance from all users. Through crowdsourcing, the involved device owners can then contribute to the further disclosure and neutralization of the botnet infrastructure. A preliminary implementation on real IoT devices exhibits the applicability of our solution in real applications. Finally, WARDOG facilitates an automatic generation of forensic documentation, thus augmenting and enhancing the international cybercrime investigation process. Our proposal offers unique features in comparison with the state-of-the-art systems and can provide runtime protection, mitigating instantly the malicious effects.

ACKNOWLEDGMENT

This work has received funding from the European Union Horizon's 2020 research and innovation programme H2020-DS-SC7-2017, under grant agreement No. 786890 (THREAT-ARREST).

REFERENCES

[1] Microsoft, 'Microsoft security intelligence report,' *Microsoft*, vols. 9-17, 2010-2014.

[2] 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*, IEEE, vol. 15, issue 4, pp. 2046-2069, March 28, 2013.

[3] S. Liu, 'Surviving distributed Denial-of-Service attacks,' *IT Professional*, IEEE, vol. 11, issue 5, pp. 51-53, 29 September, 2009.

[4] S. M. Specht and R. B. Lee, 'Distributed Denial of Service: taxonomies of attacks, tools and countermeasures,' *17th International Conference on Parallel and Distributed Computing Systems (ICPADS)*, San Francisco, CA, USA, pp. 543-550, September 15-17, 2004.

[5] H. Asghari, M. J. G. van Eeten and J. M. Bauer, 'Economics of fighting botnets: lessons from a decade of mitigation,' *IEEE Security & Privacy*, IEEE, vol. 13, issue 5, pp. 16-23, October 28, 2015.

[6] A. K. Sood, S. Zeadally and R. J. Enbody, 'An empirical study of HTTP-based financial botnets,' *IEEE Transactions on Dependable and Secure Computing*, IEEE, vol. 13, issue 2, pp. 236-251, March-April 1, 2016.

[7] T. Moore, R. Clayton and R. Anderson, 'The economics of online crime,' *The Journal of Economic Perspectives*, vol. 23, no. 3, pp. 3-20, 2009.

[8] J. J. C. de Santanna, R. M. van Rijswijk, R. J. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville and A. Pras, 'Booters – an analysis of DDoS-as-a-Service attacks,' *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, IFIP/IEEE, Ottawa, Canada, pp. 243-251, May 11-15, 2015.

[9] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed and S. A. Khayam, 'A taxonomy of botnet behavior, detection, and defense,' *IEEE Communications Surveys & Tutorials*, IEEE, vol. 16, issue 2, pp. 898-924, October 2, 2014.

[10] I. Brown and C. Marsden, 'Co-regulating Internet security: the London Action Plan,' *Academic Symposium on Global Internet Governance Academic Network (GigaNet)*, SSRN, Rio de Janeiro, Brazil, pp. 1-18, November 11, 2007.

[11] A. Garcia and B. Horowitz, 'The potential for underinvestment in Internet security: implications for regulatory policy,' *Journal of Regulatory Economics*, Springer, vol. 31, issue 1, pp. 37-55. February, 2007.

[12] J. Habibi, D. Midi, A. Mudgerikar and E. Bertino, 'Heimdall: mitigating the Internet of Insecure Things,' *IEEE Internet of Things Journal*, IEEE, vol. 4, issue 4, pp. 968-978, May 17, 2017.

[13] Z. Lu, W. Wang and C. Wang, 'On the evolution and impact of mobile botnets in wireless networks,' *IEEE Transactions on Mobile Computing*, IEEE, vol. 15, issue 9, pp. 2304-2316, September 1, 2016.

[14] A. Karim, S. A. A. Shah, R. B. Salleh, M. Arif, R. M. Noor and S. Shamshirband, 'Mobile botnet attacks – an emerging threat: classification, review and open issues,' *KSII Transactions on Internet and Information Systems*, TIIS, vol. 9, no. 4, pp. 1471-1492, April 30, 2015.

[15] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. Mc Daniel and T. L. Porta, 'On cellular botnets: measuring the impact of malicious devices on a cellular network core,' *16th ACM Conference on Computer and Communications Security (CSS)*, ACM, Chicago, Illinois, USA, pp. 223-234, November 9-13, 2009.

[16] M. Antonakakis et al., 'Understanding the Mirai botnet,' *26th Usenix Security Symposium (SS)*, Vancouver, BC, Canada, pp. 1093-1110, August 16-18, 2017.

[17] J. Fruhlinger, 'The Mirai botnet explained: how teen scammers and CCTV cameras almost brought down the Internet,' *CSO Online*, article 3258748, https://www.csoonline.com/article/3258748/security/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html, March 9, 2018.

[18] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi and A. Y. Zomaya, 'An efficient data-driven clustering technique to detect attacks in SCADA systems,' *IEEE Transactions on Information Forensics and Security*, IEEE, vol. 11, issue 5, pp. 893-906, May, 2016.

[19] B. Stone-Gross, M. Cova, B. Gilbert, R. Kemmerer, C. Kruegel and G. Vigna, 'Analysis of a botnet takeover,' *IEEE Security & Privacy*, IEEE, vol. 9, issue 1, pp. 64-72, September 2, 2010.

[20] S. Shin, G. Gu, N. Reddy and C. P. Lee, 'A large-scale empirical study of Conflicker,' *IEEE Transactions on Information Forensics and Security*, IEEE, vol. 7, issue 2, pp. 676-690, April 1, 2012.

[21] A. M. Konovalov, I. V. Kotenko and A. V. Shorov, 'Simulation-based study of botnets and defense mechanisms against them,' *Journal of Computer and Systems Sciences International*, Springer, vol. 52, no. 1, pp. 43-65, January 2013.

[22] Y. Bekeneva, N. Shipilov, K, Borisenko, and A. Shorov, "Simulation of DDoS-attacks and protection mechanisms against them," *IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (ElConRusNW)*, IEEE, St. Petersburg, Russia, pp. 49-55, February 2-4, 2015.

[23] E. Cooke, F. Jahanian and D. Mc Pherson, 'The zombie roundup: understanding, detecting, and disrupting botnets,' *Usenix Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, Cambridge, MA, USA, pp. 1-6, July 7, 2005.

[24] Y. Nadji, R. Perdisci and M. Antonakakis, 'Still beheading Hydras: botnet takedowns then and now,' *IEEE Transactions on De-*

pendable and Secure Computing, IEEE, vol. 14, issue 5, pp. 535-549, October 29, 2015.

[25] Y. G. Dantas, V. Nigam and I. E. Fonseca, 'A selective defense for application layer DDoS attacks,' *Joint Intelligence and Security Informatics Conference (JISIC)*, IEEE, The Hague, Netherlands, pp. 75-82, September 24-26, 2014.

[26] O. Osanaiye, K.-K. R. Choo and M. Dlodlo, 'Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework,' *Journal of Network and Computer Applications*, Elsevier, vol. 67, issue C, pp. 147-165, May, 2016.

[27] C. Basile and A. Lioy, 'Analysis of application-layer filtering policies with application to HTTP,' *IEEE/ACM Transactions on Networking*, IEEE/ACM, vol. 23, issue 1, pp. 28-41, February 2015.

[28] E. Cambiaso, G. Papaleo and M. Aiello, 'Taxonomy of slow DoS attacks to web applications,' *International Conference on Security in Computer Networks and Distributed Systems (SNDS)*, Springer, CCIS, vol. 335, pp. 195-204, Thiruvananthepuram, India, October 11-12, 2012.

[29] P. Wang, B. Aslam and C. C. Zou, 'Peer-to-peer botnets,' *Handbook of Information and Communication Security*, Springer, pp. 335-350, 2010.

[30] P. Wang, S. Sparks and C. C. Zou, 'An advanced hybrid peer-to-peer botnet,' *IEEE Transactions on Dependable and Secure Computing*, IEEE, vol. 7, issue 2, pp. 113-127, July 18, 2008.

[31] D. J. Sanok Jr, 'An analysis of how antivirus methodologies are utilized in protecting computers from malicious code,' *ACM Information Security Curriculum Development (InfoSecCD) Conference*, ACM, Kennesaw, GA, USA, pp. 142-144, September 23-24, 2005.

[32] O. Sukwong, H. Kim and J. Hoe, 'Commercial antivirus software effectiveness: an empirical study,' *Computer*, IEEE, vol. 44, issue 3, pp. 63-70, March 2011.

[33] J. Oberheide, E. Cooke and F. Jahanian, 'CouldAV: N-version antivirus in the network cloud,' *17th Usenix Security Symposium (SS)*, San Jose, CA, USA, pp. 91-106, July 28 – August 01, 2008.

[34] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei and L. T. Yang, 'Internet traffic classification using constrained clustering,' IEEE Transactions on Parallel and Distributed Systems, IEEE, vol. 25, issue 11, Novermber, 2014.

[35] P. Bazydlo, K. Lasota and A. Kozakiewicz, 'Botnet fingerprinting: anomaly detection in SMTP conversations,' *IEEE Security & Privacy*, IEEE, vol. 15, issue 6, pp. 25-32, November 28, 2017.

[36] S. N. Shiaeles, V. Katos, A. S. Karakos and B. K. Papadopoulos, 'Real time DDoS detection using fuzzy estimators,' *Computers & Security*, Elsevier, vol. 31, issue 6, pp. 782-790, September, 2012.

[37] W. Xiong, H. Hu, N. Xiong, L. T. Yang, W.-C. Peng, X. Wang and Z. Qu, 'Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications,' Informatics and Computer Science, Intelligent Systems, Applications: An International Journal, vol. 258, pp. 403-415, February, 2014.

[38] L. F. Maino, A. L. P. Gomez, F. J. G. Clemente, M. G. Perez and G. M. Perez, 'A self-adaptive deep learning-based system for anomaly detection in 5G networks,' *IEEE Access*, Special Section on Cyber-Physical-Social Computing and Networking, IEEE, vol. 6., issue 1, pp. 7700-7712, February 7, 2018.

[39] O. Y. Al-Jarrah, O. Alhussein, P. D. Yoo, S. Muhaidat, K. Taha and K. Kim, 'Data randomization and cluster-based partitioning for botnet intrusion detection,' *IEEE Transactions on Cybernetics*, IEEE, vol. 46, issue 8, pp. 1796-1806, August 2016.

[40] M. Albanese, S. Jajodia and S. Venkatesan, 'Defending from stealthy botnets using moving target defenses,' *IEEE Security & Privacy*, IEEE, vol. 16, issue 1, pp. 92-97, February 6, 2018.

[41] S. Khattak, Z. Ahmed, A. A. Syed and S. A. Khayam, 'BotFlex: a community-driven tool for botnet detection, 'Journal of Network and Computer Applications, Elsevier, vol. 58, pp. 144-154, December, 2015.

[42] J. Feng, L. T. Yang, G. Dai, W. Whang and D. Zou, 'A secure higher-order Lanczos-based orthogonal tensor SVD for Big Data reduction,' IEEE Transactions on Big Data, IEEE, Early Access, pp. 1-14, February 2018.

[43] A. Aqil, A. O.F. Atya, T. Jaeger, S.V. Krishnamurthy, K. Levitt, P. D. Mc Daniel, J. Rowe and A. Swami, 'Detection of stealthy TCP-based DoS attacks,' *Military Communications Conference (MILCON)*, IEEE, Tampa, FL, USA, pp. 348-353, October 26-28, 2015.

[44] S. Newman, 'Service providers: the gatekeepers of Internet security,' *Network Security*, Elsevier, vol. 2017, issue 5, pp. 5-7, May 2017.

[45] B. Rowe, D. Wood, D. Reeves and F. Braun, 'The role of Internet Service Providers in cyber security,' *Institute for Homeland Security Solutions*, IHSS Cyber Reports, Duke University, pp. 1-12, June, 2011.

[46] X. Li, J. Wang and X. Zhang, 'Botnet detection technology based on DNS,' *Future Internet*, MDPI, vol. 9, issue 4, article 55, pp. 1-12, September 25, 2017.

[47] L. Xu, X. Yu, Y. Feng, F. Han, J. Hu and Z. Tari, 'Comparative studies of router-based observation schemes for anomaly detection in TCP/UDP networks,' *IEEE International Conference on Industrial Technology (ICIT)*, IEEE, pp. 1832-1837, Taipei, Taiwan, March 14-17, 2016.

[48] P. Chatziadam, I. G. Askoxylakis and A. Fragkiadakis, 'A network telescope for early warning intrusion detection,' *2nd International Conference on Human Aspects of Information Security, Privacy, and Trust (HCI)*, Springer, Heraklion, Greece, LNCS, vol. 8533, pp. 11-22, June 22-27, 2014.

[49] P. Chatziadam, I. G. Askoxylakis, N. E. Petroulakis and A. G. Fragkiadakis, 'Early warning intrusion detection system,' *7th International Conference on Trust and Trustworthy Computing (TRUST)*, Springer, Heraklion, Greece, LNCS, vol. 8564, pp. 222-223, June 30 – July 2, 2014.

[50] R. Hunt and S. Zeadally, 'Network Forensics: an analysis of techniques, tools, and trends,' *Computer*, IEEE, vol. 45, issue 12, pp. 36-43, July 30, 2012.

[51] J. I. James and P. Gladyshev, 'A survey of international cooperation in digital investigations,' *7th International Conference on Digital Forensics and Cyber Crime (ICDF2C)*, Springer, Seoul, South Korea, LNICST, vol. 157, pp. 103-114, October 6-8, 2015.

[52] P. Du and A. Nakao, 'DDoS defense deployment with network egress and ingress filtering,' *IEEE International Conference on Communications (ICC)*, IEEE, Cape Town, South Africa, pp. 1-6, May 23-27, 2010.

[53] S. Chen and Y. Tang, 'Slowing down Internet worms,' *24th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, Tokyo, Japan, pp. 1-8, March 26, 2004.

[54] M. Nasu, 'Communication device, software update device, software update system, software update method, and program,' *US Patents*, US7555657B2, Ricoh Co Ltd, pp. 1-51, June

30, 2009.

[55] C. R. Wysocki and A. Ward, 'Secure software updates,' *US Patents*, US9489496B2, Apple Inc., pp. 1-21, November 8, 2016.

[56] S. Yu, 'Network identity certificate pinning,' *US Patents*, US9473487B2, Bank of America Corp, pp. 1-16, October 18, 2016.

[57] C. Evans, C. Palmer and R. Sleevi, 'Public key pinning extension for HTTP,' *IETF*, RFC7469, Google Inc., pp. 1-28, April, 2015.

[58] D. Barrera and P. C. van Oorschot, 'Secure software installation on smartphones,' *IEEE Security & Privacy*, IEEE, vol. 9, issue 3, pp. 42-48, May-June, 2011.

[59] B. Blanchet, "Automatic verification of security protocols in the symbolic model: the verifier ProVerif," *Foundations of Security Analysis and Design (FOSAD) VII*, Springer, LNCS, vol. 8604, pp. 54-87, 2014.

[60] I. Ristic, 'OpenSSL cookbook,' *Feisty Duck*, pp. 1-94, March, 2016.

[61] N. Kobeissi and N. Kulatova, 'Ledger design language: designing and deploying formally verified public ledgers,' *Cryptology ePrint Archive*, IACR, report 416, pp. 1-5, May 4, 2018.

[62] K. Taha and P. D. Yoo, 'Using the spanning tree of a criminal network for identifying its leaders,' *IEEE Transactions on Information Forensics and Security*, IEEE, vol. 12, issue 2, pp. 445-453, February, 2017.

[63] M. Kumar, 'Biggest-ever DDoS attack (1.35 Tbs) hits GitHub website,' *The Hacker News*, March 1, 2018.

[64] K. I. Sgouras, A. N. Kyriakidis, and D. P. Labridis, "Short-term risk assessment of botnet attacks on advanced metering infrastructure," *IET Cyber-Physical Systems: Theory & Applications*, IET, vol. 2, issue 3, pp. 143-151, October, 2017.

[65] P. Piegg, J. M. de Oca, F. Lgg, P. Cazenave, A. Haydock, J. Satirom and N. Newky, "PyGeoIpMap," *GitHub*, https://github.com/pieqq/PyGeoIpMap, 2017.

[66] D. Dagon, G. Gu and C. P. Lee, 'A taxonomy of botnet structures,' *Botnet Detection*, Springer, ADIS, vol. 36, Boston, MA, USA, pp. 143-164, December 10-14, 2008.

[67] G. F. Riley and T. R. Henderson, 'The ns-3 network simulator,' *Modeling and Tools for Network Simulation*, Springer, edition 1, pp. 15-34, June 29, 2010.

[68] G. Hatzivasilis, I. Papaefstathiou, D. Plexousakis, C. Manifavas and N. Papadakis, 'AmbISPDM: managing embedded systems in ambient environment and disaster mitigation planning,' *Applied Intelligence*, Springer, vol. 48, issue 6, pp. 1623-1643, August 30, 2017.

[69] K. Fysarakis, G. Hatzivasilis, I. Papaefstathiou and C. Manifavas, 'RtVMF – a secure real-time vehicle management framework with critical incident,' *IEEE Pervasive Computing Magazine (PVC) – Special Issue on Smart Vehicle Spaces*, IEEE, vol. 15, issue 1, pp. 22-30, January 21, 2016.

[70] G. Hatzivasilis, I. Papaefstathiou and C. Manifavas, 'SCOTRES: secure routing for IoT and CPS,' *IEEE Internet of Things Journal (IoT-J)*, IEEE, vol. 4, issue 6, pp. 2129-2141, September 15, 2017.

[71] The legion of the Bouncy Castle, 'Bouncy Castle Crypto APIs,' *GitHub*, https://github.com/bcgit and bouncycastle.org, January 27, 2014.

[72] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger and S. Shenker, 'DDoS defense by offense,' *ACM Transactions on Computer Systems*, ACM, vol. 28, issue 1, article 3, pp. 1-54, March, 2010.

[73] M. Mehta, K. Thapar, G. Oikonomou and J. Mirkovic, 'Combining speak-up with DefCOM for improved DDoS defense,' *International Conference on Communications (ICC)*, IEEE, Beijing, China, pp. 1708-1714, May 19-23, 2008.

[74] A. S. Sairam, S. Roy and S. K. Dwivedi, 'Using CAPTCHA selectively to mitigate HTTP-based attacks,' *Global Communications Conference (GLOBECOM)*, IEEE, San Diego, USA, pp. 1-6, December 6-10, 2015.

[75] K. J. Singh and T. De, 'A novel approach of detection and mitigation of DDoS attack,' International Conference on Computer Science, Data Mining & Mechanical Engineering (ICCDMME), Bangkok, Thailand, pp. 62-67, April 20-21, 2015.

[76] S. Gao, M. Mohamed, N. Saxena and C. Zhang, 'Emerging-image motion CAPTCHAs: vulnerabilities of existing designs, and countermeasures,' *IEEE Transactions on Dependable and Secure Computing*, IEEE, Early Access, pp. 1-14, June, 2017.

[77] S. S. Brown, N. DiBari and S. Bhatia, 'I am 'totally' human: bypassing the reCAPTCHA,' *13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, Jaipur, India, pp. 9-12, December 4-7, 2017.

Dr. G. Hatzivasilis holds a PhD from the ECE department of the Technical University of Crete, Greece. He received his M.Sc. and B.Sc. in Computer Science from the University of Crete. He also received a certificate in Management of Information Systems (MIS) from the National and Kapodistrian University of Athens. His research interests include IoT and CPS systems, lightweight cryptography, trust management, and ambient intelligence.

O. Soultatos is a PhD candidate at the City University of London and researcher at ICS-FORTH. He received his BSc in Computer Science from the University of Crete. He is interested in the design and implementation of IoT and SDN security and privacy solutions.

P. Chatziadam holds a BSc in Computer Science from the City University of New York. He is a key member of FORTHcert where he fulfills the role of Network Security Specialist. Areas of expertise include Intrusion Detection & Prevention, Governance & Audit, Ethical Hacking, System & Network Forensics, Security Architecture & Compliance as well as Incident Handling & Analysis.

Dr. K. Fysarakis holds a PhD from the ECE department of the Technical University of Crete, Greece. He received an MSc in Information Security from Royal Holloway, University of London, is an IRCA certified ISO 27001:2005 auditor and a member of the IEEE. His interests revolve around the security of embedded systems and the challenges that arise with the wider adoption of ubiquitous computing.

Dr. I. Askoxylakis holds a Diploma in Physics from the University of Crete, a Master of Science in Communication Engineering from the Technical University of Munich and a PhD in Engineering from the University of Bristol. He is Head of the FORTHcert. His research interests lie in the fields of system and communication security.

Dr. S. Ioannidis is a Principal Researcher at FORTH-ICS, an Adjunct Professor at the Computer Science Department of the University of Crete and a Marie-Curie Fellow. He received his Doctorate degree from the University of Pennsylvania in 2005. His areas of interest include systems security, privacy and security policy, where he has dozens of publications in international, peer-reviewed, journals and conferences. He has also served as a member of the Technical Program Committee of leading conferences, such as ACM CSS, Usenix Security, etc.

G. Alexandris holds a Diploma in Electrical Engineering from the Technical University of Munich, Germany, and an MSc in Information Networking from Carnegie Mellon University, USA. His research interests include scalable architectures and secure cloud systems integration, where he is currently pursuing a PhD at the Bournemouth University, UK.

Dr. V. Katos obtained a Diploma in Electrical Engineering from Democritus University of Thrace in Greece, an MBA from Keele University in the UK and a PhD in Computer Science (network security and cryptography) from Aston University. He is a certified Computer Hacking Forensic Investigator (CHFI). He has worked in the Industry as Information Security Consultant and served as an expert witness in Information Security for a criminal court in the UK and a misdemeanor court in Greece. His research falls in the area of digital forensics and incident response.

Dr. G. Spanoudakis. Bsc, Msc, PhD. George Spanoudakis is Professor at City, University of London. His research interests are in software systems security and service oriented systems.