Cyber Threats Monitoring: Experimental Analysis of Malware Behavior in Cyberspace

Clara Maria Colombini[1], Antonio Colella[2⋆]
Marco Mattiucci[3], Aniello Castiglione[4]

[1] University of Milan, External Researcher, I-20122 Milano, Italy
cmcolombini@email.it
[2] Italian Army, Via XX Settembre, 123, I-00187, Rome, Italy
antonio.colella@esercito.difesa.it
[3] High Tech Crime Department (RTI), Arma dei Carabinieri
Caserma Palidoro, Viale di Tor di Quinto, 119, I-00191, Rome, Italy
marco.mattiucci@carabinieri.it
[4] Dipartimento di Informatica, Università di Salerno
Via Ponte don Melillo, I-84084, Fisciano (SA), Italy
castiglione@ieee.org

Abstract. Cyberspace is a borderless new universe in which all actors, including States, share information and communications technologies, now indispensable to the modern lifestyle. Since the beginning of the 21st century, the ability to leverage cyberspace has become the most important source of power. Due to the proliferation of ICT systems into all aspects of life, the importance of information for political matters has increased awfully. State and non-State actors can use this power to achieve objectives into cyberspace and physical world. Low cost and high potential impact make cyber-power attractive to all actors. In fact, cyber threats have grown exponentially with the proliferation of the cyberspace infrastructures. Consequently, cyberspace has become a warfighting domain with the potential to destroy or make useless logical, physical, technical, and virtual infrastructure, damaging in fact critical National capabilities.
This scenario forces all national institutions to a review of their defense strategies, because of the difficulties to identify the actors of a cyber-attack. It then becomes necessary to gain a broader view of the problem to acquire more detailed information, useful to identify such sources of cyber-attacks. This new point of view can be achieved by using the analytical method developed by the authors and applied to data streams flowing across the cyberspace. In this way we can collect, detect, isolate and analyze the behavior of those malware that are acting as cyber weapons, through the implementation of an honeypot-based system such as the one presented in this paper.

Keywords: Cyberspace, Digital Profiling, Malware, Cyber Threat, Honeypot, Cyber Weapon, Digital Behavior

⋆ Corresponding author: Antonio Colella, Italian Army, Via XX Settembre,123 I-00187, Rome, Italy, email: antonio.colella@esercito.difesa.it.

1 Introduction

Cyberspace is a unique domain that does not occupy a physical space. It does, however, depend on physical nodes, servers, and terminals that are located in Nations that exert control and sometimes ownership, as described by the definition of the U.S. Department of Defense: "Cyber-space is a global domain within the broader universe of information, and consists of a network of interdependent infrastructures, including telecommunications networks, computer systems, processors, and embedded controllers" [1]. This definition let to discern between the place *cyberspace* and the activities that occur within it. This means that cyberspace, unlike the well-known physical space, has no national boundaries. In fact, while it is possible to isolate or disable one or more parts of a compromised network, its functions and data continue to exist. This unique feature of the cyberspace influences any defensive strategy we want to implement [2]. In such a scenario, cyber threats have grown exponentially. Consequently, cyberspace has become a war-fighting domain with the potential to destroy or make useless logical, physical, and virtual infrastructure, and to damage critical national capabilities [3].

Threats within cyberspace are disparate, diffuse, and some may also be disproportionate in the harm they could cause: this means that the correct description of cyber weapons becomes primarily important to assess, on the one hand the level of threat from cyber-attacks, and on the other hand the most appropriate countermeasures to adopt, for both preventive and defensive purposes. Weapons in general are instruments through which, within a specific context, a person can bring harm to another person or object, or defend themselves from attacks. Attacks made by means of cyber weapons, in the same way as conflicts of conventional type, are designed to cause damage only to a specific opponent, often in a situation of tension or crisis already underway or about to be born, in order to obtain some kind of advantage [4].

In this paper we develop a method based on the extrapolation of the digital behavior from data streams flowing over the Internet [5]. The entire tests are carried out by implementing a set of virtual honeypots, specifically configured with different known vulnerabilities. The purpose is to collect log files, detect malware, and finally isolate those one that are acting as cyber weapons through the application of the obtained information. The purpose of the tests is to obtain valuable information about the actors of a cyber conflict, giving the vision in real-time of possible attack situations and the ability to implement an effective system of cyber defense that is pre-configurable on specific threats that are to be contrasted.

Clearly, the problem of containing large-scale malware and worms over the Internet has been addressed by several works, some of one by using a cooperative distribution of traffic filtering policies [6], others by using automatic security assessment [7]. Moreover, a part of the literature considers as a best practice to adopt audit-based access control [8] or use network anomaly detection methods [9], [10].

Section 2 shows the scenario in which we operate, the cyberspace, its unique characteristics, with respect to the proposed analytical method, while Section 3 describes briefly the method of analysis with its components: the study of the characteristics of cyber weapons, the analysis of their lifetime in cyberspace, the problems of a cyber defense, and the implementation of the filters. Section 4 introduces the experimental tests and Section 5 explains how to construct the profile of an attacker. Finally, in Section 6 are drawn the conclusions.

2 Profiling the Cyberspace

One of the biggest problems of cyber defense is represented by its anonymity with the resulting non-imputability that cyberspace can offer to the responsible of a cyber threat. In that case, it is difficult, if not impossible, to identify the enemy, because many of the challenges of traditional warfare are highlighted and amplified into cyberspace. One of the most important aspect is the challenge of situational awareness, which is defined as "the continuous extraction of environmental information, the integration of this information with previous knowledge to form a coherent mental picture, and the use of that picture in directing further perception and anticipating future event." [11]. It is therefore essential to gain the view that allows to acquire those information. This can be achieved by using the method of analysis proposed in several studies about the Digital Profiling paradigm [12], [13], [14], that gives a more detailed description of a threat in the cyberspace. This approach is based on the assessments made by the behavioral analysis of the cyber threat's main actors. We extrapolate their characteristics, in relation to their lifetime in the cyberspace. This action is made starting from two points of view: the ICT one, which analyzes software properties, and the strategic one, which reveals the strategic/military use of them as real weapons of offense. The union of the two aspects allows us to reveal new additional properties. The analysis of these new properties, together with the old ones, allow us to extrapolate the behavior of a cyber weapon. Therefore, the results of the cyber-profile are composed by a series of information that can be used as "filters" for the monitoring and the analysis of data streams [5], [15], in order to have a more efficient identification of the actors of a cyber conflict. In fact, this type of profile allows a real-time awareness of possible situations of attack and facilitate the implementation of an effective dynamic system of cyber defense.

3 The Method of Analysis

The method to extrapolate the behavior of a cyber weapon consists of the following four steps. The first one is to analyze the properties of a cyber weapon, which provides detailed features. This is followed by the analysis of the timeline of a cyber-attack. It uses the feature resulted from the previous step to extrapolate comprehensive information that help to delineate the behavioral pattern of an attack. The third step is to analyze the cyber defense, considering the

information gathered in the second step. Finally, the fourth step is the implementation of filters for monitoring and analysis of cyber threats, through the profile obtained from all the previous steps.

3.1 Analysis of Properties of a Cyber Weapon

The study of the characteristics of cyber weapons is based on the properties coming from an evaluation performed by using two different points of view. The *ICT* point of view, which describes the malware as any set of computer instructions designed to unlawfully damage a computer system. The cyber weapons are in fact an evolution of malware with all their properties. The *strategic/military* point of view that reveals the impact of cyber-attacks and the expected damages brought at the enemy target. This perspective adds further information about targets, such as critical infrastructures, data or programs contained therein or pertaining thereto, by using the common methods of military strategy.

A cyber weapon is a set of instructions compiled into a programming language, and thus can be disassembled, analyzed and modified. Unlike the common malware that affects either any computer system, without any type of control or advantage, it is specially customized for the characteristics of the systems to hit, with the aim to reach a specific advantage. The program code of the cyber weapons differs for each attack and is able to deal with different form of attacks simultaneously.

The impact of the caused damage is publicly revealed with a lag: similarly to all crimes, the victim is not willing to reveal his vulnerability. Furthermore, source and path are difficult to find, because their authors can take advantage of the anonymity offered by the Internet architecture. A cyber weapon can destroy itself after the attack, leaving no traces in the infected system. Any trace eventually left after the attack can easily be created ad hoc for deceiving any attempt to identify it.

Cyber weapons are often used as part of a larger conventional attack in support of it within a conflict, to gain more advantage over the enemy. They may act at a certain time, remaining "silent" until the right moment for the actions of attack comes, adapting themselves to the state of the systems in which it is introduced, and changing in response to the variables that meets. These properties make them intelligent agents, similar to "fire and forget" weapons [16]. Usually, they have a very short life, just the duration of the attack. Its discovery decrees an immediate reaction to correct any exploited vulnerability. For such a reason, cyber weapons should not be reused at a later time without substantial changes. The implementation of a cyber weapon is a very complex task. Differently from a common malware that can be created and launched by a single individual, it requires a C4ISTAR command & control (C&C) structure [1] such those one present in some advanced botnet architectures [17].

3.2 Analysis of Lifetime of a Cyber Weapon

The above features make possible to describe in detail when a cyber weapon was introduced in the wild. Analyzing the following six steps, we can exploit the actions that characterize the cyber weapon's lifetime.

Target Choice Often the design of a cyber-attack takes place in a strategic way, from the originating motivations to the management of the entire attack. Initially, the choice of the targets is related to the enemy structure and its criticality and is closely linked to the reasons of attack. It can be possible to describe the targets choice with the answer to the following four questions. *Where* is physical location of the target? *What* is the target functions? *Who* are the owner and the users of the target? *Why* such attack is performed? In this respect, it can be determined the type of damage to cause, which can be *digital* with the unauthorized access to confidential data, delay or interruption of service, modification, damage or destruction of a computer code or *physical* with the destruction of the devices and the equipment. In addition, the damage is measured in terms of *severity* of the effects caused by attack [18] and of the *persistence* of the effects that can be permanent, temporary or transient [19].

Acquisition of Information The phase related to the acquisition of information about the chosen target is essential for the construction of the weapon itself, since its ability to effectively hit a specific target is proportional to the nature and quality of the collected information. Such information can be derived mainly from the *intelligence* point of view (e.g., information on choosing the target, its location, any access roads, systems of physical protection, best time for attack, etc.) or from the *technical* point of view (e.g., technical characteristics of the selected target, its vulnerability and protection systems hardware and software).

Source Code Analysis The majority of cyber weapons are specifically built for their purpose: more and more often we find specific cyber weapons for specific targets to hit. This makes a cyber weapon more effective. In fact, when a cyber weapon is discovered, specific defensive countermeasures are taken. This makes it no longer able to act also if the quality of the cyber weapon was high. The code that composes it is implemented by considering the type of intrusion, which can be:

- *direct*: connection to the target system with a device that transmits it (USB mass storage, CDROM, etc.);
- *semi-direct*: sent over the network from a non-critical location;
- *indirect*: sent through cyberspace.

In addition, this "armed code" must implement those properties that distinguish it from a common malware and make it an effective weapon, anonymous and difficult to detect. Namely, an effective implementation of a good cyber weapon having the above characteristics should consider when it have to be

launched (immediately, delayed or repeated) and should adapt itself to the conditions of the targeted system, including a mechanism of self-destruction and the possibility to connect to a C&C server. In addition, no unwanted and uncontrollable collateral damage as well as no traces are left on the attacked system and in the cyberspace.

Simulation and Testing A cyber-attack must succeed at the first attempt, otherwise it can be easily neutralized. Its realization must include, as with any other software, a test phase, before the real attack. Initially it takes place in a virtual environment, in order to test the functionalities of the implemented code, but it then need to be tested into the cyberspace, to correct any eventual error, and especially to adapt it to the changes that may have occurred in the configuration of the security measures taken by the target system. The aim is to gain information on the effectiveness of the penetration methods and on the intended damages, in order to ensure the success of the attack against the real target. At this stage the type of attack is similar to a real one. In fact, the target system is composed of a set of systems similar to the chosen one.

Attacks The most important phase in the timeline analysis is the attack, in which all the prepared actions, tested in previous stages are implemented. The aim is to effectively hit the chosen target and get a response as close as possible to the expected result in the prescribed manner and time, avoiding any unwanted side-effect.

Results Evaluation The last phase consist on evaluating, both in the actual state and in the near future, the success of the attack by comparing the expected results against the real obtained ones. The first step verifies the successful reaching of the intended target, followed by the assessment of the time of the attack, the type, the duration, and costs of caused damage on the target system. Later on, have been also considered the effects of damage on the infected system, the building that houses it, and any impact in the short, medium and long term, such as side effects inside and outside the target system, impact of the attack (military/political/social), and the eventual countermeasures (active/passive). The assessments in the above paragraphs lead to an overall evaluation of the attack in terms of analysis of cost/benefit as well as in term of real gained advantages.

3.3 Cyber Defense Analysis

In order to have a more comprehensive analysis of cyber-attacks, the evaluation of its characteristics from the point of view of the structure responsible for the defense is of fundamental importance. The study of known cyber weapons (from DDoS attacks in Estonia up to the Stuxnet worm) [20], [21], [22], [23], confirmed that the weapon *computer* is mostly often used as part of a larger conventional attack in support of it. This observation leads to the creation of a monitoring system that can be useful to extrapolate those indicators that show the possibility of a cyber-attack on critical infrastructures, through the analysis

of available information from different type of sources. Such sources can be *open*, if publicly available (such as national reports coming from companies producing antivirus, national and international newspapers, websites dealing with political, economical and social analysis), *semi-open* when consist on websites of hackers' groups, antagonists, extremists, fundamentalists, and *closed* when it is part of a strategic/military documentation.

The obtained information should be able to answer to the following seven questions:

- *who*: the identification of possible attackers;
- *why*: the reasons of the attack;
- *where*: the identification of critical infrastructures that are possible targets;
- *how*: the intrusion mode;
- *what*: the damage type;
- *when*: the attack time;
- *results*: the damage extent and possible disadvantage;
- *reaction*: response actions.

3.4 Implementation of Filters to Monitor Data Flow

The information obtained from the above mentioned analysis constitutes a first set of filters applicable in the analysis of data streams to detect those signals that indicate the possibility of a cyber-attack in the near future. The main step consists on detecting the presence of a cyber weapon through the analysis of characteristics of its behavior, which distinguish it from a common malware. The possible identification of targets may be exploited detecting properties in common among different malware. In particular, the possible targets are: limited in number and restricted to a particular type, geographically distributed, with similar processes or critical data, with the same OS, with similar policy and security systems and finally with the same vulnerabilities.

In Table 1 are listed the behavioral characteristics of the malware detected by the analysis that reveals the activity of cyber weapons. Furthermore, the real target (according to the properties of the cyber weapons) undergo the highest number of attacks, is repeatedly attacked in different times, can be identified in a later time upon an intrusion or an attack, reveals stepwise refinements in the malware code and is related to the reasons of tensions/crisis/conflicts/antagonisms, either national or internationa).

The information obtained upon analyzing the content of its source code, can provide the profile of the detected cyber weapon. The indicators that can be extracted from it, can be used as filters for the recognition of a cyber-attack that is in progress or about to be launched. Such filters can be applied to the log files related to the attempts of intrusion into the domain of interest.

4 The Experiments

The main goal of the experiments is to illustrate the application of the method introduced in the previous sections. This will be used to implementing a system

Feature	Meaning
Incomplete code	- developing code
Simultaneous diffusion of the same code in a limited number of objectives	- malware test on a controllable number of objective similar to target - refine tuning of malware code - deception - reduction of target reaction response time
Repeated attacks over time for the same purposes	- code corrections - deception
More attacks on the chosen target	- customize code on actual configuration of real target
No major damage caused as a results of the intrusion or attack	- decrease the possibility of detection by antivirus softwares - reduction of target reaction response time

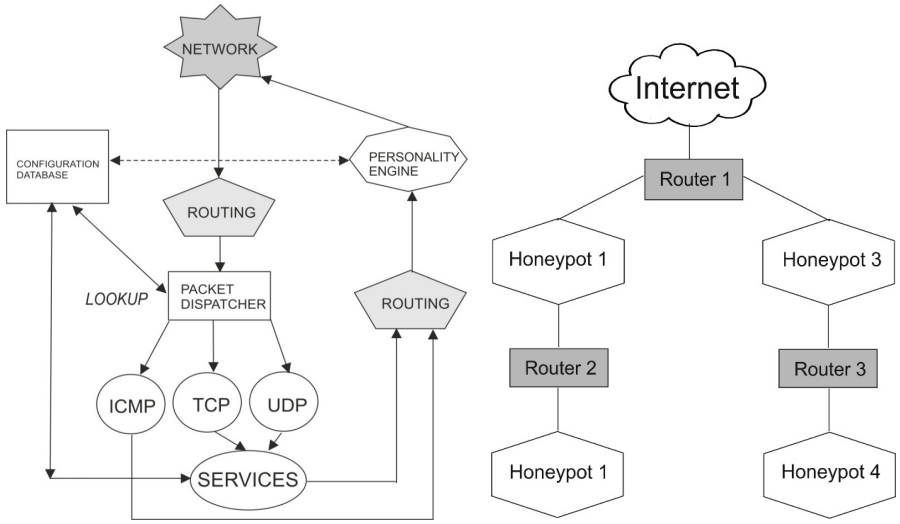
Table 1. The meaning of each detected feature in the malware behavior.

to monitor and analyze data streams flowing through Internet. The experiments were conducted on a small scale only from a technical point of view, applying those filters derived from the information extracted from the analysis of the characteristics of the detected cyber-attacks. To develop the experiments, we implemented a network of *honeypot* (called *honeynet*) through which collect, detect, extract and analyze malicious codes launched against it. A honeypot is a machine connected to a network that emulates system vulnerabilities in order to attract, capture and analyze cyber-attacks. If a connection occurs, it can be, at best an accidental connection or, more likely, an attempt to attack the machine. Briefly, we can classify honeypots firstly into two groups, based on their deployment. The *production honeypots* that are used in a company's internal network to improve the security of the whole network. In addition, the *research honeypots* that are more complex of the production ones, and provide a detailed information about the attacks and are used by research, military and government organizations.
The second criterion classifies honeypots based on their design criteria. The *pure honeypots* are full production systems, so no other software needs to be installed. The *high-interaction honeypots* use non-emulated OSes with multiple services which can be exploited by the attacker. Also, the *low-interaction honeypots* emulate the part of the system and services most frequently used [24].

4.1 Honeypots Implementation

We used the tools contained into the "Mercury Live DVD" [25]. It comprises valuable tools for digital forensics, data recovery, network monitoring, spoofing, reverse engineering, and four different type of honeypots: `Honeyd`, `Nephentes`, `Dionaea`, and `Kippo`. In particular, Honeyd is a low-interaction honeypot that comprises several components (see Figure 1(a)): configuration database, a central packet dispatcher, protocol handlers and a personality engine. Incoming packets

first go through the central packet dispatcher. It is able of dealing with three protocols, TCP, UDP and ICMP. The dispatcher queries the configuration corresponding to the destination address. Then it passes the packet to the protocol-specific handler. On receiving a TCP or UDP packet, the handler manages the connections to different services. The framework checks if a specific packet is part of an already started service application. If so, all packets are redirected to the service, otherwise a new service is started. The handler also helps in connections' redirection. Then the packet is sent to the personality engine which manipulates its content to make it appear similar to the one originated from the network stack. Through Honeyd we implemented a network with three routers and four



(a) Honeyd components (b) the implemented honeynet

Fig. 1. Architectural sketch of the Honeyd components (a) and the scheme of the implemented honeynet (b)

simulated hosts, as in Figure 1(b). The Honeyd implementation also includes two hosts configured with two different versions of Microsoft Windows, one as a server and the other as a client. What follows is an example of configuration.

```
# Windows 2000 Server SP3 WebServer
create windows2000
set windows2000 personality "Microsoft Windows 2000 Server SP3"
add windows2000 tcp port 80 "perl scripts/iis-0.95/iisemul8.pl"
add windows2000 tcp port 139 open
add windows2000 tcp port 137 open
add windows2000 udp port 137 open
add windows2000 udp port 135 open
set windows2000 default tcp action reset
set windows2000 default udp action reset
```

To improve the reality of the implemented honeynet, Honeyd allows to simulate all the standard devices connected to a network, such as "Cisco" routers as shown in the following example:

```
# Cisco Router
create routerCisco
set routerCisco personality "Cisco IOS 11.3 - 12.0(11)"
set routerCisco default tcp action reset
set routerCisco default udp action reset
add routerCisco tcp port 23 "/usr/bin/perl scripts/router-telnet.pl"
set routerCisco uid 32767 gid 32767
set routerCisco uptime 1327650
```

All configurations are contained in a simple text file (`name.conf`) that must be read by the program, and according to which all details of the simulated network are created. This is in fact a sort of *false digital profile* offered to the attackers, to increase the realism of the honeypots. This concept is similar to the one of the *false digital alibi* in which it is shown how simple is to setup false digital evidence on different systems (such as Mac OS X [26], on Android devices [27], and different flavor of the Microsoft Windows OSes [28], [29], [30]) in order to claim a false alibi to be used in several scenarios. In the case of the honeynet, since the attackers often try to remotely fingerprint OSes by using tools as `nmap` or `X probe`, Honeyd takes the same fingerprint database used by `nmap` to spoof the responses of any OS it is emulating by providing false evidence about the running OS.

In order to present simple but effective experimental results, we focuses our attention on the study of the behavior of malicious attacks performed against the SSH service. Also, the experiments let to inspect the activities performed by the attackers once they gain access to the system and try to progress in their intrusion [31], configuring the machine to record the password along with the account name that was used in the login attempt [32], [33]. In order to better analyze the behavior of the attacks, we implemented two identical honeypots, into two different subnets, with two different SSH user account configurations, in order to obtain two different profiles of the same attack to compare. In the first one (see Figure 2(a)) there exist 8 user accounts and their relatives passwords composed by very common words, in order to offer a high level of vulnerability. On the contrary, the second one ((see Figure 2(b))) also contains the same 8 user accounts, but with 8 complex passwords, composed by letters, digits, and special symbols, to resemble to a more protected system.

Here we present the results of the analysis of captured data in the two honeypots during 30 days, focusing in particular on the log files containing the authentication requests to the SSH server: date, time, the IP address from which the login attempt originated, the result of the request (failure or success), the account name and the password used for the authentication request as follows:

```
Jan 16 03:36:45 basta sshd[2308]: PW-ATTEMPT: 1234
Jan 16 03:36:45 basta sshd[2308]: Failed password for root from 10.0.160.14 port 39529 ssh2
Jan 16 03:17:11 basta sshd[2310]: Illegal user password from 10.0.160.14
Jan 16 03:17:11 basta sshd[2308]: PW-ATTEMPT: password
Jan 16 03:17:11 basta sshd[2308]: Failed password for illegal user password from 10.0.160.14 port 40444 ssh2
```

honeypot1	
Account Name	*Password*
root	root
admin	1234
user	0000
guest	password
password	123456
test	qwerty
administrator	654321
webmaster	abc123

(a) weak account names and passwords

honeypot2	
Account Name	*Password*
root	JotCR4E->
admin	mC3bum@:
user	ZR?s25{_
guest	k6r@bPr6
password	[Ea~K^#_
test	{Q};Dced
administrator	:3h!t>VD
webmaster	c)isWAr?

(b) weak account names with strong passwords

Fig. 2. Configurations of the SSH service on `honeypot1` (a) and on `honeypot2` (b)

4.2 Experimental Results: Statistical Aspects and Analysis

In this section are presented the results of our experiments, that starts with a statistical overview of the activities observed on the two honeypots continuing with the analysis of the activities performed after the intrusions.

In the examined period of 30 days, the two honeypots were contacted by 237 different IP addresses. They recorded 74201 login attempts on SSH, capturing in total 2548 different account names and 4231 passwords. We processed the raw data in order to use them as filters to extract relevant information. Such data range from usernames and passwords, the attack types and also the activities performed after the intrusion. As stated above, `honeypot1` contained weak accounts names and passwords, while `honeypot2` contained weak accounts names but complex passwords. Referring to the SSH login account of the `honeypot1`, the first success occurred with the same username and password: "root", after only 23 attempts by only one attack. Thus the remaining 7 accounts were all detected and used to access the machine after about 50-100 attempts. In relation to `honeypot2`, only one account was successfully detected after 4452 attempts, the one with username "root" and password JotCR4E->.

Regarding the date and time of the connections, considering the database of the 74201 login attempts on SSH, filtering them by date, we analyzed the distribution of the attack in the 30 days (see Fig. 3), in which we observed that `honeypot1` was hit with 57457 login attempts with a rate of 1915 attempt per day (with an increasing trend), while for `honeypot2` there were 16744 login attempts with a rate of 558 attempt per day (showing an initial increase, followed by a decrease, probably due to the complexity of passwords).

Analyzing duration and frequency of the attempts, we can split them into two separate groups. The first one comprises attacks performed without interruptions for a period of time (days), with an high frequency and the same interval of time among them. In addition, the second one is composed by attacks realized from time to time, with a low frequency and different intervals of time among them.

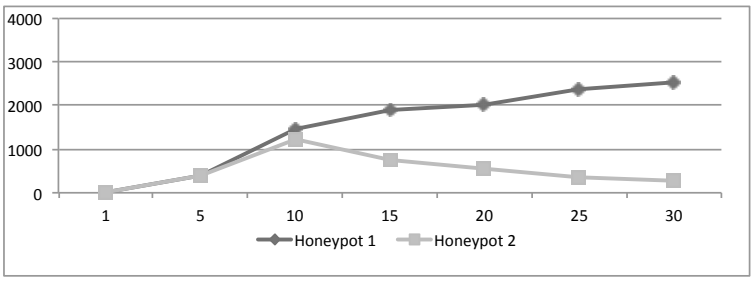


Fig. 3. Trend of access attempts on the two honeypots along the 30 days timeline

Regarding the location (i.e., the IP address) of the attacker, during the examined period of 30 days, the two honeypots were contacted by 437 different IP addresses. All of them attacked the `honeypot1`, but only 76 attempted to access the `honeypot2`. Using the tool GeoIP, we could geographically locate the machines performing the attacks, not necessary the real origin of them. We realized that the intrusion tentatives come from several countries, that is 21% from USA, 19% from China, 15% from Netherlands 11% from Romania, 9% from the United Kingdom, 7% from Germany, and so on.

Analyzing the total number of login attempts, we recognized from one side a 9% of real-time intrusions, recognized by their behavior, processing username and password in a slow way with different breaks, containing also some typing errors, while on the other site, the 91% were performed by dictionary attacks. Applying "`IP addresses`" and "`honeypot1`" as filters to the list of dictionary attacks, we found that 106 of their attempts had these characteristics, which let us to recognize them as performed by automatic scripts. In fact, such connections were only targeted against port 22, thousands of usernames and passwords were processed in a very short time, no pauses were found between attempts and weak usernames and passwords were found in a very short time. The login attempts against the `honeypot2` were performed from 76 IP addresses, which used 233 dictionary attacks, 17 real-time intrusions, and 52 scanning activities. Referring to `honeypot1`, only 8 real-time attacks, performed by two different IP addresses, were able to compromise the system, while all the 106 dictionary attacks violated the machine.

4.3 Analysis of the Activities Performed After the Intrusions

After a successful intrusion, a series of activities were performed by the attackers on the violated host. The attackers first of all change the password of the hacked account and try to acquire the *root* privileges. After that, start exploring the filesystem and start downloading files by means of the commands `wget` and `sftp`. Also, create and hide new directories where to store malicious software that usually is used to scan the networks and to create backdoors. Often, such software includes an IRC client to join a botnets and some tools useful to execute lot of scanning activities.

5 On the Construction of a Profile of Attack

Upon performing the analysis of the attacks, it is possible to extract lot of interesting features that can be useful to start constructing the profile of the attack. First of all, the IP address of the attacker machine and the associated "owner" of such IP together with the geographical location of the attacker machine are extracted. Clearly, the type attack can be a real-time attack or one based on a dictionary. All the temporal information, such as attack lifetime, duration of the attack with the complete hour and day time are easily calculated since the honeypots have been synchronized with a trusted external source of time. From such temporal data it is possible to have the frequency of the attempts of intrusion giving an idea of its regularity and occurrence. Analyzing whether the attack was successful or not, it can be seen the activities performed after the intrusion that may range from (internal/external) network scan, download of files, system exploration, directory creation, malicious software upload and installation. At last but not least, it can be analyzed the type of malicious software installed, the activities performed by the malicious software and, more importantly, the traces and evidence left on the attacked host.

The information extracted from above mentioned features allow us to build the profile of the attack. First of all, we recognized two main groups: *real-time* or *automatic* attacks. We want here to focus attention on the second group, to which we applied, as filters, the following features: high frequency attacks, fast guessing of usernames and passwords, network scanning, creation of new directories/files, successful passwords guessing, upload of files, no errors encountered, success of the intrusion and no traces left.

5.1 Profile Analysis to Detect the Presence of a Cyber Threat

The obtained results brought to our attention one intruder, the only one able to compromise the `honeypot2`. Its IP origin seems to be in Shaoxing, located in the province of Zhejiang, China. Here we show a detailed timeline of its activity:

- Jan 6 2013, 02:00 A.M.: the attacker machine launched a dictionary attack against `honeypot1`, breaking one login account in 21 minutes (username: `test` password: `0000`).
- Jan 6 2013, 02:22 A.M.: it changed the password in `N!ka@mikk@2112`, then it closed the connection.
- Jan 7 2013, 02:00 A.M.: the same machine entered the system with its new account and began to scan the network finding `honeypot2` and its open port 22.
- Jan 7 2013, 02:07 A.M.: It began a dictionary attack to SSH login account on the `honeypot2`, finding the username `root` in about 10 minutes.
- Jan 7 2013, 02:18 A.M. it continued its attack against the password, processing thousands of words in a very fast way with high frequency attempts, and stopped at 5:00.
- Jan 8 2013, 01:00 A.M.: it resumed its attack stopping it at 5:00.

He attacked the system in a continuous way for 11 days, from 0:00 to 6:00 A.M. until the right password `JotCR4E->` was guessed. After the intrusion, it did not changed the password, but created into the home directory a new directory named "`MY_OLD_DOCS`", in which a file named "`11022012doc_old.pdf`" was uploaded before stopping the connection. We did not detected other any other connection until the end of the experimentation on the two honeypots. Submitting that file (named "`11022012doc_old.pdf`") to a forensic analysis, we extracted the MD5 (0xD1E7C8A8D857E097EEF8922F41074E80), the SHA-1 (0xA1339C48B7D8A9F8C7358DA6C3C620F63BE25A51) and filesize (253.952 bytes). This allow us to discover that it was a known cyber threat, named IXESHE [34], that is a backdoor/trojan born in China. This malware communicates with remote servers and receives instructions, acting as in a botnet. It may download and run other malware. The Trend Micro reported [35] that such a trojan is often attached to email messages as a simple PDF file, coming from a compromised or spoofed account. Once opened, the PDF either displays a blank or dummy page, but the code inside it starts the malware. Once installed, IXESHE starts communicating with compromised machines hosted on previously infiltrated networks. Such a dangerous backdoor is the trojan horse named IXESHE, hidden into a PDF file with a very common name capable to connect to a remote C&C server [36], [17] to transmit and receive information to be used during future attacks. It is worth to highlight that very often PDF files are used to convey different kind of malware. The reader can find an interesting study on some security issues that can be exploited by means of PDF files in [37].

Although this type of malware is almost sent by email messages as attachment, here we saw that it was uploaded, but not executed, into a directory with a very ordinary name, probably for several reasons. In fact, spreading such a malware in the ordinary way may not be effective because it could be easily detected by antivirus checking emails and attachments. Also, the attacker did not started the malware immediately for several reasons: to observe how long it will remain undetected, to wait for some user to open such file resulting in the malware installation, or to test a new infection method on some compromised machines to improve it and use at a later time on other targeted systems. Moreover, due to the massive spread of mobile devices and its ubiquitous nature, particular attention should be also paid in protecting such mobile equipment from malware attacks [38], [39].

6 Conclusions

In this paper, we have presented the application of an analytical method to monitor the data streams flowing in cyberspace. The experimental tests were performed by implementing an honeynet system. The results have been obtained from tests conducted for 30 days by using two honeypots, configured to trap SSH intrusion attempts in two different ways. This let us to analyze the behavior of the attacks, step by step, from login attempts to the activities after the intrusion. Clearly, this represented a simple scenario that can be expanded in further

researches. However, this test, although small, allowed us to detect, extract, and analyze the behavior of one cyber-attack, that is used to compromise systems, that are well protected, in a very little time by means of a dictionary-based attack against the SSH service, acting only during the night in a increasing way to stay inconspicuous.

It is important to highlight that, in order to study attacks against a predetermined system, it could be also useful to use tools and methods that record all the packet traffic going to the system under attack [40]. Any way, such approach could not be useful when there is the need of a broader point of view. The information deriving from the use of our method can be used to implement a series of effective countermeasures for protection and prevention cyber-attacks, by improving the physical protection of critical infrastructures, as well as the resolution of digital vulnerabilities of critical systems and the implementation of new security policies for protecting critical data. From a defensive point of view, the possibility to be aware, in real-time, of the presence of forthcoming attacks allows the implementation of an effective system of cyber defense in a dynamic way.

References

1. U.S. Department of Defense: Joint Publication 1-02, Dictionary of Military and Associated Terms. `http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf` (Nov. 2010)
2. Fahrenkrug, D.T.: Countering the Offensive Advantage in Cyber-space: An Integrated Defensive Strategy. In: 4th International Conference on Cyber Conflict, NATO CCD COE Publications, Tallinn. (2012) 197–207
3. Klimburg, A.: National Cyber Security Framework Manual, NATO CCD COE Publications. `http://www.ccdcoe.org/369.html` (Dec. 2012)
4. Saalbach, K.: Cyber-war. Methods and Practice, version 6.0. `http://www.dirk-koentopp.com/downloads/saalbach-cyberwar-methods-and-practice.pdf` (Jan. 2013)
5. Colombini, C., Colella, A., Mattiucci, M.: Cyber-war Profiling, a new Method for the Analysis of a Cyber-Conflict. to appear on NATO CCD COE, Tallinn (Jan. 2013)
6. Palmieri, F., Fiore, U.: Containing large-scale worm spreading in the Internet by cooperative distribution of traffic filtering policies. Computers & Security **27**(1-2) (2008) 48–62
7. Palmieri, F., Fiore, U., Castiglione, A.: Automatic security assessment for next generation wireless mobile networks. Mobile Information Systems **7**(3) (2011) 217–239
8. Palmieri, F., Fiore, U.: Audit-Based Access Control in Nomadic Wireless Environments. In Gavrilova, M., Gervasi, O., Kumar, V., Tan, C., Taniar, D., Laganà, A., Mun, Y., Choo, H., eds.: Computational Science and Its Applications - ICCSA 2006. Volume 3982 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 537–545
9. Palmieri, F., Fiore, U.: Network anomaly detection through nonlinear analysis. Computers & Security **29**(7) (2010) 737–755

10. Fiore, U., Palmieri, F., Castiglione, A., De Santis, A.: Network anomaly detection with the restricted Boltzmann machine. Neurocomputing (0) (2013)
11. Vidulich, M., Dominguez, C., Vogel, E., McMillian, G.: Situation Awareness: Papers and Annotated Bibliography, U.S. Department of Defense, Defense Technical Information Center (DTIC). http://www.dtic.mil/dtic/tr/fulltext/u2/a284752.pdf (Jun. 1994)
12. Colombini, C.M., Colella, A.: Digital Profiling: A Computer Forensics Approach. In Tjoa, A., Quirchmayr, G., You, I., Xu, L., eds.: Availability, Reliability and Security for Business, Enterprise and Health Information Systems. Volume 6908 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 330–343
13. Colombini, C., Colella, A., Castiglione, A., Scognamiglio, V.: The Digital Profiling Techniques Applied to the Analysis of a GPS Navigation Device. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. (2012) 591–596
14. Castiglione, A., De Santis, A., Fiore, U., Palmieri, F.: Device Tracking in Private Networks via NAPT Log Analysis. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. (2012) 603–608
15. Colombini, C.M., Colella, A., Mattiucci, M., Castiglione, A.: Network Profiling: Content Analysis of Users Behavior in Digital Communication Channel. In Quirchmayr, G., Basl, J., You, I., Xu, L., Weippl, E., eds.: Multidisciplinary Research and Practice for Information Systems. Volume 7465 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 416–429
16. Matrosov, A., Rodionov, E., Harley, D., Malcho, J.: Stuxnet Under the Microscope, rev. 1.31, ESET LLC. http://ece.wpi.edu/~dchasaki/papers/Stuxnet_Under_the_Microscope.pdf (2012)
17. Castiglione, A., De Prisco, R., De Santis, A., Fiore, U., Palmieri, F.: A botnet-based command and control approach relying on swarm intelligence. Journal of Network and Computer Applications (0) (2013) –
18. Ziolkowski, K.: *Ius ad bellum* in Cyberspace - Some Thoughts on the "Schmitt-Criteria" for Use of Force. In: 4th International Conference on Cyber Conflict, NATO CCD COE Publications, Tallinn. (2012) 295–309
19. Fanelli, R., Conti, G.: A methodology for cyber operations targeting and control of collateral damage in the context of lawful armed conflict. In: Cyber Conflict (CYCON), 2012 4th International Conference on. (2012) 1–13
20. CrySyS Lab: sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks. http://www.crysys.hu/skywiper/skywiper.pdf (May 2012)
21. Bencsáth, B., Pék, G., Buttyán, L., Félegyházi, M.: Duqu: A Stuxnet-like malware found in the wild. http://www.crysys.hu/publications/files/bencsathPBF11duqu.pdf (Oct 2011)
22. Kaspersky Lab, Global Research and Analysis Team: Gauss: Abnormal Distribution. http://www.securelist.com/en/analysis/204792238/ (Aug 2012)
23. Kaspersky Lab, Global Research and Analysis Team: The Mahdi Campaign. http://www.securelist.com/en/blog/208193691/The_Madi_Campaign_Part_II (Jul 2012)
24. Infosec Institute: Honeypots Resources. http://resources.infosecinstitute.com/honeypots/ (Oct 2012)
25. Moore, J.: Mercury Live DVD. http://mercurylivedvd.sourceforge.net/ (2013)

26. Castiglione, A., Cattaneo, G., De Prisco, R., De Santis, A., Yim, K.: How to Forge a Digital Alibi on Mac OS X. In Quirchmayr, G., Basl, J., You, I., Xu, L., Weippl, E., eds.: Multidisciplinary Research and Practice for Information Systems. Volume 7465 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 430–444
27. Albano, P., Castiglione, A., Cattaneo, G., De Maio, G., De Santis, A.: On the Construction of a False Digital Alibi on the Android OS. In Xhafa, F., Barolli, L., Köppen, M., eds.: INCoS, IEEE (2011) 685–690
28. Castiglione, A., Cattaneo, G., De Maio, G., De Santis, A.: Automated Production of Predetermined Digital Evidence. Access, IEEE **1** (2013) 216–231
29. De Santis, A., Castiglione, A., Cattaneo, G., De Maio, G., Ianulardo, M.: Automated Construction of a False Digital Alibi. In Tjoa, A., Quirchmayr, G., You, I., Xu, L., eds.: Availability, Reliability and Security for Business, Enterprise and Health Information Systems. Volume 6908 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 359–373
30. Castiglione, A., Cattaneo, G., De Maio, G., De Santis, A., Costabile, G., Epifani, M.: The Forensic Analysis of a False Digital Alibi. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. (2012) 114–121
31. Nicomette, V., Kaâniche, M., Alata, E., Herrb, M.: Set-up and deployment of a high-interaction honeypot: experiment and lessons learned. Journal in Computer Virology **7**(2) (2011) 143–157
32. Li, C., Parsioan, T.: Profiling Honeynet Attackers. Proceedings of the Class of 2006 Senior Conference, (2005) 19–26
33. Seifert, C.: Analyzing Malicious SSH Login Attempts. `http://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts` (Nov 2010)
34. Threat Expert Ltd: Backdoor:Win32/Ixeshe.E. `http://www.threatexpert.com/report.aspx?md5=d1e7c8a8d857e097eef8922f41074e80` (2013)
35. Sancho, D., dela Torre, J., Bakuei, M., Villeneuve, N., McArdle, R.: IXESHE An APT Campaign. `http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_ixeshe.pdf` (20102)
36. Tyugu, E.: Command and control of cyber weapons. In: Cyber Conflict (CYCON), 2012 4th International Conference on. (2012) 1–11
37. Castiglione, A., De Santis, A., Soriente, C.: Security and privacy issues in the Portable Document Format. Journal of Systems and Software **83**(10) (2010) 1813–1822
38. Armando, A., Merlo, A., Migliardi, M., Verderame, L.: Would You Mind Forking This Process? A Denial of Service Attack on Android (and Some Countermeasures). In Gritzalis, D., Furnell, S., Theoharidou, M., eds.: Information Security and Privacy Research. Volume 376 of IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg (2012) 13–24
39. Armando, A., Merlo, A., Migliardi, M., Verderame, L.: Breaking and fixing the Android Launching Flow. Computers & Security (0) (2013) –
40. Castiglione, A., Cattaneo, G., De Maio, Giancarlo, De Santis, A.: Forensically-Sound Methods to Collect Live Network Evidence. In: Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on. (2013) 405–412