Analysis and Visualization of SSH Attacks Using Honeypots

Ioannis Koniaris, Georgios Papadimitriou and Petros Nicopolitidis

Department of Informatics,
Aristotle University of Thessaloniki
54124 Thesssaloniki, Greece
[ikoniari, gp, petros@csd.auth.gr]

Abstract—**In the field of computer security, honeypots are systems aimed at deceiving malicious users who launch attacks against the servers and network infrastructure of various organizations. They can be deployed as protection mechanisms to an organization's real systems, or as research units to study and analyze the methods employed by individual hackers. In this paper we present the results of a research honeypot's operation, which undertook the role of a web trap for attackers who target the SSH service in order to gain illegal server access. The fake system has remained online and fully operational during a course of several consequent months, capturing attacks and logging all malicious activity. During this assessment it was shown that honeypots remain very effective tools in gathering information about SSH attacks. Furthermore, we observed that attackers are constantly targeting servers in the wild employing ready-to-use tools and dictionaries, while their post-compromise actions include mostly pivoting and IRC-related activities. Lastly we present a visualization tool aimed at helping security researchers during the analysis and conclusions drawing phases, for use with the same SSH honeypot implementation software as outlined in this work.**

Keywords: honeypot, secure shell, security visualization, cyber attack analysis, cyber crime

I. INTRODUCTION

The massive use of computer systems and internet technologies has made information security one of the top concerns for today's reality. Malicious users and software launch attacks on a regular basis against specific or broad targets. The motivation behind these attacks varies and can be the simple act of a cyber vandal, corporate espionage for financial gain or state-sponsored attacks with geopolitical impact.

Attackers are searching the internet for servers that can be used for their malicious activities. One of the most prominent targets is servers on which the administrator has set up a remote access service, for example Secure Shell (SSH). Many times this can be exploited by malicious users if a weak password is in a place as an authentication mechanism. Whenever an attacker finds such a server that runs the particular service, he will try to connect to it using various combinations of authentication credentials. If a successful login attempt is made, the attacker gains remote access to the server and then uses it for malicious activities such as malware installation or pivoting; using the compromised server to launch attacks against other systems.

Identifying and classifying these attacks along with their actors is a crucial step in developing security policies and strategies to effectively defend against them. To stay one step ahead of malicious users and get early warnings of new vulnerabilities and exploits, one can use devices known as honeypots. In their simplest form these devices act as decoy-based intrusion detection systems with full logging capabilities.

Many definitions for honeypots exist, but one of the most accurate belongs to Lance Spitzner who defines a honeypot as an information system resource whose value lies in unauthorized or illicit use of that resource [1]. A honeypot is a system with no production value. Its operation is based on the following concept [2]: there is no reason for a legitimate user to use it directly or interact with it, therefore any communication attempt with the system is automatically considered malicious and it is usually classified as one of the following: detection, network scanning or attack. Respectively, a honeypot that tries to connect to an external network resource has been probably compromised by an attacker [3].

Honeypots are both deceit tools and traps. They can lure malicious users by acting as systems that contain valuable data or interesting services. They allow to be exploited and then offer a simulated or real environment to the attacker to interact with, while logging all of his actions and activities. In such a way they help security professionals and researchers in the process of learning the techniques and methods used by attackers to compromise computer systems. Honeypots cannot prevent cyber attacks against the network by themselves, but they can help in identifying and detecting them when used alongside with other defense-oriented tools such as firewalls.

Honeypots often generate a small amount of data of high value [4] which is considered an advantage, but depending on the circumstances the analysis of this dataset can be a challenge for information security professionals. As the number of attacks grows very large over time, it becomes impossible to manually analyze or compare each and every captured session. This is where data visualization solutions and visual analytics [5] can play a vital role in helping the defenders get a quick and detailed overview of a honeypot's operation.

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

In this paper we focus specifically on SSH brute-force and dictionary attacks. In particular, we analyze data collected from a large number of SSH attacks against a Virtual Private Server (VPS) which was set up as a honeypot in order to log malicious internet activities. These show the vast number of attacks being constantly launched by malicious adversaries, and their favorite post-compromise activities which mostly included pivoting; using the captured systems to attack other servers, most of the time without even examining the compromised machine's contents. We then proceed to the visualization of the captured data using a tool that was created for this purpose, employing standard web technologies and services.

II. Related work

A substantial number of studies of SSH attacks have been carried out in recent years. Some of them have been the result of academic affiliated work [6]–[8], while others have stemmed from the effort of information security professionals or companies [9]–[11]. In some of these cases the study of SSH attacks has been a portion of a bigger study, which mostly included profiling attackers or following their activities after they gained illegal system access. In our research we have included post-compromise activities in our scope as well. This gives us a better understanding of the actors and motivations behind the attacks, while the captured SSH login traffic serves the goal of developing a deeper understanding of the techniques employed in SSH attacks.

Regarding the visualization of attacks on computer networks, a lot of research has been done, but it is mostly focused on visualizing NetFlow data coming from attacks logged by an Intrusion Detection System (IDS). For example, the tool NFlowVis [12] was created in 2008 and it can be used to visually analyze attacks in large-scale networks using NetFlow data from IDS attack logs. While SSH-related visualizations were presented alongside, only connection attempts were shown. In 2009 the tool VIAssist [13] was created, which can provide the details of specific network flows in need to be examined further. However, VIAssist has no valuable practical use for the analysis of SSH attacks since it only visualizes NetFlow data.

Focusing more on visualizing malicious activities using honeypots, a project that was started by security professional J. Blasco resulted in the creation of a visualization tool for the Nepenthes honeypot [14]. Nepenthes is a malware honeypot; a utility to assist malware researchers in the process of gathering and securely storing infectious binaries of malicious software. The aforementioned visualization tool [15] uses the AfterGlow and Graphviz software libraries in order to create several directed graphs. These depict the correlations between IP addresses, malware samples and geographical data.

Another honeypot-related visualization tool is called carniwwwhore [16] and was developed for the Dionaea malware honeypot [17]. Dionaea is considered to be the successor of Nepenthes honeypot and the aforementioned visualization tool has similar capabilities in comparison to our tool that was developed during the course of this work, although each employ different technologies.

III. Project Overview

A. Theoretical Background

In this section we provide a brief overview of the theory behind our experiment. More specifically, we first discuss a number of concepts related to different types of honeypots, and secondly the functionality of the Secure Shell protocol and its security challenges.

1) *Types of Honeypots based on Purpose:* Honeypots can be classified into two distinct categories based on the purpose they serve, namely production and research honeypots. Production honeypots are systems that help an organization mitigate risk in a networked environment. These types of honeypots are most of the times similar to the real systems of an organization and they are deployed as deceit mechanisms. Thus, an attacker wastes time on these decoys while the administrators can examine his methods to harden the real valuable systems. On the other hand, research honeypots are not used with a specific end in mind, other than that of capturing as many malicious data as they can. They are deployed by information security researchers with the primary purpose to capture extensive information that can be later analyzed and used in designing strategies and countermeasures against the attacks. They also help information security professionals identify new and unknown vulnerabilities, exploits or tools used by malicious users.

2) *Types of Honeypots based on Interaction Level:* Honeypots can be classified into three distinct categories based on the level of interaction they provide to the attacker. The first of these categories is low interaction honeypots. This type of honeypot, as the name implies, offers limited interaction between the system and the attacker. The primary goal of such a honeypot is the detection and logging of unauthorized connections. Low interaction honeypots are the easiest to deploy and maintain. While the added risk to a network is low, their information gathering capabilities are limited. The second category is medium interaction honeypots. A medium interaction honeypot offers a higher interaction level to the attacker. When a connection attempt is made to a system port, the honeypot can reply back with specially crafted messages or network packets that emulate those of a real network service. Medium interaction honeypots can also offer a simulated or virtual operating system to the attacker where he can enter commands or download files, while all of these activities are monitored and logged by the underlying real operating system. The level of added risk to the network is considered to be medium, along with the level of information gathering capabilities they provide. The last category is high interaction honeypots. These types of honeypots offer the highest possible level of interaction with the attacker. They are actual systems with real operating systems and vulnerable services, offered to attackers for compromise and takeover. High interaction honeypots add the

greatest risk to a network but they offer the highest level of data capturing and information gathering capabilities. High interaction honeypots are hard to maintain and must be secured using tools such as firewalls in order to restrict outbound connections to external resources.

3) *Secure Shell and SSH Attacks:* Secure Shell, or SSH, is an encrypted remote connection mechanism, commonly used in Linux and Unix-based operating systems. The protocol was defined by Ylonen and Lonvick in Internet Engineering Task Force's RFC 4254 [18], and allows users to authenticate to remote machines gaining access to an interactive shell. By default, Secure Shell uses the well known port 22 and it implements a username and password authentication mechanism, although this can be substituted with other more secure methods like public key authentication. Due to this fact, attacks against the SSH protocol have become quite common as an attacker can simply guess or brute-force the correct credentials used by a legitimate system user or administrator. The SSH protocol is open and well defined and many software libraries exist allowing the creation of SSH clients. This has enabled malicious users to create easy-to-use and automated attack tools against the specific service, that employ either brute-force or dictionary-based attack methods.

B. Experimental Setup

To test the effectiveness of SSH honeypots and gather as much data as possible we deployed an SSH honeypot using a Virtual Private Server (VPS). This system served the role of a research honeypot as described in the previous section. It was connected to the internet using a static IP address and run specialized software in order to implement the web trap. Specifically, we used the Kippo SSH honeypot [19], which is a piece of software written in Python programming language. Kippo is a medium interaction honeypot as described in the previous section, as it allows interaction with the attacker. It can bind to Secure Shell's default TCP port 22 and log each connection attempt with the server. The software can also store these attempts to a MySQL database along with useful information. For this reason a standalone MySQL server was set up in order to collect the aforementioned data. Furthermore, Kippo allows a list of credentials to be defined, which give access to a fake operating system giving to the intruder the ability to interact with it. Every command that is entered in this interactive shell is logged in the MySQL database as well, and every file downloaded is saved for future analysis. The program responds to these commands as a real operating system based on Debian Linux would do.

Upon the implementation of the SSH honeypot using Kippo, we discovered two flaws in its design that could serve as signatures for malicious users, identifying the system as a decoy. These two flaws were reported to the development team along with suggestions for mitigating their risk, and were successfully fixed during our testing phase.

This SSH honeypot operated for a period of 4 months, from early December 2011 until early April 2012. This time period was considered sufficient to gather the data we needed, extract information from them and draw conclusions.

TABLE I
OVERALL HONEYPOT ATTACK ACTIVITY

Time Period	Login Attempts	Source IPs	Attacks
9 Dec. 2011 – 9 Apr. 2012	23.271	298	87

C. Visualizing the Results

An important step towards more meaningful analysis of the data we collected was the effort of visualizing the results. This can give to an information security professional the ability to get a quick and detailed overview of the honeypot's operation.

For this reason a visualization tool called Kippo-Graph was created, with the goal to extract data stored in the MySQL database and present them in a meaningful graphical way. Kippo-Graph is written in PHP programming language and uses some software libraries for its operation. Specifically, the Libchart drawing library to create the charts it displays, the QGoogleVisualizationAPI library to utilize and import content to the maps service offered by Google, and the free web based geoPlugin geolocation service. As of this writing, Kippo-Graph can create 24 charts in total and it is consisted of three distinct modules: kippo-graph itself to present the basic charts, kippo-input for post-compromise information and kippo-geo for correlating attacks with their geographic origins.

IV. OVERVIEW OF ATTACK ACTIVITY

In this section we begin with a basic overview of the brute-force and dictionary-based attacks we observed. Over the course of approximately 4 months, the honeypot was subjected to at least 87 distinct brute-force or dictionary-based attacks, consisting of 23.271 login attempts, originating from 298 distinct IP addresses. These overall statistics are presented in Table I.

A. Common Usernames, Passwords and Combinations

Overall the honeypot logged 2.844 distinct usernames used against it. As one might expect, the username observed most often in malicious login attempts was root, which was used 10.523 times, almost in half of the total number of logins. This is no surprise as malicious users would have wanted to gain illegal system access with full privileges. Other usernames attempted are often associated with temporary user accounts such as test, user or guest. Another category of usernames attempted belongs to known system accounts and services such as oracle, nagios, postgres or tomcat. Fig. 1 and Table II present the "top 10" usernames observed, along with their respective login attempts.

Beyond root, system and temporary account names, the majority of usernames used in attacks were first names, such as michael, alex or amanda. Some very rare usernames such as w1ckedshootings or fedorapraxis2823 were rather complicated and could have been the result of previous system compromise.

In total, our honeypot recorded 8.556 distinct passwords. The most common one was 123456 which was tested 1.507

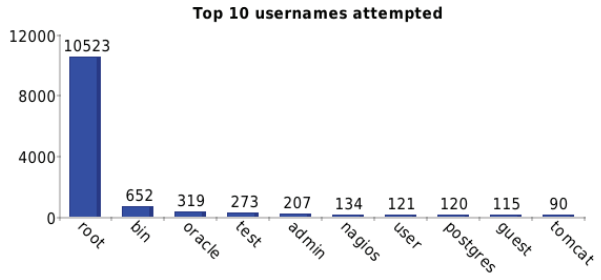

Fig. 1. Top 10 usernames attempted against the SSH honeypot system.

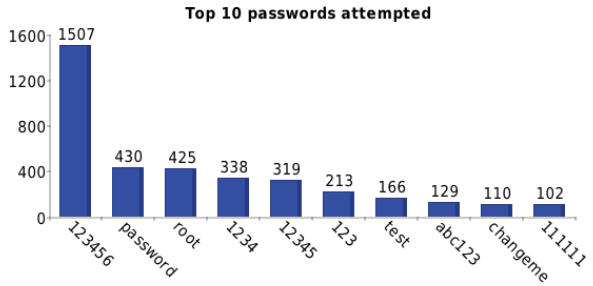

Fig. 2. Top 10 passwords attempted against the SSH honeypot system.

TABLE II
TOP 10 USERNAMES OBSERVED IN SSH ATTACKS

Username	Login Attempts
root	10.523
bin	652
oracle	319
test	273
admin	207
nagios	134
user	121
postgres	120
guest	115
tomcat	90

TABLE III
TOP 10 PASSWORDS OBSERVED IN SSH ATTACKS

Password	Login Attempts
123456	1.507
password	430
root	425
1234	338
12345	319
123	213
test	166
abc123	129
changeme	110
111111	102

times. The second and third choices among malicious users were the word password itself, and root, which were seen 430 and 425 times respectively. Once more temporary user accounts such as test and system accounts such as oracle, redhat or postgres are present in the top list. There are also some interesting password attempts such as iamhacker22 and worlddomination, along with passwords that can be entered by following simple keyboard patterns such as q1w2e3 or 1qaz2wsx3edc. Lastly, passwords such as danielsm300385 or jurca4ileana can be considered quite complicated and could have been the result of previous system compromise. Fig. 2 and Table III present the "top 10" passwords observed, along with their respective login attempts. With regards to password length, 36.21% consisted of one to six characters, 61.46% consisted of one to eight, while 38.54% had more than eight characters. Passwords' character set results are not very surprising, since 45.74% of all passwords consisted of plain strings and 22.42% consisted of strings combined with digits.

The above usernames and passwords were entered in pairs for each login attempt forming credential combinations. Our honeypot logged 12.269 distinct combinations of usernames and passwords. One out of three combinations contained the username root, which is no surprise since malicious users wanted to gain illegal system access with full privileges. The combination root/123456 is the top one, seen 142 times. It is followed by root/password and root/root with 99 and 81 attempts respectively. Other interesting combinations are some that come from a dictionary containing many variations

TABLE IV
TOP 10 CREDENTIAL COMBINATIONS OBSERVED IN SSH ATTACKS

Username	Password	Login Attempts
root	123456	142
root	password	99
root	root	81
root	p@ssw0rd	66
root	111111	63
root	qwerty	60
root	1234	57
test	test	53
root	passw0rd	53
root	1q2w3e	47

of the word password such as p@ssw0rd or pa$$word. A list of the "top 10" results can be seen in Table IV, while Fig. 3 presents them graphically.

Kippo SSH honeypot has a special list of credentials that allows a malicious user to gain access to a simulated operating system. The combination root/123456 is present in this list by default, thus at least 142 break-ins took place. If we query the MySQL database we can count a total of 152 successful logins. This means that apart from the 142 aforementioned connections, attackers manually entered in our honeypot 10 more times and interacted with the fake Linux system during our experimental period. These 10 additional break-ins are not

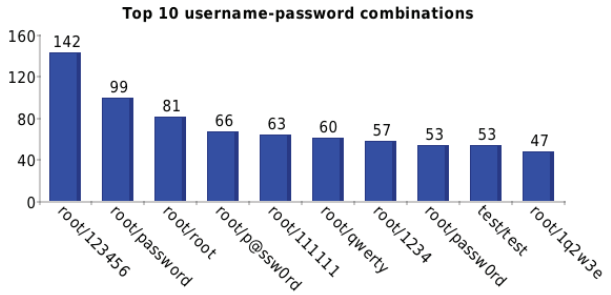

Fig. 3. Top 10 credential combinations attempted against the SSH honeypot system.

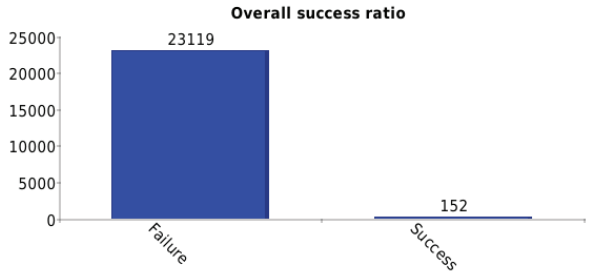

Fig. 4. Summary of failed and successful login attempts.

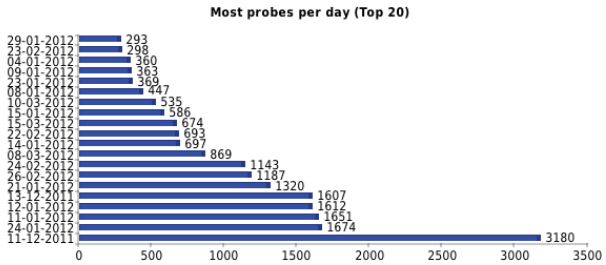

Fig. 5. Top 20 days ordered by greatest number of probes.

attributed to the sum of root/123456 combination because the attackers have changed the account's password in the meantime. With regards to failed login attempts, we observed that 23.119 out of the total 23.271 connections resulted in a failure, as depicted graphically in Fig. 4.

B. *Attack and Intrusion Frequency*

Our honeypot logged the greatest number of attacks on the 11th of December 2011. During the aforementioned day, 3.180 attacks were observed against the system. Four additional dates follow in the top list, each of them near the 1.600 attacks mark. These dates include the 24th of January 2012 with 1.674 attacks, the 11th of January 2012 with 1.651 attacks, the 12th of January 2012 with 1.612 and lastly the 13th of December 2011 with 1.607 attacks. Another significant date was the 21th of January 2012 with 1.320 logged attacks. The "top 20" results

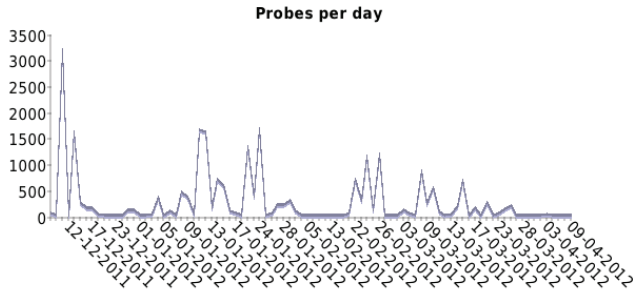

Fig. 6. Probes per day during our honeypot operation period.

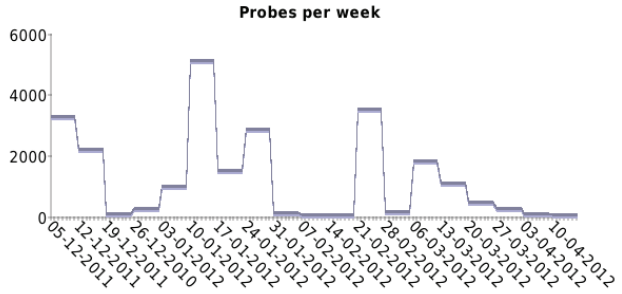

Fig. 7. Probes per week during our honeypot operation period.

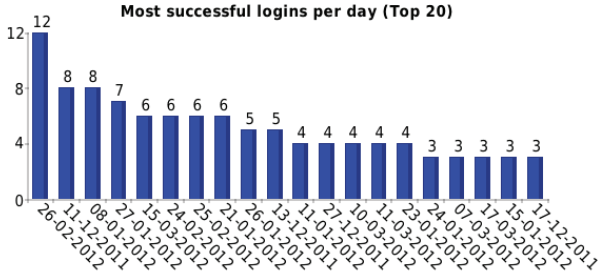

Fig. 8. Top 20 days based on the number of successful break-ins.

for the most active dates during the operation of our honeypot are depicted graphically in Fig. 5.

A general overview of the attack frequency against the honeypot is shown in Fig. 6, where our visualization tool presents the "probers per day" chart, while Fig. 7 summarizes the attacks per week during our honeypot operation. Despite the relative big set of distinct username and password combinations that were tried against the honeypot system, a direct correlation between the number of daily attacks and the number of daily successful intrusions is not clearly observed.

The most successful day from the attackers' perspective was the 26th of February 2012, when 12 distinct intrusions were observed. This means that at least 12 different attackers managed to guess the correct authentication credentials, or that fewer than 12 attackers managed to find more than one correct combinations. Other days with a significant number of successful intrusions are the 11th of December 2011 and the 8th of January 2012, when 8 successful hacking attempts were

TABLE V
TOP 10 SOURCE IP ADDRESSES OBSERVED IN SSH ATTACKS

Source IP	Login Attempts
67.205.111.24	2.795
88.191.152.31	1.594
195.56.150.12	1.508
67.23.14.194	1.286
86.110.198.227	1.194
108.60.201.66	1.052
60.171.214.30	910
91.93.35.68	725
70.89.106.145	628
31.15.104.83	628

observed. Those two aforementioned dates are indicative of the absence of direct correlation between attacks and successes, as the former was the top day based on volume of attacks with 3.180 tries, while the latter was only at the 15th position with 447 tries. Fig. 8 shows the "top 20" days based on the number of successful attacks against our honeypot.

C. *Source IP Addresses*

Our honeypot was attacked by 298 distinct IP addresses. Each one of them participated in a varying number of login attempts against the system. On average, we can say that 78.5 attacks are accounted for each source IP address. Naturally though, this is not observed in our results, where there are big deviations in the number of attacks per distinct IP. For example, there were 4 IP addresses with a four-digit sum of attacks each, and more than 190 IP addresses with a single-digit sum. This does not necessarily mean that there were some more persistent attackers than others, but it could simply be the case that the former had a bigger attack dictionary in their possession.

The top IP address observed was 67.205.111.24 and it is located in Montreal, Canada. This IP address was the source of 2.795 attacks and belongs to a netblock owned by a relative known web hosting company. If one performs a reverse DNS lookup he or she could see that the server is leased by an internet radio station. It could be possible that an attacker has gained illegal access to that server and is using it as a pivot to launch attacks against other systems. The second IP in the top list was 88.191.152.31 which was the source of 1.594 attacks. This address is located in Paris, France and belongs to a netblock owned by a relative known French Internet Service Provider (ISP). With further analysis one can see that it is used by the dedicated servers division of the company, and thus it is perhaps another already compromised server used by an attacker. The same applies to the 3rd top IP which was 195.56.150.12, located in Hungary. This address was the source of 1.508 attacks against our honeypot and it is used by a web design company. Once more, one could assume that an illegal action has previously taken place against the system, which is now used in further attacks by a malicious user. A list of the "top 10" source IPs can be seen in Table V, while

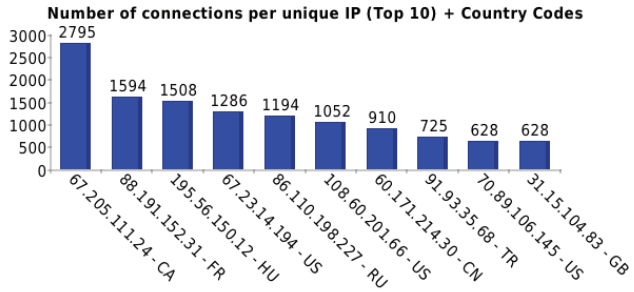

Fig. 9. Top 10 source IP addresses along with their respective country codes.

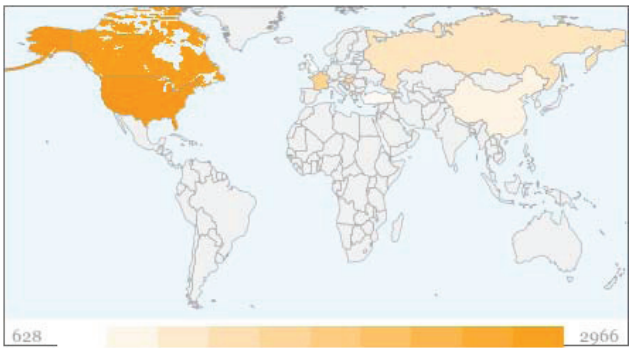

Fig. 10. An intensity map summarizing the attacks based on the top 10 source IP addresses.

Fig. 11. The geographical location of the top 10 source IP addresses drawn on a Google Map.

Fig. 9 depicts the same results graphically along with the ISO 3166 country code for the origin of each IP address.

If someone was to perform an analysis like the one mentioned previously for the top three source IP addresses, he or she would find that it can be a challenging and time consuming task if performed manually. Our visualization tool Kippo-Graph was programmed among others to automate this analysis phase. In Fig. 10 and Fig. 11 we can also see some of the tool's geolocation presentation capabilities, where it depicts the geographical sources of attacks.

D. *Post-Compromise Commands and Downloaded Files*

In total, 453 commands were entered inside the fake operating system simulated by Kippo honeypot. From these, we observed 212 distinct commands. This does not necessarily mean that each of the aforementioned entries perform a drastically different operation. Some of them are simply the same command entered with extra parameters or

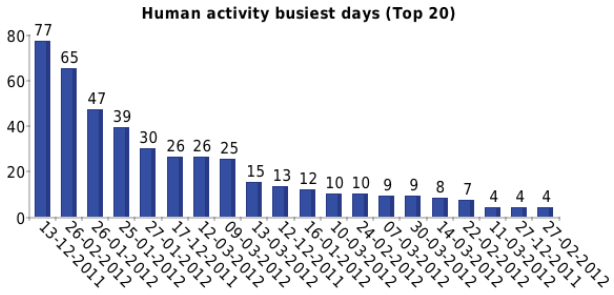

Fig. 12. Top 20 days based on human interaction between the honeypot and successful intruders.

TABLE VI
TOP 10 POST-COMPROMISE COMMANDS

Command	Logged Attempts
w	38
ls –a	26
ls	17
chmod +x *	14
uname –a	12
cat /proc/cpuinfo	12
exit	9
wget	7
ps x	7
passwd	7

different target directories and files. Kippo logs the whole terminal line as a string upon pressing the Enter key. Thus, the same command with and without parameters is counted as a unique entry. Fig. 12 presents the "top 20" days based on human activity inside the honeypot, i.e. how many commands were logged by the system during each day of operation as given by successful intruders.

The top command that was logged the greatest number of times, specifically 38, was "w" which generally returns system information like the server's uptime duration and the currently logged in users. Second and third in the top list were the commands "ls –a" and plain "ls" which return the contents of the current directory in Unix-based systems. They were logged 26 and 17 times respectively. The "-a" parameter is used to make the command return every folder and file in the directory, even the hidden ones that are not normally shown when using the plain version. The fourth most frequent command observed was "chmod +x *" which is an interesting one. In total, it was logged 14 times and it gives execution rights to all the files in the target directory. It was mostly used by intruders to enable the direct execution through the terminal of their various malicious scripts they had previously downloaded to the system. The next two most frequent commands were "uname –a" that also returns system information and "cat /proc/cpuinfo" that shows data concerning the server's CPU, logged 14 times each. A list of

TABLE VII
FINAL LIST OF AUTHENTICATION CREDENTIALS

Username	Password
root	123456
root	tosoigoceebatse
root	xp007dan
root	mereubazatsmecher
root	mixmixtalentat
root	diariaalinamisu123
root	spiri123spiri
root	invincib2012

the "top 10" commands entered by malicious intruders inside the honeypot is shown in Table VI.

Some of the malicious users proceeded to change the password used for logging in to the fake operating system in order to secure it or prevent other attackers from guessing it as well. In total, there were 7 changes of the default password using the "passwd" command. Kippo automatically added those new authentication credentials to the special list that gives access to the fake operating system. The final contents of this list are shown in Table VII. Apart from the first combination root/123456, which is Kippo's default one, all others were added by successful intruders.

Kippo can also store the files downloaded by malicious users when they use the "wget" command inside the fake operating system. In reality these files are in fact downloaded and saved in a special directory belonging to the real operating system running in the background. In total, the "wget" command was logged 33 times. From these, 7 times the command contained no target file, thus 26 files were finally stored in the system. The analysis of those 26 files showed that 19 were unique.

The greatest number of downloads were observed for the Windows 2000 Service Pack 3 file, hosted by Microsoft. This particular file was downloaded 5 times by different intruders. In general it has been noted to appear often in compromised systems and it is presumed that this happens because it is one of the few files left in the company's servers with a direct download link available for it. Given the company's resources, the aforementioned file has become an effective way for an intruder to measure the compromised server's network connectivity speed. A same need was fulfilled by another similar file hosted by CacheFly Content Deliver Network (CDN) which was downloaded a single time.

The rest of the downloaded data are malicious files which aid mostly in launching further attacks. Several of the programs downloaded contained phrases in languages of Eastern European origin, such as Romanian or Russian. This does not necessarily indicate the country of origin for successful intruders, but it can probably indicate the country of the programs' original developers.

The downloaded software contained psyBNC [20]; a well known Internet Relay Chat (IRC) open source bouncer, FloodBot which is used to send mass messages in IRC servers, and other programs related to IRC in general. Other

files included a SYN network scanner written in C and other port scanners like a binary with Romanian strings named "ss" and the pscan2 tool [21]. One interesting piece was an old version of SHOUTcast [22] software which is used to host internet radio stations. Lastly, other significant files observed were an exploit for a vulnerability found in phpMyAdmin [23]; a well known graphical database management interface, UDP flooders used in Denial of Service (DOS) attacks such as fudp [24], and a tool to perform attacks against the SSH service along with a file containing an attack dictionary of 627 username and password combinations. It was interesting to observe that the same dictionary was previously used in attacks against our honeypot as well.

V. Future Work

While the deployment and management of SSH honeypots with hardware and software such as those used in our study is considered a straightforward process, the work of properly analyzing the collected data can present a challenge to information security professionals and researchers. A toolkit that would bundle honeypot software along with useful and ready-to-use utilities such as visualization scripts could be a reliable asset for an organization's system administrators. Such a toolkit does not currently exist in an easily distributed medium such as a virtual appliance, but its development is underway by our team. Furthermore, many such honeypot appliances can be spread across one large or many different networks, while a centralized database could collect all the activity from each one of them. The data logged from an operation of this sort, especially the source IP addresses responsible for the attacks, could be used in the creation of a blacklist database. This in turn could be used in conjunction with software implementing countermeasures, in order to block malicious connections before they are processed by the SSH service itself.

VI. Conclusion

Honeypots present a unique security concept and they are a powerful technology. In this paper we presented a practical implementation of a specialized honeypot for the Secure Shell (SSH) service, which is commonly targeted by attackers. Many Linux distributions install an SSH server by default, many times without proper security in place or additional protection mechanisms such as firewalls. Thus, otherwise fully patched and updated systems can be compromised by malicious users, due to a carelessly chosen password.

In this paper we have run a relative lengthy experiment using the aforementioned SSH honeypot system, which yielded interesting results. It was shown that common credential dictionaries exist and they are exchanged between malicious users for repeated use in SSH attacks. Intruders seem to have in place specific tools they download and utilize immediately, aiding them mostly in conducting further attacks against other servers. Thus, the compromised systems are used as pivots; launch pads for more intrusion attempts. In such cases the affected organizations run the additional risk of law accusations for liability and damages, which is another important factor to consider when designing network and security architectures. System administrators should also use password-checking tools to ensure the security of their chosen credentials and implement password security policies for the users of their systems.

Lastly, a visualization tool was presented for use with the specific honeypot software fielded in our study. This type of utility aids in the graphical presentation of a honeypot's operation and can give a quick and detailed overview of the activity to information security professionals and researchers.

References

[1] L. Spitzner, "Honeypots: Catching the Insider Threat," in *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003.

[2] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA: Addison Wesley, 2003.

[3] L. Spitzner, "Strategies and issues: Honeypots - sticking it to hackers," *Network Magazine*, 2003.

[4] A. Obied, "Honeypots and Spam." 2007.

[5] D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," *Visual Data Mining*, pp. 76–90, 2008.

[6] E. Alata, V. Nicomette, M. Kaaniche, M. Dacier, and M. Herrb, "Lessons learned from the deployment of a high-interaction honeypot," in *Proc. Dependable Computing Conference (EDCC06)*, 2006, pp. 39–46.

[7] D. Ramsbrock, R. Berthier, and M. Cuckier, "Profiling Attacker Behavior Following SSH Compromises," in *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2007, pp. 119–124.

[8] J. Owens and J. Matthews, "A Study of Passwords and Methods Used in Brute-Force SSH Attacks." 2008.

[9] "Observations of Login Activity in an SSH Honeypot," *Cisco Security Intelligence Operations*, 2009. [Online]. Available: https://www.cisco.com/web/about/security/intelligence/ssh-security.html.

[10] J. C. Klein Keane, "Using Kojoney Open Source Low Interaction Honeypot to Develop Defensive Strategies and Fingerprint Post Compromise Attacker Behavior," *HITB Magazine, Volume 1, Issue 3*, pp. 4–14, 2010.

[11] C. Seifert, "Analyzing Malicious SSH Login Attempts," *Security Focus, Infocus 1876*, 2006. [Online]. Available: http://www.securityfocus.com/infocus/1876.

[12] F. Fisher, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel, "Large-scale network monitoring for visual analysis of attacks." 2008.

[13] J. R. Goodall and M. Sowul, "VIAssist: Visual analytics for cyber defense," in *IEEE Conference on Technologies for Homeland Security, HST'09*, 2009, pp. 143–150.

[14] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The Nepenthes Platform: An Efficient Approach to Collect Malware." 2006.

[15] J. Blasco, "An approach to malware collection log visualization." 2008.

[16] "carniwwwhore." [Online]. Available: http://carnivore.it/2010/11/27/carniwwwhore.

[17] "Dionaea honeypot." [Online]. Available: http://dionaea.carnivore.it/.

[18] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Connection Protocol," *Internet Engineering Task Force, RFC 4254*, 2006.

[19] "Kippo honeypot." [Online]. Available: https://code.google.com/p/kippo/.

[20] "psyBNC." [Online]. Available: http://www.psybnc.at/about.html.

[21] "pscan2." [Online]. Available: http://packetstormsecurity.org/files/22277/pscan2.c.html.

[22] "SHOUTcast." [Online]. Available: http://www.shoutcast.com/.

[23] "phpMyAdmin." [Online]. Available: http://www.phpmyadmin.net/home_page/index.php.

[24] "fudp." [Online]. Available: http://sourceforge.net/projects/usoft/.