How to build a honeypot System in the cloud

Marius Alin Lihet
PhD Student: Technical University of Cluj Napoca
QSL, Information Security Officer
Cluj Napoca, Romania

Vasile Dadarlat
Professor: Technical University of Cluj Napoca
Information technology Dept
Cluj Napoca, Romania

***Abstract* — This article will explain and will act as a guide to implement a cloud based honeypot using the Kippo honeypot application suite. It will also underline the importance of having an honeypot and will illustrate statistical and real data collected during the implemented system used for this article.**

Keywords — Honeypot, Honeypot systems in the cloud, Information Security, Information Security Appliances, Kippo

I. Introduction

Computers are now valued tools due to the ability of communicate with each other and exchange information with other computer using the telecommunication networks. Nowadays we do have different large WAN networks that communicate between each other using different medium types like copper networks, Fiber networks or radio networks. One of the biggest and well known of such networks is the Internet Trough Internet millions of computers communicate and facilitate services like website and email to billions of people all around the world. It is a fact that because the Internet is such a huge place to discover, internet users have different intentions, and it is quite common to find user with bad intentions around all Internet. So there are users that try communicate with other computer systems and use them without being allowed to access those systems.

These users are so called hackers. There are different types of hackers and they do have different intentions when they look after a system. A very common example is (for hacker) trying to access and steal confidential data or financial data from a big company's servers. Some different type of hackers called hacktivists will try only to deface the websites and spread their word and their beliefs.

The Internet at the beginnings was designed as a mean of communication and sharing the resources, the malicious aspect of it was not taken into consideration. That's why some new types of security devices and security software have to be implemented these days in order to make it safe

But if an infrastructure has value in a view of a hacker regular information security appliance will not be enough to mitigate the hacker attacks. Definitely it is a must to have Firewall, Intrusion Detection Systems or Intrusion Prevention Systems if the Information Security Infrastructure has high value and the services that are using it are important from the confidentiality point of view. We can implement a new layer of security by having a honeypot system implemented. The honeypot system is a "trap", a bogus system that has no production value, but in the same time replicates an existing system but using fake data. This article describes how to implement a honeypot system. Also we will be covering the most common information security hardware and software used these days, and the way a honeypot interacts with it

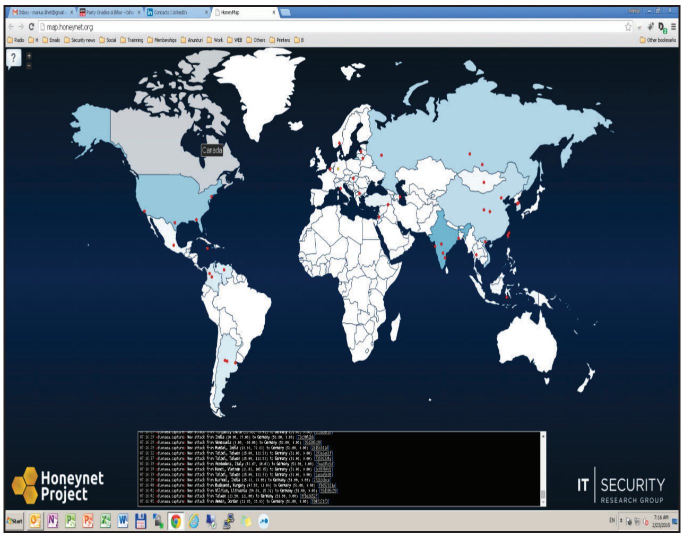
The article will cover all the steps needed to implement a honeypot system in the cloud using a Linux Ubuntu based Virtual Private Server (PVS).. The option of using a cloud VPS has become an affordable and safe option in regards of having in house capabilities. We can opt also for node balancers so we have an increased workload.

Another advantage is that the backup and restore options of a VPS in the cloud are fast and complex. Scaling the systems is also an easy task to do, is a matter of minutes to increase

II. Honeypots Systems, Basic Concepts and Classification

A honeypot is a fake system, a rogue system that accomplishes the same tasks like a real one that is in the production environment. The data from this system is completely fake. This system has no value for the production environment so if an attacker compromise it, the infrastructure is not affected in any way.

Honeypot systems can be classified according to the level of interaction in



```
07:16:29 <dionaea.capture> New attack from India (20.00, 77.00) to Germany (51.00, 9.00) [illegible]
07:16:29 <dionaea.capture> New attack from Venezuela (8.00, -66.00) to Germany (51.00, 9.00) [illegible]
07:16:29 <dionaea.capture> New attack from Mumbai, India (18.98, 72.83) to Germany (51.00, 9.00) [illegible]
07:16:32 <dionaea.capture> New attack from Taipei, Taiwan (25.04, 121.53) to Germany (51.00, 9.00) [illegible]
07:16:37 <dionaea.capture> New attack from Taipei, Taiwan (25.04, 121.53) to Germany (51.00, 9.00) [illegible]
07:16:39 <dionaea.capture> New attack from Pontedera, Italy (43.67, 10.63) to Germany (51.00, 9.00) [illegible]
07:16:39 <dionaea.capture> New attack from Hanoi, Vietnam (21.03, 105.85) to Germany (51.00, 9.00) [illegible]
07:16:39 <dionaea.capture> New attack from Taipei, Taiwan (25.04, 121.53) to Germany (51.00, 9.00) [illegible]
07:16:39 <dionaea.capture> New attack from Kurnool, India (15.83, 78.05) to Germany (51.00, 9.00) [illegible]
07:16:40 <dionaea.capture> New attack from Budapest, Hungary (47.50, 19.08) to Germany (51.00, 9.00) [illegible]
07:16:42 <dionaea.capture> New attack from Vilnius, Lithuania (54.68, 25.32) to Germany (51.00, 9.00) [illegible]
07:16:42 <dionaea.capture> New attack from Taiwan (23.50, 121.00) to Germany (51.00, 9.00) [illegible]
07:16:45 <dionaea.capture> New attack from Amman, Jordan (31.95, 35.93) to Germany (51.00, 9.00) [illegible]
```

Fig. 1. An example of cloud honeypot - Honeypot.org

A. Low interaction Honeypot

These systems will only emulate some important services like SSH, HTTP or FTP. They will be very easy to be discovered by attackers and also they do provide the lowest degree of security overall. They still have an advantage, they are quite easy to be installed and maintenance/monitoring part is quite easy and fast. They are used more for demonstration and educational levels.

B. Medium Interaction Honeypot,

Kippo Honeypot is a medium interaction honeypot. That means that it is still a software instance running on an operational system but it will blend so well into the operating system that will be very hard to be discovered by the attackers. These are the most common honeypots in use and this article will explain in deep how we do install such a system using the Kippo software.

C. High Interaction Honeypot

The main characteristics of a high interaction honeypot is that it will be using a real operating system, as in real life, but all will be operated, analyzed and monitored as being a honeypot system. It does use also more hardware resources and also does inflict a major grade of risk in the infrastructure where implemented. In order to get results and minimize the risk they are monitored 24/7.

Based on the implementation criteria we can classify in the following categories

A. Production Honeypots

Are those honeypots that are deployed in production having an active part in in the overall cyber security defense of an organization. They will be closely monitored and maintained and will have an important part into an ISMS (Information Security Management System). The risk will be kept to a moderate level and all the gathered information will be analyzed and based on the findings they will be used as inputs for other information security devices.

B. Research Honeypots

Are those honeypots used to gather statistical data; mostly of them are used for research only purpose. They are intentionally kept running with a highest level of risk attached to them, in order to expose them to a highest spectrum of attacks and situations.

According to the location of the honeypots:

A. Local Honeypots

Honeypot systems installed in a local domain , being part of a local infrastructure. They are difficult to implement and usually it takes a medium to long period of time from planning to testing phase. Based on the defined scope they will be in various configurations interacting directly with different other security appliances. They are the most common type of honeypots for a company that doesn't rely on SAS (Software as Service) or they don't use the Cloud services at a*ll.*

B. *Cloud Honeypots*

are the honeypots that are deployed in the Cloud, with multiple advantages, but also restrictions. They are used mostly by companies that have at least part of the infrastructure located in the Cloud. They do pose a major advantage that one that are easy to be installed, fast to be deployed or restored in case of corruption. They will usually have an indirect connection to the monitoring component of itself. Applying encryption to all the communications between the active component (the honeypot system deployed) and the monitoring component is crucial.

Honeynet.org (Fig. 1.) is one of the the first big honeypot project online. It is a benchmarks in the field of honeypots and they managed to have their own honeypot software and a huge community to support them. By participating into the honeynet.org project you can install the honeypot system into a local system or buy a separate device that runs the honeypot by itself. These options did make possible that honeynet.org have thousands of installation around the world.

III. CASE SCENARIO: IMPLEMENTING A HONEYPOT IN A CLOUD VPS

Hardware and prerequisites:

The easiest way to implement a honeypot for test reasons and without compromising the security of your infrastructure is to do it in a cloud. For this example we will use a cloud VPS loaded with Ubuntu 14.4 LTS distribution. As a honeypot software we will use one of the most simple to implement honeypots: KIPPO.

KIPPO is able to emulate a SSH service able to listen and log all the login attempts. But before we go deeper into the technical side of it, first we will cover the 3 major phases that a hacker usually follows in a hacking scenario.

A. Reconnaissance & Scanning

Mostly of the time the hackers already have a target in mind, but there are also plenty of hackers that just try looking for targets randomly. Usually they search for weak or not well maintained servers (are easy targets), and can be transformed into proxies or can be used as part of more complex botnets. In this article we will cover the scenario that is the most commonly used in the hacking world: using automated tools to do the scanning part. There are plenty of scanning tools available for free, that are able to scan entire class of ip's and identify the live hosts in a matter of minutes. When we are conducting a scan more in deep we will also be able to identify the open ports and figure out the Operating system of the host.

B. Exploiting phase

After the hackers have the details (IP address and open ports) of the potential hosts that are live in the Internet, will go to the next phase and will try to exploit some vulnerabilities of the existing victims. In our case our Linux based VPS will have SSH port open. One of the most common strategy is to run again automatic tools like brute forcing or dictionary attacks (the most common passwords).

C. *Maintaining access and hiding tracks*

In case the hacker manages to hack into the system, he will try to hide his track by deleting the logs but also in the same time will try to leave a backdoor, so in the future he will be able to access the compromised system in an easier way.

IV. Important aspects of implementation

For purposes related to this article, the KIPPO Honeypot will be installed into an Ubuntu VPS in the cloud. The VPS will be setup to have the ssh port open. Because the majority of the hackers tools scan for default ssh port :22 , we will assign this port for KIPPO and we will change our ssh port to 5436. By default KIPPO listen to port 2222 so we will also need to change this one to 22.

A. *Update and Upgrade the Ubuntu Cloud VPS*

It is really important to start and also maintain an updated server in order to avoid potential hacks of the server using exploits that are not so well know or using 0-days attacks.
Log into the Cloud VPS:

```
ssh root@ipaddresshere
```

The following commands will be used to be sure that our server is updated to the latest official release

```
apt-get update
```

and

```
apt-get upgrade
```

B. *Editing the ssh configuration*

Using the nano or another editor we will modify the ssh configuration. From the console, run the follow command: nano /etc/ssh/sshd_configWe do need to change the port number, in this example will be 7389:# What ports, IPs and protocols we listen for Port 7389. Next we need to save the file using ^O and exit to terminal. To complete the ssh port changing operation we do need to restart ssh using this command: reload ssh

C. *Installing Kippo dependencies*

There are 2 important dependencies that we need to be sure that we have installed before installing Kippo: phyton dependencies (kippo is a phyton script) and subversion (we will use it to install kippo), because our VPS is an Ubuntu VPS, we will use the apt-get package manager; so type the following commands: apt-get install python-dev openssl python-openssl python-pyasn1 python-twisted apt-get install subversion

D. *Creating the Kippo User*

With the following command we will add the kippo username, assign the login shell and create in the same time the home directory:

```
useradd -d /home/kippo -s /bin/bash -m kippo -g sudo
```

E. *Changing kippo server port to 22*

Now we will have to change the kippo port to 22 so an attacker will see the server as a normal one with the SSH (port 22 open) By default kippo does listen on port 2222. But on linux servers only the root username is able to run ports under 1024. Also the new version of kippo does not allow to run under root username by default for security reasons. In order to be able to run kippo on port 22 we will use AuthBind; to accomplish that use the following commands:

```
apt-get install authbind | touch /etc/authbind/byport/22
```

In this moment still kippo doesn’t have the permisions to this file so we will allocate them by running the command from bellow: chmod 777 /etc/authbind/byport/22

F. *Download and configure KIPPO*

As explain before we will be using the new created username kippo from now on to install it in the home folder. We will type the following commands:

```
su kippo | cd
svn checkout http://kippo.googlecode.com/svn/trunk/ ./kippo
```

At this point we will change the default port of kippo 2222 to 22.Remeber from now one the real ssh port of our server is the one that we did setup before : 7389, so for future we will need to use this port to connect via ssh. So to do that we will edit the kippo.cfg file from kippo directory:

```
cd kippo
mv kippo.cfg.dist kippo.cfg
```

Now we will edit the file . It is also a good ideea to change also the hostname, if we leave it the default one (NAS3) an experienced attacker that is aware of the technical characteristic of Kippo will understand that the server is an honeypot. We can change it into a webserver , mail server . In our case I did change it to **websrv.** It is important to have the services running in this case. It will be hard to believe that it is an webserver if for example the port 80 is closed and only the ssh port will be open. For that we will edit it using nano:

```
nano kippo.cfg
# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment.
# (default: nas3)
hostname = websrv
```

Here we do also modify the line with the ssh_port=2222 to:

```
# Port to listen for incoming SSH connections.
# (default: 2222)
ssh_port = 22
```

At this point we will need to restart the ssh service in order to take effect

```
Ssh service restart
```

G. *Starting KIPPO*

The last step before starting the script is to modify the start.sh file by adding the authbind feature. The start.sh file will look like this:

echo -n "Starting kippo in background..."
twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid
we will change it to
echo -n "Starting kippo in background..."
authbind --deep twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pi

H. *KIPPO logging capabilities (Kippor-graph)*

The main log in which all the activity of the honeypot will be logged is located into /kippo/log and it it the kippo.log file. From here we will be able to read and understand from the date and time, ip's and combinations of the User/Passwords that have been tried. It is not the most easy way but sometime it is enough.

By installing the kippo-graphs we will unleash the real value of the data interpretation. From a text file will be able to aggregate and visualize by using different templates.

Fig. 2. Kippo Honeypot graphs

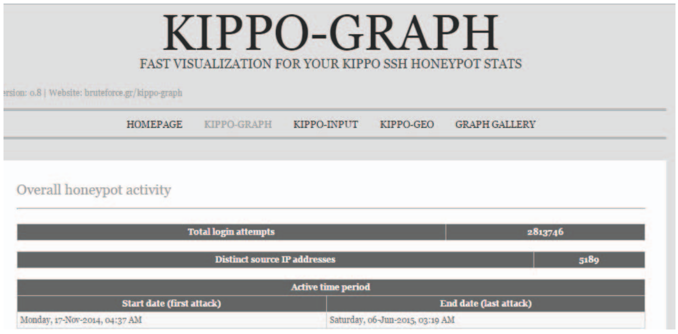
V. TESTS AND RESULTS

After the installed VPS was in production for 8 months we already have been able to capture a significant amount of data having a number of almost **6000 unique IPs** from where attackers tried to login into the system that pretend to be a FeeNAS. More than **3 millions** login attempts have been made, **4000** of them trying to connect using the combination of Admin and 123456 as login credentials.

We can go deeper and setup the Kippo with a common password like **Admin123456** and then we will be able to see what the attackers are doing after they are inside of the honeypot, but this scenario is beyond the scope of this article.

In order to see the graphs we will use our browser and browse to the: http://ip-of-the-vps/kippo-graph. Bellow are some examples of graphs that we can generate. We do have also the option to download the statistics as csv file containing the full activities of the honeypot. The data in this format can easy be fed into different other systems.

Main page of the Kippo System



KIPPO-GRAPH

FAST VISUALIZATION FOR YOUR KIPPO SSH HONEYPOT STATS

ersion: 0.8 | Website: bruteforce.gr/kippo-graph

HOMEPAGE KIPPO-GRAPH KIPPO-INPUT KIPPO-GEO GRAPH GALLERY

Overall honeypot activity

Total login attempts	2813746

Distinct source IP addresses	5189

Active time period	
Start date (first attack)	End date (last attack)
Monday, 17-Nov-2014, 04:37 AM	Saturday, 06-Jun-2015, 03:19 AM

Fig. 3. Main page of Kippo- Graph

Top 10 username-password combination

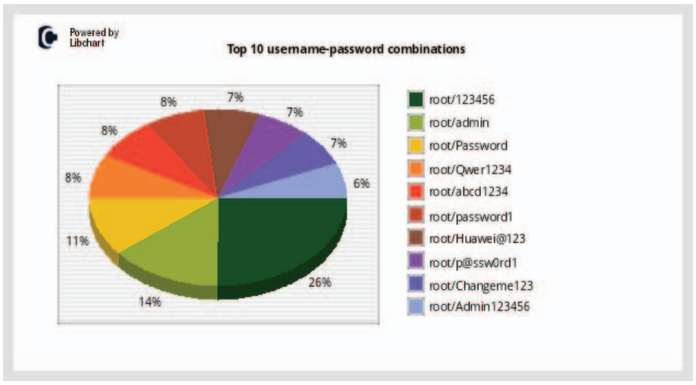


Fig. 4. Top 10 most tried username/password combinations

Number of attempts per unique IP

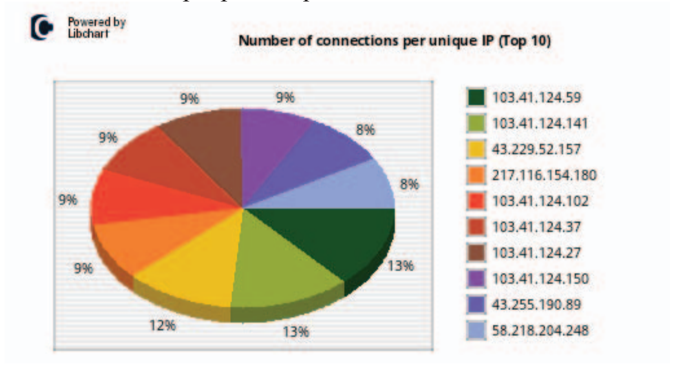


Fig. 5. Number of connections per unique IP

VI. CONCLUSIONS AND FUTURE WORK

Does a honeypot system installed bring an extra layer of security?. Defintelly that it does and is just playing a a small part in a information security infrastructure with different layers and multiple redundant appliances. A honeypot can bring some value to maintain and increase the information security momentum. It can bring business value only when the data produced is having value to the owner of the honeypot system.

There are pro and cons in having a Honeypot System installed and used in production.

The biggest advantage is that the data collected by a owned honeypot system is not just a result of a statistical data but can be a custom set of data, tailored to custom scenarios. So if we are using a honeypot system we can automate the activity of it and use the outputs of it in form of lists of ip's as inputs for other security devices, having a blacklist which is updated live and it does contain a small number of false positives. For sure we are getting these lists from different appliances providers that are being uploaded transparently to the devices. But these set of data is just a statistically sorted data and might not apply to your needs. Combating a Advanced Persistent Threat will be more easy if you will have an active honeypot system in your ISMS. Another advantage is the list of real usernames and passwords, a custom list that usually is hard to get and trust. In conjunction with a firewall it will take the security to the next level having real live data it is the real advantage.

One of the biggest disadvantages is the fact that it really has to be closely monitored and maintained. Also if not properly configured can leave traces to the administration part of it. If an hacker gain access of an honeypot it will be used mostly of the time as proxies or starting points of launching attacks resulting in having a system that does more bad than good.Another disadvantage is that a honeypot alone without no other security devices IDS or firewall bring no advantage and cannot be used standalone. They will do more bad than good in

References

[1] Honeypot.org [ONLINE] Available at:http://www.honeypot.org. [Accessed 22 February 2015]

[2] Sans.org [ONLINE] Available at :http://www.sans.org/security-resources/idfaq/honeypot3.php [Accessed 22 February 2015]

[3] Niels Provos and Thorsten Holz (2007) Virtual Honeypots: From Botnet Tracking to Intrusion Detection. London: Addison-Wesley Professional

[4] Maria Cruz-Cunha; Irene Portela (2014) Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance : IGI Global

[5] Al-Sakib Pathan (2014) The State of the Art in Intrusion Prevention and Detection : Auerbach Publications

[6] Corey Schou; Steven Hernandez (2014) Information Assurance Handbook: Effective Computer Security and Risk Management Strategies : McGraw-Hill Osborne Media