

Information Security in Healthcare Organizations using Honeypot Intrusion Detection System

Aastha Yadav¹, Sarthak Raisurana², N.Ch.S.N. Iyengar³

^{1,2,3}Vellore Institute of Technology University, Vellore-632014, Tamil Nadu, India

¹aasthay1705@gmail.com, ²sarthak.raisurana@gmail.com, ³nchsnir@vit.ac.in

ABSTRACT

Healthcare Organizations have seen an alarming rise in cyber-attacks in the recent years. One way a hacker could get control was by breaking into a medical network to gain access over the active medical devices that patients rely on for their survival. The network model in this paper proposes a low-interaction and a medium-interaction honeypot based intrusion detection system using Dionaea and Kippo SSH to secure our internal network and study the activities of the intruders. The paper also looks at a possible Metasploit attack and Brute force attack logged by Dionaea and Kippo SSH which prepares the Malware Analysis report of the suspicious file downloaded.

Keywords: *low-interaction honeypot, medium-interaction honeypot, Dionaea, Kippo SSH, Metasploit attack, Brute force attack*

1.0 INTRODUCTION

Healthcare organizational networks are vulnerable to a variety of exploits that can compromise the patient data and the hospital's internal operations. They are especially vulnerable to malware and other cyber-physical attacks. The challenges of securing networks in the face of intruders have become overwhelming and are still growing. Intruders can attack hospital data networks and medical device data systems. The transfer of diagnostic information could be tampered with by a cyber-attack on the medical device data system. Vital utilities of a hospital building, such as water, oxygen, electricity, etc. could be shut down by a cyber-attack on the controls. Honeypot Intrusion Detection System (IDS) is an attempt to overcome shortcomings of traditional systems towards a more safe and secure environment. A honeypot is a program, machine, or system put on a network as bait for attackers. The idea of the system is to deceive the attacker by making the honeypot seem like a legitimate system. Honeynets are useful to expose current vulnerabilities of the organization. Honeypots return highly valuable data that is much easier to interpret than that of IDS (Intrusion Detection System). The information gathered from honeypots is used by the administration to protect their system from potential attacks [2].

Kippo is a honeypot written in Python. It is a medium-interaction honeypot and is used to log brute force attacks. The entire shell interaction performed by the attacker is logged by the honeypot. Dionaea is the successor of Nepenthes and is used as a malware capturing honeypot initially developed under The HoneyNet Project's 2009 Google Summer of Code. Dionaea's job is to trap malware exploiting vulnerabilities exposed by services offered over a network, and ultimately obtain a copy of the malware in the binaries and analyze them. Server Message Block (SMB) is the main protocol offered by Dionaea. SMB has been able to capture remote exploitable bugs, and is a very popular target for worms. The system supports HTTP and HTTPS on port 80. Dionaea provides a basic FTP server on port 21. It allows creation of directories, and uploading and downloading of files. Dionaea provides a TFTP server on port 60 which can be used to serve files. Dionaea implements the Tabular Data Stream protocol which is used by Microsoft SQL Server., Dionaea uses LibEmu to detect and evaluate payloads sent by attackers in order to obtain a copy of the malware. LibEmu is used to detect, measure, and if necessary, execute the shellcode. Shellcode measurement / profiling are performed by executing the shellcode in LibEmu VM, and recording API calls and arguments [3].

2.0 MOTIVATION

Ethical health research and privacy protections both provide valuable benefits to society. Health research is vital to improving human health and health care. Protecting patients' sensitive data involved in research is

essential to ethical research. The primary reason for protecting personal privacy is to protect the interests of individuals. Protecting the security of data in health research is important because healthcare organization requires the collection, storage, and use of large amounts of personally identifiable health information, much of which may be sensitive and potentially embarrassing. Healthcare data has been breached to sell for ransom as the recent trends of cyber-crimes have shown. Recent trends have also shown attacks being performed requires a complete analysis of hacker's activity to be tracked to secure the very sensitive healthcare information. This calls for building an Intrusion Detection System (IDS) such as honeypots that keeps track of hacker's moves and to move towards more secure network architecture.

3.0 BACKGROUND RESEARCH

- Existing IDS and their gaps
- Honeypot as an IDS
- Existing uses of honeypots

3.1 Healthcare Organizations: Vulnerable sector of cyber-attacks

About 93 major cyber-attacks hit healthcare organizations in 2016, up from 57 in 2015, new research shows. Sophisticated attackers are now responsible for 31% of all major HIPAA data breaches reported in 2016, a 300% increase over the past three years, according to the report. Cybercriminals were responsible for 10% of all major data breaches in 2014 and 21% in 2015 [1].

According to a research earlier this year with the Ponemon Institute, within healthcare and pharmaceutical companies, an average of 30% of outbound web traffic is encrypted today and these organizations expect that percentage to increase to 48% over the next 12 months. Indeed, healthcare organizations have been taking a multipronged approach, using a combination of people, policies and technical controls to combat cyber-attack and protect information, with encryption being considered as a best practice for protecting the electronic medical records (EMR) and personal health information (PHI) of patients [4]. A study by Experian found that Electronic health records remain likely to be a top target for hackers [10]. Despite the fact that healthcare organization will more likely be taking far harsher cyber security precautions which will ward off lethal attacks, the growing proliferation of cheap, connected Internet of Things (IoT) devices will provide an easy gateway for criminals to illegally access critical information and personal data [4].

In a recent SANS survey, the findings of this study indicate that 7 percent of traffic was coming from radiology imaging software, another 7 percent of malicious traffic originated from video conferencing systems, and another 3 percent came from digital video systems that are most likely used for consults and remote procedures. In this study, most of the malicious traffic passed through or was transmitted from VPN applications and devices (33 percent), 13 whereas 16 percent was sent by firewalls, 7 percent was sent from routers and 3 percent was sent from enterprise network controllers (ENCs). This indicates that the security devices and applications themselves were either compromised, which is a common tactic among malware families, or that these "protection" systems are not detecting malicious traffic coming from the network endpoints inside the protected perimeter. If they are not detecting, they are not reporting—and that means they are out of compliance with privacy and security regulations for patient data. This report, however, shows that the systems were compromised for long periods of time, and even when alerted to their system's actions, the organizations did not repair the vulnerabilities [5].

Here's a list of sample medical equipment which when hacked by an intruder can cause serious situations.

X-ray generator: Medical X-ray machines are used to take pictures of dense tissues. The radiation from X-ray machines are highly penetrating, ionizing radiation, therefore they can be very dangerous. If a hacker is able to increase the dose or radiation exposure, a patient could be overexposed to enough radiation that it results in permanent destruction of either hair or sweat glands, or skin with a resulting scar. The irreversible changes are categorized as radiation dermatitis, chronic radiation dermatitis and radiation cancer. Acute exposure is a one-time event with high-level dose (>100 rem or 1 Sievert) and symptoms appear quickly (within days or weeks). Also, Long-term effects of chronic exposure to ionizing radiation include carcinogenesis, life span shortening,

and cataract formation with the principle delayed effect being an increased incidence of leukemia and other cancers.

Dialysis Machine: A dialysis machine is designed to replace many of the kidney's important functions and restore a patient's blood to a normal, healthy balance by filtering out harmful wastes, salt, and excess fluid. A doctor creates a vascular access into the patient's blood vessels so the patient can be connected to the filtering machine during each hemodialysis session. In a dialysis machine, two independent failures are required before the machine operates in an unsafe condition. A single failure is probably not life threatening. Two independent failures, however, would be life threatening-not a coincidence. If the hacker spoofs extracorporeal venous pressure decrease, biomedical technician assumes disconnect of needle from patient vascular access.

Magnetic Resonance Imaging (MRI): A common breach of MRI safety occurs when a metal object is attracted by the static magnetic field. Although generally considered very safe, if a hacker is able to tamper with the MRI controls, a person can be struck, injured, or trapped against the magnet by the object. The MRI may even be damaged by slamming into the magnet or struck by a rapidly accelerating object. These very high-strength magnetic field (or "missile-effect") accidents, where ferromagnetic objects are attracted to the center of the magnet have resulted in injury and death. In one case, a six-year-old boy died during an MRI exam, when an oxygen tank was pulled across the room and crushed his head. An intruder can spoof the MRI to drain the magnetic field, causing the MRI to cease its operation. The MRI machine can also be manipulated to associate a patient's medical file with another patient's image, resulting in the delivery of the diagnosis to the wrong patient. The strength of the magnetic field of the MRI can be overridden possibly causing heat and tissue burns in the patient.

There are many more such attacks that can be performed on medical equipment such as infusion pump, barcode scanning systems, Medical imaging systems by simple spoofs and altering or modifying the frequencies or data causing fatal damage to the health of a patient [9]. The sensitive information collected of patients upon admission in hospitals is also at risk of theft and modification, along with their private medical records.

3.2 Existing Intrusion Detection Systems

In order to monitor the devices for possible points of attack, an Intrusion Detection System (IDS) can be used to identify a threat and notify someone in the event of an intrusion^[16]. An IDS is more like an alarm system that makes an organization aware of threat. It doesn't prevent threats or remove malicious files. It helps the organization to better defend itself^[17]. Additionally, an IDS can help improve security configurations of the network. The difference between different types of IDSs lies in their placement in the network. The different models are classified by the process the system uses to determine an attack.

Network-based Intrusion Detection Systems (NBIDS) are just what the name implies, "Network Based". This system uses a device that is directly connected to a network segment to monitor traffic flows. The device uses these traffic flows as it's data source to determine whether the traffic matches a known attack signature or pattern. The three main signatures that the NBIDS uses are; attack text string, port signatures, and header signatures. By using the network as a data source, the NBIDS give the ability to monitor entire segments of the network for malicious behavior. Although the NBIDS is good for detecting broad network attacks or threats, it does have some drawbacks. Because the system is monitoring the network, it may not detect isolated attacks or threats. Therefore, NBIDS isn't as effective for detecting things such as trusted-insider attacks that may only target specific devices. So if one individual machine is compromised, it may not be detected if it isn't passing suspicious traffic over the network. Also, if an attack is disguised in legitimate network traffic such as HTTP, FTP, SMTP, etc., it could potentially be missed. So although the NBIDS does have drawbacks, it can be an effective security monitoring device to complement existing security measures.

Host-Based Intrusion Detection Systems (HBIDS) typically consist of loading software on the system being monitored. The software monitors the system for changes resembling an attack or threat. HBIDS uses log files, auditing agents, communication traffic, system file integrity, suspicious processes, and user privileges to determine threats and attacks. Because the system is monitoring the individual host, it is effective in detecting isolated attacks including trusted-insider attacks [12]. One drawback of the Host-based system is software

must be installed and monitored on individual devices. In a large environment, this could become overwhelming [13].

Intrusion Detection Systems also vary in way they determine an attacks and threat. The most prevalent models used to detect attacks include algorithms for statistical anomaly detection, rules-based detection, and a hybrid of the two. As with the type of IDS, the different models have advantages and disadvantages associated with each. The concept is to deploy the model that is most effective in the environment in which it will be used.

Statistical-anomaly model looks for statistical abnormalities. This model runs under the assumption that abnormal behavior is indicative of a threat. The Statistical-anomaly model uses factors such as log files, audits, file/folder properties, and traffic patterns to determine normal system behavior. The key to the statistical-anomaly model is what the systems considered normal behavior. Also, we must determine how much suspicious behavior must deviate for the normal profile to be considered an attack. Deviation from normal activities is the basis for this IDS model. The driving force in anomaly IDS is the use of abnormalities for detection. For this detection to occur, normal behavior must be identified. This normal behavior profile can either be manually created or can be adaptively learned. If the profile is created manually, it must be updated as the system evolves so it doesn't become outdate. Alternatively, if the profile is adaptively learned, there is an increased risk of false positives indicating an attack when one isn't present. Because the anomaly based system works off of a normal profile to detect abnormalities, it is a very customizable model for an organization to use. Along with the customization come high false-positive rates as well as high maintenance to update the "normal" profile.

Rule or Signature-based model: Most attacks are characterized by a sequence of events, making it is possible to create signatures to define these threats. The Signature-based system examines its data source for matches to predefined signatures or activities. The system alarms attack matches to the signatures are found in the data. This model is easier to implement and maintain than the anomaly model. As the system has very specific events that it is searching for, it has a very low false positives rate in comparison to Anomaly based IDS [13]. It can only detect attacks for which it has signatures. This need for signatures causes the system to be unable to detect new threats or "Zero Day" attacks [14].

3.3 Honeypots as IDS

Deploying an IDS in a network and analyzing the traffic is complicated. Identifying the malicious activities and separating it from the normal traffic is time consuming and difficult, which is made simple by making use of honeypots. Most of the traffic to a honeypot is suspicious. A honeypot is a fake computer asset that exists only to alert its owner if it is touched. The only people touching a honeypot or attempting to log into one have malicious intentions. Because all activity is illegitimate, no analysis is needed to tell good traffic from bad.

4.0 Proposed Architecture of Network Design

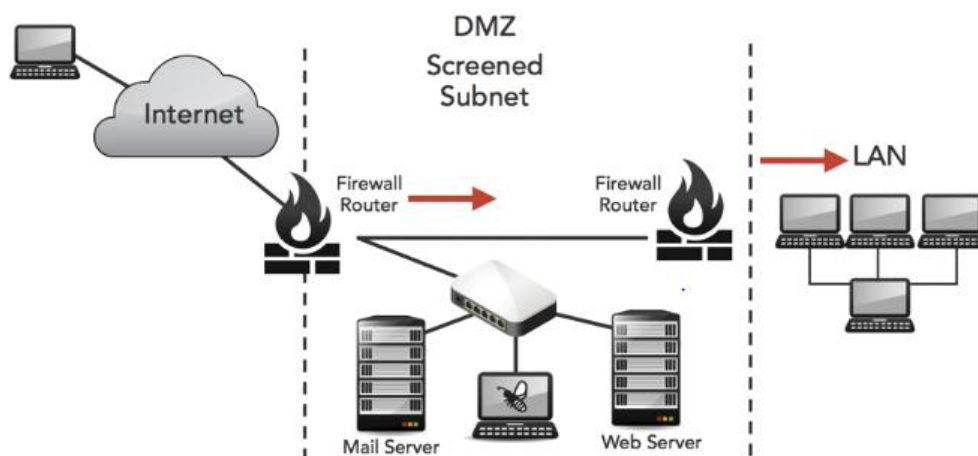


Fig. 1. Architecture of proposed Network Design

The above proposed model will make use of **low-interaction and a medium- interaction Virtual Honeypot** to collect malware in an automated way. A low-interaction honeypot provides only limited access to the operating system. By design, it is not meant to represent a fully featured operating system and usually cannot be completely exploited. As a result, a low-interaction honeypot is not well suited for capturing zero-day exploits. Instead, it can be used to detect known exploits and measure how often your network gets attacked. The advantages of low-interaction honeypots are manifold. They are easy to set up and maintain. They do not require significant computing resources, and they cannot be compromised by adversaries. The risk of running low-interaction honeypots is much smaller than running honeypots that adversaries can break into and control. On the other hand, that is also one of the main disadvantages of the low-interaction honeypots. They only present the illusion of a machine, which may be pretty sophisticated, but it still does not provide an attacker with a real root shell [6].

The aim of this project is to look at an approach to collect malware with the help of honeypots and to monitor activities of the intruder on the shell of a fake system. This will help us build an Intrusion Detection System (IDS) against malware which in the form of botnets can bring down almost any server through Distributed Denial of Service (DDoS), the combined power of many compromised machines is a constant danger even to uninfected sites.

Honeypots in the DMZ, which are exposed to external traffic, will detect external attacks and probes. Given the current amount of noise in the Internet this will probably amount for lots of unimportant probes and scans together with the important ones. On the other hand, honeypots in the internal network would detect internal attacks, either true malicious activity or just bad traffic generated by infected or misconfigured systems [11]. As most of the attacks in a healthcare organization are performed by insiders, it would be ideal to place the honeypot inside LAN to catch any malware or malicious activity.

5.0 Honeypot: An approach to collect malware

The proposed system uses Dionaea and Kippo SSH as the honeypot system to collect malware. Dionaea's intention is to trap malware exploiting vulnerabilities exposed by services offered to a network and to achieve the goal of downloading malware. Dionaea uses libev to get notified once it can act on a socket, or begin to read or write. It can offer services via tcp/udp and tls for IPv4 and IPv6, and can apply rate limiting and accounting limits per connections to tcp and tls connections - if required. Dionaea works on a number of modules. Honeypot listens on all available interfaces in the network on the organization. The curl module is used to transfer files from and to servers, it is used to download files via http as well as submitting files to 3rd parties. The emu module is used to detect, profile and to execute shellcode.

Nowadays worms use API to access services, before sending their payload. To allow easy adjustments to the protocol, dionaea implements the protocols in python. There is glue between the network layer which is done in the c programming language and the embedded python scripting language, which allows using the non-blocking connections in python.

Attackers attempt is not to seek your service, but exploit you; they'll chat with the service for some packets, and afterwards send a payload. Dionaea has to detect and evaluate the payload to be able to gain a copy of the malware. In order to do so, dionaea uses libemu. The part of dionaea which takes care of the network io can create a copy of all in/output run for a connection, this copy is passed to the detection facility, which is a tree of detection facilities, at this moment there is only a single leaf, the emu plugin. The emu plugin uses threads and libemu to detect and profile/measure shell code [7]. Dionaea is considered to be the successor of Nepenthes and the aforementioned visualization tool utilizes its XMPP backend in order to present some statistics of the honeypot's activity. The creator of Dionaea has included some rather basic visualization tools in the software, but these proved to be ineffective when the dataset grows large in size [8].

Kippo is a medium interaction SSH honeypot designed to log brute force attacks and, most importantly, the entire shell interaction performed by the attacker. SSH protocol can be used to perform a secure remote login over an insecure network to access a remote shell. In order to study the activities performed by attackers after they compromised a system with an SSH server, we can use a Kippo honeypot. Kippo allows an attacking entity to attempt a login to the system, believing it is entering into a legitimate SSH session with the server. Now the attacker tries to guess the password and upon successful guessing of the password, the attacker is then moved into a fake system with which they can interact. Kippo SSH honeypot is placed before any administrative systems so the attacker is logged on to it assuming it's a legitimate system. In this fake system,

all interactions with the shell are monitored and recorded. The system also allows the use of wget and other commands commonly used to fetch or download files. The main objective of the implementation is to bring to the attacker the impression of navigating the real system of the organization. IT security can give a view on the basis of the command given by him – for what purpose it was hacked, which data the attacker was trying to get, and what techniques were used [15].

6.0 Implementation

We run our simulations to capture malware on our honeypot system. We present statistics about the collected binaries by analyzing the activity and behavior of malware using dockers to study the types of malware. VmWare workstation is used to create the virtual system and Ubuntu 14.04 is installed on it. The computer and operating system instance that executes the VMware process is referred to as the host machine. In order to run Linux-based honeypots, we are running **Ubuntu 64-bit** as the guest operating system. For each virtual machine, we allocated budget at least 256MB RAM or, even better, **512MB**.

Dionaea gains location of the file the attacker wants it to download from the shell code and it tries the download the file. The protocol to download files via tftp and ftp are implemented in python (ftp.py and tftp.py) as a part of Dionaea. Dionaea can then http/POST the file to several services like CWSandbox, Norman Sandbox or VirusTotal.

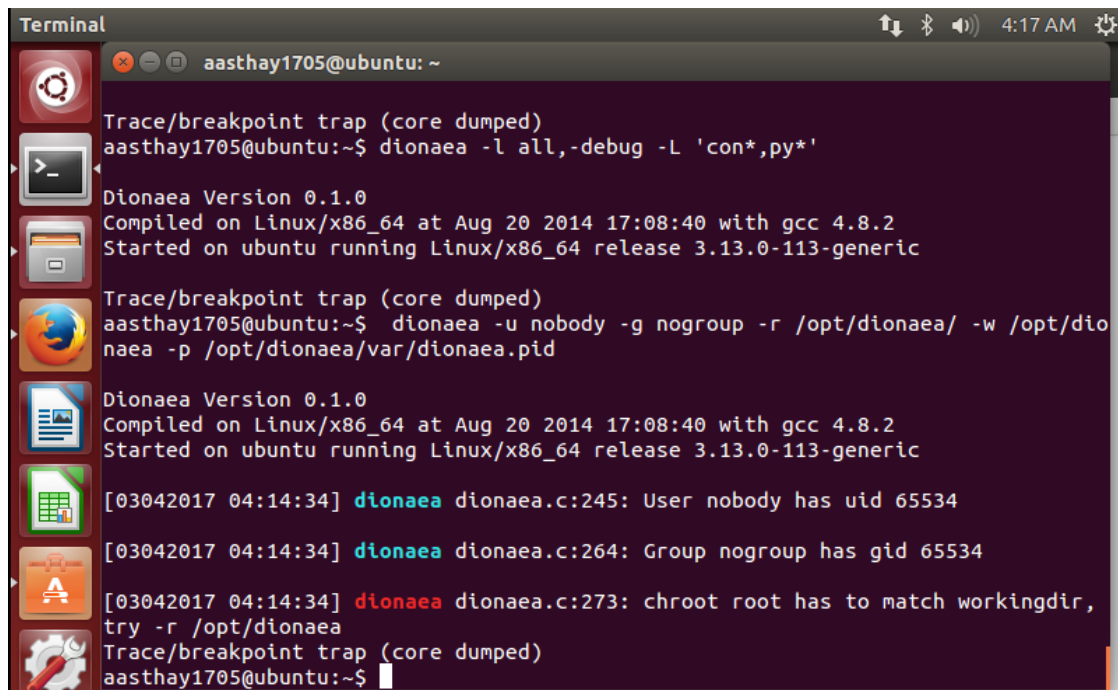
A terminal window titled 'Terminal' with a dark background and light text. The prompt is 'aasthay1705@ubuntu: ~'. The user enters 'dionaea -l all, -debug -L 'con*,py*'' and the output shows 'Dionaea Version 0.1.0', 'Compiled on Linux/x86_64 at Aug 20 2014 17:08:40 with gcc 4.8.2', and 'Started on ubuntu running Linux/x86_64 release 3.13.0-113-generic'. The user then enters 'dionaea -u nobody -g nogroup -r /opt/dionaea/ -w /opt/dionaea/ -p /opt/dionaea/var/dionaea.pid' and the output shows the same version and compilation info, followed by three log messages: '[03042017 04:14:34] dionaea dionaea.c:245: User nobody has uid 65534', '[03042017 04:14:34] dionaea dionaea.c:264: Group nogroup has gid 65534', and '[03042017 04:14:34] dionaea dionaea.c:273: chroot root has to match workingdir, try -r /opt/dionaea'. The terminal ends with a 'Trace/breakpoint trap (core dumped)' message and the prompt 'aasthay1705@ubuntu: ~\$'.

Fig. 2. Dionaea Setup to create user and group

We used our Dionaea setup to perform a metasploit attack and check the dionaea.log to see it logs the information. Attackers do not seek your service, attackers want to exploit you, they'll chat with the service for some packets, and afterwards sent a payload. dionaea has to detect and evaluate the payload to be able to gain a copy of the malware. In order to do so, dionaea uses libemu.

```

[*] Started reverse handler on 192.168.22.128:4444
[*] Using URL: http://0.0.0.0:8080/8yaE0zDnBvrI
[*] Local IP: http://192.168.22.128:8080/8yaE0zDnBvrI
[*] Server started.
[*] Run the following command on the target machine:
python -c "import urllib2; r = urllib2.urlopen('http://192.168.22.128:8080/8yaE0zDnBvrI'); exec(r.read());"
msf exploit(web_delivery) > [*] 192.168.22.129 web_delivery - Delivering Payload
[*] Sending stage (18558 bytes) to 192.168.22.129
[*] Meterpreter session 1 opened (192.168.22.128:4444 -> 192.168.22.129:39843) at 2017-04-24 13:32:33 +0530
sessions -l

Active sessions
=====

```

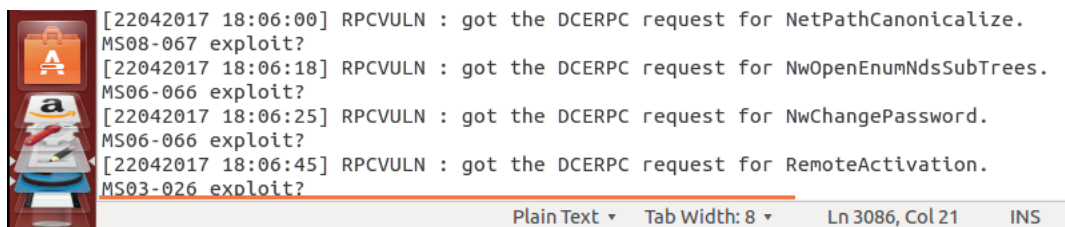
Id	Type	Information	Connection
1	meterpreter	python/python	root @ ubuntu 192.168.22.128:4444 -> 192.168.22.129:39843 (192.168.22.129)

```

msf exploit(web_delivery) > sessions -i1

```

Fig. 3. Metasploit Exploitation performed



```

[22042017 18:06:00] RPCVULN : got the DCERPC request for NetPathCanonicalize.
MS08-067 exploit?
[22042017 18:06:18] RPCVULN : got the DCERPC request for NwOpenEnumNdsSubTrees.
MS06-066 exploit?
[22042017 18:06:25] RPCVULN : got the DCERPC request for NwChangePassword.
MS06-066 exploit?
[22042017 18:06:45] RPCVULN : got the DCERPC request for RemoteActivation.
MS03-026 exploit?

```

Plain Text ▾ Tab Width: 8 ▾ Ln 3086, Col 21 INS

Fig. 4. Dionaea.log spits out the information

The honeypot Dionaea also already supports shell emulation and ftp/http/tftp downloads of malware. From figure 5, the script command generated by this exploit on the target, we are able to get complete control of the system including keystroke logging, turning on microphone to hear and reading or deleting any files on the system. SMB is the main protocol offered by Dionaea. SMB has a decent history of remote exploitable bugs, and is a very popular target for worms. And, Figure 6 is the log file of Dionaea that logs a possible MS08-067 exploit.

Dionaea also has a virustotal module to automatically submit the suspicious files and prepare a malware analysis report for the same. In the figure below (7), the file downloaded from the url in Figure 5, undergoes virustotal scan for malware behaviour.



SHA256:

127d1cd82f4c6697eb371dcf5619481606cdef43139b07692f4beb2b59b3ea8c

File name:

Qarawlfy

Detection ratio:

3 / 55

Analysis date:

2017-04-24 11:43:17 UTC (0 minutes ago)

Analysis

Additional information

Comments

Votes

Antivirus	Result	Update
Avast	JS.Downloader-EQA [Trj]	20170424
ClamAV	Legacy.Trojan.Agent-37025	20170424
Qihoo-360	Script/Trojan.Downloader.4e1	20170424

Fig. 5. VirusTotal scan of the file

Next we track malicious activity of an intruder with Kippo SSH set up on port 22.

From the figure 8, we can see the intruder's activity logged on kippo.log. This includes username and passwords entered including the add, modify and deleting commands run by the intruder on the files of the fake file system. The system also allows the use of wget and other commands commonly used to fetch or download files, as well as a base set of utilities. It saves the files downloaded with wget for later investigation dl folder of logs.

```
root@ubuntu: /home/kippo/kippo/log
GNU nano 2.2.6 File: kippo.log
2017-04-29 18:01:39+0530 [kippo.core.ssh.HoneyPotSSHFactory] New connection: 127.0.0.1:55937 (127.0.0.1:22) [session: 1]
2017-04-29 18:01:39+0530 [HoneyPotTransport,1,127.0.0.1] Remote SSH version: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8
2017-04-29 18:01:39+0530 [HoneyPotTransport,1,127.0.0.1] key alg, key alg: diffie-hellman-group-exchange-sha1 ssh-rsa
2017-04-29 18:01:39+0530 [HoneyPotTransport,1,127.0.0.1] outgoing: aes128-ctr hmac-md5 none
2017-04-29 18:01:39+0530 [HoneyPotTransport,1,127.0.0.1] incoming: aes128-ctr hmac-md5 none
2017-04-29 18:01:42+0530 [HoneyPotTransport,1,127.0.0.1] NEW KEYS
2017-04-29 18:01:42+0530 [HoneyPotTransport,1,127.0.0.1] starting service ssh-userauth
2017-04-29 18:01:42+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] root trying auth none
2017-04-29 18:01:42+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] root trying auth keyboard-interactive
2017-04-29 18:01:51+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] login attempt [root/aasthayad95] failed
2017-04-29 18:01:51+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] root failed auth keyboard-interactive
2017-04-29 18:01:51+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] unauthorized login:
2017-04-29 18:01:51+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] root trying auth keyboard-interactive
2017-04-29 18:01:55+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] login attempt [root/123456] succeeded
2017-04-29 18:01:55+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] root authenticated with keyboard-interactive
2017-04-29 18:01:55+0530 [SSHSservice ssh-userauth on HoneyPotTransport,1,127.0.0.1] starting service ssh-connection
2017-04-29 18:01:55+0530 [SSHSservice ssh-connection on HoneyPotTransport,1,127.0.0.1] got channel session request
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] channel open
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] got global no-more-sessions@openssh.com request
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] pty request: xterm (31, 81, 0, 5
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Terminal size: 31 81
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] request_env: '\x00\x00\x00\x04LS
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] getting shell
2017-04-29 18:01:55+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Opening TTY log: log/tty/201704$
2017-04-29 18:01:56+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] /etc/motd resolved into /etc/mo$
2017-04-29 18:03:49+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] CMD: useradd manthangandhi
2017-04-29 18:03:49+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Command found: useradd manthang$
2017-04-29 18:09:37+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] CMD: ls
2017-04-29 18:09:37+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Command found: ls
2017-04-29 18:13:04+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] CMD: wget http://ftp.gnu.org/gn$
2017-04-29 18:13:04+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Command found: wget http://ftp.$
2017-04-29 18:13:04+0530 [SSHSchannel session (0) on SSHService ssh-connection on HoneyPotTransport,1,127.0.0.1] Starting factory <HTTPProgressD$
2017-04-29 18:13:05+0530 [HTTPPageDownloader,client] Saving URL (http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz) to dl/20170429181304 http_ftp$
2017-04-29 18:13:09+0530 [HTTPPageDownloader,client] Updating realfile to dl/20170429181304 http_ftp_gnu_org_gnu_wget_wget-1.5.3_tar.gz
2017-04-29 18:13:09+0530 [HTTPPageDownloader,client] Stopping factory <HTTPProgressDownloader: http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz>
```

Fig. 6. Intruder's activity on Kippo on the fake file system

Table 1: Kippo's logged contents

Essential Information	Log File
wget commands	dl/
Username attempt	mysql.sql
Password attempt	mysql.sql
Session ID	log/tty/
Session Timestamp	log/tty/
Fake file system contents	honeypfs/

Table 1 includes the location of all the essential information logged about the intruder's activity on the fake file system .

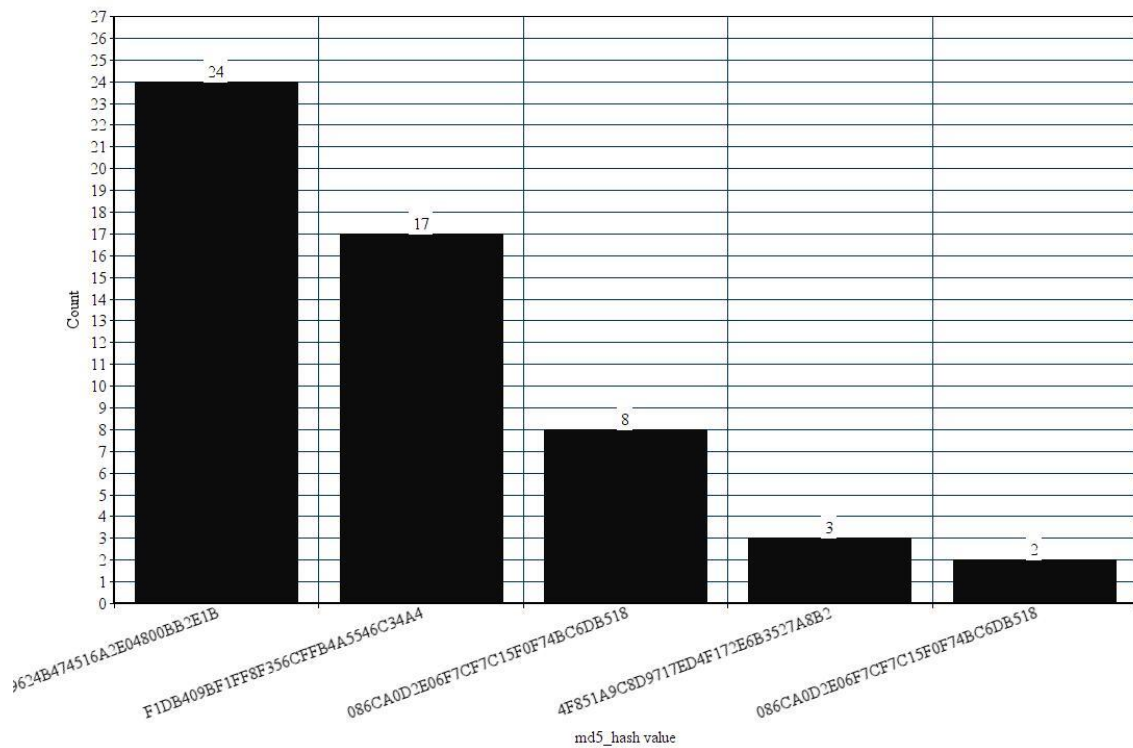


Fig. 9. md5_hash value of the malware downloaded

From the above Fig. 9, we can get the md5_hash value downloaded over a period of 2 months by the honeypot Dionaea up and running in the binaries folder. Dionaea successfully logs all the payloads and shell binds and downloads the malware for further analysis.

Table 2. Honeypot Systems as security mechanism

Honeypot System	Hacker Activity	Potential Vulnerability/Hackable Device	Honeypot Solution
Dionaea	Send an email with a payload using Metasploit Exploit that runs on the medical device	Exposed Networking gear and admin computers of Healthcare organizations to exploit critical medical devices	Logs the downloaded payloads and performs the analysis on malware detected using VirusTotal, CWSandbox etc.
Kippo SSH	Successful SSH and web logins on critical medical devices	Secure Server of the organization	Logs the activity of the intruder including IP activity, geological location, inputs, password and usernames tried on the fake file system.

Table 2 explains how honeypots act as a security mechanism to intruder's activity in an attempt to gain access to medical devices to modify or sweep away patient's critical information.

7.0 Conclusion

The work concludes that Dionaea and Kippo SSH honeypot system act as an effective security mechanism placed in the DMZ to trap malware and to perform reports of malware analysis on logged binaries. They can also be used to log all activities of an attacker on the shell. This system can also be used to send emails to the administrative department in the healthcare organization so that they may be notified of the intruder's activities. It is possible to reduce the damage potential by reducing the number of attack vectors. It is possible for a cyber-physical attack to be detected quickly thereby permitting equipment to shut down before any damage is done. The next step is rapidly restoring hospital systems and equipment to normal operation. Cybersecurity knowledge for healthcare organizations is important for the hospital staff. They have to be trained to understand the aspects of it. A secure and safe environment for patients is the goal of cybersecurity.

References

- [1] Major Cyberattacks on Healthcare Grew 63% in 2016 [Online]. Available: <http://www.darkreading.com/attacks-breaches/major-cyberattacks-on-healthcare-grew-63--in-2016/d/d-id/1327779>
- [2] Bill Hancock, John W. Rittinghouse, "Cybersecurity Operations Handbook: The Definitive Reference on Operational Cybersecurity", Elsevier Science Inc., New York, NY, 2003
- [3] Dionaea – A Malware Capturing Honeypot [Online]. Available: <https://www.edgis-security.org/honeypot/dionaea/>
- [4] Why healthcare is a vulnerable sector for cyber attack – and what can be done about it [Online]. Available: <http://www.appstechnews.com/news/2017/jan/17/why-healthcare-vulnerable-sector-cyber-attack-and-what-can-be-done-about-it/>
- [5] Health Care Cyber threat Report: Widespread Compromises Detected, Compliance Nightmare on Horizon [Online]. Available: <https://www.sans.org/reading-room/whitepapers/analyst/health-care-cyberthreat-report-widespread-compromises-detected-compliance-nightmare-horizon-34735>

- [6] Niels Provos , Thorsten Holz, “Virtual honeypots: from botnet tracking to intrusion detection “, Addison-Wesley Professional, 2007
- [7] Dionaea Documentation [Online]. Available: <http://dionaea.readthedocs.io/en/latest/old/exploitation.html>
- [8] Ioannis Koniaris, Georgios Papadimitriou, Petros Nicopolitidis, Mohammad Obaidat, “Honeypots Deployment for the Analysis and Visualization of Malware Activity and Malicious Connections”, June 2014
- [9] Cybersecurity for Hospitals and Healthcare Facilities: A Guide to Detection Facilities, Luis Ayala, pp 21-29
- [10] Healthcare top target for cyberattacks in 2017, Experian predicts [Online]. Available: <http://www.healthcareitnews.com/news/healthcare-top-target-cyberattacks-2017-experian-predicts>
- [11] David Pérez Conde, “Deploying Honeypots and the Security Architecture of a Fictitious Company”, February 2005, pp 11
- [12] Lata, Indu Kashyap, “Study and Analysis of Network based Intrusion Detection System”, Vol. 2, Issue 5, May 2013
- [13] Intrusion Detection Systems in Hospitals - Infosec Writers [Online]. Available: http://www.infosecwriters.com/text_resources/pdf/IDS_JBarnes.pdf
- [14] Anomaly Based Intrusion Detection System: Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Anomaly-based_intrusion_detection_system
- [15] Tracking Attackers with a Honeypot – part 2 (Kippo) [Online]. Available: <http://resources.infosecinstitute.com/tracking-attackers-honeypot-part-2-kippo/#gref>
- [16] Abimbola, A., Merabti, M., Qi, S., “Nethost-sensor: A Novel Concept in Intrusion Detection Systems”, Eight IEEE International Symposium on Computers and Communications, June 2003. (pp.232-237).
- [17] Allen, J; Christie,A, McHugh, J.,” Defending yourself: The roll of Intrusion Detection Systems”, Software IEEE, 17(5),42-51.

Authors



Aastha Yadav, is a student, currently pursuing Masters in Cybersecurity at Syracuse University, Syracuse NY. Her research interests include Network Security and Knowledge based Artificial Intelligence. She is a computer enthusiast and has keen interest in developing secure software solutions for problems. She loves working on projects with real life application and scope using computer applications.



Sarthak Raisurana is a Business Technology Analyst at ZS Associates, Pune, Maharashtra, India. His interests lie in the fields of Mathematical Cybersecurity (cryptography) and Machine Learning. He likes thinking of abstract and Hypothetical problems, and coming up with solutions to them.



N. Ch. S. N. Iyengar, he is a Professor of the School of Computer Sciences and Engineering at VIT University, Vellore, TN, India. His research interests include Distributed Computing, Information Security, Intelligent Computing, and Fluid Dynamics (Porous Media). He has had teaching and research experience with a good number of publications in reputed International Journals & Conferences. He chaired many International Conferences delivered Keynote lectures, served as PC Member/Reviewer. He is an Editorial Board member for many International Journals like *Int. J. of Advances in Science and Technology*, of SERSC,

Cybernetics and Information Technologies (CIT)-Bulgaria, Egyptian Computer Science Journal-Egypt, IJConvC of Inderscience-China, IJCA (USA) etc., Also Editor in Chief for International Journal of Software Engineering and Applications(IJSEA) of AIRCC, Advances in Computer Science (ASC) of PPH, Guest editor for “Cloud Computing and Services” IJCNS.