International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017
Issue 02, Volume 5 (February 2018) www.ijiris.com

HOST PROTECTION USING PROCESS WHITE-LISTING, DECEPTION AND REPUTATION SERVICES

Rahul Tulaskar*, Sujata Pathak*
Department of Information technology
K.J. Somaiya College of Engineering, Mumbai University, India
rahul.tulaskar@somaiya.edu , sujatapathak@somaiya.edu

Manuscript History
Number: **IJIRIS/RS/Vol.05/Issue02/FBIS10080**
DOI: 10.26562/IJIRAE.2018.FBIS10080
Received: 09, February 2018
Final Correction: 18, February 2018
Final Accepted: 22, February 2018
Published: **February 2018**
Citation: Rahul & Sujata (2018). HOST PROTECTION USING PROCESS WHITE-LISTING,DECEPTION AND REPUTATION SERVICES. IJIRIS:: International Journal of Innovative Research in Information Security, Volume V, 01-12. doi: //10.26562/IJIRIS.2018.FBIS10080
Editor: Dr.A.Arul L.S, Chief Editor, IJIRIS, AM Publications, India
Copyright: **©2018 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited**

Abstract— The Internet or World Wide Web has become prominent platform for business and commerce and is witnessing user growth with increased penetration of mobile Internet. Huge traffic is being generated, some of it being legitimate and the rest being malicious. Hence the implementation and maintenance of Information Security programs is been done .In the age of the Internet, protecting our information has become just as important as protecting our property. Malware authors have found and exploited new zero-day vulnerabilities resulting in damage to end-user system. Ransomware, a malware that has taken malware attacks to a new level by locking files of the affected user and demand Bitcoin payment to unlock those files. On the other hand the Volume and frequency of Distributed Denial of Service (DDoS) attacks have increased. Many unpatched machines without the knowledge of its owners have become a part of Botnets which carry out DDoS attacks. This paper focuses on strategies to be adopted to protect individual hosts from malware attacks and other types of intrusions using Deception, White-Listing and Reputation Services.

Keywords—Deception; Process White-Listing; Botnet; Malware; Command and Control (CnC); Reputation; Security;

I. INTRODUCTION

The Internet or World Wide Web has become prominent platform for business and commerce and is witnessing user growth with increased penetration of mobile Internet. .In the age of the Internet, protecting our information has become just as important as protecting our property. Information security (InfoSec) is the practice of protecting both physical and digital information. We might have personal medical or financial records that we want to keep private from destruction or unauthorized access. Personally Identifiable Information like our Internet passwords, credit card numbers, and banking information should be guarded from getting into the wrong hands. Information is a critical resource for all organizations since information supports business continuity and commerce and helps managers and staff to make appropriate and effective decisions. Securing organizational information and its critical elements, including the systems and hardware that use, store, and transmit that information have become more and more important. Data and the infrastructure on which it resides are assets for organization. The attacker's motive is to steal or cause damage to these assets. To achieve this objective they need access and also need to maintain it. The attackers achieve their objective by generating payload using various exploit kits.

© 2014- 18, IJIRIS- All Rights Reserved

The exploit can be in the form of word, pdf or excel documents. Such malicious documents can be hosted on a compromised website or a crafted link can be sent as a part of phishing mail campaign. The purpose of this research is to present techniques that could be adopted for a Desktop or a Server machine to protect itself from cyber security breach and limit damage to the information stored.

II. LITERATURE SURVEY

When an Anti-Virus (AV) scans a file, it compares the file characteristics with the signature database. If a match is found the file is flagged as malicious. It may happen that AV may not have a signature thus allowing a malware to go undetected until the database is updated. There is a need for a unified threat management and knowledge sharing platform which can provide AV engines from multiple vendors having different signatures. The malware are being created and launched at an alarming rate .In 2016 a new malware specimen emerged every 3.6 seconds and that has reduced to 3.2 in 2017 [1]. The overall number of phishing attacks reported in 2016 was 1,220,523, a 65% increase over 2015[2]. Phishing continues to be preferred method for spreading malware and stealing sensitive data. Phishing can be done through email or a page can be inserted in a vulnerable website. Hence any website presented to the user must be thoroughly checked for any malicious activities even if does raise a suspicion. Thus website reputation is valuable information to stay safe against drive-by-downloads, malvertising, click-jacking and other online frauds.

Malwares manifesting as legitimate code executable programs that spread from computer-to-computer via e-mail, web browsers, etc. are a popular way to launch malicious attacks against users. The Windows operating system is often vulnerable to such attacks, in part because of its popularity among non-business users and in part because o f the vast functionality it provides [3]. Hackers are using malware which targets Linux to build botnets to launch distributed denial of service (DDoS attacks) [4].XOR DDoS, a Trojan malware and the so-called Bill Gates Trojan botnet family of malware are some of them. The malicious Web sites attempt to compromise machines by launching drive-by-download attack which redirects users to exploiting sites and install malware in a stealthy manner, in their machines by exploiting vulnerabilities of their Web browser or plugins. In order to ensure protection from these malicious Web sites, blacklisting such URLs or domains that launch drive-by downloads is carried out by the organization's security team .However, attackers ofuscate the URLs or domains by using URL shorteners, encodings etc to bypass the blacklist. Thus, a universal solution which can detect, block or generate intelligence of even unknown malicious Web sites is critical. Many email campaigns in the form of spear-phishing are launched, where the content of the email successfully convinces the recipient to click a URL redirecting to a malicious Web site or downloading a malicious attachment [5].

Similar to website reputation, some website also generate IP reputation [6]. Poor IP reputation generally means that there's a high probability that a machine communicating to such IP getting infected by malware. Also the infected machine may act as a bot and participate in Distributed Denial of Service (DDOS) attacks. Such connections must be monitored and terminated. Alternatively a process initiating such a connection must be killed. A system affected by a botnet malware will spawn a process that would try to connect to a command and control (CnC) server. From this server it will get commands to perform actions. Information gathering is the first step in any penetration test. The goal of this step is to find and collect information which would help in designing the exploit or discovering vulnerability. Information gathering can be done using search engines like Google, Bing, visiting job forums of the target, querying who is a database. Another way could be visiting the target's website and looking for some information which gives idea about the usernames, roles, locations and technologies used.

It may happen that developers of the website may not have removed comments in the webpage source code that is in production environment. These comments serve as vital pieces of information that may give idea about user accounts for testing, unused URLs, internal IP addresses etc. The good practice is to remove such pieces of information. Alternatively such channel could be used be used as proactive defence to mislead the attacker. Deception uses false information and tricks designed to stop, or block, an attacker's intellectual processes, disrupt an attacker's automated scripts or scanners, delay an attacker's lateral movement or disrupt breach progression. Threat deception is not a way of preventing or stopping an attack but a method to monitor and disrupt the attackers from their motives [7].

III. SYSTEM DESIGN AND IMPLEMENTATION

The proposed system has 3 modules namely Email Attachment Analysis and Site Reputation Checker, Process Monitor using White-listing Approach accessible and Threat Deception. The logic was implemented in Python [12] programming language. The modules need python interpreter installed on user machine .Version of the python interpreter can be between 3.0 to 3.5x.

© 2014- 18, IJIRIS- All Rights Reserved

3.1 Email Attachment Analysis and Site Reputation Checker:

The attachment analysis module checks the inbox of the user for unread emails. It downloads the file(s) in the mail to a predefined folder .This module utilizes API service of VirusTotal as well as information from sites like urlvoid.com. The API of VirusTotal requires a key that can be obtained by creating account on the site. Different Vendors on VirusTotal may give different reputations for a website. For file and URL analysis the public key was generated by creating account on www.virustotal.com. The other way of obtaining website reputation is by using website scraping .Web scraping, also known as web harvesting or web data extraction is a method used for extracting data from websites. Web scraping software obtains publicly available data by accessing the World Wide Web directly using the Hypertext Transfer Protocol, or via a web browser. For obtaining the site reputation using web scraping, the website www.urlvoid.com was used. Even if a malicious URL is shortened, the module expands it and then checks the reputation. The imaplib[13] module in python is used to connect to the Mailbox of the user. Here Google's Gmail service is used for testing purposes. Figure 3.1.1 shows the flowchart for email attachment and site reputation checker. The script checks for every unread mail after 2 minutes of time interval .The script searches for unseen emails and if found determines it's content i.e whether message contains a file or a link.

Following regular expression searches for URL in the message body:
<href="?\'?([^"\'>]*

If the content is a file then, it is downloaded to a predefined folder. After downloading, the script uploads the file to VirusTotal[14] for analysis. Based on the response, if the file is malicious then it is purged from the system after displaying appropriate message to the user. If URL is encountered, the script checks its reputation on VirusTotal using API call and also on urlvoid.com using web scraping. If a shortened URL is encountered, it is expanded to its equivalent original equivalent long URL and then checked for its reputation. The malicious URL is blocked on the host with the following Iptables[10] command:

$iptables -A OUTPUT -d <malicious_url> -j REJECT
Non- malicious files are not purged from the system.
Figure 3.1.1 how the working of the module:

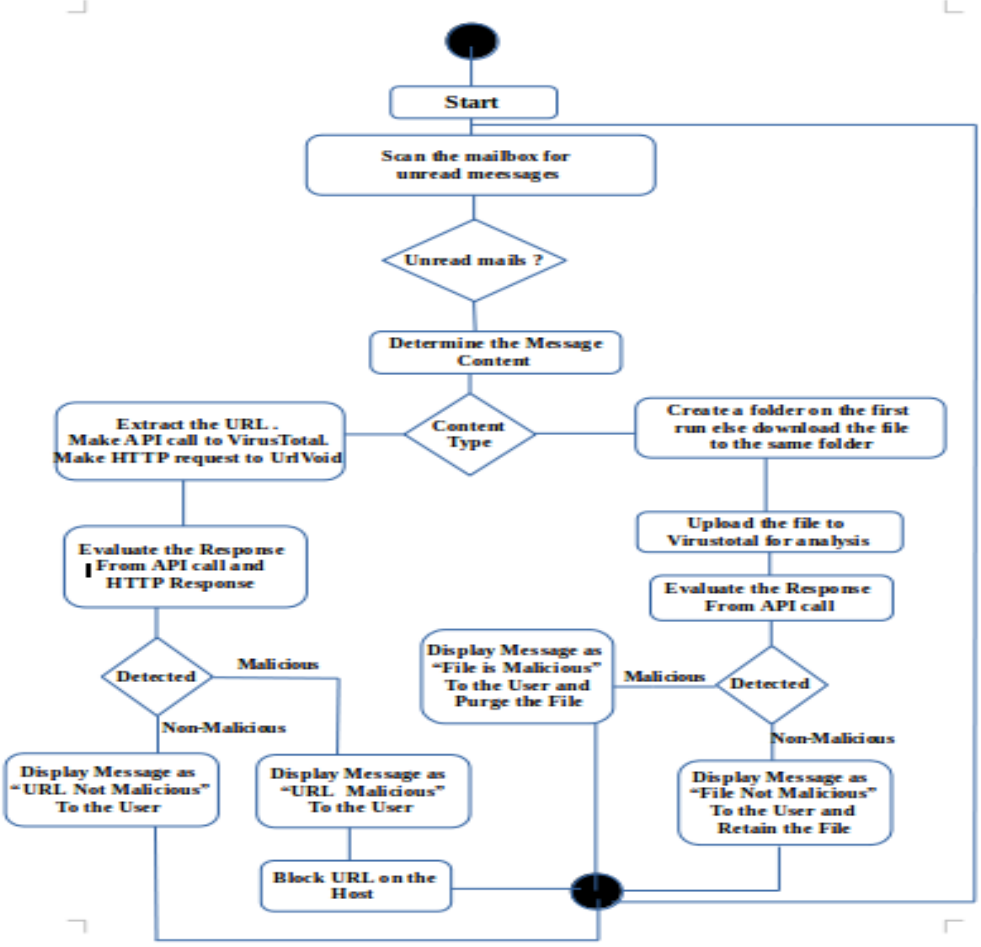

Figure 3.1.1: Flowchart for Email Attachment and Site Reputation Checker

© 2014- 18, IJIRIS- All Rights Reserved

3.2 Process Monitor using White-listing Approach:

The process monitor module first obtains a list of all processes. It then obtains the networking state of the process. The networking state of a process indicates whether the process has opened a socket connection to the Internet. The script in the module checks the reputation of all the IP addresses that the processes are communicating with. The IP reputation is obtained from the website http://www.reputationauthority.org/ by Watch Guard technologies [6]. Based on the reputation and the network state, the process is assigned a threat score and appropriate action is taken .The threat score is assigned on the following basis and corresponding action is to be taken:

Table 3.2.1: Criteria for Assigning Threat Score and Corresponding Action

Networking State	IP Reputation	Threat Score	Action	Process State	Network state
None	-	0	N.A	Good	No Connection
Connection Established	Clean	5	N.A	Benign	Connection to Clean IP
Connection Established	Malformed messages or Suspicious	7.5	Block the IP address on the Host	Suspicious	Connection to Suspicious IP
Connection Established	Spam(poor email reputation) or Malicious	10	Kill the process and block the IP address on the Host	Rogue	Connection to Blacklisted IP

The malware variants are continuously being improvised to evade behaviour or signature-based modes of detection. One of the principal advantages of blacklisting lies in the simplicity of its principle of identifying everything bad causing harm to the operating system, detect and mitigate the same, then allow the free flow of everything else. It has been and continues to be the basis on which signature-based anti-virus and anti-malware software operates. Against unknown menaces like zero-day threats which have yet to be discovered and isolated by security professionals, blacklisting is of very limited or no value. The Anti-virus solution kills the process when the path of the process matches the signature in its database. This is called blacklisting [8]. The main advantage of blacklisting is that it is very easy to maintain a list of a few bad things, identify and take action on it, and accept the rest.

Network connections to a local address (local host/127.0.0.1) are assigned a threat score of zero. If public IP address has good reputation then the process is assigned a threat score of 5 and is termed as benign process with network state as “Connection to Clean IP”. If the public IP address has suspicious or malformed reputation then the process is assigned a threat score of 7.5 and is termed as suspicious process with network state as “Connection to Suspicious IP”. Such IP addresses are blocked on the host. If the public IP address has Spam/Malicious reputation then the process is assigned a threat score of 10 and is termed as rogue process with network state as “Connection to Blacklisted IP”. Such processes are killed and the corresponding IP address is blocked on the host. The script has exempted the process ‘firefox’ from its monitoring. This is called as process whitelisting. In different environment different processes can be whitelisted. For example Anti-virus process or an email client process can be whitelisted. This process is shown in figure 3.2.1.

For blocking the IP address in case of malicious/spam or suspicious reputation, following IP-tables[11] command is executed:

$iptables -A OUTPUT -d <ip_address> -j REJECT

3.3. Threat Deception:

It has been observed that traditional security perimeter devices like IDS and firewall generate considerable number of false positives. Generally manual analysis distinguishes the false positives from real attacks. In order to reduce false positives and to be pro-active in terms of security defence, the concept of deception is used. Consider a web server hosting a website which could be of any category. The site’s source code has a hint embedded as a html comment which is actually a deceptive information. The attacker tries to access an html page mentioned in the comment thinking that it has admin panel and ends up getting caught and eventually locked out from accessing the site. This process basically involves embedding deceptive information into the WebPages, monitoring and analysis of server logs for access to such non-existent resources and blocking such malicious users. This technique of deception is known as Diversity & Randomness [10].

© 2014- 18, IJIRIS- All Rights Reserved

Figure below shows working of the module:

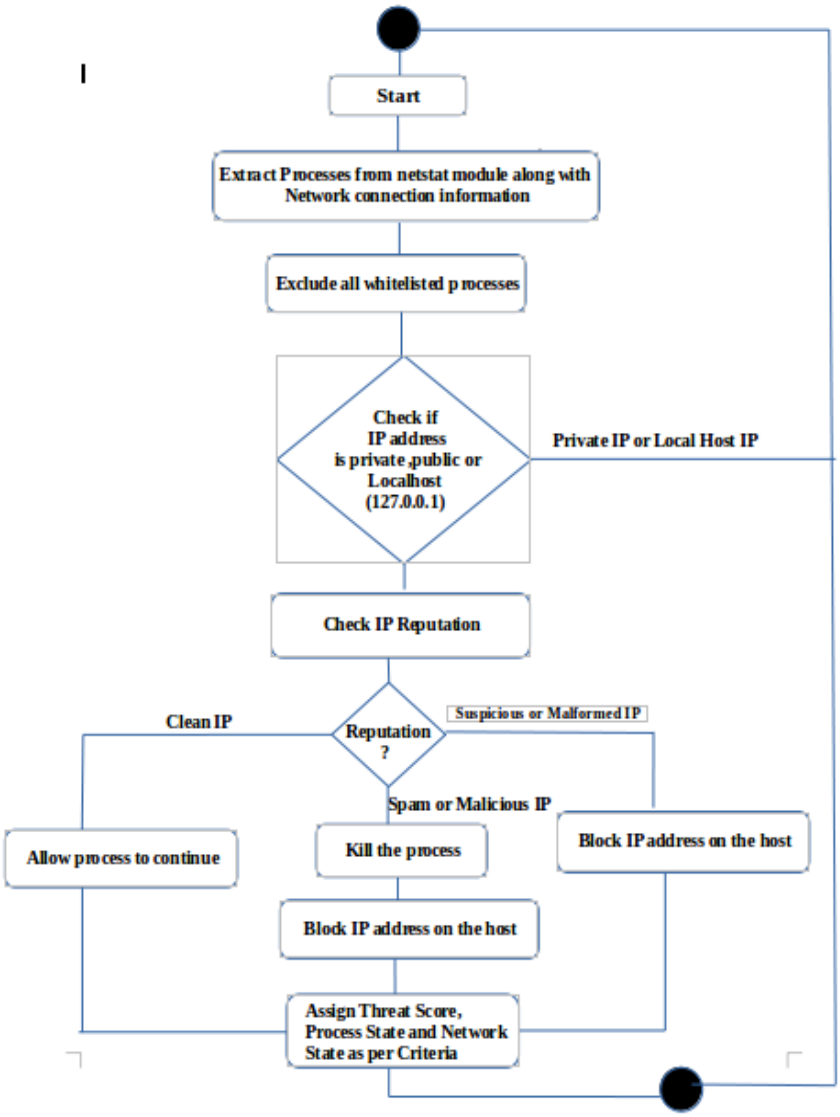

Figure 3.2.1: Flowchart for Process Monitor using White-listing Approach

A website's source code can reveal useful information to the attacker such as folder/file names and their corresponding path. Running automated tools for directory listing is often risky for attackers as their IP addresses can get logged. An attacker can discover information in source code of the web-pages which may make it easier to carry out an attack. Such channels can be used by organizations to their strength by leaking false information. Non-existent folders, files or accounts can be mentioned as a part of comments in the webpage. This is one way of deceiving the attacker and makes his attempts unfruitful while monitoring his actions simultaneously. This concept is called Threat Deception. The existing security controls perform the task of only a limited number of response actions, such as log, reject, drop and quarantine. These response actions generate large number of false-positives and in case of misconfiguration cause a security breach. Attackers may detect DROP and REJECT actions. Here the concept of deception can be used to confuse the attacker by presenting false-information. To create a deception environment, a sample website is chosen. The website implements banking functionalities and is developed using HTML-PHP. The website uses My-SQL database for storing information. The website is hosted on XAMPP server running on Ubuntu operating system. The site with deception information can be accessed using following URL: http://localhost/bank_td

The logs generated by the server can be found at the location/opt/lampp/logs/access_log. The admin.html file can be viewed by the attacker in his browser by sending following HTTP request:
http://localhost/bank_td/admin.html
The admin.html file contains a html code to redirect the attacker to the home page of the site. This can be achieved by adding a code snippet containing a meta tag within a header tag.

© 2014- 18, IJIRIS- All Rights Reserved

The meta tag has three attributes: “http-equiv” having value “refresh”,“content” having value “1” and “url” having value “http://site_url/bank_td”. This snippet will redirect the attacker to the homepage after one second. To analyze the logs, a variable holding the path of admin.html file has been declared and the HTTP response code to be checked will be 304 as the attacker will be directed to homepage. The IP addresses from the rows that satisfy the above condition will be blocked on the host. Here following Iptables[11] command has been used for blocking:
$iptables -A INPUT -s <attacker_ip_address> -j REJECT
Figure below shows the homepage of the website:

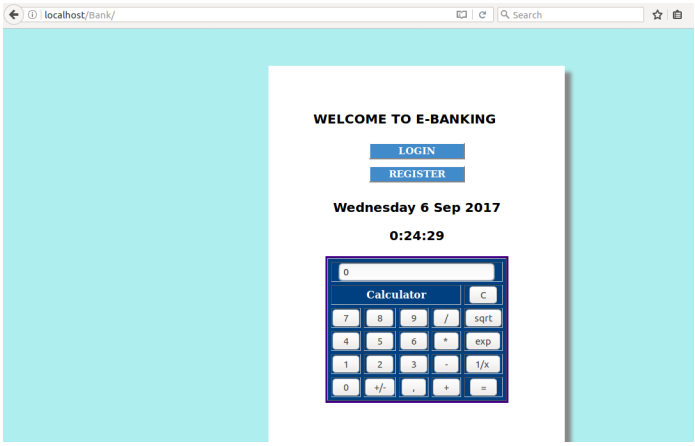

Figure 3.3.1: Website Homepage

IV. RESULTS

4.1 Email Attachment Analysis and Site Reputation Checker:

In the figure below a non-malicious file present in the mailbox is uploaded to check for infection. The result gives a SHA-1 hash of the file and lists all the AV Vendors who have scanned the file followed by the result. Here all AVs mention “None” which means that the file is clean. Also a message “File Not Malicious” is displayed to the user. Figure below shows a PDF file being uploaded for analysis:

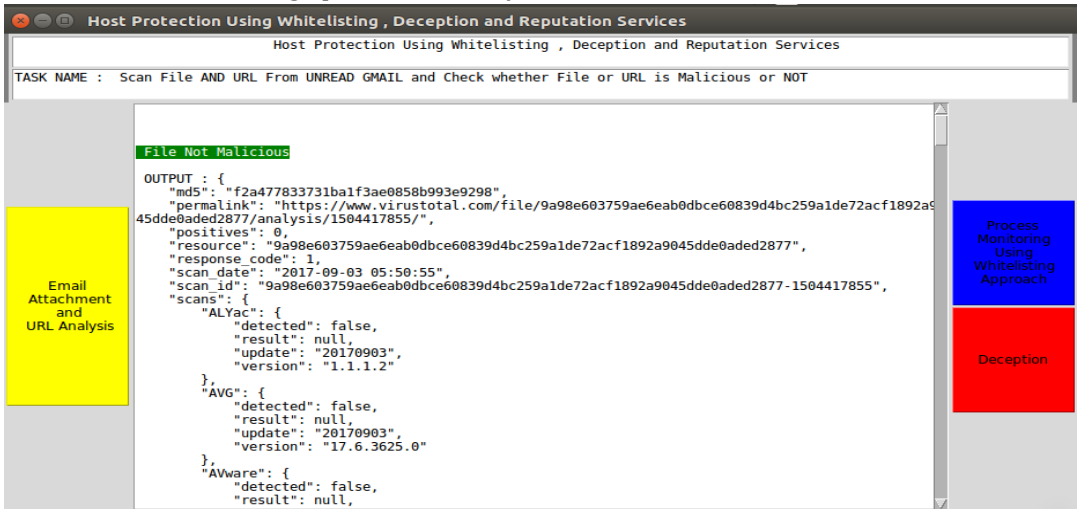

Figure 4.1.1: Scan Result of non-malicious PDF file on Virus Total

Following is the malicious sample qw.iso used to check the file reputation on VirusTotal:

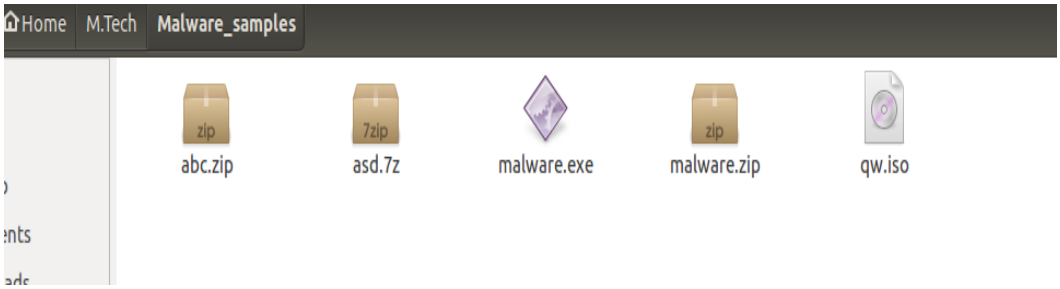

Figure 4.1.2: Malicious Sample Used for Analysis

© 2014- 18, IJIRIS- All Rights Reserved

Figure below shows analysis of malicious attachment:

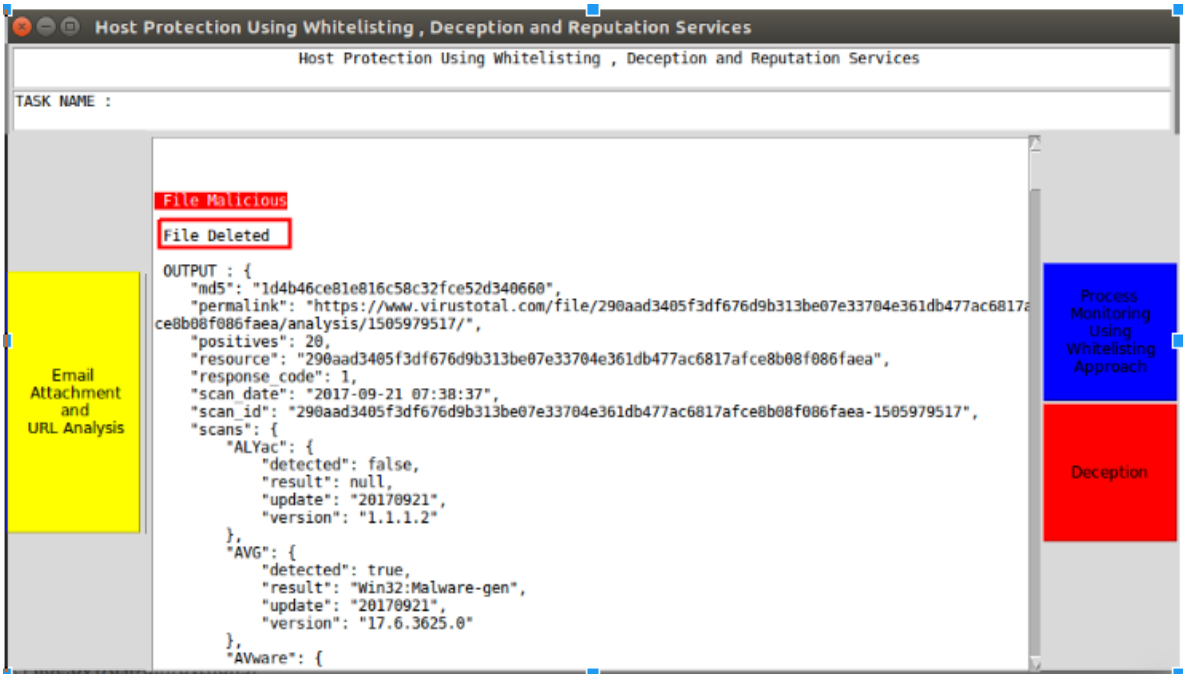

Figure 4.1.3: Malicious Sample in the Mailbox

The result gives a SHA-1 hash of the file and lists all the AV Vendors who have scanned the file followed by the result. Here most of the AVs mention some string other than "none" which indicates that the file is malicious. For example, the Anti-virus AVG identifies the file as malicious indicated by "detected: true" and classifies the file as "Win32:Malware-gen". Also the file gets deleted and the message is displayed to the user as shown below:

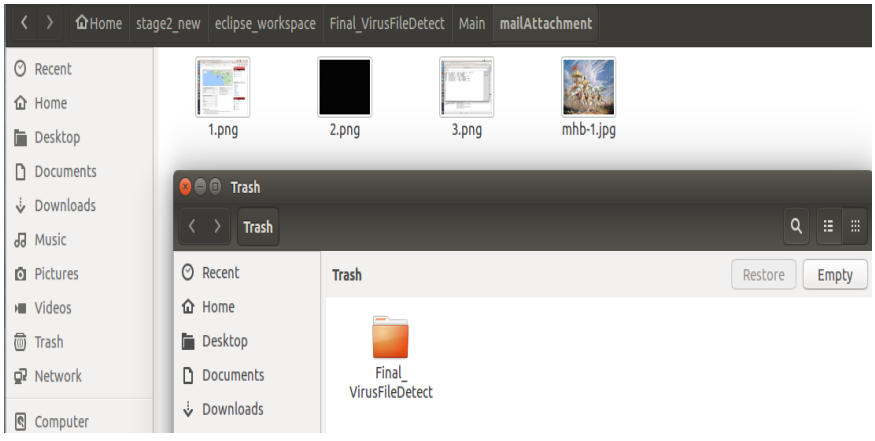

Figure 4.1.4:

Verification of File Purging Action

Figure below shows the reputation score given by urlvoid . 0 indicates that the site is clean.
Below is the email containing shortened URL?

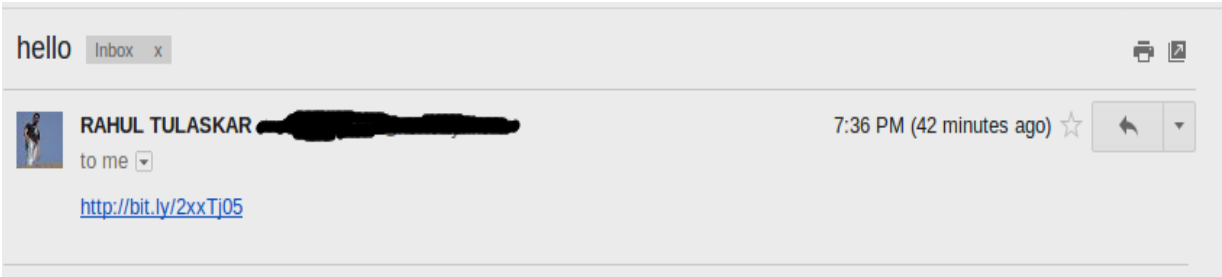

Figure 4.1.6: Email Containing Shortened URL

© 2014- 18, IJIRIS- All Rights Reserved

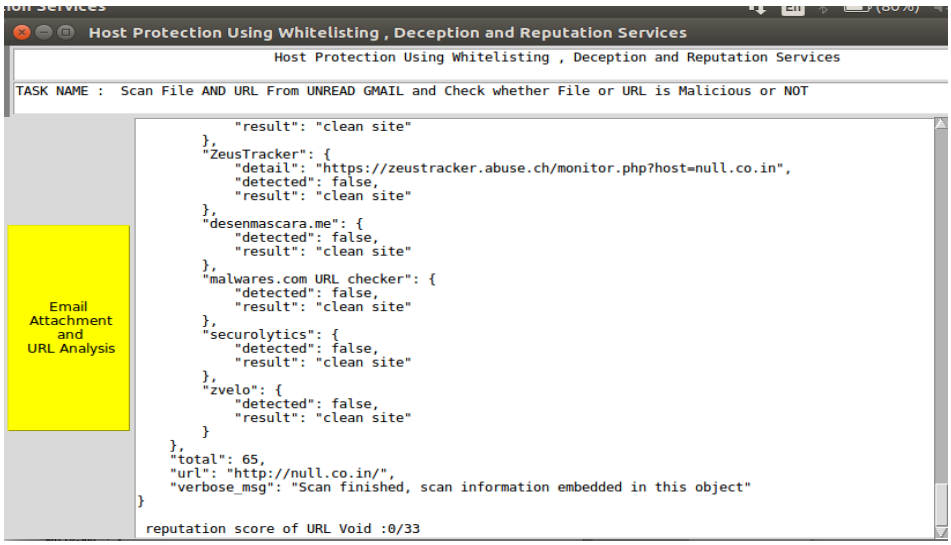

Figure 4.1.5: Site Reputation of a non-malicious website by scrapping urlvoid website

Figure below shows the reputation of a shortened malicious URL generated by VirusTotal:

Figure 4.1.7: Site Reputation of a malicious shortened URL

Also the expanded URL is shown in the above figure.

Figure below shows the reputation of a shortened malicious URL generated by urlvoid:

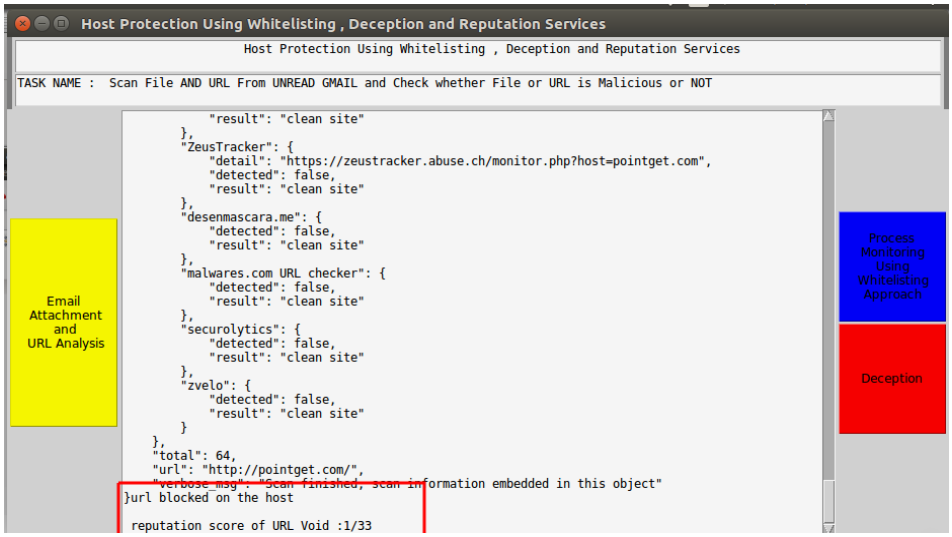

Figure 4.1.8: Site Reputation of a malicious shortened URL by scrapping urlvoid website

© 2014- 18, IJIRIS- All Rights Reserved

Also the expanded URL is shown in the above figure. After checking the reputation, the script blocks the URL on the host. When the iptables are viewed, the redirected domain of the URL is blocked. Figure below shows the blocked URL:

```
root@leorahul-Inspiron-3537:/home/leorahul/pydemo# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             www3012.sakura.ne.jp  reject-with icmp-port-unreachable
```

Figure 4.1.9: URL Blocked On Host

4.2 Process Monitor using White-listing Approach:

For Proof of Concept, a HTTP request to an IP address having poor reputation is initiated from the Google Chrome browser. This will make chrome browser process rogue and as it is not white listed, it will be assigned a threat score of 10. Following figure shows the HTTP request made to a blacklisted IP:

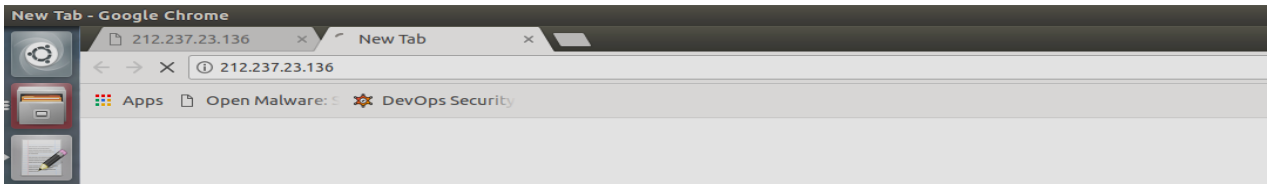

Figure 4.2.1: HTTP request to blacklisted IP via chrome browser

Following figure shows the IP reputation given by Reputation Watch:

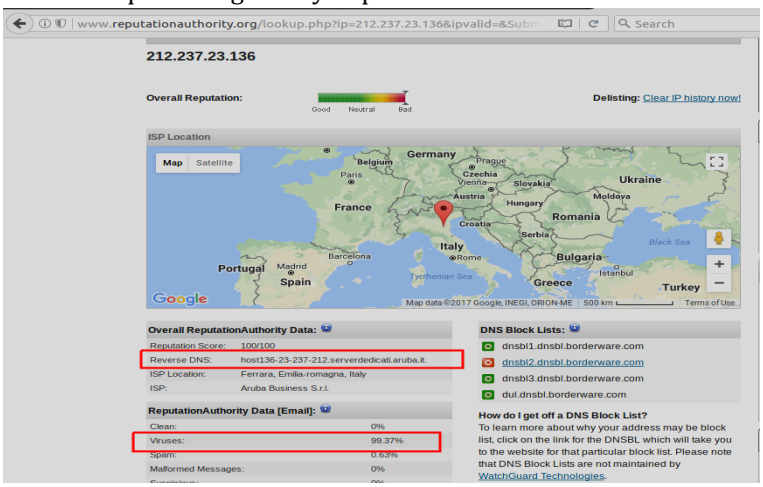

Figure 4.2.2: IP reputation given by Reputation Watch

Following figure shows the threat score assigned to the process by the script:

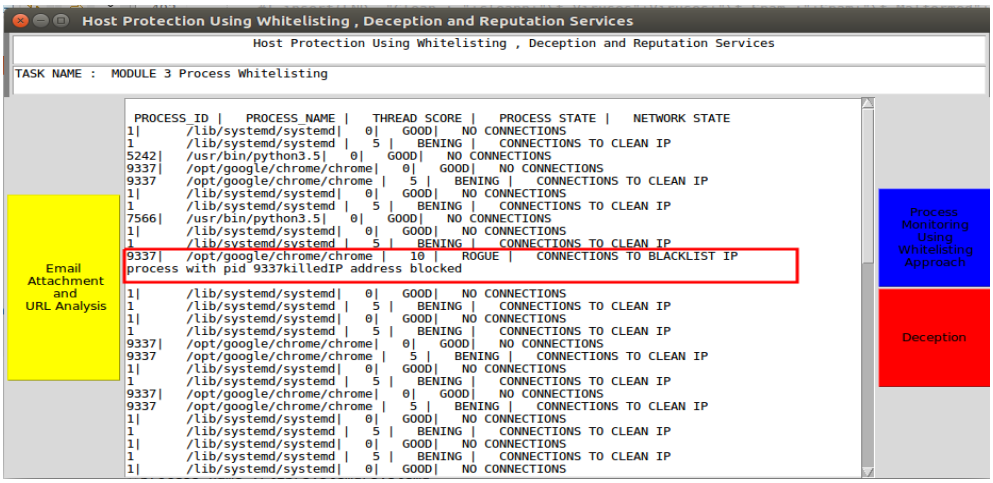

Figure 4.2.3: Identification of Rogue Process

© 2014- 18, IJIRIS- All Rights Reserved

Google chrome has process id as 9337 and network state is 'Connections to blacklist IP' .On the next line the script prints a message confirming killing of chrome process. Following figure shows that there is no chrome process running on the host. The script blocks the IP that was accessed using Chrome. Following figure shows the same:

```
root@leorahul-Inspiron-3537:/home/leorahul# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
REJECT     all  --  anywhere             host136-23-237-212.serverdedicati.aruba.it  reject-with icmp-port-unreachable
```

Figure 4.2.5 : Reverse DNS blocked on the host

4.3 Threat Deception:

The attacker will visit the website and would eventually come to the login page. Now the attacker views the source code of the above webpage in the browser by clicking on "View Page Source" .Following figure shows the same:

view-source:http://localhost/bank_td/login.php

```
}
a{
    color:#f00f53;
    text-decoration: none;
    align-content: right;
}
.button{
    width :150px;
    margin :10px;
    padding:5px;
    font-weight: bold;
    background-color: #04ae3a;
    text-align: center;
    position:relative;
    right:-100px;
    color:white;
}
.button:hover {
  background: #fa6538;
}
body{
    background-color: PaleTurquoise;
}
    body
    {
    background-color : PaleTurquoise     ;
    </style>
    </head>
    <body>
<!-- Changes pending in admin.html  -->
    <div class="container">
        <h2 >The Login Page</h2>
        <a href="index.php" >Click Here to Go Back.</a><br/>
        <form action="checklogin.php" method="POST">
            Enter Username : <input type="text" name="username" required="required"/><br/>
            Enter Password : <input type="password" name="password" required="required"/><br/>
            <input type="submit" value="Login" class="button"/>
```

Figure 4.3.1: Source Code Containing Vital Information

Looking at the comments the attacker will initiate following request:
http://localhost/bank_td/admin.html
For Proof of Concept (POC) purpose, consider a Ubuntu VM with the following IP address:

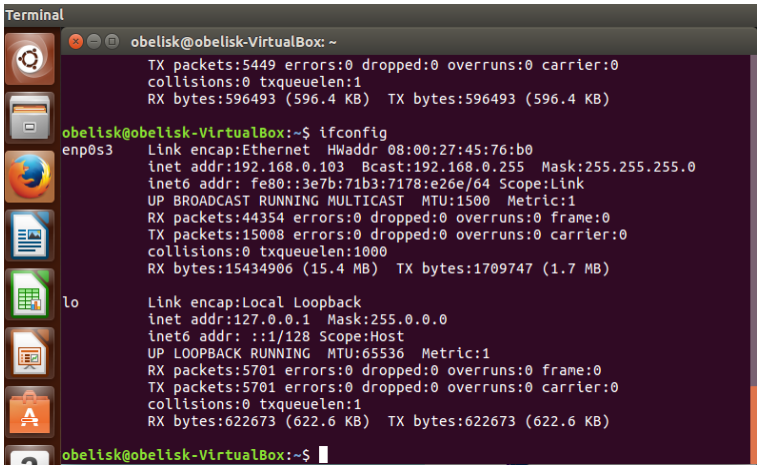

Figure 4.3.2: IP address of the Ubuntu (Attacker's) VM

The attacker's request is logged in the access log file of XAMPP server. Figure 3.3.3 shows the logged attacker's request:

© 2014- 18, IJIRIS- All Rights Reserved

login.php × admin.html × access_log

```
192.168.0.103 - - [23/Sep/2017:23:02:26 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:02:26 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.103 - - [23/Sep/2017:23:03:39 +0530] "GET /bank_td/admin.html HTTP/1.1" 304 -
192.168.0.103 - - [23/Sep/2017:23:03:40 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:03:40 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.103 - - [23/Sep/2017:23:04:20 +0530] "GET /bank_td/admin.html HTTP/1.1" 304 -
192.168.0.103 - - [23/Sep/2017:23:04:21 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:04:21 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.103 - - [23/Sep/2017:23:04:37 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:04:37 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.103 - - [23/Sep/2017:23:05:36 +0530] "GET /bank_td/admin.html HTTP/1.1" 304 -
192.168.0.103 - - [23/Sep/2017:23:05:37 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:05:38 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.102 - - [23/Sep/2017:23:07:08 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.102 - - [23/Sep/2017:23:07:08 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.102 - - [23/Sep/2017:23:07:10 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.102 - - [23/Sep/2017:23:07:10 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.102 - - [23/Sep/2017:23:07:50 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.102 - - [23/Sep/2017:23:07:50 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.103 - - [23/Sep/2017:23:08:22 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:08:22 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
192.168.0.102 - - [23/Sep/2017:23:09:51 +0530] "GET /bank_td/login.php HTTP/1.1" 200 1297
192.168.0.102 - - [23/Sep/2017:23:09:53 +0530] "GET /bank_td/index.php HTTP/1.1" 200 6790
192.168.0.102 - - [23/Sep/2017:23:09:53 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1195
192.168.0.102 - - [23/Sep/2017:23:09:54 +0530] "GET /bank_td/register.php HTTP/1.1" 200 1261
192.168.0.103 - - [23/Sep/2017:23:11:19 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [23/Sep/2017:23:11:19 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
127.0.0.1 - - [24/Sep/2017:11:12:34 +0530] "GET /bank_td HTTP/1.1" 301 233
127.0.0.1 - - [24/Sep/2017:11:12:34 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
127.0.0.1 - - [24/Sep/2017:11:12:34 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1165
127.0.0.1 - - [24/Sep/2017:11:12:34 +0530] "GET /favicon.ico HTTP/1.1" 200 30894
127.0.0.1 - - [24/Sep/2017:11:12:37 +0530] "GET /bank_td/login.php HTTP/1.1" 200 1287
127.0.0.1 - - [24/Sep/2017:11:13:01 +0530] "GET /bank_td/login.php HTTP/1.1" 200 1287
192.168.0.103 - - [24/Sep/2017:16:13:31 +0530] "GET /bank_td/admin.html HTTP/1.1" 304 -
192.168.0.103 - - [24/Sep/2017:16:13:32 +0530] "GET /bank_td/ HTTP/1.1" 200 6790
192.168.0.103 - - [24/Sep/2017:16:13:32 +0530] "GET /bank_td/clock_style.css HTTP/1.1" 404 1177
```

Figure 4.3.3: Attacker's Request in the Log File

Figure below shows the output of the script blocking attacker's IP:

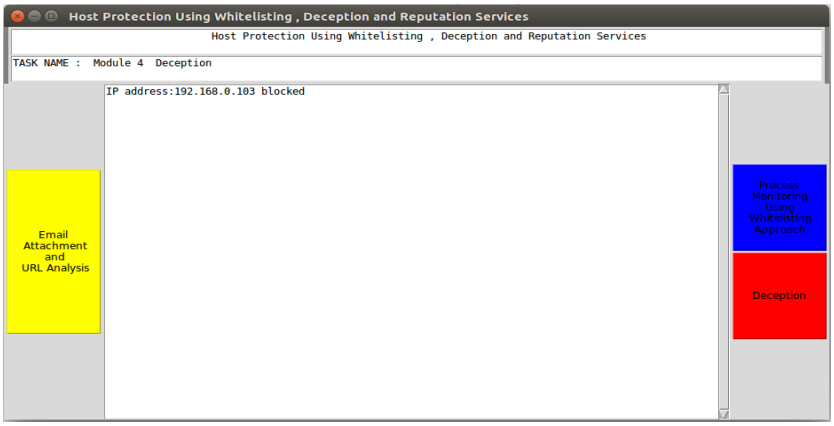

Figure 4.3.4: Logged Attacker's Attempts Processed by the Script

Post executing the script, message is displayed as " IP address: 192.168.0.103 Blocked" which is the same IP as shown in figure 3.3.2 Now the attacker may again proceed to check other WebPages , but cannot do so as his IP has been blocked which is indicated in the following figure:

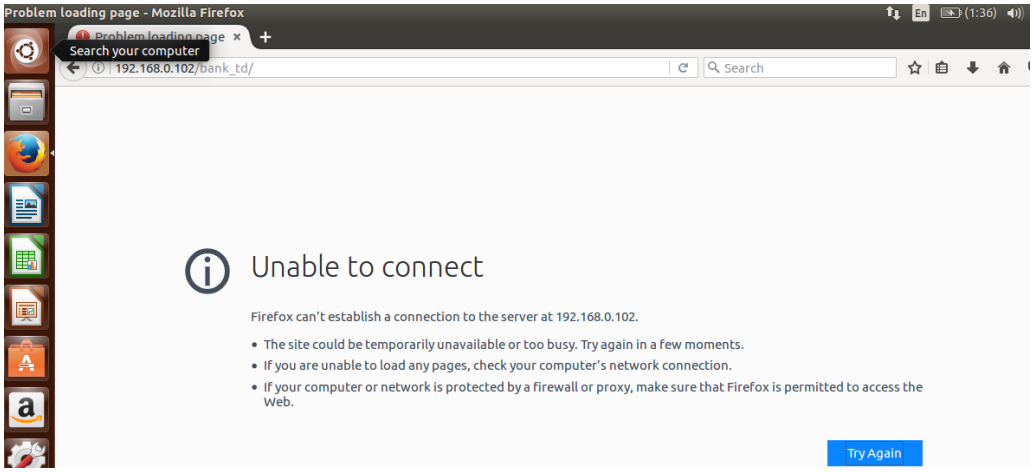

Figure 4.3.5: Verification of IP Address Blocking

The figure below shows the attacker's IP address in the iptables list.

© 2014- 18, IJIRIS- All Rights Reserved

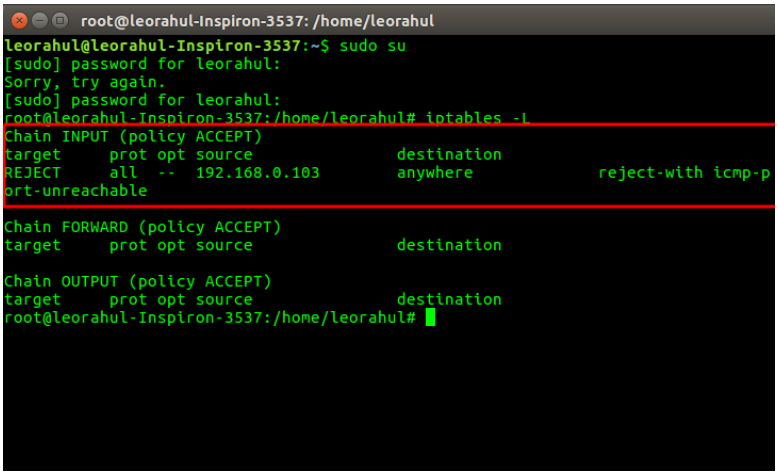

Figure 4.3.6: IP Address of Attacker Appended In Iptables

V. CONCLUSION AND FUTURE WORK

The Attachment Analysis module will allow a non-technical user to analyze any suspicious file in his email box. The project modules provide an abstraction over online sand box technologies like VirusTotal. The age of the data is important in case of an attachment and URL reputation. Also URL reputation tends to change frequently .Hence the user will get the correct reputation. This system is not an attempt to disregard paid AV services but merely to stay away from becoming a victim of cybercrime. The process monitoring module has the flexibility of exempting white-listed process from its monitoring set. The user can specify more than one process which he is sure of as a part of whitelist. The module also prevents a machine from becoming a bot and participates in DDoS attacks. Botnet IPs will have bad reputation causing the script to terminate the process that initiated a connection to such IP address. Threat deception technique will confuse and divert the attacker from his goals and also allow organizations to monitor the attackers. As the attacker IP is available in the access log file there is no need of a honeypot or honeynet. Thus resources are saved and the solution is economic and feasible. The URL reputation checker module can be integrated into a SIEM solution or into Host based IDS that runs this script frequently. Also a Proxy can use the Site Reputation Checker to get the reputation of the website and use the same for its internal categorization .The process monitoring module can be used to study the impact and efficiency of whitelisting approach. Also by using more advanced paid reputation services this module can be used in detection and cleaning of botnet machines in the organization. IP addresses with bad reputation could be populated into a list that will be updated in the Threat IP set of SIEM. This will cause the SIEM to trigger alerts whenever a connection to bad IP addresses is made. The IP address list generated by the threat deception module can be given to the Intrusion Prevention System to silently drop the packets and keep attackers away from the organization's networks.

ACKNOWLEDGMENT

I would like to thank my guide Prof. Sujata Pathak for her invaluable guidance and advice throughout this project.

REFERENCES

1. https://www.gdatasoftware.com/blog/2017/04/29666-malware-trends-2017
2. http://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf
3. Neutralizing Windows-Based Malicious Mobile Code, James A. Whittaker, Andres De Vivanco. http://www.zdnet.com/article/linux-botnet-attacks-increase-in-scale/
4. Malicious Spam Emails Developments and Authorship attribution, Mamoun Alazab,Robert Layton ,Fourth Cybercrime and Trustworthy Computing Workshop,2013.
5. WatchGuard Technologies : http://www.reputationauthority.org/
6. AHEAD: A New Architecture for Active Defense,Fabio De Gaspari,Sushil Jajodia,
7. A Practical Business Security Framework to Combat Malware Threat ,P.R.Lakshmi Eswari,N.Sarat Chandra Babu,World Congress on Internet Security,2012 http://voorloopnul.com/blog/a-python-netstat-in-less-than-100-lines-of-code/
8. Cyber Resilience for Mission Assurance , Harriet Goldman, Rosalie McQuaid, Jeffrey Picciotto https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands https://www.python.org/
9. https://rickypc.github.io/robotframework-imaplibrary/doc/ImapLibrary.html
10. www.urlvoid.com
11. www.virustotal.com

© 2014- 18, IJIRIS- All Rights Reserved