Mehran University Research Journal of Engineering and Technology
Vol. 40, No. 1, 230-240, January 2021
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2101.20

Detection of Malicious Servers for Preventing Client-Side Attacks

Khuda Bux[1a], Muhammad Yousaf[1b], Akhtar Hussain Jalbani[2], Komal Batool[1c]

RECEIVED ON 06.03.2019, ACCEPTED ON 26.07.2019

ABSTRACT

The number of client-side attacks is increasing day-by-day. These attacks are launched by using various methods like phishing, drive-by downloads, click-frauds, social engineering, scareware, and ransomware. To get more advantage with less exertion and time, the attackers are focus on the clients, rather than servers which are more secured as compared to the clients. This makes clients as an easy target for the attackers on the Internet. A number of systems/tools have been created by the security community with various functions for detection of client-side attacks. The discovery of malicious servers that launch the client side attacks can be characterized in two types. First to detect malicious servers with passive detection which is often signature based. Second to detect the malicious servers with active detection often with dynamic malware analysis. Current systems or tools have more focus on identifying malicious servers rather than preventing the clients from those malicious servers. In this paper, we have proposed a solution for the detection and prevention of malicious servers that use the Bro Intrusion Detection System (IDS) and VirusTotal API 2.0. The detected malicious link is then blocked at the gateway.

Keywords: Malwares, Client-Side Attacks, Cybersecurity, Malicious Servers, Phishing, Ransomware, Script Injection Attacks

1. INTRODUCTION

As Internet usage increased in the current era, as we are doing online shopping, booking rides, freelancing for clients, the number of client-side attacks also increased, so the cybersecurity has emerged as a challenging task. A significant number of client systems are affected by various sort of cyber-attacks. Different kinds of attacks like web-based attacks, phishing, spam, click-frauds, and scareware/ ransomware are common that target the client machines on the Internet. These sort of malicious exercises are carried out by the attackers through drive-by download approaches. A large number of malicious servers on the Internet forms the foundation for cybercriminals and thus the underground market which is known as the dark web. To detect such malicious servers can be a great contribution to overcome the problem of many cybercrimes. Even Peer to Peer (P2P) [1] botnets require "server-like" remotely available accomplices for starting late contaminated hosts to join the botnet. This work uses passive and active detection approach for detection and prevention of the malicious servers. Different tools and systems are utilized to detect malicious servers such as CyberProbe, Safe Browsing, SpyProxy, and RevProbe. Some tools use passive detection to block blacklist Internet Protocol (IP) addresses. Others use active detection for

[1] Institute of Systems Engineering, Riphah International University, Islamabad, Pakistan
Email: [a]bux.khuda@gmail.com (Corresponding Author), [b]muhammad.yousaf@riphah.edu.pk, [c]komal.batool@riphah.edu.pk
[2] Department of IT, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Sindh, Pakistan
Email: jalbaniakhtar@gmail.com

Command and Control server [2] by checking the response of the modified packet. But none of them is preventing clients from the attacks. We have used a passive detection method with Bro IDS for detecting malicious traffic on signature base. To minimize the processing of system, we use blacklist IP addresses at the gateway. We use VirusTotal and Cuckoo sandbox for checking of malicious server with binary malware analysis to avoid the false-positive ratio of blacklist IP addresses.

The rest of the paper is structured is follows. Section 2 briefs about the few types of client-side attacks. Section 3 gives detailed related work on techniques of detecting malicious servers. Section 4 gives a description of the proposed methodology for detecting malicious servers. In Section 5, the implementation and results are discussed. Lastly, Section 6 concludes the paper.

2. OVERVIEW OF CLIENT-SIDE ATTACKS

In the current era, websites are an important part of our everyday life. Almost every business is shifted to online service from booking of air tickets, bus tickets, metro city transport, online shopping, taxi booking, online banking, online study, outsourcing of jobs, and much more. For this research, we have focused on the web-based attacks. The well known client-side attacks are described below:

2.1 Cross-Site Script Attack

In Cross-Site Script (XSS) attack [3], the malicious code is executed by attackers on the legitimate website or any application for a client-side attack. The cross-site script is a well-known web application vulnerability and it is compromised when a web application has weak validations or gets client code as input into their web sites or web applications. By using a cross-site script, an attacker does not directly focuses on a client. Instead of it, attackers look for the vulnerable websites or web application visited by the clients, they use these vulnerable websites with the cross-site script as the main source to compromise the client browser with malicious code.

2.2 SQL Injection

The Structured Query Language Injection (SQLi) [4] is an insertion attack which may execute malicious SQL queries. These queries manage database servers at the backend of the web application. The attackers target the web application with SQL Injection vulnerabilities to bypass web application security weaknesses. Attackers can bypass the authentication and authorization of a web application, after that extract the data from the database. The SQL Injection attack can be used to add, modify and delete the records from the database. Any website or web application which uses an SQL database such as MySQL, Oracle, SQL server, or other databases can be affected. By using SQL Injection attack the attackers can get unauthorized access of confidential data like as user information, personal data, business secrets, intellectual property, and health record. The SQL Injection is a common and oldest attack and a dangerous vulnerability in the web application.

2.3 Client Side Exploits

The client-side exploits [5] are used to build a botnet network and target any specific company with a combination of content with malicious payloads and social engineering. This type of exploit looks for browsers, browser plugins, and email clients. There are a number of client software packages that will open up a socket to run the service. This will make a connection on the network, so it is known as clients with exposed services. If any software opens a socket on the network and makes a connection with it, it can be exploited. The Simple Message Block (SMB) service was targeted by the hackers in May 2017 for the WannaCry ransomware attack [6]. If any client is connected to the internet without any firewall, it can be attacked without any interaction of user such as open an email or click on any malicious link in the website. So that all vulnerabilities are not coming with any operating system, by monitoring the network in the passive mode we have to scan the clients for vulnerabilities into Domain Name Server (DNS) lookup, unencrypted File Transfer Protocol (FTP) traffic, website queries, email protocols, Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP), SMB, and many more client services or applications.

2.4 Phishing

A phishing attack [7] is used to send a malicious message that seems to be coming from a reliable source. Most of the time this is done by sending an email to the victim. The main goal of this attack is to get the confidential data of victims like credit card information, login details, or to install malware on the system. The phishing is a common type of attack so that from this type of attack client can be prevented by awareness. The phishing is used to cheat the victims and they are manipulated to provide the confidential information to attackers, mostly on malicious websites. In this attack both any person and organization are at risk, so that any kind of personal or organization data can be theft, with this attacker can do fraud, blackmail or he can access the network of the organization. Now a day this type of attack is carried out at the state level also.

2.5 Ransomware

The ransomware attacks [8] are carried out with too many types such as cryptowall, locky and cryptolocker these all have common functionality and they come with the same pattern. In this attack, users are receiving an email with attachments like as an invoice, a word file, a package alert, any report which come with a message to convince the victims it is legitimate attachment. As the client open attachment, the ransomware malicious file will run that will encrypt files and folders on the victim system. After that, he will see a message on his screen in which the attacker demands money from him for decryption key and to get back access of his files. In the last few years' numbers of ransomware attacks are increasing which is alarming for the security professionals.

3. RELATED WORK

The malicious servers are more damaging threat to Internet. To protect the clients or users of the Internet from malicious servers, the cybersecurity experts are doing an effort to detect them and protect the clients. There are two methods for detection, first is passive detection and second active detection [9]. The safe browsing tools/plugins are depending on the blacklist IP address databases such as Google safe browsing, Spoof Guard and Site Adviser [10-11]. For detection of malicious webpage links, the researchers have used Support Vector Machine (SVM) for the malicious Universal Resource Locator (URLs) detection and the RAkEL, MLkNN techniques have been used for detection of attack carried out by that malicious URL's [12]. The disadvantage of SVM technique is, it slowly processes large data sets. The SpyProxy [13] have used two techniques, first, the content of the website is executed in real-time on virtual environment developed by them to protect the local area network from that malware propagation, to detect and block the malicious website before it can be accessed by users. Another technique used is by inserting malicious website contents into client system such as "modification into Windows registry or new process creation", by that malicious website. In CyberProbe [14], the researcher has proposed a novel dynamic probing technique for the detection of malicious servers and botnet systems. Those listen for and do any action against that incoming instruction from that network. By this technique, they send a modified packet to remote system and examination the reaction of that received packet. For choosing whether the remote systems are malware contaminated or not. The autoprobe [15] will create a unique signature of those command and control servers which are using client-side rationale. This tool will utilize a dynamic malware investigation to get more profound information of packet how it carries on for the request to the remote server and the reaction of the client. The autoprobe deal with unique finger printing and filtering. To identify malicious server infrastructure, the Antonio Nappa have presented a tool RevProbe [16] which is working as an active probe. The RevProbe is an active probe to send an inquiry to the remote IP address and after that check, its reaction for any anomaly and spillage demonstrate that the IP address has a belongs with the reverse proxy intermediary veiling with another server.

One shortcoming of figure print [17] approach is that a replayed request may neglect to induce a reaction from a remote server, e.g. if a field in the request ought to be a checksum of the sender's IP address or if the request is encoded using the IP address as the initialization vector. The CyberProbe tool will not detect the command and control servers which are coming online for a short time to give instruction to botnets. Such semantic information cannot be easily gotten from the

system action. Another drawback of blacklist IP address databases updates which may take some time in between that update time more damage can be done by malware on the network. In some tools the privacy of clients exposed, and username or passwords can be saved by the safe browsing plugins which may be used for illegal activities. To overcome the limitations of existing work, we have proposed a passive detection method along with binary malware analysis of website and files. We are blocking URLs on the network via cache engine server. Through this proposed technique, we shall save bandwidth. As the functionality and working of existing tools in Table 1. The new features are added in malicious server detection solution such as offline binary malware analysis via Cuckoo sandbox and prevented the client from malicious servers. Which are detected as malicious servers by our monitoring server?

Table 1: Comparison of Existing Techniques

Features	Cyber Probe	Auto Probe	Spy Proxy	Safe Browsing Plugins	Rev Probe
Passive Detection	No	No	No	Yes	No
Active Detection	Yes	Yes	Yes	No	Yes
Anomaly Based	Yes	Yes	No	No	No
Signature Based	No	No	No	No	No
Blacklist IPs	No	No	No	Yes	Yes
Binary Malware Analysis	No	Yes	Yes	No	No
Client-Side Attacks Prevention	No	No	No	No	No

4. PROPOSED METHODOLOGY

In this paper, we have used a passive method to detect malicious servers for preventing client-side attacks. The tool and techniques used for this are Bro IDS which depends on passive approach for detecting malicious servers by analysis of network traffic at the gateway. We are focusing on websites/ URL's. Our method monitors network traffic passively as any client on network access the malicious website with any payload like as EXE file, pdf file, and file with any other extensions, it will alert us for that malicious website. After that malicious URL/website will send for further analysis to well-known dynamic malware analysis tool VirusTotal for scanning of that URL/website. If that system also detects any malware in that URL then our system will tag it as malicious link. This will be sent to the gateway for blocking it at network to avoid more damage from that malware.

The four methods are proposed for the detection of malicious servers for preventing client-side attacks: first, this will monitor the network traffic at gateway. When client access any URL with payload of EXE file, pdf file, image file or any other extension. Based on signature, the monitoring system of Bro IDS will send an alert on Sguil dashboard and will generate an email to system administrator. In second step, that malicious URL will be forwarded automatically to our malware analysis system on which VirusTotal Application Program Interface (API) 2.0 integrated. URL detected as malicious in the first step will be scanned and if found malicious, it will be saved in MySQL database. The report of this malicious URL will be sent to system administrator via another email. In third step, we have used Cuckoo sandbox for minimizing the false-positive ratio of Bro IDS and VirusTotal. By this we try to avoid a zero-day attack on clients if its signature is not found in VirusTotal. The Personal Home Page (PHP) custom program is used to extract file from BroIDS http logs. That malicious file will be sent to Cuckoo sandbox for offline binary malware analysis. In the fourth step, the automated process has been developed with the help of PHP custom program to scan URL or file in VirusTotal and Cuckoo sandbox. If it is flagged as malicious it will be blocked at the gateway/cache engine by an auto process. We have used the blacklist IP address resources (https://www.abuseipdb.com/account#api-settings, https://www.neutrinoapi.com/api/ip-blocklist/, https://lists.blocklist.de/lists/dnsbl/allinone.list) to decrease the computing process. [17, 18]. NFS (Network File Server) is used for file sharing between the monitoring server, Cuckoo sandbox server, VirusTotal scanning server and gateway/cache engine.

4.1 Detection Topology

We have used tools and techniques for detection of malicious servers in our topology as shown in Fig. 1. VMware workstation is sued to isolate from campus local area network. The CentOS 7 is installed to be deployed as gateway/cache engine along with the blacklist IP addresses. Bro IDS is used in passive mode

for monitoring of network traffic on signature base. To minimize the false-positive ratio of Bro IDS the dynamic binary malware analysis has been used with the help of Cuckoo sandbox. Another open source community tool VirusTotal is used for malware analysis of extracted URL and files. URL and file extraction from monitoring server log and its feeding to Cuckoo sandbox and VirusTotal is done through automated process. Reporting to system administrator and URL blocking at gateway/cache engine is also done through automated process without any human intervention. We have used Windows 7/10 as client machines.

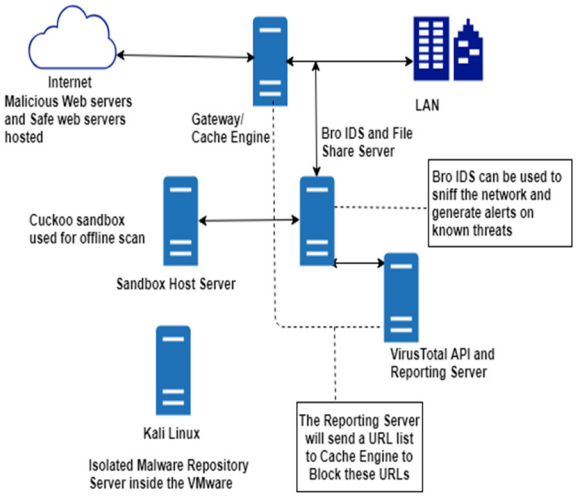

Fig. 1: Architecture of Malicious Servers Detection Schema for Preventing Client-Side Attacks

4.2 Bro IDS

Bro IDS [19] accompanies default policies for known system protocols and gives low-level event handlers to permit network administrators to characterize security policies. We characterize the accompanying activating events for the security strategy of the local network. The Bro IDS will monitor likewise incoming traffic on the network, the default logs of Bro IDS will be converted into Java Script Object Notation (JSON) format. Extracted URL and files are sent into VirusTotal API or Cuckoo sandbox for malware analysis.

4.3 Virus Total API 2.0

The scanning of malicious URL and offline files will be processed by the VitusTotal [20] API 2.0.

4.4 Cuckoo Sandbox

The Cuckoo sandbox [21] is an open source dynamic malware binary analysis tool, which is used for malware analysis in virtual environment. It can analyze number of applications as API calls and different types of malicious files. Furthermore, it analyze network traffic and perform advanced memory examination of the tainted virtualized systems through volatility just as on a procedure memory granularity utilizing YARA.

4.5 Gateway and Cache Engine

For preventing clients from the malicious websites or minimize the damage from that malware, we have used a gateway or cache engine. It blocks those URL by the help of received data from our system or blacklist IP addresses.

4.6 URLs Extraction, Scanning and Reporting

The URL or file extraction program developed in PHP. This program is used for scanning, reporting and email notification to system/network administrator. MySQL database is used to save the reports for future use. The main lines of code in PHP given below:

```
Function fetch_url ()
{
        $file_path = '/nsm/brother/logs/current/http.json';
        $contents = file_get_contents($file_path);
        $json_decode = json_decode($contents);
        $uri = $json_decode->URI;
        $id_resp_h = $json_decode->id_resp_h;
        $resp_mime_types = $json_decode->resp_mime_types;
Return $uri;
}
```

4.7 Proposed Solution

As client access any website in his/her system its traffic will be observed by Bro IDS on signature base. The monitoring system will send an alert if any malicious file has been downloaded or accessed by the client. After that our system will extract that domain name from the Bro IDS http.json log.

That URL will be forward to VirusTotal system for malware analysis and report will be saved in database. If that URL or file is detected malicious it will be sent to cache engine server for blocking to prevent clients. URL and file extraction will be done automatically

and forwarded to Cuckoo sandbox and VirusTotal for malware analysis. We have built a malware repository downloaded from Zoo/Master. To secure the network we have separated the Kali Linux server. These malicious files are placed into PDF, pictures, and other formats to swindle the client for drive-by download. We have utilized distinctive systems for fudging the malicious files and hosted them on free hosting providers www.000webhost.com. We have utilized meterpreter as payload with the latest contents for Window 7/10 alongside most recent endeavors. By utilizing these advanced contents, we have bypassed the typical security software like as AVG free antivirus, and Windows Defender on the client system. The meterpreter is a gathering of such a large number of little contents those are utilized to perform different sorts of exercises on a target system and it can send the payload to the target system with no blockage or alert to the client by this free antivirus software.

4.8 Malicious Server Detection Flow

Our framework will work as the passive detection on signature-based by utilizing Bro IDS. The details of its flow is depicted in Fig. 2. In initial step the client will request for a URL from the Internet. The Bro IDS will investigate the response of that URL. Bro IDS monitors the ingress traffic on network. In the event that Bro IDS identify any malicious URL they will generate alert on local system. The URL will be extracted from the Bro IDS logs http.json. For extraction we have created URL extraction application in PHP. This program will save that URL into file on local disk. Furthermore, we have installed NFS server to share those files for malware analysis. After that, the URL will be scanned by VirusTotal API and it will create a report. If that URL is malicious the report will be saved in MySQL database and email notification will be sent to the system administrator. If URL is tagged malicious this will be additionally processed for the prevention of clients. On the bases of Bro IDS alert and malware analysis system report, our system will act accordingly. If that URL is found malicious, reporting server will forward that URL to cache engine server to block it on the network. After that, if any client inside the network attempt to access that URL the message "It is a Malicious URL" will be displayed on his screen.

5. EXPERIMENTS AND RESULTS

In the current era of the world, the attackers are focusing on clients rather than servers for increasing significantly more financial profits. The Unites State of America elections was controlled by the attackers through phishing attack [22]. Attackers may utilize another sort of attacks, for example, malware spreading, drive-by downloads, scareware/ransomware, and Click-Frauds. The security personals are endeavoring to obstruct these sort of attacks and the attackers are finding another method for an attack on clients. The vast majority of antivirus software relies upon signatures of malicious files. So these signatures or anomaly base antivirus can be effectively circumvented by an attacker.

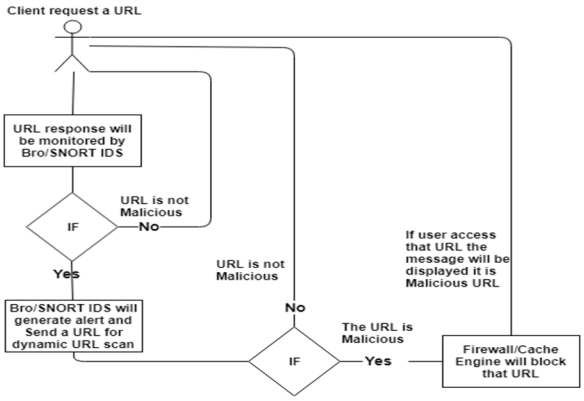

Fig. 2: Malicious Servers Detection Flow

5.1 Client-Side Attacks

There is a different type of attacks carried out on clients. Maybe couple of them are run undetectable with existing tools and procedures on client-side. The client-side attacks those are fudged with various family types are yet difficult to be detected in a brief time as any client affected by any malware on the network. Client-side attacks detection with fudged huge test because of its distinctive sort of payloads like as drive-by download, email attachments, website scripting, and ransomware. These type of attacks need client interaction, for example, click on any link which is known as the drive-by download or click on any image. As that link clicked, opened in browser or file is downloaded the malware will be installed on the client system. After that client can be diverted to another link or advertisement popups, information

steal, and data encryption for ransomware. The primary thought process of the attacker is to compromise the system of any company/target network to get more benefit in the shape of financial profit, harassments, or data theft of that company/target. The attacker utilizes payloads with substance and obscurity to sidestep the antivirus or safe browsing. Exploits are used to the target system by payload content attack which dynamic quiet download to the system. The principle motivation behind attack to run malware stealthy on the targeted system.

5.2 Weaponized Attacks

The weaponized attacks [23] are known as authentic files, links, or email attachments that stow away malware in them. At the point when the client endeavors to open those files, the malware will be installed on the system. That malware exploits the word, flash, or some other program weakness. The weaponized attack is completed by social engineering, or by utilizing phishing to spread it. At the point when a client downloads any pdf file, email attachment, or fake link, genuine site, malware will be installed on the system. These weaponized attacks are done as PE (Portable Executables), which may have reverse shellcode. Malware can be considered "weaponized" when it acquires a specific level of sophistication and demonstrates a reasonable thought process and intent.

5.3 Drive-by-Downloads

The drive-by download [24] will exploit the program shortcoming by malware installation. At the point when client access to any site has malware with concealed EXE in pdf or image, as client open that pdf file or image the malware will be installed on the targeted system by that drive-by download. Attacks, known as "drive-by-downloads," focus on the application layer. They abuse browser vulnerabilities, embedding malicious programs, which are launched and executed, without the client's assent or notice. When the customer interfaces with the server to recover a contaminated website page.

5.4 Watering hole Attack

The watering hole attack [25] is same as drive-by downloads the main distinction is it is done by employee or company workers. The attacker will utilize that site as an exploit vault that will abuse the browser or its plugins. The single exploit with multi-browsers usefulness can keep running on Firefox, Chrome, and Internet Explorer and so on. The attackers are endeavoring to build up these sort of exploits to acquire advantage with a solitary exploit. As attackers know the significantly more utilized applications with browsers are Adobe Flash and Oracle's Java because of this are creating exploits for these applications. The attackers are utilizing divert traps to their own sites for theft of client's username and passwords. In a watering hole attack, the enemy cautiously chooses a lot of sites as often as possible visited by his targets and by trading off these locales picks up chances to infiltrate the targets' systems, in a way much like the predator prowling around a watering hole to trust that its prey will appear.

5.5 Repository of Malware Samples

From the repository of each malicious file is executed on each firewall, antivirus, and operating system are the up-to-date and along with signatures. In lab the meterpreter of metasploit [26] on kali Linux is used to embed payload in pdf file, images and other weakness of client system exploited. The well-known free antivirus (Windows Defender and AVG free) is installed on client's operating system. The Windows 7/10 deployed in virtual lab environment for the experiment of detection of malicious servers.

5.6 Malware Embedded into PDF/IMAGE

We have downloaded the malwares from Zoo/Master. The Offensive Security Linux Distribution (Kali Linux) is used in disconnected mode on VMware. Metasploit [27] framework is used for the attack on clients in lab, sample commands are given below:

start msfconsole
"use" command will be used for exploit:
use exploit/windows/file format/adobe_pdf_embedded_exe
After that set a payload with the command"
set EXENAME win33.exe
set FILENAME test1.pdf

5.7 Detection of Malicious Servers

The attackers are using different type techniques to attack on clients, for example, drive-by downloads,

click-frauds, ransomware, phishing attack, and social engineering. To counter these attack, we have proposed passive detection with signature-based Bro IDS. The work as state-full in this sort of correspondence monitoring server keep up the signature base association. The malware analysis server is utilized for URL or file scanning. This all process is automated to limit the impact of client-side attacks. The cache engine server is deployed for preventing clients from those attacks. For the detection of malicious servers, we have deployed Bro IDS, Reporting Server (binary malware analysis) and cache engine server as a gateway to block the client's accesses to those malicious websites. The cache engine server is configured as transparent proxy [28], due to this client browsers will not be configured manually.

5.8 Client Accessing URL

As client is accessing any URL from the Internet on his system. These client are using Windows 7/10 up-to-date, the Windows Defender is enabled and AVG free antivirus installed on these systems. As the client has access to URL msiskb.000webhostapp.com and downloaded pdf file. However, the client have not received alert for malicious file from Windows Defender, AVG free antivirus.

5.9 Bro IDS HTTP Logs

The Bro IDS utilizes a passive detection strategy and it is an open-source network traffic monitoring. Be that as it may, it supports another capacity of system analysis out of security area, performance check, and valuable for troubleshooting. The more vital utilization of Bro IDS recording of log documents for network activities in the abnormal state. The logs are not recorded just for layer-1 however, it will likewise record the layer-7 logs for HTTP sessions with URL name, header data, MIME type, and response from the server, DNS request for with answers, SSL certificate, the key content of SMTP session and so forth. The log details of HTTP appeared in Fig. 3. In this the full detail of HTTP response from Web server are shown, the destination IP address 192.168.186.146 as id.oirg_h, Date and time response to client, URL, status code 200 (alright), file name in resp_filename, client system time March 2 2019 18:17:21 at response, MIME type, type operating system. To extract URL from the Bro IDS logs, we have converted default logs into JSON format.

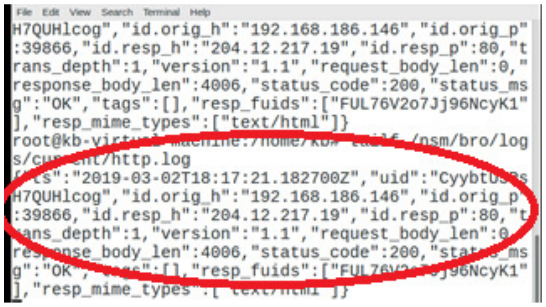

Fig. 3: Bro IDS Logs for HTTP

5.10 URL Extraction

We are extracting URL from the Bro IDS logs. For this, we have changed the default log format of Bro IDS to JSON format to extract the URL or file. The log location of network monitoring server "/nsm/brother/logs/current/http.json". According to our perception, a large number of client-side attacks are done by the attacker with the social engineering like as installing malicious link in a site or embedding any malicious document (malware inserted into PDF or IMAGE) which is known as drive-by download. On this investigation, we are extracting URLs just with the response_filename or type application or pdf, application or xdosexe, and application or image. After the extraction of URL or file from http.json logs, it will be saved in a file on local system. The extracted URL file will be sent to the malware analysis system by utilizing the NFS. We have installed the cron-jobs to check that file, if it is updated then it will be forwarded for further processing of scanning.

5.11 Dynamic URL Scanning

As we get alert on Bro IDS server for the malicious URL, on that bases URL or file will be extracted. For this program is developed in PHP. On the malware analysis system URL/File will be analyzed. The automated process is developed for analyzing of these files in VirusTotal and Cuckoo sandbox. For report response from VirusTotal, we have added the delay of one minute for scanning another URL, to scan it through automated process code given below:

Function scan_url($scan_url){

```
$api_key = "fa0af441a6cf987ef94373703eabc9abd1e40dcec03e13d4a2401d8c50d6c181";
$scan_url = $_POST['url'];
$post = ['apikey' => $api_key, 'url' => $scan_url];
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://www.virustotal.com/vtapi/v2/url/scan');
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_VERBOSE, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
}
```

As we get a response from malware analysis system for its report whether it belongs to malicious family or not. As it is detected as malicious link its report will be saved in database. The alert will be sent to the system administrator through an email (we have utilized SendGrid free API key for email alerts). In the meantime, the malicious URL will be saved into a file on the local system for further processing to keep the clients safe from these attacks.

5.12 Preventing Clients from Malicious URL

The Squid proxy [29] offers an excessive number of options for access control, authentication, rich logs, content-based applications, and caching. Squid proxy gives high traffic streamlining by empowering cache by this the DNS traffic will be decreased for external request at every time and practically about 50% bandwidth will be saved by this also. The Squid proxy is deployed with a transparent proxy for blocking the malicious URL or IP addresses on the network. Which are received from malware analysis system or Cuckoo sandbox. The transparent proxy will forward port 80 and 443 to its default port 3128. The clients will be unaware that their all requests are going through cache engine server. The customized page for client is configured to show them a message on screen why any URL is not opened at their system. The transparent proxy settings commands are given below:

```
iptables –t nat –A POSTROUTING –i ems33 d 0/0 –j MASQUERADE
iptables –t nat –A PREROUTING –i ems33 –p tcp –dport 443 –j REDIRECT –to-port 3128
iptables –t nat –A PREROUTING –i ems33 –p tcp –dport 80 –j REDIRECT –to-port 3128
```

The URL blocking is done by utilizing file regex in the configuration of the cache engine server. Since we are getting these file from malware analysis system and store that file on the local disk of the cache engine server to block them. This all procedure is automated by utilizing of NFS server and with the help of cron-jobs on cache engine server.

6. CONCLUSION

Prevention of client-side attacks is not a simple task. By utilizing some advanced methods for payload, for example, meterpreter to exploit the clients, the traditional firewalls or antivirus can be effectively circumvented. According to our study, there is a requirement for more work to be done for the detection of malicious servers. So we have attempted to build up a decent and dynamic tool which will detect malicious servers that will be free and fewer resources required for it. We are not saying our tool will give 100% grantee to identify every single malicious server, yet it is an addition to current systems/tools utilized for the detection of malicious servers. As we want the outcomes of our tool gives us 100% results to detect malicious servers. The technique we have utilized for the detection of malicious servers is passive detection and signature base. The automated system is developed which will extract the URL from the log files. That extracted URL file will be sent to malware analysis system for scanning. If it is detected as malicious then it will be sent to cache engine server to block it on network.

7. FUTURE WORK

For the future work and recommendation, the monitoring and detection of the malicious servers. The HyperText Transfer Protocol Secure (HTTPS) traffic on the network can be detected via the deployment of cache engine server with our own SSL (Secure Sockts Layer) certificates in between with local network and Internet. As per the growing number of client-side attacks, day by day to keep safe clients from these attacks the awareness to clients should be given in the organization.

ACKNOWLEDGMENT

Authors want to say thanks to Dr. Muhammad Saleem Abid, Visiting Faculty Member, Riphah Institute of System Engineering, Islamabad, Pakistan, for offering their pearls of knowledge to us over the span of this

research, and we thank 2 "unknown" analysts for their great insight review of this paper.

REFERENCES

[1] Niazi M.A., Hussain A., "Complex Adaptive Communication Networks and Environments: Part 1," *Simulation*, Vol. 89, No. 5, pp. 559–561, May 2013.

[2] Nachum S., Schuster A., Etzion O., "Detection in the Dark – Exploiting XSS Vulnerability in C&C Panels to Detect Malwares", *Cyber Security Cryptography and Machine Learning*, pp 227-242, June 2018.

[3] Bukhari S.N., Dar M.A., Iqbal U., "Reducing attack surface corresponding to Type 1 cross-site scripting attacks using secure development life cycle practices", *4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics* (AEEICB-18), October 4, 2018.

[4] Khanna S., Verma A.K., "Classification of SQL Injection Attacks Using Fuzzy Tainting", *Springer*, July 13, 2018.

[5] Cao Y., Qian Z., Wang Z., Dao T., Krishnamurthy S.V., Marvel L.M., "Off-Path TCP Exploits of the Challenge ACK Global Rate Limit", *IEEE/ACM Transactions on Networking*, Vol. 26, No. 2, pp. 765 - 778, April 2018.

[6] Schirrmacher N-B, Ondrus J., Tan F.T.C., "Towards a Response to Ransomware: Examining Digital Capabilities of the WannaCry Attack", *Proceedings of the Pacific Asia Conference on Information Systems (PACIS)*, June 26, 2018.

[7] Meyers J.J., Hansen D.L., Giboney J.S., Rowe D.C., "Training Future Cybersecurity Professionals in Spear Phishing using SiEVE", *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, pp. 135-140, October 03 - 06, 2018.

[8] Min D., Park D., Ahn J., Walker R., Lee J., Park S., Kim Y., "Amoeba: An Autonomous Backup and Recovery SSD for Ransomware Attack Defense", *IEEE Computer Architecture Letters*, Vol. 17, No. 2, pp. 245 - 248, July-Dec. 1, 2018.

[9] Bartlett G., Heidemann J., Papadopoulos C., "Understanding passive and active services discovery", *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pp. 57-70, October 24-26, 2007.

[10] Arora R., Arora A.K., "Phishing Web Pages detection Using Feature Selection and Extraction Method", *International Journal of Scientific Research in Civil Engineering*, Vol. 2, No. 4, 2018.

[11] Qabajeh I., Thabtah F., Chiclana F., "A recent review of conventional vs. automated cybersecurity anti-phishing techniques", *Computer Science Review*, Vol. 29, pp. 44-55, August 2018.

[12] Veni R.H., Reddy A.H., Kesavulu C., "Identifying Malicious Web Links and Their Attack Types in Social Networks", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2018.

[13] Moshchuk A., Bragin T., Deville D., Gribble S.D., Levy H.M., "SpyProxy: Execution-based Detection of Malicious Web Content", usenix.org, 2007.

[14] Nappa A., Xu Z., Rafique M.Z., Caballero J., Guy G., "CyberProbe: Towards Internet-Scale Active Detection of Malicious Servers", http://citeseerx.ist.psu.edu, 2014.

[15] Xu Z., Nappa A., Baykov R., Yang G., Caballero J., Gu G., "Autoprobe: Towards Automatic Active Malicious Server Probing Using Dynamic Binary Analysis", *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[16] Nappa A., Munir R.F., Tanoli I.K., Kreibich C., Caballero J., "RevProbe: detecting silent reverse proxies in malicious server infrastructures", *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 101-112, 2016.

[17] Zeng X., Kang C., Shi J., Li Z., Xiong G., "A Novel Website Fingerprinting Method for Malicious Websites Detection", *Information and Communication Technology for Intelligent Systems*, Springer, Vol. 107, pp 723-730, December 15, 2018.

[18] Ali, S.A., "Cloud Based Remote FPGA Lab Platform: An Application of Internet of Things", *Mehran University Research Journal of Engineering and Technology*, Vol. 37, No. 4, p. 535-544, oct. 2018.

[19] Chen B., Lee J., Wu A.S., "Active Event Correlation in Bro IDS to Detect Multi-stage Attacks", *Proceedings of the Fourth IEEE International Workshop on Information Assurance (IWIA'06)*, 2006.

[20] Masri R., Aldwairi M., "Automated malicious advertisement detection using VirusTotal, URLVoid, and TrendMicro", *Proceedings of the 8th International Conference on Information and Communication Systems (ICICS)*, May 11 2017.

[21] Jamalpur S., Navya Y.S., Raja P., Tagore G., Rao G.R.K., "Dynamic Malware Analysis Using Cuckoo Sandbox", *Proceedings of the Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, September 27, 2018.

[22] Mansfield-Devine S., "The ever-changing face of phishing", *Computer Fraud and Security*, Vol. 2018, No. 11, pp. 17-19, November 2018.

[23] Knapp E.D., Langill J.T., *Industrial Cyber Security History and Trends, Industrial Network Security*, 2015.

[24] Narvaez J., Popovsky B.E., Seifert C., Aval C., Frincke D.A., "Drive-by-Downloads", *Proceedings of the 43rd Hawaii International Conference on System Sciences*, 2010.

[25] Alrwais S., Yuan K., Alowaisheq E., Liao X., Oprea A., Wang X.F., Li Z., "Catching Predators at Watering Holes: Finding and Understanding Strategically Compromised Websites", *Proceedings of the 32nd Annual Conference on Computer Security Applications*, Pages 153-166, 2016.

[26] Xie M., Hu J., Yu X., Chang E., "Evaluating Host-Based Anomaly Detection Systems: Application of the Frequency-Based Algorithms to ADFA-LD", Network and System Security. NSS 2015. Lecture Notes in Computer Science Springer, Cham, Vol. 8792, 2015.

[27] Xia H., Xi Y., Pei Q., "The Research of Advanced Evasion Attack Method Based On Metasploit And Fragroute", *Proceedings of the 6th International Conference on Information Engineering for Mechanics and Materials*, November 2016.

[28] Mani A., Vaidya T., Dworken D., Sherr M., "An Extensive Evaluation of the Internet's Open Proxies", *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 252-265, December 03 - 07, 2018.

[29] Mushtakov R.E., Silnov D.S., "New approach to detect suspicious activity using HTTP-proxy honeypots", *Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (EIConRus), April 27 2017.