Volume 3, Issue 6
December 2014

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Malware Contaminated Website Detection by Scanning Page Links

[1] Mehdi Dadkhah, [2] Amin Dadkhah, [3] Jie Deng

[1] Lecturer of Tiran Branch, Islamic Azad University, Isfahan, Iran.

[2] Student of Tiran Branch, Islamic Azad University, Isfahan, Iran.

[3] Queen Mary University London, London, UK.

E-mail: [1]mdt@dr.com, [2]amin@iautiran.ac.ir, [3] j.deng@qmul.ac.uk

ABSTRACT

With increasing growth of communication networks, social interactions and financial transactions have been migrate to virtual environments. Internet is one of the most substantial platform for most people's social interactions and transactions. However, the notable challenge in online transactions is security in cyber environments and to understand the hazards accompanied with this communication platform. Because of the increased use of Internet and virtual environments in daily affairs such as financial transactions, this platform has become the focus of attackers and swindlers, for example the stealing of users' passwords. In this paper, we introduce main methods which attackers use to contaminate websites by malware. Even though several articles have been written on the subject, our main goal is introducing the advanced type of these frauds conducted by professional attackers which includes contaminating websites by any kind of malware like using phishing attacks, security vulnerability in web, social engineering and click hijacking. Finally we present our approach for confronting frauds conducted by installing spyware by contaminated websites attack and malware behavior.

Keywords: *Online fraud, cyber environment, phishing, web vulnerability, spyware.*

1. INTRODUCTION

Malware is a program which intends to do unwanted or disrupting tasks in operating system without user's permission [1]. The first malware was a virus which was written in early 1980s with the purpose of disrupting stored information in computer systems. Then first network worms were born [2] in 1988 for contaminating SunOS and VAX BSD systems. It attacked these systems through network vulnerability and after inserting, ran a disruptive program on the system. From 2000, new techniques were invented for jobbery by malware and the main goal of malware was misusing computers as zombies [3]. In that time backdoor software was massively used. Since 2003 spying becoming popular and other goals like stealing passwords were performed by attackers. Such malware is called spyware [4]. Different techniques are developed to confront spyware which are adopted in security software especially antiviruses. In [5] provides a profile of suspicious behavior for confronting key loggers and spyware, then suspicious files are detected according to this profile. Despite this approach may fail because many attackers write their spyware so that they have a similar behavior to other network related software. [6] Suggests using a virtual keyboard to confront spyware. Since the input information is entered by virtual keyboard, spyware is not able to record input information. However, advanced types of spyware take consecutive pictures from user's screen and discover the passwords entered by virtual keyboard. [7] applies a technique which combines input characters with random characters so that recorded information by spyware will be blocked; however, this technique uses a simple mechanism which can be recognized by attackers. Many researches have been conducted showing many spyware can be unknown for a long time and continue to run [8, 9]. Therefore, the harms related to malware can be reduced by preventing their entrance to computers. Our goal in this paper is presenting an approach to prevent entrance of these spyware into user's operating system through contaminated websites. The rest of the paper is structured as follow: in section 2 we will introduce methods that attacker use to contaminate victims' computer thro internet and in section 3 we present our approach to prevent entrance of these spyware into user's operating system through contaminated websites.

2. ATTACKER'S METHODS FOR SPREADING SPYWARE THROUGH WEB

Today attackers use various methods to contaminate victims' computer with different malware. This includes deceiving users using social engineering, hacking websites and putting links containing malware in them, phishing attacks in order to contaminate victim computers, using vulnerability in operating systems and applications, Internet click hijacking, email spoofing which are used by attackers recently to contaminate computers by malware through web.

2.1. SOCIAL ENGINEERING

Social engineering is the art of using people's vulnerable behavior to creating security trap without raising any suspicions from the victim. Social engineering deceives people in different ways and forth putting them by convincing them to access information. Instead of using common and direct influence, like data collection methods and passing firewall to access the organization systems and its databases, attackers target people who have access to this information and deceive them to get what they want [10]. Malware developers need to encourage victims to run their disruptive file or access the contaminated websites; therefore they need social engineering techniques. Forging famous institutions and companies by email addresses or embed the disruptive file as attachments in emails [11] are common methods. The point which should be considered here is the fact that attacker's behavior is planned in advance so that user won't be aware of running a malware on his computer. In all stages of the attack, user thinks that he is running a legal software like a test or free software from a famous company. It will increases users' ability to confront these attacks significantly if they are familiar with methods which swindlers use to deceive their victims.

2.2. PHISHING ATTACKS

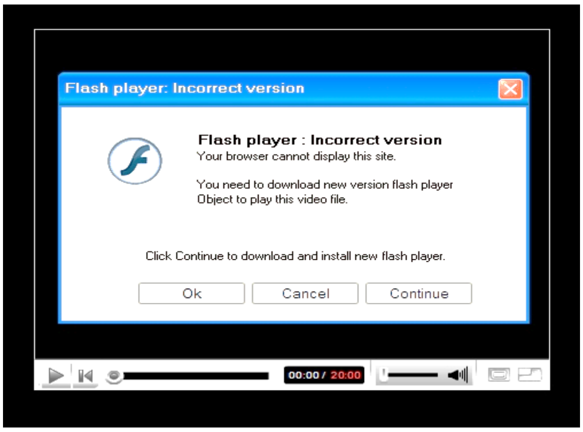
Attempting to access valuable information such as username, password, and credit card information by using social engineering techniques and fake websites is called phishing attacks [12]. Normally, phishers (attackers or people who do phishing attacks) begin their work by designing an identical website. Then deceive their victims to enter these websites and enter their confidential information. In other words, attackers' goal in phishing attacks is let victims entering information in the fake websites [13]. Recently, attackers also use phishing techniques to contaminate users' computers by malware. They design fake websites contaminated by malware similar to navigate websites and encourage victims to enter these websites. After users enter these websites, their computers will be contaminated by malware. Therefore attackers' chance of success increases because attack is done as soon as victims enter the contaminated website and there's no need to enter information. Also a wider range of websites can be used which mainly includes financial websites.

2.3. EMAIL SPOOFING

In this method, attackers send an email to the victim (these emails are often sent massively) which appears to have an important and valid address. In the email content victims are demanded to go to a particular link or use a file which is in the email attachment [14]. In fact, a defect in TCP/IP protocol is used in this method which allows the attacker to send emails from any arbitrary address, but attacker is not able to receive replies so he puts his fake links in such emails. The best way to confront such attacks is to reply to the email instead of using links which are in the email.

2.4. CLICK HIJACKING

Click hijacking is a smart method in which hackers use social engineering to encourage users to click on the virus icon or file (advertisements, for example). Also the disruptive file may be sent as an ordinary email which contains a link to a video file of important or interesting news but they are redirected to the hacker's desirable page by clicking on the play button of the video file [15]. Figure 1 illustrates an example of a fake video player which requests to install flash player, but if user clicks on the install button, malware will be installed on the victim's system.



Flash player: Incorrect version

Flash player : Incorrect version
Your browser cannot display this site.

You need to download new version flash player
Object to play this video file.

Click Continue to download and install new flash player.

Figure 1: Fake Video Player.

2.5. WEB VULNERABILITY

There are various web vulnerabilities which can be used by attackers. After influencing a website, attackers will have a chance to contaminate it by disruptive code and can continue contaminate their viewers' personal computers. The most important security vulnerability are introduced as follow.

2.5.1. XSS ATTACKS

Script writing structure is used in this kind of attack and its main goal is to get information from users and clients who connected to the server. If web software is problematic and hacker can insert his disruptive script into software input fields, when users or clients run the disruptive code after they connect to the contaminated server, their system will be affected as well. Browsers are not able to detect these contaminating scripts [16]. When the disruptive script is executed on the target computer, the contaminated client's browser will follow instructions sent out by hacker; so hacker can send any instruction he wants and receive his required information. Most of codes entered by hackers in web software run at the client side and are usually written in JavaScript and so it can be injected to the server. In this kind of attack, hacker is looking for the input fields of information which is located in the website. If proper security concerns are not considered in web servers, the website will be affected. Most information obtained by XSS is cookies which client left on its browser. If validation is not conducted in information input fields, that website will be exposed to XSS attacks [17]. XSS attacks are divided into two groups which are known as reflection and reservation. In reflection attacks, hacker finds a security vulnerability and a way to use it to redirect the unaware user to a web application which has XSS vulnerability [18]. At this point the attack has been done. This attack is done by a series of URL parameters which are sent by URL. Hacker sends the disruptive URL to the users using available parameters in URL. This URL is usually sent to user by emails, weblogs, forums or any other possible method.

You may imagine if user does not click on unknown links; therefore he won't have a problem. But you should note that by using JavaScript, an XSS attack is done even with opening an email or visiting a website. In addition, in such attacks URLS are coded by methods such as Hex or any other coding method which display URLs as valid [19]. In reservation attacks, hacker stores disruptive codes which will be called by a user in future. In fact user will encounter disruptive codes and executed unconsciously.

2.5.2. SQL INJECTION ATTACKS

SQL is acronym for Structured Query Language which is a language for communicating with databases. This language can be used for queries from databases such as MSSQL, MySQL, Oracle or any other similar database. The operating structure of SQL injection attacks is when there is information input and output fields which needs a database in its background, by entering unusual SQL commands in these fields, responses can be received from the servers which may include sensitive information [20]. Login pages usually contain fields which get ID and Password form users and then process them using a SQL command. A simple example of such command is as follows:

SELECT COUNT(UserID) FROM tblUsers WHERE UserID=’” & UserID.Text & “‘ AND Pass=’” & Password.Text & ““”

In this example, UserID and Password are two textbox controls which their values should be processed. Now we suppose user enter these values as Mehdi and 123, and then SQL sentence is generated as:

SELECT COUNT(UserID) FROM tblUsers WHERE UserID=’mahdi’ AND Pass=’123

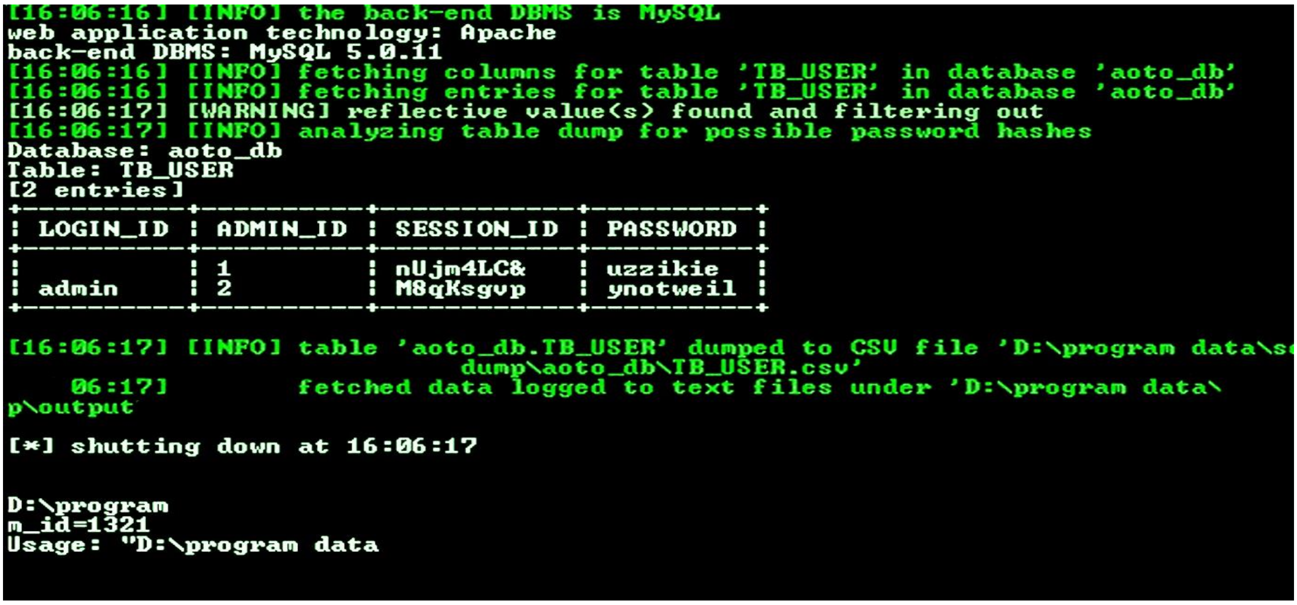
The performed query is a normal command. Now suppose that instead of username, attacker enter the following expression:

'OR 1=1-

Then the following SQL expression will be generated:

SELECT COUNT(UserID) FROM tblUsers WHERE UserID=” OR 1=1 – AND PASS=”

“-“ character in SQL is comment sign and its following expression will not be processed, consequently when processing the command, because it is always 1=1 and considering OR it will be adequate regardless of the value in the User ID and all records of information bank will be returned, in other words the attacker will be authenticated. In order to confront such attacks, website input fields should be filtered against meta characters and if it is required to enter meta characters in input fields, they should be replaced with numerical values and then processed [21, 22]. Figure 2 shows the extracted information from a site containing SQL injection bug. In these attacks, attackers replace some existing links in website with disruptive links after obtaining the website administrator's password; therefore the computers of website viewers will be attacked and contaminated.



```
[16:06:16] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0.11
[16:06:16] [INFO] fetching columns for table 'TB_USER' in database 'aoto_db'
[16:06:16] [INFO] fetching entries for table 'TB_USER' in database 'aoto_db'
[16:06:17] [WARNING] reflective value(s) found and filtering out
[16:06:17] [INFO] analyzing table dump for possible password hashes
Database: aoto_db
Table: TB_USER
[2 entries]
+----------+----------+------------+----------+
| LOGIN_ID | ADMIN_ID | SESSION_ID | PASSWORD |
+----------+----------+------------+----------+
|          | 1        | nUjm4LC&   | uzzikie  |
| admin    | 2        | M8qKsgvp   | ynotweil |
+----------+----------+------------+----------+

[16:06:17] [INFO] table 'aoto_db.TB_USER' dumped to CSV file 'D:\program data\s
                             dump\aoto_db\TB_USER.csv'
   06:17]         fetched data logged to text files under 'D:\program data\
p\output

[*] shutting down at 16:06:17


D:\program
m_id=1321
Usage: "D:\program data
```

Figure 2: Extracted Information from a Site Containing SQL Injection Vulnerability.

3. OUR APPROACH FOR SPYWARE CONTAMINATED WEBSITE DETECTION

Our goal is to confront the frauds which attempt to install malware on users' operating systems through contaminated websites. Installing malware can be done by click hijacking or influencing vulnerable websites in order to put links containing malware to contaminate vulnerable website to viewers' computers or forging emails and put contaminated websites' links in them.

Another way attackers use to contaminate their victims is called JDB Key logger. In fact this kind of key logger is a combination of software key logger and web key logger. In this method, the creators turn their final executable file to a java file. Then they put the created java file in the website and write the code for executing java file in that website. If someone visits the website, the key logger will be activated on his operating system even without the requirement for downloading and running malware file on victim's operating system.

In all described methods, social engineering is also used. Files with exe, jar, or bat extensions can illustrate contaminated websites. According to examination on 100 legal website and 40 malware contaminated website, links are always referring to database (such as 127.0.0.1?id=214324) when users need to download an executable file in legal websites , instead of links referring to file (such as 127.0.0.1/malware.exe) in most malware website. For example, installing flash player on online video players uses links referring to database. Figure 3 show our approach, after user's URL is received, the contents of the entered page are saved in a file with txt extension. The reason for using this saving method is to inactivate existing scripts in the page such as scripts containing JDB attack and reducing the size of the file. After saving the file in text format, the saved file content is scanned for links which are connected to executable files. If such links exist, the website is suspicious to be contaminated because existence of such links in a website seems unnatural as we described above. If WebCrawler methods are used for this purpose, there is a possibility that attackers bypass security concerns because WebCrawler only investigates links in current domain and if the malware is in another domain or it is linked externally, it cannot be discovered.

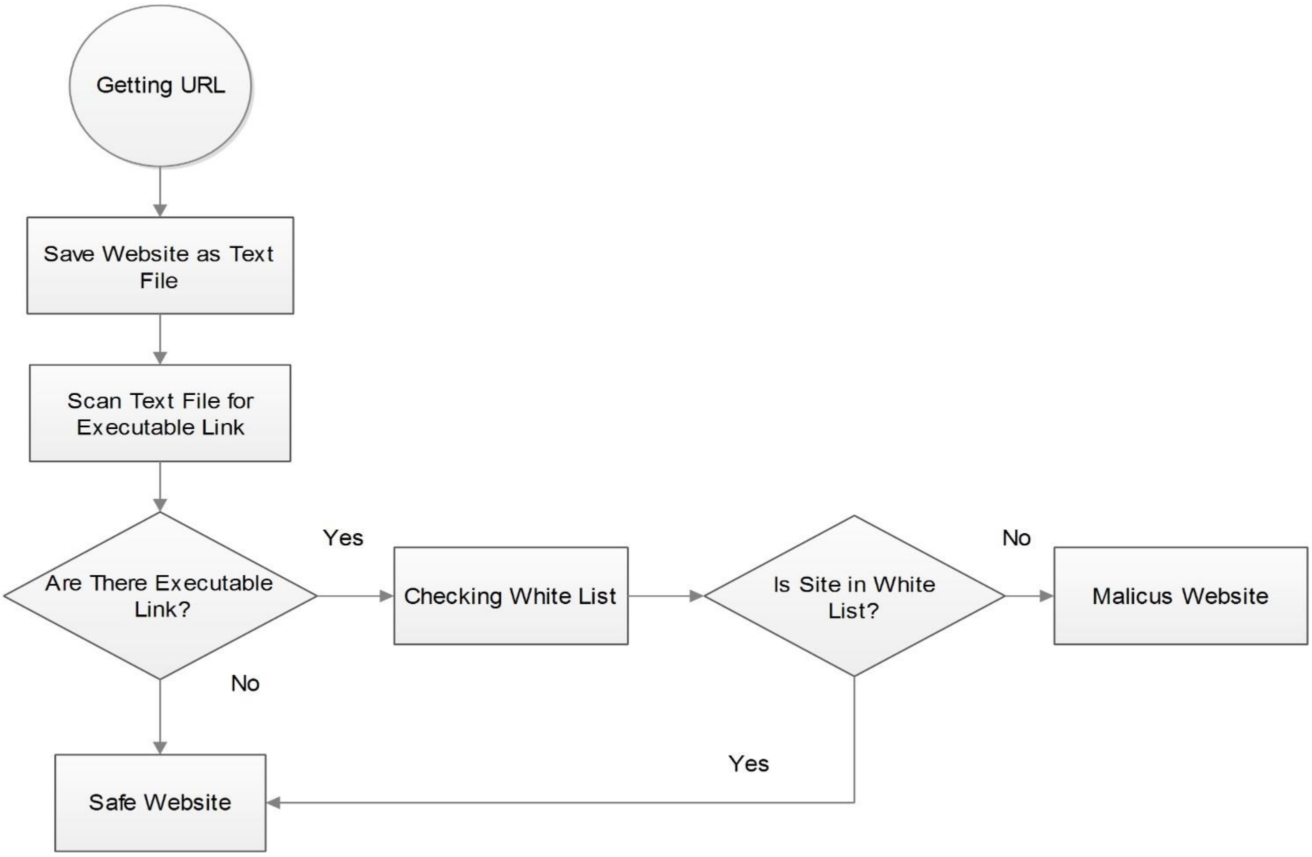


Figure 3: Suspicious Websites Detection Procedure.

However, existence of such links may be necessary in a website. In this case, we could use white list in our approach which includes secure and safe websites. Because white list approach is used, the approach needs less updating compared to black list approach. It can also be updated by user.

We use KMP [24] algorithm to search websites in white list. This algorithm examines the existence of string W in substring S, so that when a mismatch occurs, the word itself contains adequate information to determine the next location which the next match may begin from by re-testing previous matched characters. This algorithm, if needed, also allows all existing addresses in white list which contain this string to be found by entering some strings of the address. In this algorithm strings are demonstrated using zero based arrays, so the character 'C' in S={A, B, C}is shown as S[2]. In most circumstance, the algorithm is in a specified status which is determined by two integer numbers m and i. respectively, they specify positions in S which are the beginning of a possible match for W and an index of W that shows which character is being examined currently. With consecutive comparing of characters of W (some entered strings by user in order to find all websites containing these strings in white list) with corresponding characters of S, we go to the next character if characters are the same. In the second stage S[2] is 'a' and

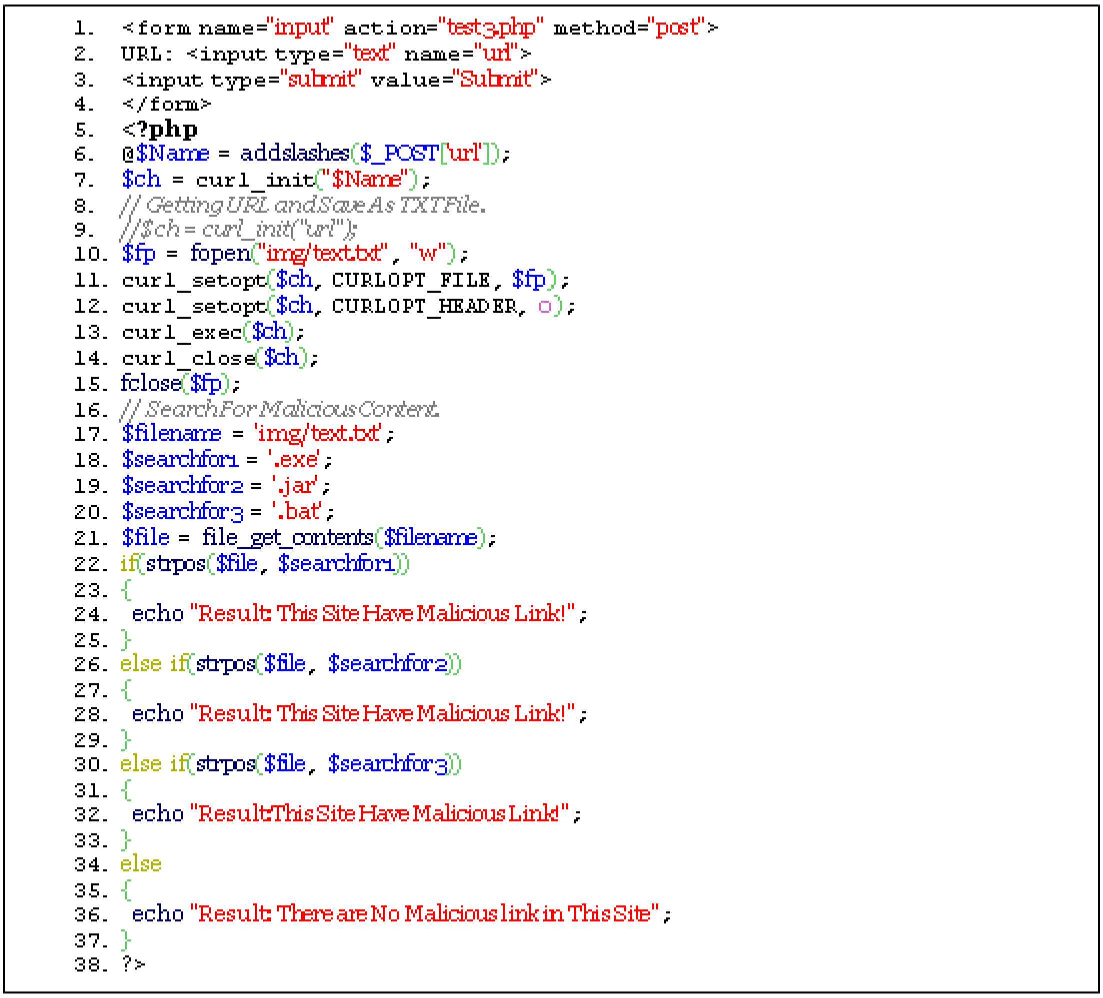
W[2] is 'b' which is a mismatch. Instead of restarting the search from S[1], we consider 's' between positions 0 and 2 is only seen in index 0 of S; knowing this and that those characters have been examined, we know if we examine them again, we will never find another match. So we go to the next character and consider m equal to 3 and i equal to 0.

```
          1       2
m: 01234567890123456789012
S: shahabisp.ir
W: shb
i: 0123456
```

If we intend to look for a complete address of a website in white list, the matched address is found by using the algorithm, but if a mismatch of one character is found in a string, that string will not be scanned again and scanning will be done for the next string (next available address in white list). It is obvious that in the sixth step S[6] is equal to 'i' and W[6] is equal to 'p', therefore the search operation in this string stops and scanning begins in the next string.

```
          1       2
m: 01234567890123456789012
S: shahabisp.ir
W: shahabpc.ir
i:    0123456
```

Figure 4 shows source codes related to scanner and website saver in text file format. In this approach user will be aware of contamination before visiting a website and regarding that website will be saved in a text file in application server (instead of user's computer), so there is no way the contamination can threaten user's operating system. In addition, this approach can be extended to detect other online malware through adding extensions of web based disruptive files.



```
<form name="input" action="test3.php" method="post">
URL: <input type="text" name="url">
<input type="submit" value="Submit">
</form>
<?php
@$Name = addslashes($_POST['url']);
$ch = curl_init("$Name");
// Getting URL and Save As TXT File.
//$ch = curl_init("url");
$fp = fopen("img/text.txt", "w");
curl_setopt($ch, CURLOPT_FILE, $fp);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_exec($ch);
curl_close($ch);
fclose($fp);
// Search For Malicious Content.
$filename = 'img/text.txt';
$searchfor1 = '.exe';
$searchfor2 = '.jar';
$searchfor3 = '.bat';
$file = file_get_contents($filename);
if(strpos($file, $searchfor1))
{
 echo "Result: This Site Have Malicious Link!";
}
else if(strpos($file, $searchfor2))
{
 echo "Result: This Site Have Malicious Link!";
}
else if(strpos($file, $searchfor3))
{
 echo "Result:This Site Have Malicious Link!";
}
else
{
 echo "Result: There are No Malicious link in This Site";
}
?>
```

Figure 4: Source Codes of Saving Website and Scanner in PHP Script Format.

4. EXPERIMENTAL TESTING AND ERROR RATE MEASUREMENT

For measuring the precision of the proposed approach, we adopted a list of 50 secure and malware contaminated websites. These websites are dividable into four groups: spyware contaminated websites, websites containing JDB attack, websites related to frequently visited sites, and websites belonged to download websites. Scanner's positive error rate is 16%, its false positive error rate is 8% and total error rate is 12%. The error rate is related to scanner without using white list. If we use white list, positive error rate will be reduced significantly.

5. CONCLUSION

In this paper, we first introduced malware and common malware detection methods. Then we briefly reviewed the new methods which attackers use to contaminate websites by malware today and presented the method to confront them. Finally we proposed our approach to detect websites contaminated by malwares which usually contaminate users' operating systems to steal their information. The proposed approach will give necessary warnings to the user before visiting a suspicious website and prevents the operating system to get contaminated. Compared to other approaches such as black list, white list, this approach requires less updating and shows a better performance.

6. ACKNOWLEDGEMENTS

This article is published by the professional team of Shahab Rayaneh Tiran Company (Iran, Isfahan, Tiran) and with technical support of Shahab Rayaneh Company. We also thank Mr. Mazaheri's support.

REFERENCES

[1] Bailey M, Oberheide J, Andersen J, Mao Z.M, Jahanian F and Nazario J, “Automated Classification and Analysis of Internet Malware”, Springer-Verlag Berlin Heidelberg, LNCS 4637, pp. 178–197, 2007.

[2] Fielding A and Connor M.J, “First Response Computer Virus Blocking”, Patent Application Publication, pp. 1-10, Sep. 2, 2004.

[3] Joukov N and Chiueh T, “Internet Worms as Internet-Wide Threat”, Stony Brook University, pp. 1-26, 2003.

[4] Ames W, “Understanding Spyware: Risk and Response”, IT Professional (IEEE Computer Society), Volume 6, Issue 5, pp. 25-29, 2004.

[5] Ortolani S, Giuffrida C and Crispo B, “Bait Your Hook: A Novel Detection Technique for Keyloggers”, Springer-Verlag Berlin Heidelberg, LNCS 6307, pp. 198–217, 2010.

[6] Sagiroglu S and Canbek G, “Keyloggers: Increasing Threats to Computer Security and Privacy”, IEEE Technology and Society Magazine, pp. 10-17, 2009.

[7] Herley C and Flor ˆencio D, “How to Login from an Internet Caf´e without Worrying about Keyloggers”, Symposium on Usable Privacy and Security (SOUPS) ’06, pp. 1-2.

[8] Skapinetz K, “Virtualization as a black hat tool”, Network Security, Volume 2007, Issue 10, pp. 4-7, 2007.

[9] Geer D, “Hackers Get to the Root of the Problem”, Computer (IEEE Computer Society), Volume 39, Issue 5, pp. 17-19, 2006.

[10] Sherly and InduShobha C.S, “An Overview of Social Engineering Malware: Trends, Tactics, and Implications”, Technology in Society, Volume 32, Issue 3, pp. 183–196, 2010.

[11] Alamgir Khan A, “Preventing Phishing Attacks using One Time Password and User Machine Identification”, International Journal of Computer Applications, Volume 68, No.3, pp. 7-11, 2013.

[12] Hong J, “The State of Phishing Attacks”, Communications of the ACM, Volume 7, No.1, pp. 74-81, 2012.

[13] James D and Philip M, “A Novel Anti Phishing framework based on Visual Cryptography”, International Conference on Power, Signals, Controls and Computation (EPSCICON), IEEE, pp. 1-5, Thrissur, Kerala, 3-6 Jan. 2012.

[14] Sharma D, “Integrated Network Electronic Warfare: China's New Concept of Information Warfare”, Journal of Defense Studies, Volume 4, No.2, pp. 36-49, 2010.

[15] Duchene F, Groz R, Rawat S and Richier J.L, “XSS Vulnerability Detection Using Model Inference Assisted Evolutionary Fuzzing”, International Workshop on Security Testing, pp. 1-3, Montreal, Canada, 2012.

[16] Weinberger J,Saxena P, Akhawe D, Finifter F, Shin R and Song D, “An Empirical Analysis of XSS Sanitization in Web Application Frameworks”, Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report, pp. 1-17, 2011.

[17] Austin A and Williams L, “One Technique is Not Enough: A Comparison of Vulnerability Discovery Techniques”, International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, pp. 97-106, Banff, AB, 22-23 Sept. 2011.

[18] Bates D, Barth A and Jackson C, “Regular Expressions Considered Harmful in Client-Side XSS Filters”, Proceedings of The 19th International Conference on World Wide Web, ACM, pp. 91-100, Raleigh, North Carolina, USA, 26–30 April. 2010.

[19] Kontaxis G, Polakis I, Polychronakis M and Markatos E, “dead.drop: URL-based Stealthy Messaging”, Seventh European Conference on Computer Network Defense, IEEE, pp. 17-24, Gothenburg, 6-7 Sept. 2011.

[20] Tajpour A, Ibrahim S and Masrom M, “SQL Injection Detection and Prevention Techniques”, International Journal of Advancements in Computing Technology, Volume 3, Number 7, pp. 82-91, 2011.

[21] Srivastava S, “A Survey On: Attacks due to SQL injection and their prevention method for web application”, International Journal of Computer Science and Information Technologies, Volume 3, Number 1, pp. 3225-3228, 2012.

[22] Das D, Sharma U and Bhattacharyya D.K, “An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching”, International Journal of Computer Applications, Volume 1, No. 25, 28-34, 2010.

[23] Madaan R, Dixit A, Sharma A.K and Bhatia K.K, “A Framework for Incremental Hidden Web Crawler”, International Journal on Computer Science and Engineering, Volume 21, No. 3, 753-758, 2010.

[24] Wikipedia, “Knuth–Morris–Pratt algorithm”, Aug 2014, Online Document available at:

http://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm