We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,000
Open access books available

125,000
International authors and editors

140M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

A Detection of Malware Embedded into Web Pages Using Client Honeypot

M. Veena, S. Upasana, S. Prathima and Sudha Senthilkumar

Abstract

In today's Internet world, web pages are facing a severe threat which uses the client-side browser attacks. The vulnerability-based attacks are based on client-side application which becomes the major threat to web pages. The spread of malware uses software vulnerabilities which attack the client application sending request to the server if whether the attack has occurred. This detection is based on client honeypot which detects the various malicious program linked with web pages. Client honeypots are active security devices in search of malicious servers that attack clients. The client honeypot poses as a client and interacts with the server to examine whether an attack has occurred. Often the focus of client honeypots is on web browsers, but any client that interacts with servers can be part of client honeypot. In this research paper, we propose a model of detecting embedded web pages using client honeypot.

Keywords: client honey pots, network, malware, cyber space, client-side attack

1. Introduction

The Internet has become the most popular medium where the increasing new trend has availability of spreading attacks. There are existing attacks as firewalls and intrusion detection system, and honeypot is also one of the technology-based security attacks. Here honeypots are playing a big role in security attacks, and it is a new kind of attacks on the cyber space. There are many attacks in the area of security. Honeypots refer to closely monitoring system which needs to be attacked. To supplement a new value it must be compromised, attacked by cyber crimes on honeypot.

Honeypots are based on servers which will not be able to detect the client-side applications in web pages. Honeypots crawl the network of any firewall that attacks the client. There are two attacks: client and server. Server honeypots is a traditional honeypot, whereas client honeypot is based on client-side scripting on web pages. This attack detects the attack from the client side which is vulnerable on the client side. It needs a source and visits and detects all activities. It spreads malware through the vulnerability in the client-side attack. Here we are using only client honeypot and vulnerability-based attack. The vulnerability is based on exploiting the attack where the security detects the services exposing on attacking the system.

Client honeypots crawl the network, interact with servers, and classify servers with malicious web pages. It does not expose server-based attacks on the client-side

attack. They have some kind of exposed attack which is vulnerable to the sender and receiver. They can be detected, if they are passive.

2. Proposed work

The security resources are production value; no resources should communicate between each other. Honeypot is compromised for outbound connections on the web pages. Honeypot collects all information about the intruder or intermediate where the community is targeting to attack. And they list the type of resources attack on the network security. Honeypots play the big role on the attacker side scripting based on web pages in client-side attack. Client honeypots are also called as active honeypots or honey client. It visits the web page as requested by the attacker and visits the web page to check whether the attack has happened or not [1–3] (**Figure 1**).

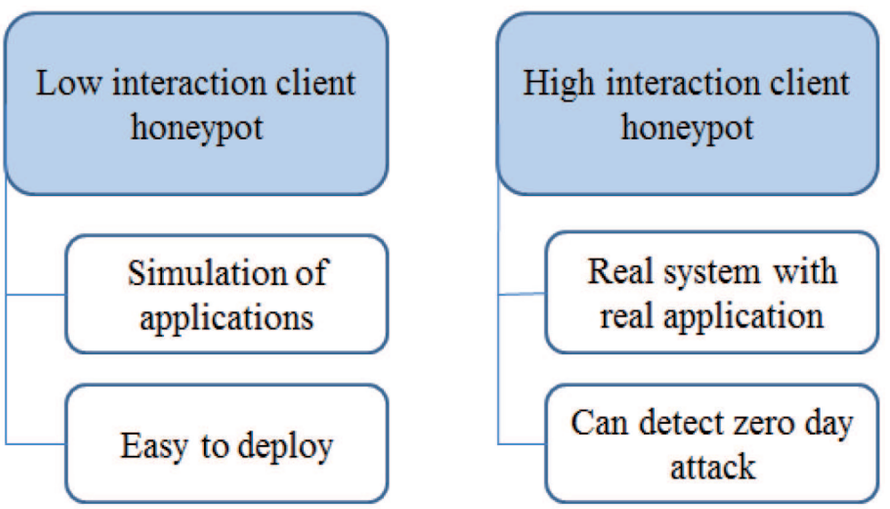

Figure 1.
Client honeypot classifications.

3. Client honeypots

A honeypot is one of the security technologies that helps an organization to catch viruses, malware, or attackers, and it acts as an alarm system that discovers the attempts to attack a network. Honeypot technology is defined as a "security resource whose value lies in being investigated, attacked, or compromised" [1]. The types of honeypot are active and passive. A technology that passively waits for attacks to detect them are called passive honeypot. Active honeypot also called as client honeypot that interacts with a target web page to find its possible effect on the honeyware.

4. Architecture

The client honeypot architecture is separated into three components, namely, queuer, the client, and analysis engine. A queuer is a process of creating a list of servers for the client to visit. A client who can able to create request to servers is recognized by the queuer. An analysis engine is a process of identifying an attack processing in client honeypot. Along with all the above components, client honeypot

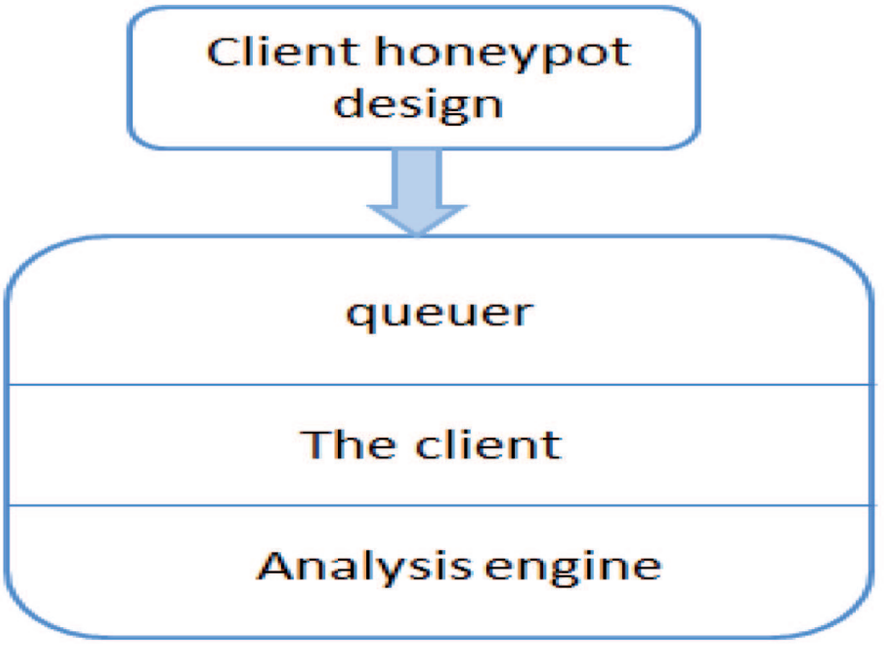

Figure 2.
Client honeypot design.

is furnished with some kind of approach to avoid successful attacks from exploring beyond the client honeypot [4, 5]. Analog to traditional server and client honeypots are classified by their high- or low-interaction level that denotes the client honeypot make utilize of functional interaction the server. This is a newly hybrid approach that uses both high- and low-interaction detection techniques (**Figure 2**).

5. Client honeypot solutions

5.1 High-interaction client honeypots

High-interaction client honeypot is a real application installed on the real systems. Real browsers and plug-ins are being browsed by the websites. Attacks are detected by checking the state of the process after a server interaction. Capture differentiates from existing client honeypots in different ways. It is designed to be fast and to be scalable. Event-based model allow to know the detection of state changes. A main capture server can able to manage several clients across the network.

Honeyclient is a web browser. It is an open-source honeypot and a mix of perl, c++. It detects attacks on Windows client by registry entries, monitoring files, and processed events. It included the capture-HPC. It also contains a crawler, so that it can be sowed with a list of URLs from start and continues to exchange web pages in search of client-side malware. HoneyMonkey is also a web browser. It is not an open source. It detects attacks on Windows client by registry entries, monitoring files, and processed events. It is a layered approach to communicate with servers to identify zero-day exploits. If the attack is still identified, one can complete the attack as no patch has been publicly released and it is dangerous [6–8]. SHELIA is a combination of the process of email received and email reader. It opens different client applications depending on the type of URL or the received attachment. It observes the executable instructions that are processing in data area of memory that indicates a buffer. UW Spycrawler is integrated; with the web browser like Mozilla, it cannot be downloaded. It detects attacks on Windows client by registry entries, monitoring files, browser crashes, and processed events. Event-based mechanism is used to detect by spcrawlers [9, 10]. It increases the

time period of the virtual machine. It is a process to overcome time bombs. WEF is an automatic implementation of drive by download that detects in virtualized environment. WEF is used as an active HoneyNet with overall simulated architecture beneath for rollbacks of compromised virtual machines.

5.2 Low-interaction honeyclient

Low-interaction honeyclient is different from high-interaction honeyclient in that they do not use the entire real system. But it uses lightweight or simulated clients to communicate with the server. Responses received from servers are scanned directly to consider whether an attack has been taking place or not. It is a platform-independent open-source framework written in Ruby [11]. It concentrates on driving a web browser emulator which interacts with the server. Mischievous server is identified by statically investigative the web server's response for mischievous string through the usage of snort signatures. Honeyclient uses many existing freely available open-source software systems. It consists of the following components:

5.2.1 Queue/seed generation

It is a source of initial set of seed URLs.

5.2.2 Web search seeding

The three web search engine application interface are Google, Yahoo, and MSN which are common keywords.

5.2.3 Spam trap seeding

Spam mails are extracted from URLs and are enlisted.

5.2.4 Blacklist seeding

It is a tool designed to automatically download blacklist from major blacklist workers and seed for crawler [12, 13].

5.2.5 Web crawling

Heritrix crawler is simulated into the monkey spider prototype with two parameters predefined:

- Maximum link hops which counts the connections to be included in crawl
- Maximum transitive hops which count the URLs extracted from seeded URLs

5.2.6 Content/malware analysis

5.2.6.1 Static analysis

The contents that are downloaded from the URL are scanned by ClamAV antivirus and it alerts using pattern matching. The terminology is provided for the downloaded binary [14, 15].

5.2.6.2 Dynamic analysis

Malware analysis tool like CWSandbox is performed [16].

It is implemented to copycat the behavior of a user-driven network client application and abused by an attacker's content. It is a virtual honeyclient which means that it is not a real application but it is an emulated client. It performs dynamic analysis of JavaScript and visual basic scripts to delete the complication from malicious pages. To analyze the malicious content, complication or encrypted JS is decrypted and reanalyzed. SPYBYE allows a web master to identify whether a website is eroded by a set of heuristics and scanning of data against the clamAV. It is a tool that communicates with a URL that is integrated with a web browser through its user agent field and downloading the response of the target website. The response is exploited using the scan engine [17, 18].

5.3 Problem statement

The problem explained in this paper is to identifying and extracting malicious web programs combined into malicious websites on the client honeypot technology. It is used to address the problem to determine malicious websites based on the maliciousness inside the websites. It helps to rectify the difficulty of identifying whether we need to visit the website or not. Client honeypot technology can able to identify the client-side attacks by surfing the web pages [19, 20].

During the implementation, the steps of generic algorithm are used in the URL separation and feed within the virtual machine for visit:

1. List of URL to be visited, say N numericals of URLs.

2. Store the list of URLs into the database.

3. Get the URL one by one from mysql database.

Figure 3 represents the process of the system listing websites which have been presented as having visited a true browser on a client honeypots. By using clean machine of client honeypots, after visiting the clean website taking snapshot of that machine and store the log created at the stage of website visit.

System logs are created in client machine that are analyzed using third-party analyzing tool like antiviruses to identify the contagions on the list of collected logs.

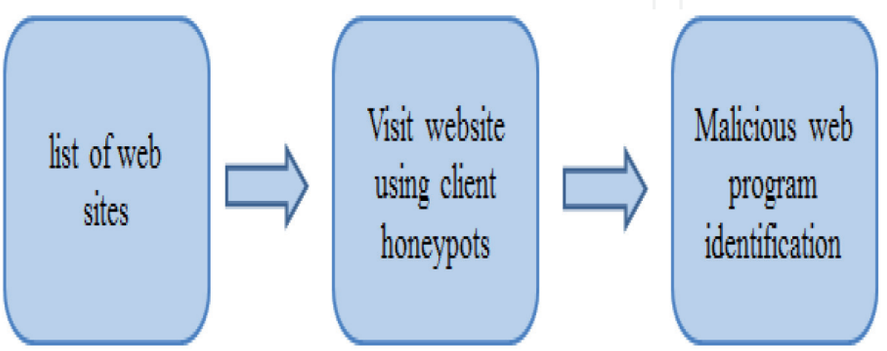

Figure 3.
Process of identifying malicious program.

6. Conclusion

In this paper, we presented a study of current solution of client honeypots for identifying the mischievous websites, and it is not applicable for closely bound and public users. So we propose a system which is able to identify the malware platforms with the help of client honeypot and put on the clever forensic inquiry of the collected network data.

Author details

M. Veena[1], S. Upasana[1], S. Prathima[1] and Sudha Senthilkumar[2*]

1 VIT University, Vellore, India

2 School of Information Technology and Engineering, VIT University, Vellore, India

*Address all correspondence to: sudha.s@vit.ac.in

References

[1] Sun X, Wang Y, Ren J, Zhu Y, Liu S. Collecting Internet malware based on client-side honeypot. In: 9th IEEE International Conference for Young Computer Scientists (ICVCS 2008); 2008. pp. 1493-1498

[2] Spitzner L. Honeypots: Tracking Hackers. Addison Wesley; 2002

[3] Honeynet Project. Know Your Enemy: Defining Virtual Honey/nets. 2003. Available from: http://old.honeynet.org/papers/virtual/

[4] Danford R. —2nd Generation Honeyclients. SANS Internet Storm Center; 2006. Available from: http://handlers.dshield.org/rdanford/pub/Honeyclients_Danford_SANS 06.pdf

[5] Available from: http://www.honeyclient.org/trac

[6] Available from: https://projects.honeynet.org/capture-hpc

[7] Available from: http://en.wikipedia.org/wiki/HoneyMonkey

[8] Available from: https://projects.honeynet.org/honeyc

[9] Available from: http://www.xnos.org/security/overview

[10] Available from: http://code.google.com/p/phoneyc/

[11] Available from: https://en.wikipedia.org/wiki/Client_honeypot

[12] Available from: http://www.spybye.org/

[13] Alosefer Y, Rana O. Honeyware: A web-based low interaction client honeypot. In: Third International Conference on Software Testing, Verification, and Validation Workshops (ICSTW); 2010

[14] Available from: http://www.honeynet.org/node/158

[15] Ramaswamy C, Sandhu R. Role-based access control features in commercial database management systems. Citeseer 1998

[16] Skoudis E, Zeltser L. Malware: Fighting Malicious Code. Prentice Hall; 2003. p. 3

[17] Available from: http://monkeyspider.sourceforge.net/

[18] Available from: http://www.cs.vu.nl/~herbertb/misc/shelia/

[19] Provos N, McNamee D, Mavrommatis P, Wang K, Modadugu N. The ghost in the browser analysis of web-based malware [Online]. 2007. Available from: http://www.usenix.org/events/hotbots07/tech/full_papers/provos/provos.pdf [Accessed: 11 February 2009]

[20] Secure Browsing. Malware Protection. Trustwave. Available from: https://www.trustwave.com/securebrowsing/