A Network-based Solution to Kaminsky DNS Cache Poisoning Attacks

Tien-Hao Tsai
Chunghwa Telecom Laboratories,
Yang-Mei, Taiwan, ROC,
skyno717@gmail.com

Yu-Sheng Su
Research Center for Advanced Science and Technology
National Central University
Taoyuan, Taiwan
ncuaddison@gmail.com

Shih-Jen Chen
Institute for Information Industry
Taipei, Taiwan
sjchen@iii.org.tw

Yan-Ling Hwang
School of Applied Foreign Languages
Chung Shan Medical University
Taichung, Taiwan
yanling@csmu.edu.tw

Fu-Hau Hsu
Department of Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
hsufh@csie.ncu.edu.tw

Min-Hao Wu
Department of Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
mhwu@csie.ncu.edu.tw

***Abstract*—In this paper, we propose a network-based solution, *Cache Poisoning Solver* (CPS), to defend an organization against the notorious Kaminsky DNS cache poisoning attack. DNS cache poisoning has been used to attack DNS servers since 1993. Through this type of attacks, an attacker can change the IP address of a domain name to any IP address chosen by him. Because an attacker cannot obtain the transaction number and port number of a DNS query sent by a DNS resolver, in order to forge the related DNS response with one of the attacker's IP address, the attacker needs to send many fake DNS responses to the related resolver. All these fake DNS responses map the target domain name to the above attacker's IP. Based on this observation, CPS solves DNS cache poisoning by detecting, recording, and confirming the IP addresses appearing in contents of fake DNS replies. As a result, CPS not only can block DNS cache poisoning attacks but also can identify the malicious hosts, which attackers plan to use to redirect target hosts' traffic. Usually, these malicious hosts are botnet members and used as phishing sites; hence, identifying these bots and disconnecting traffic to them can provide further protection to the hosts in a network. Besides, through the utilization of Bloom Counter and host confirmation, CPS maintains its detection accuracy even when it is bombarded with tremendous fake DNS replies. Experimental results show that with low performance overhead, CPS can accurately block DSN cache poisoning attacks and detect the related bots.**

Keywords-*DNS; resolver; cache poisoning attack*.

I. INTRODUCTION

Domain Name System (DNS) is an important part of the Internet. DNS provides mapping between domain names and IP addresses. With its assistance, network applications, such as web browser, FTP client, and E-mail client and server, can find the location of their communication targets easily. To reduce the processing time, DSN-related payload is usually delivered through UDP packets [10]. However, UDP is a less reliable protocol than TCP. In addition, it is difficult to check the correctness of UDP packet payload. To enhance the reliability of DNS, DNS only accepts answers in a DNS query whose IP address, port number, and Transaction ID (a random 16-bit number) match the related DNS query. DNS cache poisoning is an attack that changes the IP address of a domain name to any IP address chosen by the attacker. In the past, due to the difficulty to obtain the transaction ID and port number of a DNS query, a DNS cache poisoning attack was usually launched through sending a large amount of packets with various port numbers and Transaction IDs to increase its chance to match the port number and transaction ID of an unsolved DNS query.

In 2008, Kaminsky [1] presented a threatening model making the attack easier. Following this model, Hubert and Van Mook [2] shows that, by sending 7000 forged packets per second (around 4.5MB/Sec) to a strict-port DNS resolver, a Kaminsky attack could have a 50% chance to spoof the DNS resolver only in 7 seconds. We call the success probability a cache poisoning attack has the *spoofing probability* of the cache poisoning attack. Fortunately, if 64,000 ports are randomly used, it will cost more than 116 hours to reach the 50% spoofing probability. However, if an attacker increases the rate of issuing forged DNS responses to 4.5 GB/Sec, it could get 50% chance after 7 minutes. Nowadays, the above transmission requirement is easy to be satisfied for most bot masters who can easily control tens of thousands of bots simultaneously. Hence, developing an anti-cache poisoning attack solution that is also robust enough to handle Kaminsky attacks becomes an important issue.

In this paper, we propose a network-based solution, Cache Poisoning Solver (CPS), to defend an organization against the notorious DNS cache poisoning attack. CPS also records IP addresses appearing in fake DNS messages. These IP addresses usually belong to the hosts that are bots of some botnets and perform malicious activities, such as phishing, launching drive-by-download attacks. Thus, CPS further blocks traffic to or from these IP addresses. CPS only records the IP addresses, which appear in many DNS responses, because an authoritative name server only uses one DNS response to notify a resolver the IP address of a domain name. By counting Bloom filter [4], we can effectively observe the incoming frequencies of fake DNS responses in a cache poisoning attack.

The rest of this paper is organized as follows. Section 2 describes the system structure of the CPS. Section 3 analyzes the effectiveness and overhead of the CPS. Section 4 discusses previous work. Section 5 concludes this paper.

 ISBN: 978-1-61208-285-1

II. SYSTEM STRUCTURE

As shown in Fig. 1, there are three major components in CPS: IP collector, analysis crawler, and traffic controller. The IP collector is inside a DNS resolver to collect IP addresses appearing in DNS responses. The traffic controller resides at a router. Based on the malicious IP addresses extracted by the IP collector and analysis crawler, the traffic controller blocks traffic to or from malicious IP addresses. The analysis crawler analyzes the hosts with malicious IP addresses to gather more information about these hosts. This section gives a detailed introduction about these components.

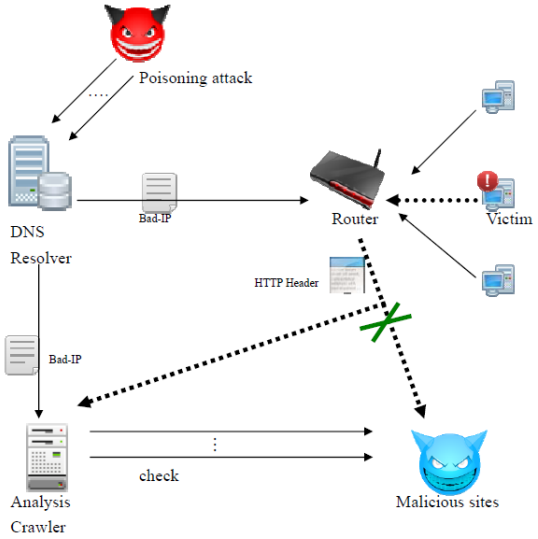

Figure 1. CPS system structure

A. IP Collector

The IP collector on a DNS resolver monitors all DNS queries received by the resolver, and looks over each DNS response to check if it matches a previous DNS query. A DNS response provides the IP address of a domain name, called *response domain name* hereafter. A DNS response matches a DNS query only if they have the same Transaction ID, port number, and IP address. The above matching rule is also adopted by most DNS software to verify a DNS response. We call the triple (Transaction ID, port number, IP address) of a DNS query or DNS response the *DNS packet ID* of the packet hereafter. After a resolver sends a DNS query to an authoritative name server to query the IP address of a domain name, before the server sends the corresponding DNS response to the resolver, the DNS query is called an *unsolved DNS query* and the queried domain name is called an *unsolved domain name*. A DNS response is called a *candidate DNS response*, if there is an unsolved DNS query whose unsolved domain name matches the response domain name of the DNS response. The DNS packet ID of a candidate DNS response may or may not match the DNS packet ID of the related unsolved DNS query. The IP collector only handles candidate DNS responses. Non-candidate DNS responses are ignored by the IP collector. A candidate DNS response, which does not have a matching DNS query is deemed as a *suspected DNS response*. A suspected DNS response could be a *fake DNS response* issued by a DNS cache poisoning attack.

A group of suspected DNS responses that try to set a domain name to the same IP address is called a *fake DNS response set*. Because an authoritative name server does not reply a DNS query with multiple DNS responses, a group of DNS responses that try to answer the same DNS query must try to set the IP address of a domain name to an IP address controlled by an attacker. We call the above IP address a *cheat IP address* of the fake DNS response set. The above domain name is usually contained in the additional section of a DNS response. Even though the domain name may also be contained in the answer section of a DNS response, it appears in old non-efficient cache poisoning attacks and it rarely happens nowadays.

IP collector maintains two lists, *suspicious IP list* and *black IP list*. The former contains the cheat IP addresses of fake DNS response sets whose sizes are greater than a threshold, called *size threshold*. The later contains the IP addresses, which have been confirmed to be used in malicious activities.

IP collector extracts information from suspected DNS responses, such as (1) IP addresses in the answer section and IP addresses (cheat IP addresses) in the addition sections of the DNS responses and (2) target name servers in the authority sessions of the DNS responses. Since fake DNS responses usually contain the IP addresses of bots, intuitively we can collect these IP addresses to unveil a partition of some botnets. After the IP collector extracts the cheat IP address from a suspected DNS response, it adds the cheat IP address to its bloom counter. If the counter of the cheat IP address is greater than the size threshold, it means that someone may be launching a cache poisoning attack to map the cheat IP address to a target domain name. The cheat IP address is added to the suspicious IP list of the IP collector. Whenever 9,000 ~ 10,000 cheat IP addresses are added to the suspicious IP list, the hash tables used by bloom counter are cleared to yield space to store new cheat IP addresses. To prevent the IP address of the target domain name from being poisoned, the IP collector performs a DNS lookup immediately to find the real IP address of the target domain. Hence, later on, even if an attacker sends a DNS response with the correct DNS packet ID, the real IP address of the target domain will not be replaced by a cheat IP address. The cache poisoning attack can be blocked.

The suspicious IP list of CPS only records cheat IP addresses whose corresponding fake DNS response sets contain more than size threshold suspected DNS responses during a period of time. Based on this strategy, CPS can decrease the amount of IP addresses to record in its *suspicious IP list*. We will discuss the size threshold later in this paper.

CPS extracts the following information from a DNS response of a suspected DNS response set, which contains more than size threshold suspected DNS responses.

1. The legal domain, name servers, and the fake IP addresses in the additional and authority session.
2. The counterfeit destination IP corresponding to the domain in the answer session.

 ISBN: 978-1-61208-285-1

B. *Analysis Crawler*

UDP packets are easy to forge and difficult to check the correctness of the sources; hence, an attacker may pollute the suspicious IP list of a resolver with IP addresses, which are not owned by the attacker. Thus, cheat IP addresses will be further analyzed by analysis crawler to avoid misjudging normal IP addresses as malicious IP addresses. Because web sites are frequently involved in various attacks, our analysis focuses on checking whether a suspicious IP is used by a malicious web site. The analysis crawler sends HTTP requests to the IP address to check whether the host with the IP address is a web server. If it is a web server, CPS utilizes [9] to check whether the web site is a benign one or a malicious one. A malicious web site may contain a phishing page or launch drive-by-download attacks. To reduce the number of IP addresses to check, IP addresses in "Alexa Top 500 Global Sites" [3] are skipped and classified as benign IP addresses. Besides, to further improve the performance overhead of the CPS, the CPS only performs the above check when an inner host tries to contact an external host with the IP address in the suspicious IP list. We call this approach *lazy confirmation*. IP addresses that are confirmed to be malicious ones will be added to the *black list* in the IP collector. After the examination, the IP address is removed from the suspicious IP list.

C. *Traffic Controller*

The traffic controller of CPS blocks any IP packet with an IP addresses listed in the black list. When the router receives an IP packet with an IP address listed in the suspicious list, the traffic controller informs the analysis crawler of this event so that the later can perform lazy confirmation to check whether the IP is a benign one.

III. ANALYSIS AND EVALUATION

This section analyzes the probabilities of successfully polluting a DNS cache under various fake DNS response rates and discusses the size threshold that CPS uses to move a cheat IP address into the suspicious IP list. This section also discusses various overhead introduced by CPS.

A. *Analysis*

In this section, we analyze the success probability a cache poisoning attack can have and the time it takes to complete an attack when various approaches are used to launch such an attack. In addition, we also discuss the thresholds of incoming rates and incoming duration of a DNS response set.

The probability that a resolver is polluted in one second of cache poisoning attacks is denoted as P_S.

$$P_S = \frac{W * R}{N * P * I} \qquad (1)$$

W: Window of opportunity, a period of time (in seconds), bounded by the response time of the authoritative servers (often 0.1 Sec)
R (*incoming rate*): Number of fake DNS responses sent per second. The fake DNS responses belong to the same fake DNS response set.
N: Number of authoritative Name Servers for the domain (around 2.5 on average)
P: Number of available UDP ports (maximum value is around 64000 as ports under 1024 are not always available)
I: Number of Transaction IDs (maximum 65536)

The probability that a resolver is polluted in T seconds of cache poisoning attacks is denoted as P_T. T is larger than or equal to TTL.

$$P_T = 1 - (1 - P_S)^A = 1 - \left(1 - \frac{W * R}{N * P * I}\right)^{(T/TTL)} \qquad (2)$$

According to the Kaminsky method, TTL is equal to W (Window of opportunity). So, equation (2) becomes:

$$P_T = 1 - \left(1 - \frac{0.1 * R}{2.5 * 64000 * 65536}\right)^{(T/W)} \qquad (3)$$

Fig. 2 shows that the probability of successfully polluting a resolver under different incoming rates. However, some domains are processed by only one authoritative name server. Under this environment, the value of N becomes one and the time it takes to pollute a resolver decreases around 40%. Fig. 3 shows the probability that a resolver is polluted under different incoming rates when the number of authoritative name servers is 1 and 2.5.

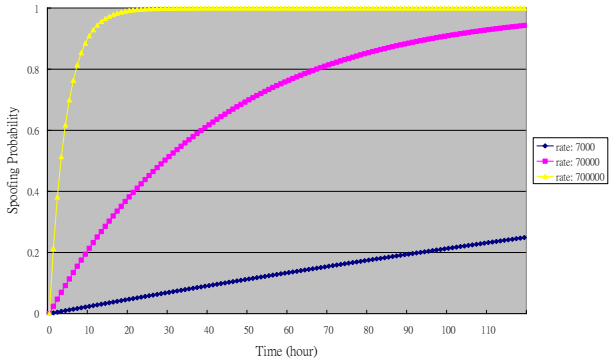

Figure 2. The probability of successfully polluting a resolver under different incoming rates

The suspicious IP list of CPS only records the cheat IP addresses in a fake DNS response set whose size is greater than 5 packets in 50 seconds. In other words, to avoid being recorded by CPS, an attacker cannot send more than 5 fake DNS responses every 50 seconds. We use *5-50 thresholds* to represent the above pair of thresholds.

The result of the 5-50 thresholds can be seen in the following paragraph. If an attacker wants to have a 0.01 success probability when launching a cache poisoning attack without being detected by the CPS, he needs to spend 490 days to continuously send fake DNS responses that map a domain name to the same IP address. However, the above price can only map the IP address of a domain name to the IP address of a bot controlled by the attacker.

 ISBN: 978-1-61208-285-1

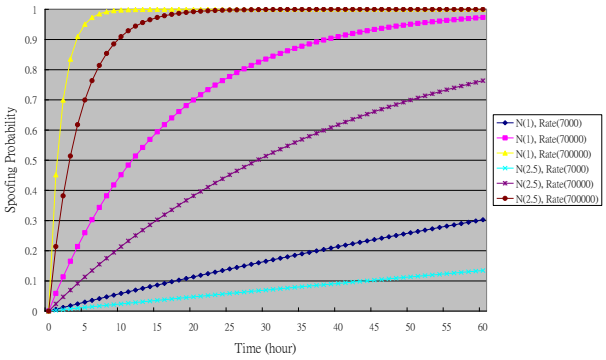

Figure 3. The probability that a resolve is polluted under different incoming rates when the numbers of authoritative name servers are 1 and 2.5

However, if an attacker controls a botnet, the attacker can reduce the attack time by launching a cache poisoning attack through issuing multiple DNS response sets from several bots simultaneously. Each DNS response set maps the same domain name to a different cheat IP address. Each different cheat IP belongs to a different bot of the attacker's botnet. Because the attacker controls all the bots whose addresses appear in the above DNS response sets, no matter, which fake DNS response set successfully changes the IP address of the target domain to the IP address of an attacker's bot, the attacker can redirect victims' traffic to that domain name to his bot.

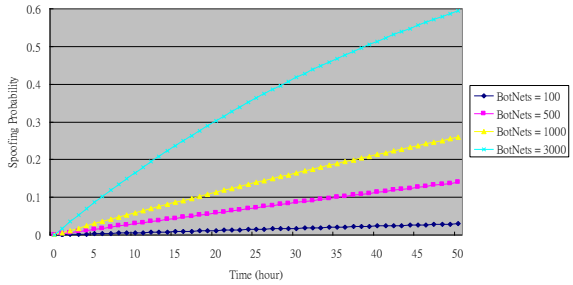

Figure 4. The spoofing probability when 100, 500, 1000, and 3000 DNS response sets are used to attack

Fig. 4 shows the success probability when multiple DNS response sets are used to attack and each DNS response set sends a fake DNS response using its highest incoming rate and incoming duration. In Fig. 4, the numbers of bots involved are 100, 500, 1000, and 3000. As shown in Fig. 4, when 1000 DNS response sets were used, the time it takes to complete a cache poisoning attack with 0.01 success probability is only 700 minutes. Hence, the lower the thresholds are, the more bots the attacker needs to use. In other words, if an attacker only wants to spend 7 minutes to have a 0.01 success probability to fake the IP address of a single domain name, he needs to use 100,000 bots, which is inefficient for the attackers.

B. *Evaluation and Discussion*

We built an IP collector on a DNS resolver with Intel Celeron 2.93GHz CPU and running the Ubuntu 9.10 operating system. To measure the performance overhead, we sent 5000 queries in different time periods of three days. We notice that the extra cost of our IP collector is very little and the usage of CPU is almost not increasing. We simulated attacks by sending fake DNS messages with the rates 0, 2000, 20000, and 120000 packets/sec. The zero rate means no attack. We use the average query time of 5000 DNS queries to represent the query time. Fig. 5 and Table 1 show that our extra overhead is around 3% in normal situation.

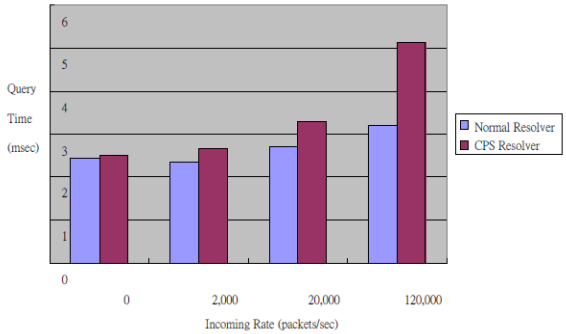

Figure 5. CPS performance overhead

TABLE I. PERCENTAGE OF CPS OVERHEAD

Incoming Rate	CPS Overhead
0	0.03
2,000	0.14
20,000	0.22
12,0000	0.60

C. *Attack Analysis*

UDP packets are easy to forge and difficult to confirm the correctness of the sources. However, sending non-candidate DNS responses (subsection III. *A*) does not have any influence on the bloom filter or of the suspicious IP list of the IP collector, because CPS ignores non-candidate DNS responses. As a result, an attacker may send plenty of candidate DNS responses to cause the analysis crawler busy confirming cheat IP addresses that appear in more than three candidate DNS responses, which in turn causes a DoS attack on CPS. However, with or without CPS, an attacker still can launch a DoS attack upon a local network. Hence, CPS does not make things worse, even though it makes the threshold to complete a DoS attack lower.

IV. RELATED WORK

This section discusses various solutions to the cache poisoning attacks. DNSSEC [6] is one of the most famous solutions of cache poisoning attacks. DNSSEC uses the asymmetric cryptography and verifies the DNS resource record by digital signature (*RRSIG*). This kind of authority needs an upper layer name server approving the public key (*DNSKEY*) by assigning the *DS*. DNSSEC provides extreme security to DNS, but it is not popularly spread.

 ISBN: 978-1-61208-285-1

A response packet often shows the correctness in the authority and additional session. Each session includes the name of the authority server and server's IP addresses. While a domain does not exist, Google name server will respond "No Such answer", but exclude the IP address of the server. Most of these attacks commit mapping a malicious IP to a target name server. Google prevents the spoofing by giving up the unreliable cache data. It's an easy way to defend poisoning but the new protocol is not deployed yet.

Kalafut *et al.* [5] use Autonomous System (AS) number to enhance history and shows that IP address may change but AS number would be stable. However, it has 0.2~3.1% false positive so it's not a robust solution.

DepenDNS [8] is built on client computers and concurrently queries multiple different resolvers to verify a trustworthy answer. It gets more robust answers by sending more queries but decreasing query times is benefit for performance. However, this work may increase much network traffic overhead.

Alexiou et al. [7] used the probabilistic model checker PRISM to model and analyze the Kaminsky DNS cache-poisoning attacks. They used PRISM to introduce a Continuous Time Markov Chain representation of the Kaminsky attack. Moreover they proposed an approach to perform the required probabilistic model checking. Finally, they demonstrated an increasing attack probability with an increasing number of attempted attacks or increasing rate at which the intruder guesses the source-port ID.

The above solutions solve DNS cache poisoning attacks through DNS servers or DNS clients or DNS protocols. There solutions improve the security of current DNS system and make current DNS system more robust against DNS cache poisoning attacks. We believe more solutions that solve the DNS cache poisoning attacks from different viewpoints will be proposed in the future.

V. CONCLUSION

In this paper, a new defending system against Kaminsky DNS cache poisoning is proposed. To solve DNS cache poisoning attacks, CPS detects, records, and confirms the IP addresses appearing in contents of fake DNS responses. The system not only blocks DNS cache poisoning attacks but also identifies the malicious hosts which may be the members of various botnets. As a result, unlike traditional anti-cache poisoning solutions whose main purpose is to protect a DNS server, CPS can also identify bots that try to attack the related network. CPS is effective in detecting cache poisoning attacks and capable of indirectly protecting other resolvers. Experimental results show that the system has low performance overhead. CPS can accurately block DNS cache poisoning attacks and reveal the related bots.

ACKNOWLEDGMENT

Our work is funded by National Science Committee of Taiwan (ROC), and the number of the Project is NSC 101-2221-E-008-028-MY2.

REFERENCES

[1] D. Kaminsky, "Black Ops 2008–It's the end of the cache as we know it," in *Black Hat USA*, 2008.

[2] A. Hubert and R. Van Mook, "Measures for making DNS more resilient against forged answers," RFC 5452, January 2009.

[3] Alexa Top 500 Global Sites, http://www.alexa.com/topsites, [retrieved: June, 2013]

[4] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," in *IEEE/ACM Transactions on Networking (TON)*, vol. 8, June 2000, pp. 281-293.

[5] A. Kalafut and M. Gupta, "Pollution resilience for DNS resolvers," in *ICC'09. IEEE International Conference on Communications*, June 2009, pp. 1-5.

[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS security introduction and requirements," RFC 4033, March, 2005.

[7] N. Alexiou, S. Basagiannis, P. Katsaros, T. Dashpande, and S. A. Smolka, "Formal analysis of the Kaminsky DNS cache-poisoning attack using probabilistic model checking," in *IEEE 12th International Symposium on High Assurance Systems Engineering*, San Jose, CA, November 2010, pp. 94-103.

[8] H. M. Sun, W. H. Chang, S. Y. Chang, and Y. H. Lin, "DepenDNS: Dependable mechanism against DNS cache poisoning," in *Cryptology and Network Security*, vol. 5888, 2009, pp. 174-188.

[9] C. S. Wang, "Shark: Phishing Information Recycling from Spam Mails," M.S. thesis, Dept. Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan, 2010.

[10] Network Ports Used by DNS, http://technet.microsoft.com/en-us/library/dd197515(v=ws.10).aspx, [retrieved: June, 2013]

 ISBN: 978-1-61208-285-1