Using a Secure Permutational Covert Channel to Detect Local and Wide Area Interposition Attacks

Jaroslaw Paduch
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
jpaduch@jjay.cuny.edu

Jamie Levy
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
jlevy@jjay.cuny.edu

Bilal Khan
Department of Math.& Computer Science
John Jay College, CUNY
New York, 10016
bkhan@jjay.cuny.edu

ABSTRACT

In this paper, we present new techniques to detect interposition attacks on stream-based connections in local and wide area networks. The approach developed here is general enough to apply *uniformly to all circumstances where the man-in-the-middle attacker achieves interposition by corrupting higher-layer to low-layer address mappings.* Thus, both the problem of local area network interposition through ARP poisoning, and the problem wide area interposition through DNS poisoning are addressed as special cases of our work. Like other solutions that reside between Layers 3 and 4 (e.g. IPSEC), our techniques enjoy the property that they *do not* require redesigning legacy software, as is the case for approaches that reside above Layer 4 (e.g. SSL/TLS). Unlike IPSEC, however, the developed system is tailored only to the *detection* of interposition attacks, and thus circumvents the overhead and complexity introduced in guaranteeing stream confidentiality and integrity. We describe the design of the system, demonstrate its efficacy, and provide a publicly accessible prototype implementation.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Protocols–*security*

General Terms

Security, Protocols

Keywords

interposition, DNS, ARP, covert channels

1. BACKGROUND

This paper addresses the problem of interposition attacks in which interposition relies on corrupting higher-layer to low-layer address mappings. We begin by describing the two most important concrete instances of this phenomenon.

IWCMC '09 , June 21-24, 2009, Leipzig, GERMANY

1.1 ARP Poisoning

Address Resolution Protocol (ARP) enables dynamic definition of the map from IP address to the link layer (MAC) address for network devices. ARP cannot be static since machines may need to change their IP addresses as needed; unfortunately, this aspect of the protocol is readily exploitable. When sending packets on the LAN, the destination MAC is resolved from the destination IP using ARP. The sender issues an ARP *Request* providing the IP, and the intended destination replies with it's MAC, which is then cached at the sender for some finite time. A "man-in-the-middle" attack based on ARP requires (1) periodically flooding the network with requests/replies for the victim's IP and the attackers's MAC address in place of the victim's and send them out on the network every few seconds, and (2) enabling packet forwarding on the attacker's machine in order to avoid a "sink hole" and allow the victim to continue new and existing conversations [1, 2]. There are many tools for packet injection [3, 4] so instrumenting this attack is straightforward. Though *ARP poisoning* is considered "old school", it still dangerous today [5]. Such attacks are increasingly prevalent and highly publicized in the media [6, 7].

1.2 Defenses Against ARP Poisoning

Many solutions are proposed in order to detect such attacks. The most direct approach is to have static ARP tables [8], and this is pursued by programs like anti-arp [9]. However opinions differ as to the merits of this strategy [10] particularly with respect to configuration overhead [8]. Another approach, taken by systems like Arpwatch [1, 11], is to monitor for changes in ARP replies, and email notifications of observed changes. However, such email may be intercepted, missed, or the recipient may be in a position where the damage of ARP Poisoning can be quickly mitigated. DHCP snooping is another solution vector and is frequently implemented as a hardware solution. In this approach, the network device keeps track of the MAC address/port bindings of clients when they request an IP address. All attempts to send false ARP requests are then blocked to prevent clients from becoming poisoned [12]. This approach has its weakness as well, since actual MAC addresses themselves can be changed or spoofed to a valid MAC address. ArpOn is another software solution that mirrors like the aforementioned architecture. ArpOn manages all ARP protocol options, and replaces ARP utilities. Since it keeps its own cache of MAC addresses, ArpOn can ignore conflicting MAC addresses [13].

been sent covertly in both directions between the endpoints. On average, this requires the transmission of approximately 1.12 megabytes of overt traffic between the hosts. If there is significant traffic between the hosts (e.g. 10% utilization on a 10Mb/s LAN) the entire verification occurs fairly quickly (approximately 8.9s). Faster link rates yield proportionately faster detection times; regretably, the technique is not as responsive when traffic rates between the hosts is low.

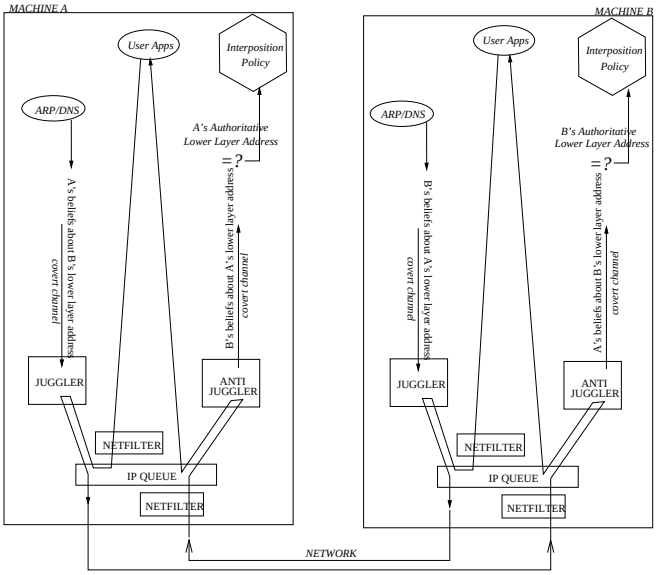

Figure 6: The architecture of PERMEATE-MITM.

5. CONCLUSION AND FUTURE WORK

In this paper, we present new effective techniques to detect interposition attacks on connection streams that apply in any circumstance where the man-in-the-middle attacker achieves interposition by corrupting higher-layer to low-layer address mappings. We demonstrated that the scheme is an effective way to address both ARP and DNS poisoning simultaneously, and does not require redesigning existing software. The proposed system addresses the *detection* of interposition attacks in order to circumvent the full overhead and complexity required for stream confidentiality and integrity. We demonstrated its efficacy, and provide a publicly accessible prototype implementation of the system. Future efforts will involve extending the system to make it more responsive in low traffic situations.

6. REFERENCES

[1] L. Loeb, "On the lookout for dsniff: Part 1." [Online]. Available: http://www.ibm.com/developerworks/library/s-sniff.html

[2] J. Erickson, *Hacking: The Art of Exploitation, No Starch Press, First Edition, 2003.*

[3] P. Biondi, "Scapy." [Online]. Available: http://www.secdev.org/projects/scapy/

[4] M. V. Alberto Ornaghi, "Ettercap." [Online]. Available: http://ettercap.sourceforge.net/

[5] R. Bejtlich, "Old school layer 2 hacking." [Online]. Available: http://taosecurity.blogspot.com/2008/06/old-school-layer-2-hacking.html

[6] H. D. Moore, "Full disclosure: Re: Metasploit - hack ?" [Online]. Available: http://seclists.org/fulldisclosure/2008/Jun/0011.html

[7] ——, "Full disclosure: Re: Metasploit - hack ? (analysis)." [Online]. Available: http://seclists.org/fulldisclosure/2008/Jun/0013.html

[8] D. Song, "Dsniff faq." [Online]. Available: http://monkey.org/~dugsong/dsniff/faq.html#How%20do%20I%20detect%20dsniff%20on%20my%20network

[9] unknown, "anti-arp." [Online]. Available: http://sync-io.net/Sec/anti-arpspoof.aspx

[10] K. Levinson, "comp.security.misc: Re: Defending arp spoofing." [Online]. Available: http://www.derkeiler.com/Newsgroups/comp.os.ms-windows.nt.admin.security/2005-11/0004.html

[11] L. N. R. Group, "Arpwatch." [Online]. Available: http://ee.lbl.gov/

[12] I. Cisco Systems, "Catalyst 4500 series switch cisco ios software configuration guide, 12.1(13)ew." [Online]. Available: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/13ew/configuration/guide/pref.html

[13] A. D. Pasquale, "Arpon." [Online]. Available: http://arpon.sourceforge.net/

[14] R. McMillan, "Dns attack writer a victim of his own creation." [Online]. Available: http://www.pcworld.com/businesscenter/article/149126/dns_attack_writer_a_victim_of_his_own_creation.html

[15] R. Naraine, "Websense reports china netcom dns cache poisoning." [Online]. Available: http://blogs.zdnet.com/security/?p=1776

[16] K. Poulsen, "Comcast hijackers say they warned the company first." [Online]. Available: http://blog.wired.com/27bstroke6/2008/05/comcast-hijacke.html

[17] C. Vulnerabilities and Exposures, "Cve-2008-1447." [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1447

[18] D. Kaminsky, "Black ops 2008: ItâĂŹs the end of the cache as we know it." [Online]. Available: http::://www.doxpara.com/DMK_BO2K8.ppt

[19] R. Bejtlich, "Thoughts on latest kaminsky dns issue." [Online]. Available: http://taosecurity.blogspot.com/2008/07/thoughts-on-latest-kaminski-dns-issue.html

[20] S. Friedl, "An illustrated guide to the kaminsky dns vulnerability." [Online]. Available: http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

[21] H. Moore and I)ruid, "Kaminsky dns cache poisoning exploit for metasploit." [Online]. Available: http://www.caughq.org/exploits/CAU-EX-2008-0002.txt

[22] Microsoft, "Microsoft security bulletin ms08-037." [Online]. Available: http://www.microsoft.com/technet/security/Bulletin/MS08-037.mspx

[23] "Opendns." [Online]. Available: http://www.opendns.com/

[24] V. Jacobson, "Congestion avoidance and control," *Computer Communication Review*, vol. 18, pp. 314–329, Aug. 1988.

[25] J. Levy, J. Paduch, and B. Khan, "Superimposing permutational covert channels onto reliable stream protocols," in *Proceedings of MALWARE 2008, Alexandria VA*, Oct. 2008.