Journal of Information Security, **2010, 1, 88-94**
doi:10.4236/jis.2010.12010 Published Online October 2010 (http://www.SciRP.org/journal/jis)

Denial of Service Due to Direct and Indirect ARP Storm Attacks in LAN Environment*

Sanjeev Kumar, Orifiel Gomez
Department of Electrical/Computer Engineering, University of Texas—PanAm, Edinburg, USA
Email: sjk@utpa.edu, sanjeevk@utpa.edu

Received October 3, 2010; *revised October* 16, 2010; *accepted October* 19, 2010

Abstract

ARP-based Distributed Denial of Service (DDoS) attacks due to ARP-storms can happen in local area networks where many computer systems are infected by worms such as Code Red or by DDoS agents. In ARP attack, the DDoS agents constantly send a barrage of ARP requests to the gateway, or to a victim computer within the same sub-network, and tie up the resource of attacked gateway or host. In this paper, we set to measure the impact of ARP-attack on resource exhaustion of computers in a local area network. Based on attack experiments, we measure the exhaustion of processing and memory resources of a victim computer and also other computers, which are located on the same network as the victim computer. Interestingly enough, it is observed that an ARP-attack not only exhausts resource of the victim computer but also significantly exhausts processing resource of other non-victim computers, which happen to be located on the same local area network as the victim computer.

Keywords: ARP Attack, Computer Network Security, Computer Systems, Direct Attack, Distributed Denial of Service Attacks (DDoS), Indirect Attack, Local Area Networks

1. Introduction

A Distributed Denial of Service (DDoS) attack [1,2] involves multiple DoS agents configured to send attack traffic to a single victim computer. DDoS is a deliberate act that significantly degrades the quality and/or availability of services offered by a computer system by consuming its bandwidth and/or processing time. As a result, legitimate users are unable to have full quality access to a web service or services. A Denial of Service attack consumes a victim's system resource such as network bandwidth, CPU time and memory. This may also include data structures such as open file handles, Transmission Control Blocks (TCBs), process slots etc. Because of packet flooding in a DDoS attack that typically strives to deplete available bandwidth and/or processing resources, the degree of resource depletion depends on the traffic type, volume of the attack traffic, and the processing power of the victim computer.

According to Computer Emergency Response Team Coordination Center (CERT/CC) [3], there has been an increase in use of Multiple Windows-based DDoS agents. There has been a significant shift from Unix to Windows as an actively used host platform for DDoS agents. Furthermore, there has been an increased targeting of Windows end-users and servers. To raise awareness of such vulnerabilities, the CERT/CC published a tech tip entitled "Home Network Security" in July of 2001 [4]. According to the CERT/CC [3], there is a perception that Windows end-users are generally less security conscious, and less likely to be protected against or prepared to respond to attacks compared to professional industrial systems and network administrators. Furthermore, large populations of Windows end-users of an Internet Service Provider are relatively easy to identify and hence the attackers or intruders are leveraging easily identifiable network blocks to selectively target and exploit Windows end -user servers and computer systems.

In this paper, we consider a Distributed Denial of Service (DDoS) attack that can be caused by a barrage of ARP-requests sent to a victim computer. In order to understand the intensity of the attack, we conduct experiments in a controlled lab environment to measure the

*Work of Dr. Kumar is supported in part by funding from CITeC, FRC, FDC, OBRR/NIH, digital-X Inc, and US National Science Foundation.

availability of the processing power and memory resources of the victim computer during an ARP-attack. Since, windows based servers are very commonly deployed, we consider a Window-XP server with a 3.06 GHz Pentium-IV processor and 512 Mbytes of RAM to be used as the victim computer in the ARP-attack experiments. Section II presents a background on ARP and how it is used to exploit vulnerability of a computer system; Section III presents detail on use of ARP requests, ARP format, types of ARP-request traffic, and the processing that needs to be done for ARP-request messages; Section IV presents the experimental-setup, systems configuration for DDoS attacks in the controlled lab environment, and attack measurement results under direct ARP attack traffic and Indirect ARP attack traffic; and Section V provides discussion on detection and prevention schemes for ARP storm attacks, Section VI concludes the paper.

2. Arp-As an Attack Bullet

The Address Resolution Protocol (ARP) requests are legitimate and essential for the operation of the network. However, ARP can be used in more than one way to exploit the vulnerability of a computer system or a network. Some of the security attacks involving ARP can cause Denial of Service (DoS) attack by sending a massive amount of ARP requests to a victim computer and tying up its resource [5]. ARP can also be used to create Denial of Service attack by sending a victim computer's outgoing data to a sink by the technique of ARP cache poisoning. Other ARP based attacks can result in unauthorized sniffing of packets, or hijacking of secured Internet sessions. The Denial of Service attacks due to ARP storms can also be caused by worms such as code red due to their rapid scanning activity [6,7]. The worm initiated ARP storms have been commonly found in networks with high numbers of infected and active computers and servers. In ARP storm, an attacked victim (the gateway or a server) may receive a constant barrage of ARP requests from attacking computers in the same sub-network, and this ties up not only the network bandwidth but also the processing resource of the victim computer.

The worm Code-Red's rapid scanning activity can result in a denial-of-service attack against a Windows NT 4.0 IIS 4.0 server with URL redirection enabled [6]. The worm Code-Red can easily spread to new vulnerable systems, and there is a patch available for this vulnerability. Applying the patch can keep a server from being infected by the worm Code-Red. Nevertheless, it is still possible for the worm in other infected computers on the network to attack the same chain of IP addresses over and over again. This can generate a high-traffic overload due to massive amount of ARP requests generated in the network, which in turn can still affect the server's performance (despite the patch).

In this paper, we investigate the brute force of ARP attack where a constant barrage of ARP requests is directed to a victim computer. In this experiment, we set out to measure how bad the effect of the ARP attack was on the victim computer. Furthermore, we also measure the extent of resource exhaustion due to the ARP attack traffic on other computers located on the same LAN segment as the victim computer. To understand the degree of resource exhaustion, we measure performance in terms of processor exhaustion, occupancy of systems' memory and the page-file size. Since Microsoft Windows-XP based computers and servers with high performance Pentium-IV processors are becoming quite affordable and popular with small businesses, we use a Windows-XP based computer as a victim computer to be stress-tested for the extent of resource exhaustion under the ARP attack.

3. Processing an Arp-Request Message

3.1. Use of ARP-Request Message

A gateway or a host on a local area network uses ARP request broadcast messages [8] for IP and hardware address bindings. The ARP message contains the IP address of the host for which a hardware address needs to be resolved (**Figure 2**). All computers on a network receive ARP message and only the matching computer responds by sending a reply that contains the needed hardware address.

3.2. ARP Message Format

ARP is used for a variety of network technologies. The ARP packet format varies depending on the type of network being used. The ARP packet format used in Ethernet is shown in **Figure 1**. While resolving IP protocol address, the Ethernet hardware uses 28-octet ARP message format [8]. The ARP message format contains fields to hold sender's hardware address and IP address, shown as SENDER-HA and SENDER-IP fields in **Figure 1**. It also has fields for the target computer's hardware and IP address, which is shown as TARGET-HA and TARGET-IP fields in **Figure 1**. When making an ARP request, the sender supplies the target IP address, and leaves the field for the target hardware address empty (which is to be filled by the target computer).

In the broadcasted ARP request message, the sender also supplies its own hardware and IP addresses for the target computer to update its ARP cache table for future correspondence with the sender. Other fields in the ARP packet format in **Figure 2** are HARDWARE TYPE of 2 Bytes (shown as 2B in **Figure 1**), which specifies the type of network being used such as Ethernet in this case. The PROTOCOL TYPE field of 2 Bytes specifies the high-level protocols address used such as the IP addresses. The fields HLEN and PLEN of one Byte each specify the length of hardware address and high-protocol address, in the case of ARP protocol use in the arbitrary networks. The OPERATION field of 2 Bytes specifies if the message is one of the four possible types *i.e.* 1 for ARP-request, 2 for ARP-reply, 3 for RARP-request and 4 for RARP-reply.

4. Types of ARP-Request Traffic on a LAN

A computer on the LAN will receive two different types of ARP-request packets from the network. The first type of ARP request packets can be named as the *direct ARP request traffic* where the IP address in the ARP request packet matches the local IP address of the computer *Pi*. The second type of ARP request traffic that is received by the computer on a LAN can be named as *indirect ARP request traffic* where the IP address in the ARP request packets doesn't match the local IP address of the computer P_i.

In other words, a computer *i* on a LAN with IP address of P_i may receive one of the two possible types of ARP request traffic during an ARP-attack –

a) *Direct ARP traffic* – it is a traffic comprising of ARP request messages with IP address = $\{\chi \mid \chi = P_i\}$

b) *Indirect ARP traffic* – it is a traffic comprising of ARP request messages with IP address = $\{\chi \mid \chi \neq P_i\}$

<table>
<tr><td colspan="2">Harware Type (2b)</td><td>Protocol Type (2b)</td></tr>
<tr><td>Hlen (1b)</td><td>Plen (1b)</td><td>Operation (2b)</td></tr>
<tr><td colspan="3">Sender Ha (Octets 0-3)</td></tr>
<tr><td colspan="2">Sender Ha (Octets 4-5)</td><td>Sender Ip (Octets 0-1)</td></tr>
<tr><td colspan="2">Sender Ip (Octets 2-3)</td><td>Target Ha (Octets 0-1)</td></tr>
<tr><td colspan="3">Target Ha (Octets 2-5)</td></tr>
<tr><td colspan="3">Target Ip (Octets 0-3)</td></tr>
</table>

Figure 1. ARP mssage frmat

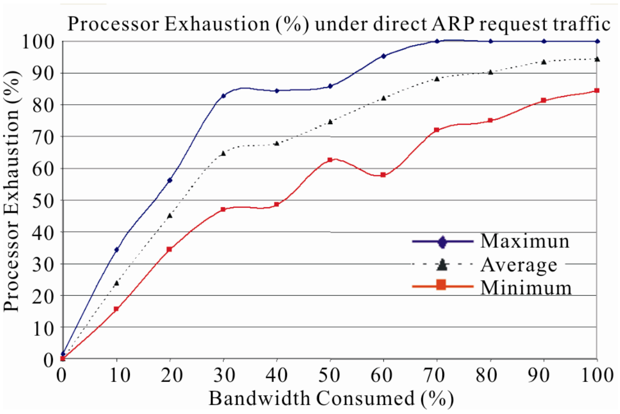

Figure 2. Processor exhaustion under direct ARP- attack traffic with IP address = {χ | χ = P_i}.

The target or victim computer will primarily be inundated with the direct-ARP attack traffic, whereas the other computers (non-victim computers) located on the same LAN segment will be inundated with the indirect ARP-attack traffic.

The main task of the processor in the target computer after receiving the ARP request message is to make sure the ARP request message is for it. In the case of direct ARP frames, the processor proceeds to fill in the missing hardware-address in the ARP request format-header, swaps the target and sender hardware & IP address pair, and changes the ARP-request operation to an ARP-reply. Thus the ARP reply carries the IP and hardware addresses of both, the sender and the target computers. Unlike the ARP request message, the ARP replies are directed to just the sender computer and it is not broadcasted. In the case of indirect ARP frames received, the computer still does some processing to determine if the ARP request message is for the local computer. In this case, once it is determined that the frame is not for the local computer, the indirect ARP message is simply dropped.

The processing needed for an ARP-request message is fairly simple, however there is more processing involved when direct ARP request frames are received by a victim computer, compared to that of the indirect ARP-request frames received by non-victim computers present on the same LAN. Even though, there is comparatively less processing involved when an indirect ARP request message is received, a barrage of such requests can still exhaust the processing power of a non-victim computer just because it happens to be sitting on the same LAN segment as the victim computer or server. The degree of processor exhaustion for a given computer will of course depend on the processor speed and the bandwidth consumed by ARP-request messages. In the following sections, we discuss our experiment to measure the extent of resource exhaustion of two different types of computers on a LAN under an ARP attack – the first type of computer, being the victim computer, which is inundated with direct ARP-request frames. We also measure the computing resource exhaustion of the second type of computers (the non-victim computers, which happen to be on the same LAN as the victim computer or server), when inundated with indirect ARP-request frames.

5. Performance Evaluation

5.1. Experimental Setup

In this experiment, an ARP-storm was generated in a controlled environment of the network security research lab at the UTPA by having different computers send a barrage of ARP-request messages to a victim computer on the same local area network. A Windows-XP based computer was used as the attack target of the ARP-storm. The computer under attack deployed a Pentium IV processor with a speed of 3.06 GHz with 533 MHz Bus, 512 kb Cache, a physical memory of 512 Mbytes (RAM), and a NIC card from 3 Com. Furthermore, other computers (that received indirect ARP request traffic) on the LAN deployed exactly the same resources as the victim computer on the LAN. Computers under attack on the LAN deployed Windows-XP Service-Pack 2 (SPK 2). We also used the network observer software to collect traffic detail and the applied load on the LAN.

This experimental setup and results obtained in this paper are much more detailed compared to the one presented in [9] where a different system was used for the victim computer, which deployed a Pentium-4 processor with a speed of 2.66 GHz. Furthermore, the NIC card used in [9] were the Intel's NIC card, which could not support full speed of 100 Mbps of network traffic. Whereas, in this experiment, the 3 Com's NIC card was used that supported full speed of 100 Mbps. Furthermore, in [9] only the effect of direct ARP traffic was measured and no indirect ARP traffic was considered.

5.2. Attack Measurements

Parameters of performance evaluation considered for this attack experiment were the applied load of the ARP-attack traffic, processor exhaustion during the attack and memory occupied while processing the attack traffic by the target computer. The DDoS attack was simulated as ARP packets coming from multiple different attacking-computers at a maximum aggregate speed of 100 Mbps towards the target server. The attack traffic (while simulating ARP storm) load was started with 0 Mbps (the background condition) and was increased by 10 Mbps *i.e.* from 0% load to 100% load (= 100 Mbps). In the ARP-storm experiment, the attacked target computer continued to receive a barrage of ARP-requests for a period of 60 minutes for a given load, and was obligated to process them by creating an ARP-reply. In this experiment, a total of 10 different loads were generated, *i.e.* 10% - 100%. A total of 10 hours of ARP attack traffic were experienced at the victim computer and another non-victim computer on the local network. The CPU time is termed as processor exhaustion in these measurements, which gives an indication of the rate of processor exhaustion for a given bandwidth consumed by the attack-traffic during the ARP storm. It is observed that as the network bandwidth is increasingly consumed by the ARP-attack traffic, the processor is exhausted at a much faster rate, and hence this type of attack can be classified under computing-resource starvation attack.

5.3. Resource Exhaustion of the Victim Computer Due to Direct-ARP Request Traffic

Direct ARP request traffic comprises of ARP-request frames that have

$$\text{IP address} = \{\chi \mid \chi = P_i\}$$

In this experiment, we measure, processor exhaustion, memory used and the page file size under direct-ARP request traffic. Page file size gives indication of virtual memory activity, if any, during the attack.

Figure 2 shows minimum and maximum CPU time observed (called processor exhaustion in the attack experiments) for a given load of the direct ARP-attack traffic. Average CPU time is also shown in the graph so that we can get an idea if the majority of observations are closer to the maximum CPU time or closer to the minimum CPU time. It can be seen that a bandwidth consumption of 40% by direct ARP-attack traffic in a fast Ethernet environment exhausts a Pentium-IV processor to up to 85% of its 3.06 GHz processing capacity. Due to the processing of a barrage of ARP-requests the CPU resource is easily consumed and this in turn can degrade the quality and availability of associated web services.

Furthermore, it is obvious that if such servers are operated in a Gigabit network deploying higher interfaces such as 1 Gbps then it will be easier for such CPU of 3.06 GHz to be completely consumed by the Gigabit-flood of ARP-attack traffic, and attacks in such Gigabit environment can completely stall the system. Complete stalling of system means that one cannot even move the cursor on the attacked computer, let alone running the security diagnostics. It is also obvious from this experiment that a lower capacity (< 3.06 GHz) processor can easily be frozen (consumed 100%) by this type of ARP-storm in commonly available fast Ethernet environment of local area networks.

Figure 3 shows the memory-usage of the victim computer under direct ARP-attack traffic, as the network bandwidth is increasingly consumed by the ARP-storm. The memory consumed due to direct ARP attack traffic is observed to be within a range of 6 Mbytes, which seems to be not much of an issue for a 3.06 GHz processor with 512 Mbytes of RAM. However, for a slower

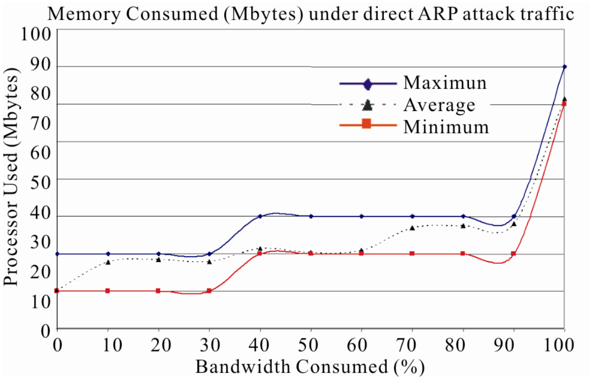

Figure 3. Memory usage under the direct ARP-attack traffic with IP address = {χ | χ = P_i}.

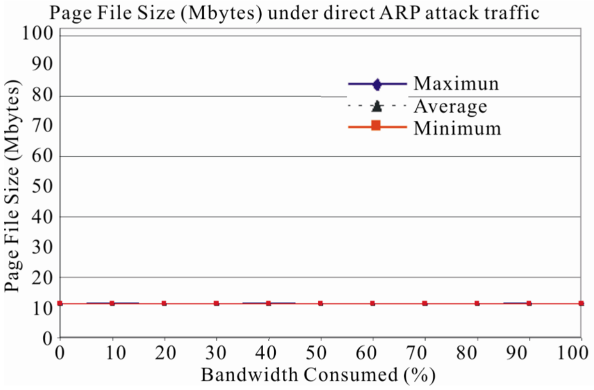

Figure 4. Page-file size is visibly unaffected under direct-ARP-attack traffic with IP address = {χ | χ = P_i}.

processor with processing power less than 3.06 GHz, a greater amount of computer's memory resource can be wasted. Slower processing power in the fast Ethernet environment can cause the queue of ARP packets to build up waiting for address resolution and computer's response. Hence a slower processor will exhaust a relatively greater amount of memory resource of the victim computer under ARP storm. In any case, the memory usage is so insignificant that it is not really a problem in these ARP attacks.

Another parameter of interest is the Page file size. Page File size is the current number of bytes that the active processes have used in the paging file(s). We measure the page file size during the attack to observe for activities in the virtual memory.

Figure 4 shows that there is no change in the page-file size before and during the direct ARP attack. Page-file size measurement at 0% load mainly provides the size due to the background processes running in the computer in the absence of any ARP request traffic. Furthermore, as the load of incoming direct ARP traffic is increased, there is really no impact on the virtual memory of the computer.

5.4. Resource Exhaustion of a (Non-Victim) Computer Receiving Indirect Frames

If i^{th} computer in the broadcast domain has an IP address of P_i then the indirect ARP-request frames arriving to the computer can be described as the frames with

$$\text{IP addresses} = \{\chi \mid \chi \neq P_i\}$$

Figure 5 shows minimum, maximum and average value for the processor exhaustion for a given load of the indirect ARP-attack traffic. It can be seen that a bandwidth consumption of 40% by indirect ARP-attack traffic in a fast Ethernet environment exhausts a Pentium-IV based non-victim computer to up to 55% of its 3.06 GHz processing capacity. Indirect ARP requests are still being processed by the computers on the network even though they are not directed towards them. Due to the processing of a barrage of indirect ARP-request messages, the CPU resource is still getting significantly consumed, however the processor exhaustion rate is relatively less intense compared to the one under direct ARP attack traffic. This is understandable as there is relatively more processing involved in direct ARP attack traffic compared to that of indirect ARP attack traffic.

Figure 6 shows the memory-usage of a non-victim computer, which is located on the same LAN segment as that of the victim computer, as the network bandwidth is increasingly consumed by the indirect ARP attack traffic. The memory consumed due to such indirect-ARP attack traffic is observed to be within 3 Mbytes, which is comparatively less than that consumed by the direct ARP attack traffic in **Figure 3.** Consumption of physical memory in the range of 3 Mbytes is not much of an issue for a 3.06 Hz computer with 512 Mbytes of RAM.

In this experiment, we also measure the page-file size before the onset of indirect ARP attack, and during the indirect ARP attack (**Figure 7**). The page-file size at 0% ARP traffic indicates the page-file size before the onset

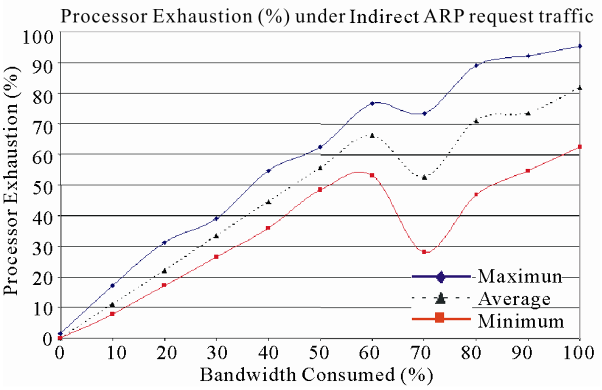

Figure 5. Processor exhaustion under the indirect ARP-attack traffic with IP address = {χ | χ ≠ P_i}.

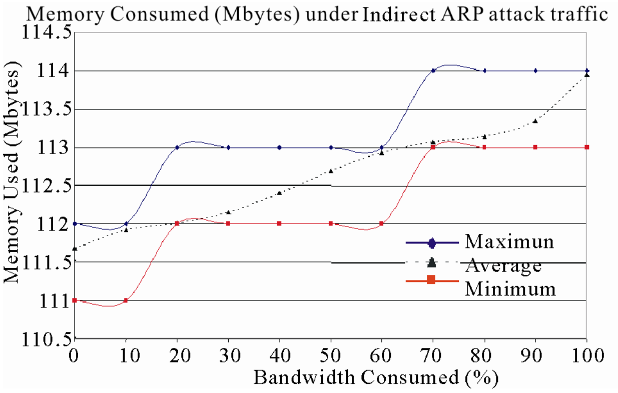

Figure 6. Occupancy of the computer's memory under indirect ARP-attack traffic with IP address = {χ | χ ≠ *P*$_i$}.

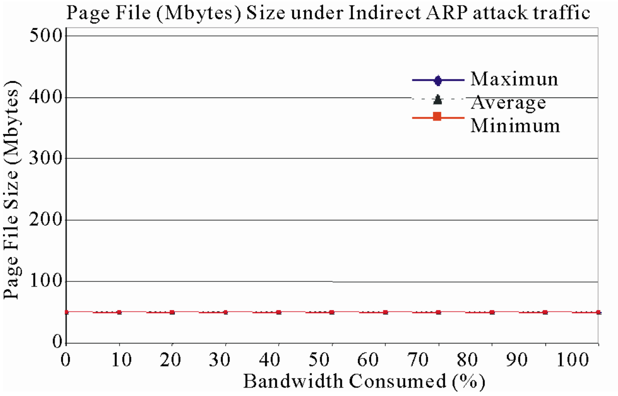

Figure 7. Page-file size is visibly unaffected under the indirect ARP-attack traffic with IP address = {χ | χ ≠ *P*$_i$}.

of the ARP-attack, which is mainly due to the processes running in the background. The page-file size of the victim computer is measured as the network bandwidth is increasingly consumed by the indirect ARP-attack traffic. **Figure 7** shows that the page-file size is not affected by the indirect ARP attack traffic, as it stays the same before and after the onset of the indirect ARP attack. This is obviously due to the fact that the memory-consumed by the indirect ARP traffic (**Figure 6**) is quite minimal, and stays within the range of 3 Mbytes (out of a total 512 Mbytes) of RAM space, and hence no incoming ARP messages spill to the page-file.

6. Detection and Prevention

It can be seen from the prior experiments that the ARP storms can consume the computing resources rapidly for all the computers on the affected LAN segment. Hence it is important to detect the ARP storms immediately and raise alarm for its possible prevention before the entire LAN segment is brought down by such ARP storms. In order to detect these types of ARP attacks, it is important to monitor the ARP traffic on each LAN segments. Programs such as ARPwatch [10] can be used to monitor ARP traffic on each LAN segments and raise alarm when ARP storms or ARP poisoning tools are detected. One can also use SNMP to monitor changes in ARP table in routers and switches to raise alarm for onset of such ARP attacks.

One way to prevent ARP storm is to involve layer-2 switches in controlling the ARP broadcast floods at the source where the storm starts building up. This can be achieved by allowing for threshold limits for broadcast/multicast traffic on a per-port basis. Furthermore, these thresholds per-port basis should be set up by limiting the bandwidth consumed by ARP broadcasts on a switch port.

In order to support multiple layers of prevention, the routers can also be used in controlling ARP storm from spreading to others LAN segments. A network manager can configure the router (using its control policy) to impose a limit on the rate of ARP requests that can be allowed for the associated LAN segments. When the imposed threshold for the ARP requests is exceeded then the ARP request packets are dropped by the router. The router hardware should be fast enough to examine and drop the ARP request packets that exceed the imposed threshold, otherwise it is possible for the router to crash or experience slowdown of its operation and itself become a bottleneck resulting in eventual denial of service (DoS).

7. Conclusions

According to Computer Emergency Response Team (CERT/CC), there has been an increased targeting of Windows end-users' computer systems and servers for security attacks. Distributed Denial of Service (DDoS) attacks due to ARP-storms can be found in local area networks where many computer systems are infected by worms such as Code Red or by DDoS agents. In this paper, we present results of our experiments to measure the impact of ARP-storms on systems resource exhaustion of a Window-XP based computer system deploying a high performance Pentium-IV processor. It is observed that ARP-storms not only waste the communication bandwidth but also exhaust a processor's resource of a victim computer even more rapidly by forcing it to reply to a barrage of ARP-request messages. It is also observed that when the network bandwidth is consumed 40% by the ARP-attack traffic in a fast Ethernet environment, a computer system with a high-performance Pentium-IV processor of 3.06 GHz speed wastes up to 85% of its (victim computer) raw CPU-time in processing direct

ARP attack traffic and 55% of its (non-victim computers) raw CPU-time in processing indirect ARP attack traffic. This attack is found to be more processor intensive which means that it exhausts processor resource more rapidly than other computing resources such as memory. The memory exhaustion is found to be not significant when compared with the corresponding processor exhaustion. Memory usage is observed to be quite insignificant compared to the memory resource deployed in the system. The virtual memory or the page file of the victim computer is observed to be completely unaffected. Based on these experimental results, the ARP-attack can be categorized as the attack that causes computing resource starvation more rapidly than the bandwidth starvation, especially that of the processor of the victim and non-victim computer systems on the affected network. It is interesting to notice the collateral damage done by this attack on a given LAN, according to which it not only exhausts the resource of the victim computer but also exhausts computing resource of other non-victim computers present on a given LAN where the victim computer resides. The rate of resource exhaustion in this type of experiment can help network security engineers design efficient flow-control and threshold based attack prevention schemes at the switches and routers used in the LAN.

8. Acknowledgements

The authors would like to thank Uriel Ramirez and Sumanth Avirneni for equipment support, data collection and verification efforts in the Network Security Research Lab (NSRL) at UTPA. The work in this paper is supported in part by funding from US National Science Foundation under grant # 0521585.

9. References

[1] L. Gerber, "Denial of Service Attacks Rip the Internet," *IEEE Computer*, April 2000.

[2] P. G. Neumann, "Denial-of-Service Attacks," *ACM Communications*, Vol. 43. No. 4, April 2000, p. 136.

[3] K. J. Houle and G. M. Weaver, "Trends in Denial of Service Attack Technology," Computer Emergency Response Team (CERT)® Coordination Center, V1.0, October 2001.

[4] Computer Emergency Response Team (CERT)® Advisory, "Home Network Security," CA-2001-20. http://www.cert.org/tech_tips/home_networks.html

[5] A. Householder, A. Manion, L. Pesante and G. M. Weaver, "Managing the Threat of Denial-of-Service Attacks," CERT Coordination Center, October 2001.

[6] CERT® Incident Note IN-2001-10, "Code-Red Worm Crashes IIS 4.0 Servers with URL Redirection Enabled," CERT Coordination Center, August 2001. http://www.cert.org/incident_notes/IN-2001-10.html

[7] Cisco Security Advisory, "Code-Red Worm—Customer Impact," Cisco Networks, July 2001. http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml

[8] D. C. Plummer, "Ethernet Address Resolution Protocol," IETF Network Working Group, RFC-826, November 1982.

[9] S. Kumar, "Impact of a Distributed Denial of Service (DDoS) Attack Due to ARP Storm," *International Conference on Networking*, to be published in *Lecture Notes in Computer Science (LNCS)*, April 2005.

[10] ARPwatch. http://en.wikipedia.org/wiki/Arpwatch