Microsoft vs. Apple: Resilience against Distributed Denial-of-Service Attacks

Sanjeev Kumar and Sirisha Surisetty | University of Texas–Pan American

Both Windows 7 and Snow Leopard claim to provide their users with safer and more reliable systems, but no work has evaluated and compared their resilience against common distributed denial-of-service attack traffic.

During a distributed denial-of-service (DDoS) attack, network-connected personal computers are made to attack other computers via the Internet. A victim computer under DDoS attack exhausts its computing resources as it is made to process a huge amount of DDoS traffic—ultimately, the victim computer either slows down or crashes. Different DDoS attacks exhaust resources in different ways, but most of them target processor, memory, and bandwidth resources. Most cyberdefense strategies involve firewalls and intrusion prevention systems (IPSs), but the attacks launched on 4 July 2009 on US and South Korean government websites demonstrated that these approaches don't always work (www.zdnet.com/blog/government/us-s-korean-websites-under-attack-n-korea-blamed/5093).[1] Another much publicized DDoS attack on Sony's Playstation network in April 2011 showed that additional firewalls were needed to combat cyberattacks (www.infosecisland.com/blogview/13558-Sony-Tells-Congress-Anonymous-DDoS-Aided-Breach.html). In fact, in these particular instances, the attacked victim computers and networks continued to experience problems several days after the cyberattack's initial launch.

Today's operating systems increasingly deploy built-in security features to provide resilience against cyberattacks, with the aim of preventing host computers from crashing when under attack and from attacking other remote computers. Apple's iMac range was originally promoted as the most secure, safe, and virus-free computer (www.apple.com/getamac/whymac) on the market. In the late 2000s, at roughly the same time Apple unveiled Snow Leopard, Microsoft released Windows 7, claiming that it had come a long way in providing its users with a safer, more reliable, and more responsive operating system (www.microsoft.com/windows/windows-7/compare/top-ten-reasons.aspx). However, no experimental work to date has evaluated the resilience of these popular operating systems, which are deployed on millions of personal computers, against the harmful DDoS traffic used in cyberattacks. We conducted experiments to test resilience of both Windows 7 and Snow Leopard in this type of scenario on the same iMac hardware platform under the same attack conditions. In this article, we describe how both operating systems fared against three common attacks on three different layers of the TCP/IP protocol stack.

Experimental Setup

Our overall experiment simulated a network condition in which multiple computers sent a barrage of DDoS attack traffic to a remote victim computer at a maximum speed of 1,000 Mbps (1 Gbps). Our victim computer was an Apple iMac with 2 Gbytes of RAM, an Intel Core2 Duo 2.4-GHz processor, and both the Apple OS X 10.6.3 Snow Leopard and Microsoft Windows 7 (professional version) operating systems were available for installation on it.

Our performance evaluation parameters for attack resilience were processor exhaustion and wired/nonpaged pool allocations in main memory. We measured

1540-7993/12/$31.00 © 2012 IEEE

processor exhaustion as CPU utilization of a victim computer under DDoS attack traffic. Complete processor exhaustion must be avoided to prevent a victim computer from crashing under a DDoS attack. Wired pages in main memory can't be paged out because they're required for execution of specific kernel tasks in Snow Leopard (http://developer.apple.com/mac/library/documentation/Darwin/Reference/ManPages/man1/vm_stat.1.html); they're similar to the nonpaged allocations in Windows 7 (http://technet.microsoft.com/en-us/library/cc778082(WS.10).aspx). The system under attack logged its performance metric values by using some of its own system activity commands.

Firewalls control the connections made to host computers from other computers on a network. The ICMP pings commonly used for diagnostic purposes can also be used by hackers to attack computers. Snow Leopard's built-in firewall controls can be set to block incoming ICMP pings by enabling "stealth mode" under the advanced settings (http://blink.ucsd.edu/technology/security/firewall/mac-snow.html#2.-Activate-the-firewall). Microsoft claims that the Windows 7 firewall also has controls that are more advanced and users can set the firewall to either block or allow incoming traffic (http://windows.microsoft.com/en-US/windows7/products/features/windows-firewall). In our experiments, we used the default firewall settings in both systems to block incoming ICMP traffic.

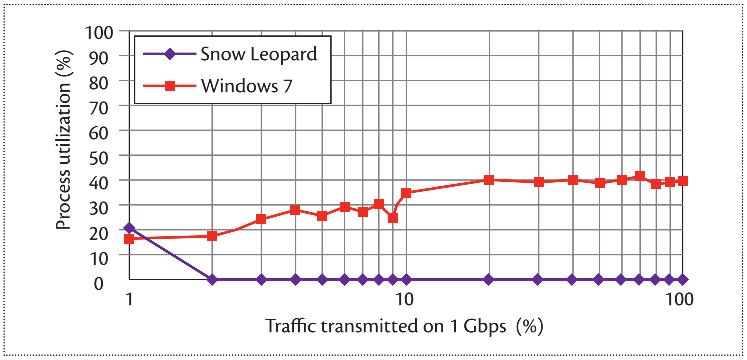
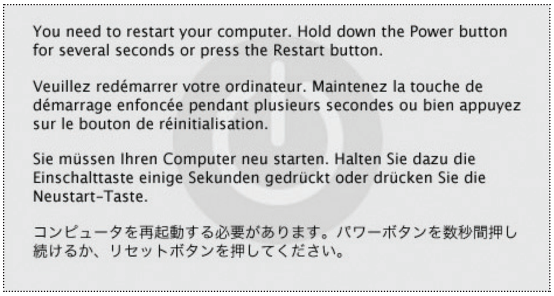


Figure 1. Processor utilization (on a logarithmic scale). Our evaluation of how Windows 7 and Snow Leopard can handle an ARP flood attack shows zero processor utilization for Snow Leopard after it crashed.



You need to restart your computer. Hold down the Power button for several seconds or press the Restart button.

Veuillez redémarrer votre ordinateur. Maintenez la touche de démarrage enfoncée pendant plusieurs secondes ou bien appuyez sur le bouton de réinitialisation.

Sie müssen Ihren Computer neu starten. Halten Sie dazu die Einschalttaste einige Sekunden gedrückt oder drücken Sie die Neustart-Taste.

コンピュータを再起動する必要があります。パワーボタンを数秒間押し続けるか、リセットボタンを押してください。

Figure 2. A detailed message appeared on the crashed iMac soon after the ARP flood attack hit Snow Leopard.

Evaluation

To evaluate overall resilience to DDoS attack traffic, we performed our evaluation on Snow Leopard and Windows 7 and measured their performance parameters. To evaluate overall resilience to DDoS attack traffic, we focused on how both operating systems performed against Address Resolution Protocol (ARP) flood (layer 2), ICMP-based ping flood (layer 3), and TCP-SYN flood (layer 4) attacks.

ARP Flood Attack

ARP is used in local area networks (LANs) to resolve IP addresses as hardware MAC addresses. Both gateways and hosts use this very basic and essential protocol to communicate in a LAN environment. The ARP request message consists of a host's IP address, and the IP and hardware MAC address of the initiator who wishes to communicate. All hosts in the LAN receive the ARP request, but only the host that has that particular IP address will respond and unicast the initiator its hardware MAC address. Upon receiving an ARP request, the system then updates its ARP cache table with the corresponding IP-MAC addresses for further communication with the initiator.[2] Attackers take advantage of this protocol and try to flood a victim computer with multiple ARP requests; as the victim computer strives to reply to the requests and update its cache table, it ultimately exhausts its computing resources. With several floods of such requests, resource starvation—in the form of either processor or memory consumption—typically worsens as the load increases for the host.

For our evaluation, we sent an ARP flood with a range from 10 Mbps to 100 Mbps in steps of 10 Mbps and again from 100 Mbps to 1 Gbps in steps of 100 Mbps over a gigabit Ethernet medium. We observed each load for 10 minutes and had the iMac reply to all ARP requests it received. Figure 1 shows processor utilization under different ARP flood attack loads. Windows 7 had a maximum CPU utilization of 40 percent, indicating its ability to provide its services to other tasks even at this level of attack traffic. We repeated the same experiment with Snow Leopard (installed on the victim iMac computer), but after just a few minutes and an initial attack load of 10 Mbps, the iMac crashed, requiring a forced reboot. Figure 2 shows the message that appeared on the screen shortly

```
1. Interval Since Last Panic Report: 938022 sec
2. Panics Since Last Report:  10
3. Anonymous UUID A8ECE62E-35DD-45FE-A5D9-AFEBD5205828 :Wed
   Nov 25 17:28:02 2009
4. panic(cpu 0 caller 0x234059): "zalloc: \"kalloc.128\" (1440640 elements)
   retry fail 3, kfree_nop_count: 0"@/SourceCache/xnu/xnu-1456.1.25/os-
   fmk/kern/zalloc.c:981
5. Backtrace (CPU 0), Frame : Return Address (4 potential args on stack)
6. 0x2fefb988 : 0x21acfa (0x5ce650 0x2fefb9bc 0x223156 0x0)
7. 0x2fefb9d8 : 0x234059 (0x5874f8 0x585efc 0x15fb80 0x3)
8. 0x2fefba78 : 0x21fe15 (0x2845bf0 0x1 0x2fefbac8 0x2ac94d)
:
:
BSD process name corresponding to current thread: kernel_task
Mac OS version: 10A432
Kernel version: Darwin Kernel Version 10.0.0: Fri Jul 31 22:47:34 PDT 2009;
root:xnu-1456.1.25~1/RELEASE_I386
System model name: iMac8, 1 (Mac-F226BEC8)
```

Figure 3. CPU error log. After the Snow Leopard forced reboot, the iMac showed the reason for the CPU panic.

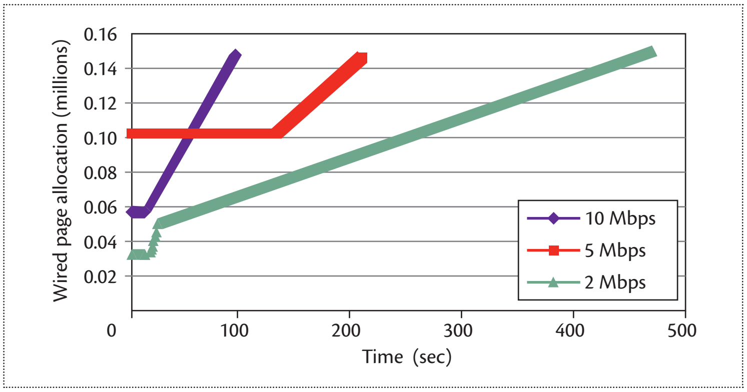


Figure 4. Wired pages allocations in Snow Leopard. With ARP flood traffic loads at 10 Mbps, 5 Mbps, and 2 Mbps, we see that the wired page allocations increased rapidly until they reached 0.15 million.

thereafter; Figure 3 shows the error log that appeared after the forced reboot.

Lines 4 and 5 of the CPU panic log in Figure 3 indicate that a "zalloc" retry failure caused the panic. Zallocs are *zone allocators* that provide an efficient interface for managing dynamically sized collections of similarly sized items (a zone is an extensible collection of items of identical size; www.gsp.com/cgi-bin/man.cgi?section=9&topic=uma_zalloc). The zone allocator works with runtime-allocated as well as preallocated zones, keeps track of which items are in use (and which are not), and provides functions for allocating items from the zone and bringing them back (thereby making them available for later use).[3]

The appearance of the word "zalloc" also points to a problem that might be related to memory allocations. Because no other application processes were running on the host computer, the ARP flood clearly created the wired pages, causing the iMac to eventually crash. We reduced the ARP attack load to 5 Mbps and 2 Mbps and observed that the same panic occurred but with longer delays. Figure 4 shows the wired page allocations in Snow Leopard for lower loads of ARP flood traffic—namely, 10 Mbps, 5 Mbps, and 2 Mbps. Figure 5 shows the average processor utilization for these loads, respectively.

As Figure 4 shows, Snow Leopard took nearly 8 minutes to crash under an ARP attack load of 2 Mbps, 3.5 minutes to crash under ARP attack load of 5 Mbps, and 1.5 minutes to crash under an ARP attack load of 10 Mbps. Figure 5 also shows that processor utilization before the CPU panic was very small—21 percent for 10 Mbps of attack traffic, 10 percent for 5 Mbps, and 5 percent for 2 Mbps—indicating that the panic didn't happen because of excessive processor exhaustion. In addition, measurements of other parameters show that the CPU panic occurred only after the wired page allocations reached nearly 0.15 million, implying that there might have been a limit on wired page allocations per process, which we exceeded in the Snow Leopard test. When we tested Windows 7 on the same iMac configuration, we didn't notice these problems, and the system didn't crash under ARP flood attack up to the tested load of 1 Gbps (see Figure 1).

So why did this happen on one system and not the other? One explanation for the exponential growth in wired page allocations in Snow Leopard is that the zalloc entries aren't able to free up, most likely due to a software flaw in the Snow Leopard operating system. After a certain preallocated reserved portion of the main memory filled up with these entries, an exception in the kernel task happened, and the CPU went into panic mode, displaying the sort of message shown in Figure 3. As Figure 6 shows, the nonpaged pool allocations in Windows 7 were bounded within 0.12 million paged allocations—that is, they didn't grow unboundedly and didn't reach the critical limit of 0.15 million paged allocations as observed in Snow Leopard prior to its crash (see Figures 4 and 5).

Ping Flood Attack

A ping is a type of diagnostic ICMP message used to determine the availability of another computer on a network. Based on RFC 0792, when a networked computer receives an ICMP echo request, it must respond with an ICMP echo reply. Attackers exploit this protocol and flood victim computers with ping requests to force that response, which ultimately consumes the victim computer's resources. An earlier work shows that a simple ping attack can keep a target host busy processing ping requests to the point where it consumes 100 percent of the CPU's utilization.[4]

Figure 7 shows processor exhaustion on the iMac for both Windows 7 and Snow Leopard during a ping flood attack. For Windows 7, the maximum processor exhaustion caused by ping flood attack traffic was approximately 30 percent; for the same attack on Snow Leopard, maximum processor exhaustion reached up to 98 percent.

Even with a lower attack traffic load of 10 Mbps, the processor utilization under a ping flood attack on Snow Leopard was nearly 75 percent, indicating that an iMac computer running Snow Leopard can be bogged down significantly even with a low load of ping flood attack traffic.

TCP-SYN Flood Attack

Layer-4 TCP uses a three-way handshake process for connection establishment prior to data transfer.[5] Under a TCP-SYN flood attack, the attacker attempts multiple TCP connections by sending a flood of TCP-SYN packets to the victim computer, forcing it to create a large number of half-open connections, which can consume considerable memory as well as processor resources. Operating systems have improved over the years in their efforts to protect their host computers against such attacks. Microsoft's XP service packs in particular mitigate this type of TCP-SYN-based DDoS attack by controlling the rate of half-open connections.[5]

In our experiment with Windows 7 and Snow Leopard, we used different TCP-SYN flood attack loads up to 1 Gbps. Figure 8 shows the systems' comparative performance (on a logarithmic scale) for processor exhaustion. For Windows 7, the maximum processor utilization under a TCP-SYN flood attack was 45 percent; for Snow Leopard, it was 98 percent. Just as for the ping flood attack, even with lower loads of TCP-SYN traffic (10 Mbps), the iMac's processor exhaustion while running its own operating system (Snow Leopard) was still quite high (nearly 80 percent).

Based on our experiments, we can conclude that Microsoft's Windows 7 operating system appears to be more capable of limiting adverse effects of DDoS flood attacks when compared to Apple's Snow

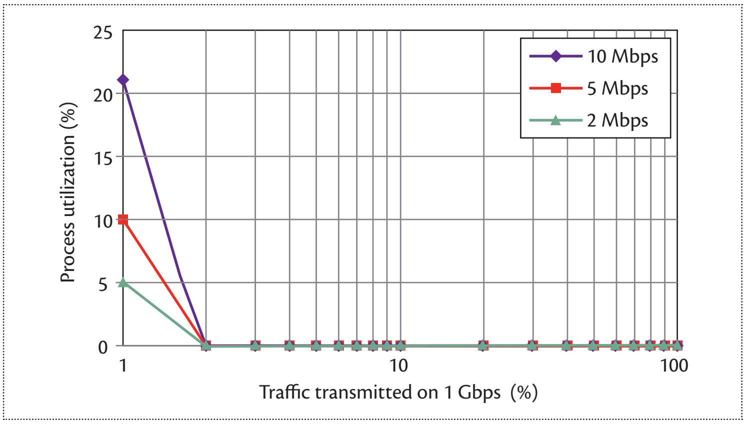


Figure 5. Processor utilization in Snow Leopard. With ARP flood traffic loads at 10 Mbps, 5 Mbps, and 2 Mbps, we see processor utilization become zero after the iMac loaded with Snow Leopard crashed.

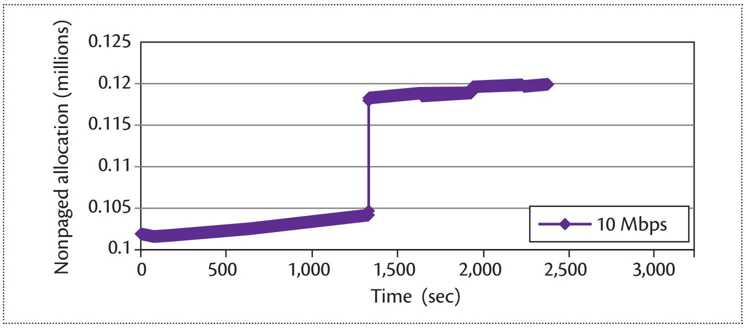


Figure 6. Wired page allocations in Windows 7. For the entire attack period, from 10 Mbps to 1 Gbps, the wired page allocations did not grow unboundedly and stayed within a limit throughout the attack period; this appeared to have prevented crashing of the host victim computer running Windows 7 OS.

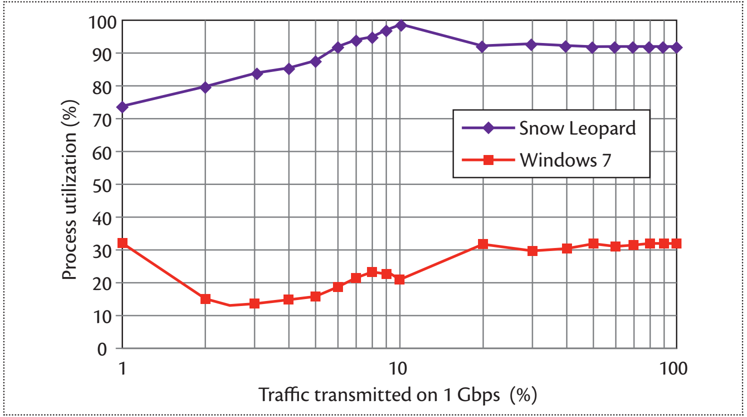


Figure 7. Processor utilization (on a logarithmic scale). With firewalls blocking ICMP packets on both Windows 7 and Snow Leopard, a ping flood attack exhausted up to 30 percent of processor utilization under Windows 7 and up to 98 percent under Snow Leopard.

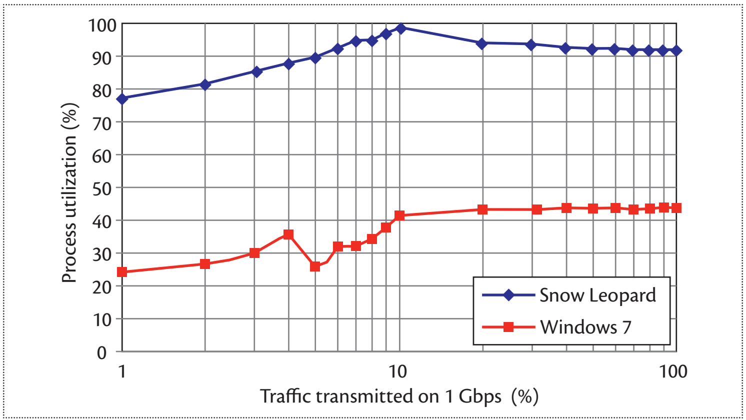


Figure 8. Processor utilization (on a logarithmic scale). Under a TCP-SYN attack, Windows 7 and Snow Leopard responded with different processor exhaustion. Windows 7 consumed fewer processor cycles under TCP-SYN attack, whereas Snow Leopard consumed almost all of its processor resources under the same attack conditions.

Leopard. This is quite remarkable, given that we ran our tests in an Apple environment (iMac), and it directly contradicts Apple's advertised superb security aspects. ■

Acknowledgments

Funding from the US National Science Foundation, grant number 0521585, partially supported this work.

References

1. S. Kumar and R. Gade, "Experimental Evaluation of Juniper Network's Netscreen-5GT Security Device against Layer4 Flood Attacks," *J. Information Security*, vol. 2, no. 1, 2011, pp. 50–58.
2. S. Kumar and O. Gomez, "Denial of Service Due to Direct and Indirect ARP Storm Attacks in LAN Environment," *J. Information Security*, vol. 1, no. 2, 2010, pp. 88–94.
3. S. Surisetty and S. Kumar, "Is Apple's iMac Leopard Operating System Secure under ARP-Based Flooding Attacks?," *Proc. 5th Int'l Conf. Internet Monitoring and Protection*, IEEE, 2010, pp. 60–64.
4. S. Kumar, "PING Attack: How Bad Is It?" *Computers & Security J.*, vol. 25, July 2006, pp. 332–337.
5. S. Kumar and E. Petana, "Mitigation of TCP-SYN Attacks with Microsoft's Windows XP Service Pack2 (SP2) Software," *Proc. 7th Int'l Conf. Networking*, IEEE, 2008, pp. 238–242.

Sanjeev Kumar is a professor in the Department of Electrical and Computer Engineering at the University of Texas–Pan American. His research interests are computer network security, high-speed Internet switching/routing, wireless ad hoc networks, computer architecture, and digital logic. Kumar has a PhD in computer engineering from North Carolina State University. He's an associate editor of the *Journal of Security and Communication Networks* and a senior member of IEEE. Contact him at sjk@utpa.edu.

Sirisha Surisetty is currently working as a software engineer in Dallas, Texas. Her research interests are computer networks and digital logic design. Surisetty has an MS in electrical engineering from the University of Texas–Pan American. Contact her at ssurisetty@broncs.utpa.edu.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*