Implementing Wake-on-LAN in Institutional Networks

Patrick Luberus
Abilene Christian University

Alfandika Nyandoro
Abilene Christian University

Wake-on-LAN (WoL) is a widely supported networking standard that allows computers to be turned on remotely. The majority of desktop hardware is WoL capable, though many organizations still do not employ this technology (despite its benefits in energy conservation). Limitations include specific configuration requirements for sending the WoL packets over multiple hops, as well as security issues that arise due to these settings. In this paper, we present a testbed that addresses these issues, and others. Our testbed utilizes standard equipment and freely available software tools, allowing network administrators and other interested parties to adapt the solution to their scenarios.

INTRODUCTION

Wake-on-LAN (WoL) is a widely supported networking standard which allows computers to be remotely woken up or turned on. Properly implemented, it is useful for administrative tasks (e.g. patch deployment or system maintenance) and for decreasing overall energy consumption. Since it allows administrators to turn on computers remotely, turning computers off when they are not in use has no adverse side effects on system administration. Being able to turn on computers remotely can be particularly useful for universities or businesses with large networks, where physically turning on all of the computers in a reasonable amount of time may not be feasible. In many cases, organizations do not turn off their computers at all, even though doing so could potentially cut their power usage and associated costs in half (e.g. if computers are rarely accessed overnight or during holidays) (Webber C et. al., 2006).

Though WoL is platform independent, it requires specific hardware and software configurations in order to work properly. Due to early alliance initiatives, WoL is supported by the majority of desktop hardware and by most major operating systems (including Microsoft Windows, Mac OS X, and Linux) ("Wake-on-LAN Explained," 2011). However, despite this support and the benefits WoL can bring, many businesses and universities still do not appear to make it an integral part of their information technology strategies. This may be attributed to some of WoL's limitations, including specific configuration requirements for sending magic packets over multiple hops (i.e. across subnets), as well as security issues that arise due to certain WoL settings.

In this paper, we outline the problems associated with WoL, and share our testbed findings and analysis in order to address these challenges. The testbed uses common enterprise equipment, as well as

freely available software tools, allowing interested parties to easily and inexpensively adapt the solution to their specific scenarios.

WAKE-ON-LAN OVERVIEW

Background

In 1996, Intel and IBM formed the Advanced Manageability Alliance (AMA) with the goal of developing non-proprietary, standards-based tools, in order to simplify PC manageability (O'Malley T., 1998). The following year, this alliance introduced WoL, which allowed computers to be turned on remotely via a special network message known as a magic packet ("Wake on LAN Technology," 2006). In 1998, the Intel based initiative Wired for Management introduced a new specification to help reduce the total cost of ownership of business computers. As a part of this initiative, baseline systems were required to have the ability to be woken up remotely by one of three remote wakeup techniques: magic packet, packet filtering, or wake-on-ring. This initiative, among others, provided additional support for WoL, but has since been replaced by the Intelligent Platform Management Interface (IPMI) and Intel Active Management Technology (AMT) standards.

On a computer that is properly configured for WoL, even if a system is completely powered down, the network interface card (NIC) remains powered on and waits for a special network message called a magic packet. Keeping the NIC on does consume some power, but this power usage is far less than normal operating power, and is normally within standby power guidelines. The magic packet is usually sent from another computer on the same network as the destination computer, and is a broadcast frame that contains six bytes of all 255 (ff:ff:ff:ff:ff:ff in hexadecimal) followed by sixteen repetitions of the destination computer's MAC address ("Wake on LAN Technology," 2006). This packet is sent to all NICs on the network using the network's limited broadcast address (i.e. 255.255.255.255), but because it contains the MAC address of a specific destination NIC, only the computer with that NIC will turn on. Magic packets can be encapsulated in an IP packet, and can thus be sent using TCP or UDP, with the latter being more commonly used.

In most situations, WoL packets are transmitted via the broadcast transmission mode. For cross-subnet transmissions, WoL packets can be transmitted in two main ways: via unicast or via subnet-directed broadcasts. Unlike subnet-directed broadcasts, using the unicast transmission method does not require network routers to be reconfigured. However, packets will not be able to find the destination computer if that computer's IP address changes (i.e. dynamic IP addresses). In addition, depending on the network adapter and its configuration, the computer may not respond to WoL packets sent using unicast ("Choose Between Unicast," 2012).

Problems and Limitations with WoL

Despite the support for WoL, and the benefits it can bring, there are a few limitations that must be acknowledged. These limitations include WoL's system requirements, the lack of magic packet delivery confirmation, network security, subnet limitations, and the necessity of destination MAC addresses. In a typical small scale home network, these limitations are less likely to affect WoL implementation. However, in larger enterprise networks, where there may be hundreds or thousands of computers (often divided into separate subnets), these limitations can make WoL challenging to implement.

System Requirements

WoL requires correct hardware and software support to function; this is typically not an issue with modern computers and networks. However, we mention hardware requirements here because it is potentially a costly change to make if an organization's hardware is not already WoL capable. However, due to early alliance initiatives, WoL is well supported, which makes this a non-issue for most organizations.

Lack of Delivery Confirmation

In order to minimize power usage and processing time, WoL was designed to be very simple. This allows the NIC to process magic packets very quickly, but this also means that it does not provide delivery confirmation. This is typically not problematic in actual WoL usage, but may cause difficulties with troubleshooting. As a part of our testbed, we used the free packet sniffer Wireshark to monitor WoL packets on target subnets.

Security Issues

Unless network hardware is configured to only allow traffic that meets specific security requirements, WoL packets can be sent by anyone on the same local area network (LAN) as the destination computer(s). Although sending an "unauthorized" WoL packet does not in itself bypass standard security measures (such as system passwords and network firewalls), once a computer is on, attackers may be able to scan it for vulnerabilities. Simply leaving the computer turned on, however, exposes the computer to vulnerability scanning even more. Whether or not WoL is implemented, strong security policies must be put in place to protect the system. Some network interface cards support passwords, but encryption is generally not supported, making such measures vulnerable to sniffers.

Subnet Limitations

Another potential security issue in implementing WoL may arise when attempting to use subnet directed broadcasts (SDBs). Also known as IP directed broadcasts, SDBs send WoL packets to a computer by using the destination's subnet broadcast address. Routers with this feature enabled will forward WoL packets to their destination router as a normal IP packet, and then the end router will broadcast it to every computer on the subnet. This is particularly useful if the networked computers frequently change their IP addresses, as unicast transmissions may be unable to deliver the WoL packets consistently.

Since computers targeted by WoL are typically off or asleep, they may not respond to Address Resolution Protocol (ARP) messages from the router, and will probably not have an IP address. This causes problems when using the unicast transmission method. Also, if there is a switch connected between the destination router and destination computer, the switch may not know the port to which the computer is physically connected.

If a router is configured to allow subnet directed broadcasts, some router security features are often disabled. These security features are intended to prevent distributed denial of service (DDoS) attacks such as the Smurf Attack. This type of attack can be particularly harmful in large networks, where unplanned network downtime could have serious consequences for users. There are two primary ways of preventing Smurf attacks, 1) configuring routers not to forward packets directed to broadcast addresses (this is now the default setting for most routers), and 2) configuring network hosts and routers to not respond to ping requests or broadcast messages. Though these solutions may work in some cases, when attempting to implement WoL neither of them is ideal. If the routers are configured not to forward subnet directed broadcasts, WoL messages sent from subnets other than the intended recipient's local network are rejected, preventing the computer from waking up.

Necessity of the MAC Address

As a part of how WoL works, the MAC address of the destination computer is required. On a local network, the MAC address can easily be found by using basic commands like *ipconfig/ifconfig,* but using this method to manually obtain the MAC addresses of a group of computers in a large network is quite tedious. MAC addresses can also be retrieved with the Windows command line tools nbtstat and getmac, but since MAC addresses are only used for point to point addressing, obtaining them from across subnets is typically not possible. However, in cases where obtaining a list of MAC addresses from a specific subnet is satisfactory, the *getmac* command can automatically create a table, list, or CSV file ("Getmac," 2012; "Nbtstat," 2012). As a part of our testbed, we wanted to determine whether or not there was a simple way to obtain MAC addresses across subnets.

METHODOLOGY

The following section describes our testbed methodology.

Network Setup

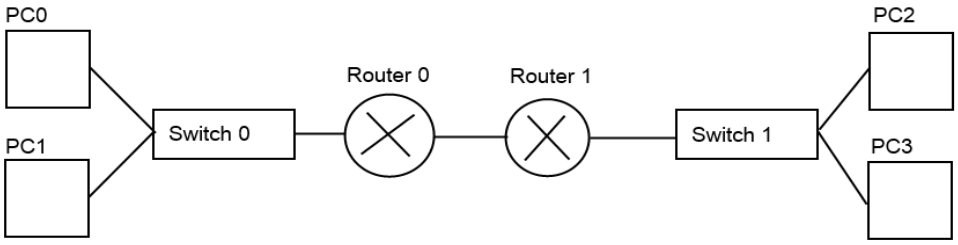
Our overall goal in creating this testbed was to demonstrate how WoL could be implemented easily and inexpensively, using existing software tools, and without the use of a local agent (software that is installed on all client computers).

In order to simulate a multi-hop (cross-subnet) network, we setup our network topology with two sets of computers, each on their own subnet. In addition to this, the router to router connection in the middle is also considered to be its own subnet, making this a multi-hop network (three subnets). However, since network routing is done on a point to point basis, the number of routers does not matter - we can easily add more hops and/or subnets without it affecting the results. The network topology model was created using Cisco Packet Tracer (see Figure 1). Our subnet mask and IP configuration was shown in Table 1.

TABLE 1
NETWORK SETUP

	Router 0	Router 1	PC0	PC1	PC2	PC3
Interface 0	15.0.0.2/24	15.0.1.2/24	15.0.0.1/24	15.0.0.3/24	15.0.2.2/24	15.0.2.3/24
Interface 1	15.0.1.1/24	15.0.2.1/24	N/A	N/A	N/A	N/A

FIGURE 1
NETWORK TOPOLOGY



Hardware

For our hardware, we used Cisco routers and switches, along with Dell enterprise desktops. The topology is shown in Figure 1. Cisco networking hardware is popular among business and university networks (Howard M., 2012) though most routers in the same price range have similar features, regardless of brand. We used Cisco's 1800 series routers and Catalyst 2960 switches, both running IOS version 12.4(1c). The desktops were Dell Optiplex 380s, with Windows XP Professional (Service Pack 3) as the operating system. The network interface card for all desktops was the Broadcom NetLink (TM) Gigabit Ethernet adapter. We also performed some initial testing with a variety of mobile devices, including Amazon's Kindle Fire, Google's Nexus One, and Apple's iPad. WoL messages were sent from these devices to the desktops over a wireless network using the Linksys WRV200 wireless router.

Software

This testbed primarily used Windows based operating systems, but since WoL is platform independent, using other operating systems would not affect our results (though the actual WoL utilities

used may be different). Both Mac and Linux support WoL ("Wake-on-LAN Explained," 2011). In large scale networks, installing new software on all computers can be difficult, time consuming, and expensive. For these reasons, we did not attempt to use any WoL utilities that required a local agent to be installed on client computers. Though using a local agent often provides additional management functionality, for the purpose of WoL, it only adds to the difficulty of successful implementation. To send the actual WoL packets, we used a variety of third-party software, all of which were free. The tools we used include: MC-WOL ("Wake-on-LAN," 2008), SoftPerfect Network Scanner ("SoftPerfect Network Scanner," 2012), Wake-on-LAN ("WakeOnLan," 2012), and WakeMeOnLAN (Sofer N., 2012). We also used Matcode's utility *MCGETMAC* to retrieve the MAC addresses of computers on a particular subnet. We also used Windows Task Scheduler ("How To Schedule Tasks," 2012) to specify the time our batch files (Windows script files) were executed.

While not usually a problem in practical WoL implementation, the lack of delivery confirmation was an issue in our testbed, as there was no way to verify what WoL messages the client computers were receiving. So in order to capture this network traffic and to verify that the computers were receiving the magic packets, we used both Wireshark ("Wireshark," 2012) and WoL Packet Sniffer ("Wake-on-LAN Packet Sniffer," 2011).

Batch Files and Scheduling

For organizations that primarily operate during the day, turning off computers for the night could be a convenient way to save on energy costs. However, many organizations leave computers on overnight so that various administrative tasks (e.g. patch deployment) can be performed. By using Windows Task Scheduler and a simple batch file, we were able to automatically run certain WoL utilities at a specific time. We also used these features to send multiple magic packets to the same machine, in order to ensure that the WoL packet was received. SoftPerfect's Network Scanner has extensive command line functionality, and Matcode's command line tool MC-Wake-on-LAN also worked for this purpose.

Process

On production WoL implementations, a packet sniffer may not be required – it is only useful as a troubleshooting tool. Also, to determine whether or not WoL packets could be sent and received without subnet-directed broadcasts, we send some WoL packets using the unicast transmission method (without implementing subnet directed broadcasts on the routers).

Steps Taken

- Enable WoL in the BIOS of the client computers
- Create firewall port exceptions for the WoL packets (usually UDP ports 7 and 9)
- Enable WoL on the NIC via Device Manager ("Allow the computer to turn off this device to save power"; "Allow this device to wake the computer")
- Install Wireshark and/or WoL Packet Sniffer on the computers
- Configure routers for IP directed broadcasts (with access lists for security)

FINDINGS AND ANALYSIS

The following section details our findings.

Software and Scheduling

Overall, the abundance of free WoL software made finding a working solution fairly easy. However, though most of these programs were adequate for sending WoL packets, none had every feature we were looking for. SoftPerfect's Network Scanner was by far the most robust, though it did not have a built-in scheduling feature ("SoftPerfect Network Scanner," 2012).

Aquila WakeOnLAN successfully sent the WoL packets to the destination computer, but did not let us specify how the magic packets were sent (i.e. unicast, broadcast, or subnet directed broadcast). Also, its built-in scheduling feature did not work with Windows XP, which limited the tool's usefulness in our testbed. On the other hand, Matcode's command line tool MC-WOL allowed us to specify how the magic packets were sent (i.e. unicast, broadcast, or subnet-directed broadcast), but it had no built-in scheduler. However, we were able to bypass this limitation in MC-WOL through the use of Windows Task Scheduler and simple batch files. Unfortunately, since most of the WoL utilities did not have command line functionality, we were only able to accomplish this with MC-WOL and SoftPerfect's Network Scanner. With the other WoL applications, if they did not have a built-in scheduling feature, turning on computers at a specific time was not possible.

WakeMeOnLAN was able to successfully send magic packets using subnet directed broadcasts, and had the option to use the NetBIOS protocol to scan for computers on other subnets. It could also generate an HTML report of IPs, computer names, and MAC addresses, which was particularly useful for creating a list of IPs and MAC addresses on the network. Like MC-WOL, it did allow us to specify how the magic packets were sent, though it did not have a built-in scheduler or command line functionality.

Using MC-WOL, we attempted to sent a magic packet using the broadcast MAC address (ff:ff:ff:ff:ff:ff) in order to turn on all of the computers at once. However, though this packet showed up as a proper WoL packet in both Wireshark and WoL Packet Sniffer, none of the computers would turn on when we sent this particular packet to them. This may have been due to improper switch configuration or limitations in WoL itself, but it requires further testing for us to be sure.

Security and Subnet Limitations

Initially, we had difficulty sending Wake-on-LAN messages across subnets, as the default WoL implementation does not allow it. However, sending the magic packets as unicast messages seemed to work, though after a few minutes of being off, the computers would stop responding to these unicast messages. This was likely due to the ARP cache table, though it could also be attributed to the MAC address table used by the switches. It is possible to extend the timer for both of these tables, but this could cause additional issues in some networks.

Rather than extending these timers, we opted to implement subnet directed broadcasts. However, in order to use subnet directed broadcasts without opening the network to a DDoS attack such as the Smurf attack, we implemented access control lists on the routers. Doing this limits the subnet directed broadcasts to those from specific computers, and also specifies the protocol and port that are to be forwarded (thus further limiting the security risks). For Cisco routers, the specific configuration instructions can be found on their website ("Catalyst Layer 3 Switch," 2007).

MAC Address Necessity

As mentioned in the literature review section of this paper, a magic packet is a broadcast frame that contains six bytes of all 255 (ff:ff:ff:ff:ff:ff in hexadecimal) followed by sixteen repetitions of the destination computer's MAC address ("Wake-on-LAN Technology," 2006). Without the destination computer's MAC address, WoL will not work. Obtaining this MAC address for all computers in a large scale network can be difficult, particularly if there are multiple subnets. For our testbed, we used the MAC address resolution feature in SoftPerfect's Network Scanner. It automatically attempts to resolve MAC addresses through ARP, NetBios, and Router SNMP MIB (Simple Network Management Protocol Management Information Base) queries, and was able to remotely retrieve the MAC address from all computers in our testbed.

General Observations

With the abundance of free WoL utilities, there is no reason for organizations not to implement it for its benefits in energy conservation and network administration. Magic packets can easily be sent across subnets by configuring routers to use IP directed broadcasts, and the security issues can be minimized by

implementing access control lists. Also, scheduling WoL could be particularly useful for universities, as it could allow administrators to turn on computers every morning before class/work starts.

FUTURE RESEARCH

Though this testbed provided some answers for network administrators interested in implementing WoL in large networks, there are still many related areas for future research. Some of these areas include: determining specific energy savings in a particular setting, creating a more robust software tool that includes all desired functions, analyzing the pros and cons of WoL with mobile devices, and analyzing the potential benefits and issues with wake-on-Wireless-LAN (WoWLAN).

Wake-on-Wireless-LAN and WoL with mobile devices could be particularly interesting areas of research, as computing has moved more and more toward mobile devices. According to one survey, 81% of global executives use a mobile device, and it was estimated that 75% of the US workforce would go mobile by 2011 (Friedman J and Hoffman D.V. 2008). However, since mobile devices typically have very conservative power settings, it would be interesting to see how much energy could actually be saved (if any) by using WoWLAN and other energy conservation strategies.

CONCLUSION

Though WoL is a relatively mature technology, many organizations do not implement it due to some of its limitations. In this paper, we discussed these limitations and implemented a simple testbed attempting to solve these problems. Our testbed shows that these limitations can be overcome fairly easily with proper router and host configuration. Moreover, the implementation of WoL need not be prohibitively expensive considering the range of freely available tools. This implies that interested network administrators can easily adapt the ideas presented in this paper to their own specific scenarios.

REFERENCES

Catalyst Layer 3 Switch for Wake-on-LAN Support Across VLANs Configuration. (2007). Retrieved from http://www.cisco.com/en/US/products/hw/switches/ps5023/products_configuration_example09186a008084b55c.shtml

Choose Between Unicast and Subnet-Directed Broadcast for Wake-on-LAN. (2012). Retrieved from http://technet.microsoft.com/en-us/library/bb632911.aspx

Friedman, J., Hoffman, D. V. (2008). Protecting Data On Mobile Devices: A Taxonomy Of Security Threats To Mobile Computing And Review Of Applicable Defenses. *Information Knowledge Systems Management*, 7.1/2 (159-180).

Getmac. (2012). Retrieved from http://technet.microsoft.com/en-us/library/bb490913.aspx

Howard, M. (2012). Infonetics Research: Cisco, Alcatel-Lucent, Juniper Gain Carrier IP Edge Router/Switch Share in Q1; N. America up 27%. *EON: Enhanced Online News.* Retrieved from http://eon.businesswire.com/news/eon/20120524005442/en/carrier-router-market-share/carrier-ethernet-switch-market/service-provider-routers-and-switches-forecast

How To Schedule Tasks in Windows XP. (2012). Retrieved from http://support.microsoft.com/kb/308569

Nbtstat. (2012). Retrieved from http://technet.microsoft.com/en-us/library/bb490938.aspx

O'Malley, T. (1998). IBM Announces Universal Management -- Industry's Most Comprehensive Tools to Lower Total Cost of Ownership. *IBM*. Retrieved from http://www-03.ibm.com/press/us/en/pressrelease/2705.wss

Remote Management using Intel AMT. (2012). Retrieved from http://www.opengear.com/SP-AMT.html

Sofer, N (2012). WakeMeOnLan v1.37. Retrieved from http://www.nirsoft.net/utils/wake_on_lan.html

SoftPerfect Network Scanner. (2012). Retrieved from http://www.softperfect.com/products/networkscanner/

Wake-on-LAN. (2008). Retrieved from http://www.matcode.com/WoL.htm
Wake-on-LAN Explained. (2011). Retrieved from http://www.datasynergy.co.uk/products/wakeman/pdfs/Wake-on-LANExplained.pdf
Wake-on-LAN Packet Sniffer. (2011). Retrieved from http://profshutdown.com/wakeonlan_troubleshoot.aspx
Wake-on-LAN Technology. (2006). Retrieved from http://www.liebsoft.com/pdfs/Wake_On_LAN.pdf
Webber, C., et. al. (2006). After-hours power status of office equipment in the USA. *Energy*, 31, 14 (2823–2838).
Welcome to AquilaWOL. (2012). Retrieved from http://aquilawol.sourceforge.net
Wireshark. (2012). Retrieved from http://www.wireshark.org/