2007

An examination of the Asus WL-HDD 2.5 as a Nepenthes malware collector

Patryk Szewczyk
Edith Cowan University

DOI: 10.4225/75/57ad5c6f7ff32
Originally published in the Proceedings of the 5th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2007.

An examination of the Asus WL-HDD 2.5 as a Nepenthes malware collector

Patryk Szewczyk
School of Computer and Information Science
Edith Cowan University
p.szewczyk@ecu.edu.au

Abstract

The Linksys WRT54g has been used as a host for network forensics tools for instance Snort for a long period of time. Whilst large corporations are already utilising network forensic tools, this paper demonstrates that it is quite feasible for a non-security specialist to track and capture malicious network traffic. This paper introduces the Asus Wireless Hard disk as a replacement for the popular Linksys WRT54g. Firstly, the Linksys router will be introduced detailing some of the research that was undertaken on the device over the years amongst the security community. It then briefly discusses malicious software and the impact this may have for a home user. The paper then outlines the trivial steps in setting up Nepenthes 0.1.7 (a malware collector) for the Asus WL-HDD 2.5 according to the Nepenthes and tests the feasibility of running the malware collector on the selected device. The paper then concludes on discussing the limitations of the device when attempting to execute Nepenthes.

Keywords

ADSL routers, Nepenthes, OpenWRT, malware, network forensics

INTRODUCTION

Trends in router technology advancements are permitting consumers to use their device as a gateway for Internet connectivity, a resource sharing point and a wireless access point. As routing devices become increasingly powerful with superior processors and significant increases in memory, developers are opting to utilise routers for a variety of applications. One router which received immense publicity was the Linksys WRT54g due to its highly manipulative firmware and basic configurable nature. Literature demonstrating the flexibility and ease of use of the Linksys WRT54g for both researchers and hobbyists (Asadoorian & Pesce 2007) is still being publicly released years after the initial product release.

Hackers and security enthusiasts may customise the Linksys WRT54g firmware (Al-Zarouni 2005) dependant on the place and purpose of use. Numerous pre-compiled firmware images are available specifically for the Linksys WRT54g and many other embedded system architectures. Innes (2005) discussed the application of some of the publicly available, pre-compiled software packages for the Linksys WRT54g operating on the OpenWRT firmware. The paper demonstrated how a Small office Home office (SoHo) router may be transformed to undertake various 802.11 wireless monitoring, intrusion detection and network forensic tasks. However, as time progresses the device which once permitted a wide range of software to be on the Linksys WRT54g is slowly becoming obsolete. One significant aspect in which the Linksys WRT54g is now insufficient is in the memory and storage availability which prevents it from being used for various network analysis and forensic activities.

Linksys released numerous versions of the Linksys WRT54g (versions 1 through to 8) with processors ranging from MIPS 125 MHz through to 240 MHz (OpenWRT 2006). However, as the processor performance increased on the high end routers the availability of flash and random access memory (RAM) decreased. Hence system performance was balanced across all the Linksys WRT54g routers. Whilst the device specifications are sound for the router's requirements any intensive third party software will halt the device and require a manual power cycle. In contrast certain network analysis and forensic software does not only consume excessive resources but also requires a medium on which to store data it collects. One specific setup which is not feasible, is operating a Snort intrusion detection system (Snort 2007) coupled with a database server to log events. As the Linksys router has minimal non-volatile storage availability, the only feasible option is to utilise a remote database server to which the Snort intrusion detection system may connect and store log files.

One of the ways to bypass the resource requirements of a router is to utilise an Asus Wireless Hard Disk (WL-HDD). Competing with the Linksys WRT54g, the Asus WL-HDD encompasses a *Broadcom 4710* – 125 MHz processor coupled with four megabytes of flash and sixteen megabytes of RAM. The device mimics the specifications of the Linksys WRT54g up to and including revision four. The device utilises a removable external dipole antenna, a 10/100 Ethernet connection and a user attachable hard disk connector for a notebook

forty gigabyte hard drive (Johnson 2005). By default the Asus WL-HDD utilises a proprietary pre-imaged firmware stored on the four megabyte flash memory. The default operating system allows end-users to: enable MAC address filtering, Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) encryption for wireless access. Further attributes include the potential to share and access resources amongst multi users using the Samba server package and the File Transfer Protocol (FTP). The device also has the added benefit of being able to copy the contents of USB devices on the fly without the need for a workstation.

MALWARE FOR FORENSIC ANALYSIS

Traditional forensic investigations have focused purely on recovering data from persistent storage mediums. However, the number of online crimes is increasing with distributed denial of service attacks and the propagation of malicious software. Thus the need to identify the source and design methods of prevention is becoming increasingly prevalent. Throughout the early 1990's forensic investigators focused predominately in attempting to identify the 'author' of viruses, Trojan horses and worms through a process of "software forensics" (Spafford & Weeber 1993). As time has progressed the same needs exist, although forensic investigators are not only attempting to identify the 'authors' or source of malicious software but are also attempting to bypass 'anti-forensic' techniques (Forte & Power 2007).

Malware

The amount of malware propagating on the web is on a steady rise, and malicious software developers are constantly advancing methods by which this concealed malware may spread and infect hosts. This is evident with the continual release of patches and updates for anti-virus and spyware applications. On a recent study initiated by Google it was found that 10 percent of the web pages indexed contained some form of malware (Anonymous 2007). In most instances the malware was stored on third party servers and hence not on the actual server hosting the web site. Whenever a user would attempt to access an infected web site they would be instructed to install an applet without any significant indication of its malicious nature. In the study Google admitted that consumers would find it difficult to protect themselves and may succumb to the associated threats of a malicious workstation.

Malware trends are steering away from traditional viruses and worms which may cause the specific workstation to become unstable and thus are moving towards botnets. Botnets are a collection of hosts under the control of a master who would generally utilise the hosts to carry out malicious tasks including the cracking of cryptographic ciphers or the undertaking of distributed denial of service attacks (Schultz 2006). Certain organisations are utilising malware for unwanted advertising targeted towards a specific host. Alternatively consumers are being victimised by malware which redirects the web browser to malicious clones of their bank's web site (Pemble 2005). As unsuspecting individual's input personal information as per usual, these details are being logged and stored for various malicious financial acts.

According to Secure Computing (2007) Trojans and targeted Spyware accounted for 78 percent of malware activity on the Internet for the month of August. Of that almost 97 percent was targeted at Microsoft Windows based workstations. Avoiding malware for consumers may prove difficult with as many as 11,906 new web sites discovered in August by Secure Computing (2007) containing malware destructive to Microsoft Windows based workstation. Combating malware may also prove potentially difficult for consumers with the need of being informed of the latest Internet scams, installing appropriate operating systems updates and patching anti-virus signatures for their workstation.

Malware Forensics

Nikkel (2006) details the use of a "portable network forensics evidence collector" built on a desktop processor utilising 128 megabytes of memory and an operating system for a desktop workstation. However, the evidence collector virtually mimics a desktop host utilising a honey pot as an intrusion detection system. As this paper demonstrates, a similar system may be developed using commercially sold pre-built hardware managed by an open source operating system in turn reducing the overall cost of producing a malware collector. Utilising a malware collector host for forensic purposes is vital as the "examination of more than 14,000 unique and valid binaries showed that current anti-virus engines have some limitations and fail to detect all malware propagating in the wild" (Baecher et al. 2006, p.183).

In order to ensure a thorough forensic capture and analysis of malware the executable binaries must be preserved in a manner which adheres to the forensic principles. One method by which forensic investigators may then attempt to analyse malicious software is by capturing the binary prior to infection of a system and then apply forensic techniques to identify the source, malicious intent, anti-forensic techniques and targeted system (Gray et al. 1997). By applying forensic techniques, investigators and researchers may understand and discover new trend patterns but most importantly develop new rule sets for intrusion detection systems (Baecher et al.

2006). In turn this research will improve the detection rate and hence allow forensic investigators to uncover new anomaly intrusion patterns in the future.

Nepenthes

An approach to capture, document and analyse malicious software in a forensic manner is through the use of a honey pot with a pre-configured up to date rule set. Nepenthes is a low interaction honey pot based on the principles of honeyd. Nepenthes operates in a passive mode emulating well documented Microsoft Windows vulnerabilities (Nepenthes 2007b). Utilising a method of deception the attacker believes they are exploiting a vulnerable machine. Nepenthes emulates and responds to the attacks as would any other vulnerable workstation. Hence the attacker believes the exploit has been successfully executed. The malicious software binary is stored safely without infecting any other connected hosts on the same subnet as the Nepenthes server. Nepenthes may operate on many workstations with lower-end hardware and minimal user interaction. The only requirement is an ADSL connection with an available Ethernet port on the router. Rule sets are then configured on the router which forwards all traffic on pre-defined ports to the malware sensor for monitoring and analysis of threats.

Developers of Nepenthes argue that the software is open source and may be recompiled to suit the environmental needs (Nepenthes 2007b).. Network administrators may monitor network activity and determine which exploits may have disrupted certain hosts on the network. Furthermore, the malware collected is free and the binaries can be reverse engineered and analysed to identify country of origin, developers and trends in malware development. As the number of malware activity circulating on the Internet is on a steady rise, Nepenthes makes it simple to counter the threat by collecting, analysing and developing methods to prevent a compromise in future attempts.

CONFIGURING AN ASUS WL-HDD

By default the Asus WL-HDD is shipped with proprietary firmware which is of no use to developers whom wish to execute third party software. Hence, the first task of installing Nepenthes onto the device is to remove the existing firmware and image an operating system which is able to execute Nepenthes successfully. The chosen firmware was OpenWRT. This operating system is highly customisable permitting end-user's to install and remove packages as desired whilst making full use of any additional features the device may incorporate including; wireless and USB interfaces (OpenWRT 2007). The developers of OpenWRT have released precompiled firmware images for the most common embedded system architectures. The selected firmware was "WhiteRussian 0.9 stable" for the Broadcom 2.4 chipset as per the hardware on Asus WL-HDD.

The Asus WL-HDD is shipped with a Firmware Restoration utility (Figure 1) used mainly for restoring the original firmware in the event of an operating system failure or upgrade. However, an alternative firmware may also be imaged utilising the same utility. Whilst there are other approaches to imaging the device including the Trivial File Transfer Protocol (TFTP), the firmware restoration utility is the simplest.

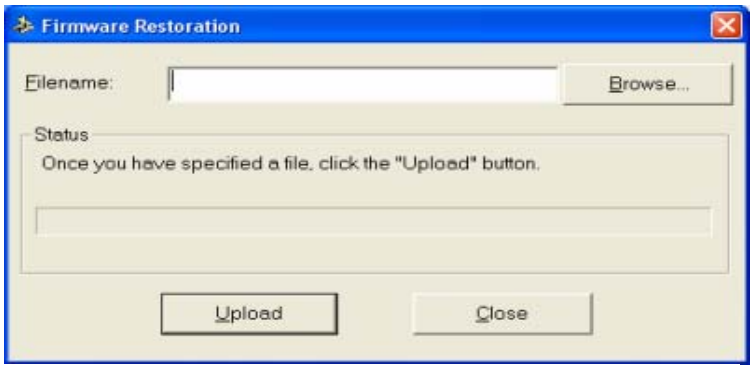

Figure 1 Asus Firmware Restoration Utility

Once the device has been successfully imaged, users may use the Secure Shell (SSH) protocol to access the device. Upon connection users are presented with a Linux console interface. The commands utilised to interact with the device are a replica of those used on a Linux workstation and may be identified by pressing the <Tab> key. By default the device does not have sufficient storage space to house the complete Nepenthes installation with all dependent files. As can be seen the dynamic partition has only two megabytes of non-volatile storage.

However, the entire Nepenthes installation requires at least six megabytes, thus additional storage is attached to the device.

root@OpenWrt:~# df

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
dev/root/	1024	1024	0	100%	/rom
none	7152	40	7112	1%	/tmp
/dev/mtdblock/4	2240	616	1624	28%	/jffs
/jffs	1024	1024	0	100%	/

The Asus WL-HDD whilst marketed as a router is in fact a Network Attached Storage (NAS) device. Hence, utilising its main feature, a forty gigabyte pre-partitioned notebook hard disk was attached as per the Asus documentation. Whilst any hard disk size may be attached to the device, the maximum partition size must be no more than forty gigabytes. The default installation of the firmware does not allow the user to mount any external storage due to missing packages. Thus two packages are installed and automatically configured by the OpenWRT package management system (IPKG). The user has the option of either installing an IDE or USB package dependant on the storage device, in conjunction with EXT2 or EXT3 file system package again dependant on the partition type.

root@OpenWrt:~# ipkg install kmod-ide

Downloading http://downloads.openwrt.org/whiterussian/packages/kmod-ide_2.4.30-brcm-5_mipsel.ipk

Installing kmod-ide (2.4.30-brcm-5) to root...

Configuring kmod-ide

Successfully terminated.

root@OpenWrt:~# ipkg install kmod-ext3

Downloading http://downloads.openwrt.org/whiterussian/packages/kmod-ext3_2.4.30-brcm-5_mipsel.ipk

Installing kmod-ide (2.4.30-brcm-5) to root...

Configuring kmod-ext3

Successfully terminated.

Once the packages are successfully installed the modules file must be edited to include the newly installed packages as per the instruction through OpenWRT developers. The modules file is located in the /etc directory of the file system.

root@OpenWrt:~# vi /etc/modules

ide-core

pdc202xx_old

ide-detect

ide-disk

wl

jbd

ext3

Once the module file has been edited and successfully saved a reboot of the device is instantiated. Upon reboot, the module file is loaded with the newly added attributes permitting the external hard disk to be recognised as a storage medium in the partitions table. In this instance the forty gigabyte hard disk has three partitions including a two gigabyte swap partition which is used further.

root@OpenWrt:~# cat /proc/partitions

major	minor	#blocks	name
3	0	39070080	ide/host0/bus0/target0/lun0/disc
3	1	18595206	ide/host0/bus0/target0/lun0/part1
3	2	10554705	ide/host0/bus0/target0/lun0/part2
3	3	2891700	ide/host0/bus0/target0/lun0/part3

Mounting a specific partition on the disk will now enable the user to interact and save specific files on the disk. As shown below OpenWRT has a slightly different path for locating physical disks and this is the only way that a hard disk may be mounted.

root@OpenWrt:~# mount /dev/discs/disc0/part1 /mnt/disk1cd /mnt/disk1

root@OpenWrt:~# cd /mnt/disk1

root@OpenWrt:/mnt/disk1#

root@OpenWrt:/mnt/disk1# ls -al

drwxr-xr-x	4	root	root	4096	Jan	1	00:49	.
drwxr-xr-x	1	root	root	0	Jan	1	2000	..
drwxr-xr-x	5	root	root	4096	Jan	1	00:05	exp

The files required for the Nepenthes installation exceed the amount of non-volatile memory that the Asus device includes onboard by default. Hence, the default installation path for package management system was altered. Rather that installing the packages into the root directory located in non-volatile memory to a specific directory located on physical disk.

root@OpenWrt:~# vi /etc/ipkg.conf

src whiterussian http://downloads.openwrt.org/whiterussian/packages

src non-free http://downloads.openwrt.org/whiterussian/packages/non-free

dest root /

dest ram /tmp

dest mnt /mnt/disk1/exp/usr/lib

Installing Nepenthes

Although Nepenthes is currently released under version 0.2.2, a precompiled version of 0.1.7 was utilised to determine if the system would successfully handle a stable version. Furthermore, Nepenthes has a number of dependencies which have been precompiled and are available as a public download specifically for version 0.1.7 (Nepenthes, 2007a).

root@OpenWrt:/mnt/disk1/exp# ipkg install –d mnt nepenthes_0.1.7-0_mipsel.ipk

Installing nepenthes (0.1.7-0) to mnt...

Configuring nepenthes

Successfully terminated.

root@OpenWrt:/mnt/disk1/exp# cd /opt/nepenthes/bin

root@OpenWrt:/mnt/disk1/exp/opt/nepenthes/bin# ./nepenthes

./nepenthes: can't load library 'libadns.so.1'

As demonstrated above, installing and attempting to execute the Nepenthes binary will lead to an error with missing library files. By default Nepenthes searches for files within (specific directory). However, due to the limited non-volatile memory resources on the device, the installation of the libraries was transferred to the disk.

ipkg install –d mnt adns_1.2-0_mipsel.ipk

…

ipkg install –d mnt curl_7.15-3_mipsel.ipk

…

ipkg install –d mnt file_4.17-_mipsel.ipk

…

The remaining library files *libgcc_s.so.1* and *libstdc++.so.*6 were manually transferred to the library location now stored on the disk. In order to ensure Nepenthes may execute successfully, symbolic links were created to the library files located on the disk. It is important to note that the symbolic links are not static by default upon reboot or shutdown of the device hence a script was created which would automatically setup all symbolic links.

root@OpenWrt:/mnt/disk1# cat nepenthes-script

ln -s /mnt/disk1/exp/usr/lib/libadns.so.1 /usr/lib/libadns.so.1

ln -s /mnt/disk1/exp/usr/lib/libmagic.so.1 /usr/lib/libmagic.so.1

ln -s /mnt/disk1/exp/usr/lib/libpcre.so.0 /usr/lib/libpcre.so.0

ln -s /mnt/disk1/exp/usr/lib/libcurl.so.3 /usr/lib/libcurl.so.3

ln -s /mnt/disk1/exp/usr/lib/libgcc_s.so.1 /usr/lib/libgcc_s.so.1

ln -s /mnt/disk1/exp/usr/lib/libstdc++.so.6 /usr/lib/libstdc++.so.6

root@OpenWrt:/mnt/disk1# ./nepenthes-script

After tweaking and configuring the Nepenthes configuration files, the program was finally ready for execution. The execution process was extracted from the Nepenthes documentation provided by the Nepenthes developers (Nepenthes 2007b).

root@OpenWrt:/mnt/disk1/exp/opt/nepenthes/bin#

./nepenthes -w /opt/nepenthes -c /opt/nepenthes/etc/nepenthes/nepenthes.conf --version

Nepenthes Version 0.1.7

Compiled on Linux/MIPS at May 23 2006 18:07:36 with g++ 3.4.4 (OpenWrt-1.0)

Started on root running Linux/mips

Limitations of Nepenthes 0.1.7

Whilst Nepenthes is capable of operating on embedded devices, the efficiency and stability is questionable. Whilst it is quite noticeable that Nepenthes is monitoring and collecting malware, after a period of approximately thirty minutes the Asus WL-HDD was no longer responding. Hence, a reboot was performed, and certain processes were killed in the hope of ensuring Nepenthes had sufficient resources to operate successfully. After a number of successive attempts the halting times of the device were random and it was concluded that too many malicious binaries were attempting to infect the system.

The OpenWRT documentation stipulates that it is possible to instantiate a swap partition for those devices which are lacking resources to execute processes successfully. After formatting a two gigabyte swap partition on the hard disk, the necessary packages were obtained to allow the operating system to manage swap storage. Further testing of Nepenthes with swap space made no significant difference to the performance of the process. It was hypothesised that the device continuously halted with swap space enabled, as the amount of volatile memory would be consumed faster than the processor was able to execute and write data to the swap.

Nepenthes Features

Nepenthes 0.1.7 features two main methods of analysing binaries once a malicious act has been detected; submit the file to a specific destination on local storage, or submitting the file to the Norman Sandbox (Norman 2007; Riden 2006). The Norman Sandbox will collect, analyse and store the binary and transmit results of the analysis to the email provided in the Nepenthes configuration file. The malware will also be stored locally and hashed to a specific folder again dependant on the configuration file. Alternatively Nepenthes may be configured to only store and log all malware locally without transmitting the binary for analysis to a third party host.

CONCLUSION:

The constant rise of malware spreading throughout the Internet is gradually becoming more difficult for consumers and anti-virus vendors to control. An analysis of captured binaries by malware honey pots shows that some anti-virus products are unable to identify the executables as malicious. Whilst organisations and research institutions are able to allocate resources towards forensically analysing malware, this paper demonstrates that a security conscious enthusiast is able to contribute to this research utilising simple and inexpensive hardware. By establishing a malware collector the binaries may then be transmitted to the central Nepenthes server for further analysis.

This paper depicts how an Asus WL-HDD can be turned into a powerful device for hosting various security applications. The onboard specifications for the Asus and Linksys product are identical however the Asus device does permit a user to attach a physical hard disk for additional storage space. Although the paper focused on a malware collector for a SoHo environment, future work could test the feasibility of executing software for example Kismet and logging intrusions natively which if executed on the Linksys WRT54g would require a second host to store log files. Furthermore, increasing the memory capacity of the device could prove useful in testing the effects this has on software which consumes more than the available memory. Since Nepenthes is becoming increasingly utilised amongst the research and security community it may also prove feasible to test the effects of Nepenthes on router's with high performance processors and increased memory capacity.

REFERENCES:

Al-Zarouni, M. (2005). Taxonomy of WRT54G(S) Hardware and Custom Firmware. Paper presented at the Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australian.

Anonymous. (2007). Google scans Web pages for malware – finds one in 10 infected. Computer Fraud & Security, 2007(5), 20.

Asadoorian, P., & Pesce, L. (2007). Linksys WRT54G Ultimate Hacking: Syngress Publishing.

Baecher, P., Koetter, M., Holz, T., & Dornseif, M. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware, URL http://honeyblog.org/junkyard/paper/collecting-malware-final.pdf Accessed 2 September, 2007

Forte, D., & Power, R. (2007). A tour through the realm of anti-forensics. Computer Fraud & Security, 2007(6), 18-20.

Gray, A., Sallis, P., & MacDonell, S. (1997). Software Forensics: Extending Authorship Analysis Techniques to Computer Programs. Paper presented at the 3rd Biannual Conference of the International Association of Forensic Linguists, Durham NC, USA.

Innes, S. (2005). Turning A Linksys Wrt54g, Into More Than Just A Wireless Router. Paper presented at the 3rd Australian Computer, Network & Information Forensics Conference, School of Computer and, Information Science, Edith Cowan University, Perth, Western Australia.

Johnson, C. (2005). ASUS WL-HDD 2.5" - NAS and Wireless AP, URL http://www.tweaktown.com/reviews/787/2/asus_wl_hdd_page_2_specifications/index.html Accessed 2 September, 2007

Nepenthes. (2007a). Nepenthes Development, URL http://nepenthes.mwcollect.org/~nepenthesdev/openwrt/ Accessed 11 September, 2007

Nepenthes. (2007b). Nepenthes finest collection, URL http://nepenthes.mwcollect.org/documentation:readme Accessed 14 September, 2007

Nikkel, B. J. (2006). A portable network forensic evidence collector. Digital Investigation, 3(3), 127-135.

Norman. (2007). Norman Sandbox, URL http://www.norman.com/microsites/nsic/ Accessed 30 September, 2007

OpenWRT. (2006). Table of Hardware – OpenWRT, URL http://wiki.openwrt.org/TableOfHardware?action=show&redirect=toh, Accessed 9 September, 2007

OpenWRT. (2007). What is OpenWRT?, URL http://openwrt.org/ Accessed 26 September, 2007

Pemble, M. (2005). Evolutionary trends in bank customer-targeted malware. Network Security, 2005(10), 4-7.

Riden, J. (2006). Using Nepenthes Honeypots to Detect Common Malware, URL http://www.securityfocus.com/infocus/1880 Accessed 2 October, 2007

Schultz, E. E. (2006). Where have the worms and viruses gone?—new trends in malware. Computer Fraud & Security, 2006(7), 4-8.

Secure Computing. (2007). Secure Computing's Trends in Email, Web, and Malware Threats, URL http://www.securecomputing.com/index.cfm?skey=1739 Accessed 5 October, 2007

Snort. (2007). Snort - the de facto standard for intrusion detection/prevention, URL http://www.snort.org/ Accessed 26 August, 2007

Spafford, E. H., & Weeber, S. A. (1993). Software forensics: Can we track code to its authors? Computers & Security 12(6), 585-595.

COPYRIGHT