3-12-2009

ADSL Router Forensics Part 2: Acquiring Evidence

Patryk Szewczyk
Edith Cowan University

DOI: 10.4225/75/57b28d6240cd4
7th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2009.

ADSL Router Forensics Part 2: Acquiring Evidence

Patryk Szewczyk
School of Computer and Security Science
Edith Cowan University

Abstract

The demand for high-speed Internet access is escalating high sales of ADSL routers. In-turn this has prompted individuals to attack and exploit the vulnerabilities in these devices. To respond to these threats, methods of acquisition and analysis are needed. The configuration data provides a wealth of information into the current state of the device. Hence, this data may be used to identify and interpret unlawful ways in which the device was used. This paper is centres around an empirical learning approach identifying techniques to address the device's acquirable limitations taking into consideration that the owner may not willingly present login credentials to directly access the device. This paper demonstrates a procedural method of obtaining data of interest from ADSL routers. It further elaborates on the methods by detailing how to extract and understand this configuration data.

Keywords
ADSL router forensics, digital forensics, embedded systems, vulnerability, forensic analysis, data extraction, JTAG, network forensics

INTRODUCTION

The demand for Asymmetric Digital Subscriber Line (ADSL) devices has increased considerably as consumers are offered fast and inexpensive methods to connect to the Internet. The plug and play nature of ADSL routers permits consumers to bypass the tedious configuration process and connect to the Internet in a streamlined manner. Numerous Internet Service Providers (ISPs) are shipping ADSL routers pre-configured with the client's username and password further eliminating the difficulties faced by novice computer users to access the Internet. Conversely, these pre-configured ADSL routers usually incorporate little or no security. As a result many Small office Home office (SoHo) ADSL routers are vulnerable to a range of emerging threats, exploiting these insecure aspects.

The number of threats to ADSL has increased significantly over the past few years. In addition the threats are sophisticated and well planned. Sajdak (2009) outlines the vulnerabilities associated with the flawed design of ADSL routers. One of these flaws permits Cross-Site Request Forgery (CSRF) attacks to take place. CSRF attacks target the ADSL router's flawed web server design. This attack succeeds since the Hypertext Transfer Protocol (HTTP) service places an automatic element of trust on internal hosts that attempt to access the web management function. As a result any attack originating from within the internal network would execute with *root* (or administrator) access.

The threats to ADSL routers are not limited to attacks from within the internal network. The default firmware of many first generation Netcomm routers initially encompassed a web server flaw which permitted the device to be accessed, compromised and controlled via a host external to the network (Sajdak, 2009). Whilst these vulnerabilities are exploiting the web server design, there are still many threats to ADSL routers that do not directly relate to the flawed web server.

In the first quarter of 2009 a new form of malware had been detected which was specifically targeting MIPS architecture - Linux based, ADSL routers. An ADSL Router based malware - Psyb0t was hijacking consumer-grade networking devices with the intention of creating a sophisticated and un-detectable botnet (Baume, 2009; Hunt, 2009; Magnus & Gassmann, 2009; Paul, 2009). The malware was capable of infecting approximately 55 different consumer grade ADSL routers in its early stages (Sajdak, 2009). In addition, the malicious binary is pre-populated with a list of 6,000 usernames and 13,000 passwords combinations commonly used on many ADSL routers – giving it a higher likelihood of compromising a vulnerable device. The malware once executed successfully resides in volatile memory. It then continually manipulates the configuration data to permit the botnet controller to communicate with the compromised device. To date evidence exists demonstrating the effects that this particular malware has had on third party web servers which have been shut down due to denial of service attacks by ADSL routers.

To date there are few protection methods that exist to prevent an ADSL router from being compromised. There is yet to be a release of an anti-malware product for ADSL routers. A simple solution includes changing the default login credentials on the device. Psyb0t cleverly exploits human-error by compromising those devices which did not have their default credentials modified. Alternatively, consumers may upgrade the firmware of their ADSL router on a regular and timely basis. However, as previous research has identified (Szewczyk & Furnell, 2009) in many instances applying a remedy is still a challenging task in that many individuals lack the skill set to appropriately download, install and apply a firmware update. This issue will remain complex until the process of firmware updating is streamlined in a similar method to the Microsoft Windows update process.

DIGITAL FORENSICS

Configuration data in ADSL routers is stored in two different memory components. ADSL routers encompass persistent storage, namely Non Volatile Random Access Memory (NVRAM). Data contained within the NVRAM remains intact during a power cycle or shutdown. In-turn this permits the end-user to store persistent configuration data such as wireless encryption keys. During the boot procedure the kernel creates a temporary storage space in Random Access Memory (RAM) which appears to the end-user and the embedded system as an extended flash storage medium. However, any changes which occur on the embedded system which are not *committed* to NVRAM or are stored in the temporary storage will be erased when a loss of power occurs.

From a forensic perspective the memory modules within the embedded system may be analysed and acquired utilising embedded system forensic methods (Breeuwsma, Jongh, Klaver, Knijff, & Roeloffs, 2007). The configuration data of an ADSL router contains a wealth of information which may aid in a legal case. Unlike a computer system or smart phone, there is little capacity within the device for the storage of illegal images, documents or multimedia files. The device does however control the flow of traffic within the network and as a result may aid in determining if a suspect has in fact accessed resources or websites of an illegal or malicious nature. Brown (2006) suggests that based on the threats currently targeting ADSL routers that the most important data to be acquired from an ADSL router consists of:

- Firmware version – This aids in determining if the device in question is utilising an illegal or maliciously altered operating system.

- System time – The ADSL router stores a simple yet effective log of significant system modifications. This may enable the forensic examiner to create a timeline of events which initiated or caused the crime to take place.

- DNS address – Acquiring the DNS address may in-turn allow the investigator to determine if any redirection of traffic is occurring.

- Wireless settings – Bandwidth theft is escalating due to insecure wireless access points. Acquiring the Medium Access Control (MAC) allow and deny lists, encryption keys, and determining if the access point is enabled may allow the forensic examiner to construct a scenario of events to do with the wireless network.

- Firewall rule sets – The ADSL router is pre-enabled with a set of firewall rule sets. Acquiring the current rules in place may aid in identifying how vulnerable the device in question was during an incident.

- Remote management – Many attacks attempt to enable remote management to simplify the controlling process. Acquiring remote management settings may aid the investigator in identifying to whom the permission has been enabled.

ACQUISITION OF POTENTIAL EVIDENCE

Breeuwsma (2006) details how the JTAG approach is feasible in acquiring data from ADSL routers. One of the most beneficial aspects is its small memory footprint and the absence of required login credentials. However, the JTAG approach is comparatively slow compared to the methods described in this paper. Furthermore, it does not immediately grant the investigator with data that could be analysed and used immediately. The procedure covered in this paper considers the benefits of the JTAG approach whilst demonstrating how configuration data could be acquired.

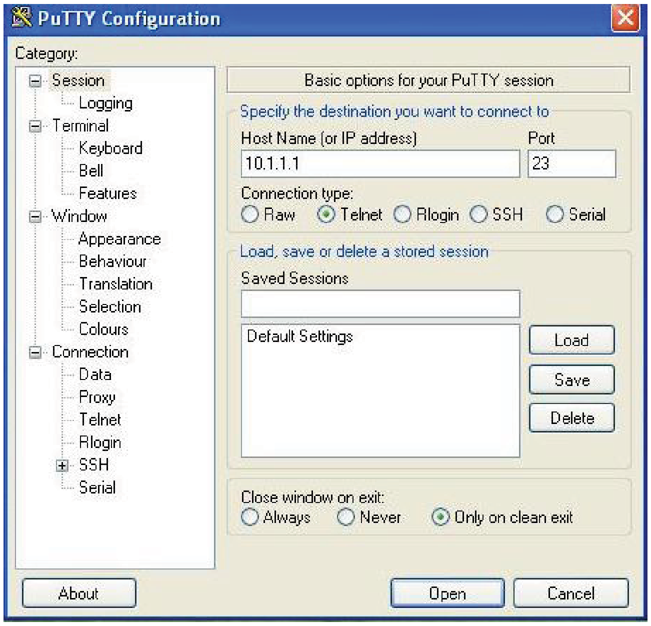
ADSL routers by default encompass few (if any) access ports – making it challenging to readily acquire data in a forensically sound manner. In addition the device is manufactured with soldered on-board memory components which are not only difficult to remove, but may also result in device malfunction due to improper handling. As a result investigating the device in a non-invasive manner is not always feasible. In "*ADSL Router Forensics: Methods of Acquisition",* Szewczyk (2009) details and compares the methods by which data could be acquired from ADSL routers to ensure little or no modification of the ADSL router state. These methods are practically applied within the procedure, through the use of a telnet or serial console approach.

ADSL routers operate through the use of partitions on flash memory. During boot time a file system is created which mimics that of a Linux based system. In typical ADSL router configurations there are four partitions consisting of; *mtd0,* which is a compressed CRAMFS or SQUASHFS file system image; *mtd1,* the MIPS Linux kernel; *mtd2,* the boot loader and; *mtd3,* the boot loader variables and configuration data (AR7 Firmware, 2009). The *mtd3* block of data is of most use to an investigator. During boot time the configuration data is pre-populated into the ADSL router permitting the device to function in the same manner during each reboot. The aim of this procedure is the actual extraction of either the *config.xml* file or the binary data contained in */dev/mtdblock/3* which encompass the configuration data for each device.

Accessing the ADSL Router

1) Accessing the device may be achieved in two distinct approaches dependant on whether or not the login credentials are available to the investigator.

   a) In the instance of a co-operative client and having access to login credentials an investigator may directly access the device utilising a telnet client through the IP addresses allocated to the ADSL router in question. In this circumstance the investigator requires both the username and password.



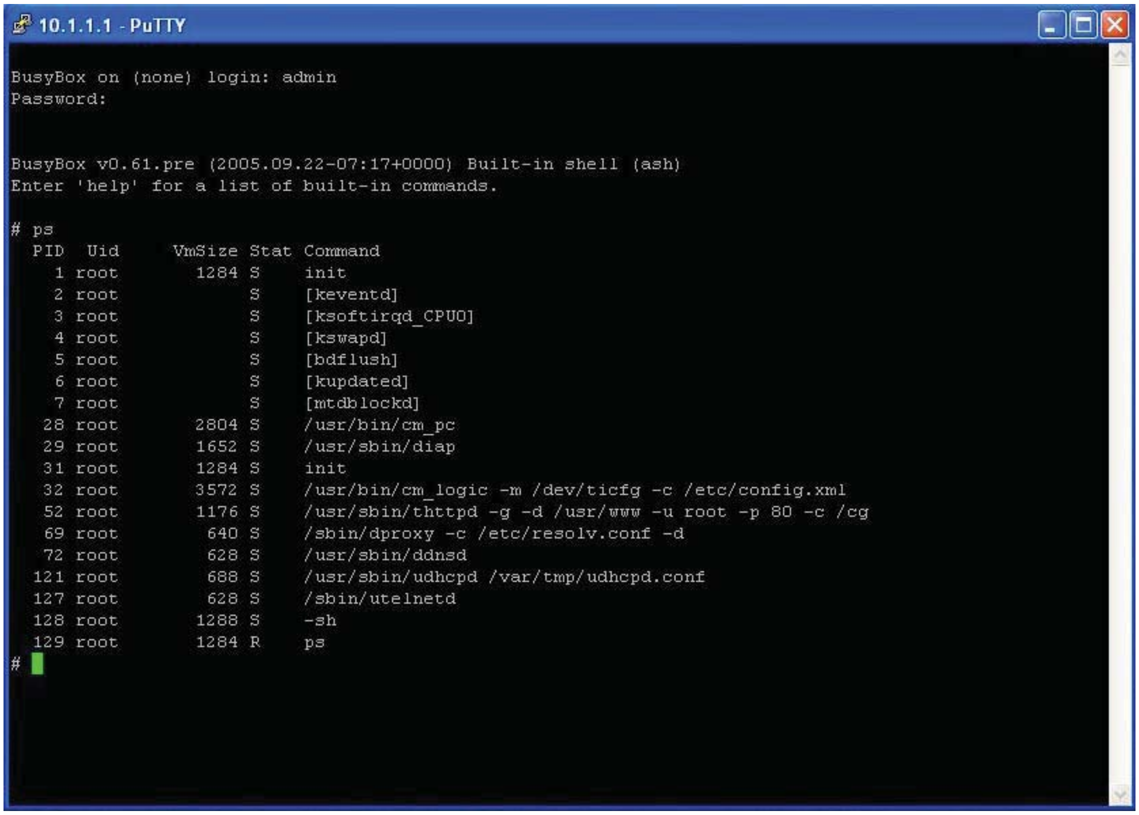
Host Name (or IP address): 10.1.1.1
Port: 23
Connection type: ○ Raw ● Telnet ○ Rlogin ○ SSH ○ Serial
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 2 Utilising PuTTY through Telnet

   b) In the event of an ADSL router requiring login credentials which were unknown, the investigator may still access the device through the serial console bypassing the authentication method. The serial console approach does not require credentials thus preventing a potential halt of the device through exceeding the number of login attempts.

In order for the serial console approach to operate three requirements must be met; access to the internal board within the ADSL router, which requires the removal of the outer case; identifying and locating the serial console pins; and, a specially designed cable which permits a connection between the workstation and the serial console ports on the ADSL router (Szewczyk, 2009). The serial client (PuTTy) must be configured to communicate with the ADSL router. ADSL router serial console settings include; 38,400 baud rate, no parity and flow control, eight (8) data bits and one (1) stop bit.

2) The HTTP daemon must be terminated as this service is required to obtain data in subsequent steps. This serves two purposes. Firstly it prevents intentional or accidental modification of configuration data through the web interface. Secondly, the HTTP services will be subsequently used to acquire the data of interest off the ADSL router. As demonstrated by Figure 2 the HTTP service in this experiment is distinguished by *Process ID* 52 which is currently hosting the web management function for administration.



```
BusyBox on (none) login: admin
Password:


BusyBox v0.61.pre (2005.09.22-07:17+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ps
  PID  Uid     VmSize Stat Command
    1 root       1284 S    init
    2 root            S    [keventd]
    3 root            S    [ksoftirqd_CPU0]
    4 root            S    [kswapd]
    5 root            S    [bdflush]
    6 root            S    [kupdated]
    7 root            S    [mtdblockd]
   28 root       2804 S    /usr/bin/cm_pc
   29 root       1652 S    /usr/sbin/diap
   31 root       1284 S    init
   32 root       3572 S    /usr/bin/cm_logic -m /dev/ticfg -c /etc/config.xml
   52 root       1176 S    /usr/sbin/thttpd -g -d /usr/www -u root -p 80 -c /cg
   69 root        640 S    /sbin/dproxy -c /etc/resolv.conf -d
   72 root        628 S    /usr/sbin/ddnsd
  121 root        688 S    /usr/sbin/udhcpd /var/tmp/udhcpd.conf
  127 root        628 S    /sbin/utelnetd
  128 root       1288 S    -sh
  129 root       1284 R    ps
# 
```

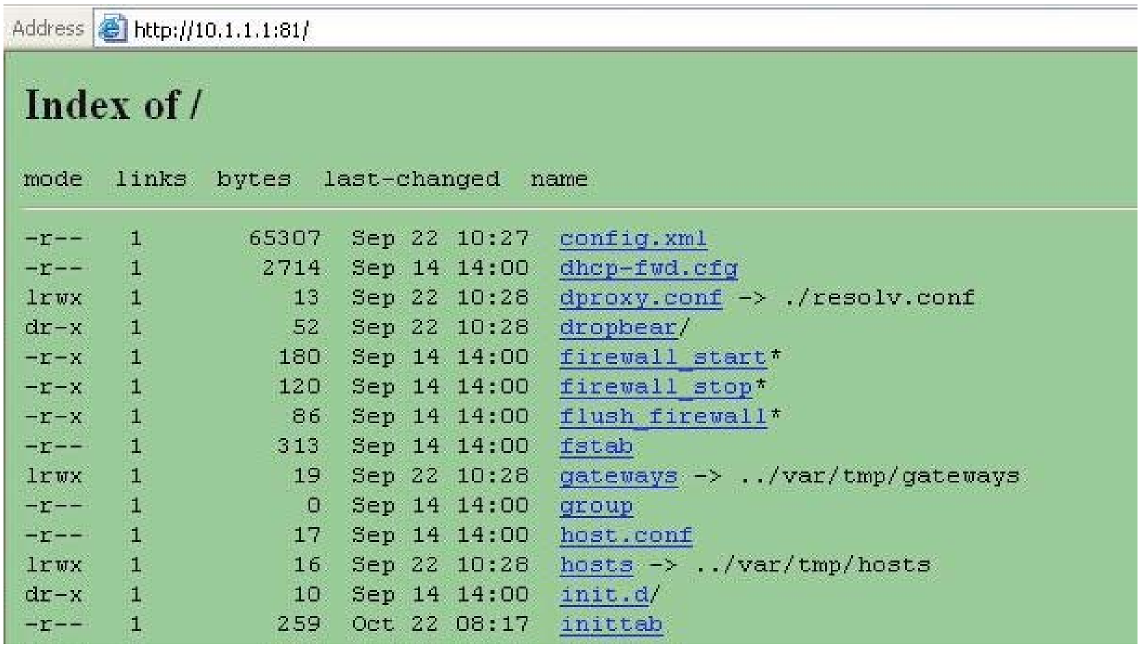
Figure 3 Default process list

Acquiring Data of Interest

3) There are two distinct configuration files which must be acquired from the ADSL router. The first configuration file is in XML format and is located in */etc.* The second data of interest element is located in */dev/mtdblock/3.* The *config.xml* file is made accessible through the following set of commands which setup a new HTTP daemon with root privileges.

```
# cd /etc
# thttpd –g –d /etc –u root –p 81
```

4) The investigator must navigate a web browser to http://10.1.1.1:81 which will present a listing of files currently contained in the */etc* directory as demonstrated in Figure 3. The investigator may then simply save the file to the desktop as if it were any other file located within a web page.



Index of /

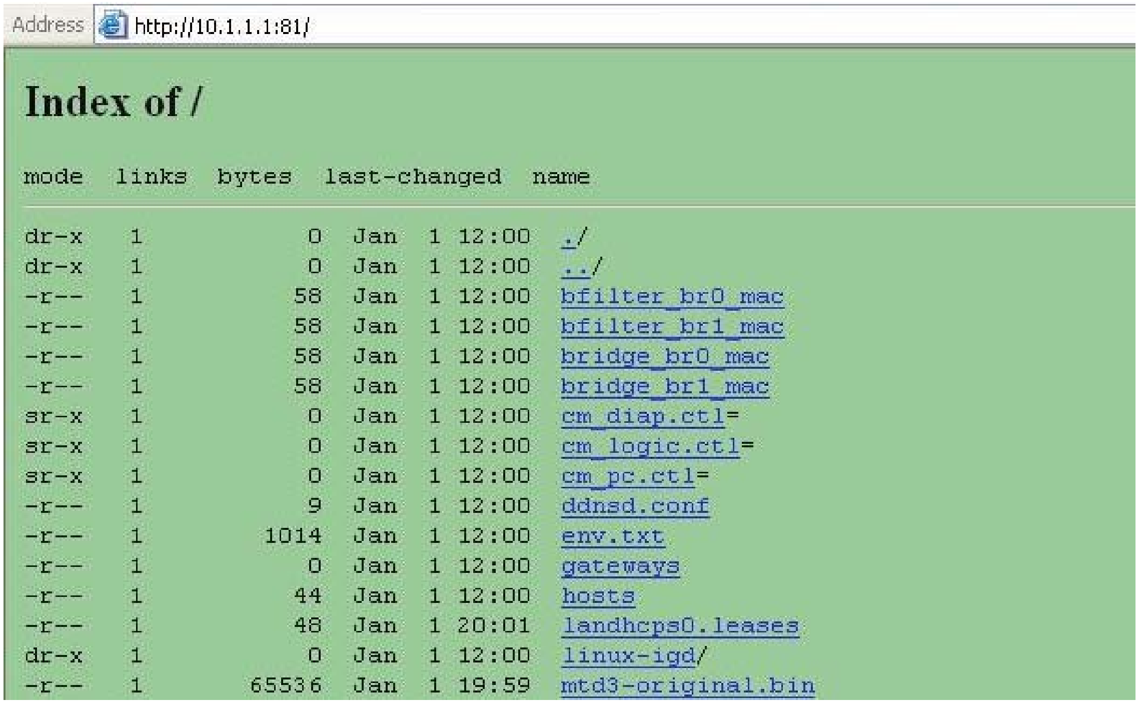
```
mode  links  bytes  last-changed  name

-r--   1       65307  Sep 22 10:27  config.xml
-r--   1        2714  Sep 14 14:00  dhcp-fwd.cfg
lrwx   1          13  Sep 22 10:28  dproxy.conf -> ./resolv.conf
dr-x   1          52  Sep 22 10:28  dropbear/
-r-x   1         180  Sep 14 14:00  firewall_start*
-r-x   1         120  Sep 14 14:00  firewall_stop*
-r-x   1          86  Sep 14 14:00  flush_firewall*
-r--   1         313  Sep 14 14:00  fstab
lrwx   1          19  Sep 22 10:28  gateways -> ../var/tmp/gateways
-r--   1           0  Sep 14 14:00  group
-r--   1          17  Sep 14 14:00  host.conf
lrwx   1          16  Sep 22 10:28  hosts -> ../var/tmp/hosts
dr-x   1          10  Sep 14 14:00  init.d/
-r--   1         259  Oct 22 08:17  inittab
```

Figure 4 Web browser access to configuration files

5) The second set of data to be acquired is the raw binary configuration file. The investigator must first terminate the existing HTTP daemon as this will not be used to acquire the subsequent data. During ADSL router boot time a small amount of RAM is set aside for the file system and user-accessible temporary storage. The Linux utility *dd* encompassed within the ADSL router permits an exact byte copy of data stored within flash memory. This data is automatically loaded and interpreted by the device's kernel during boot time and the data is used to create the default state of the router. The following command may be used to acquire the data of interest and make it accessible via the web browser.

```
# kill 132
# cd /var/tmp
# dd if=/dev/mtdblock/3 of=mtd3-original.bin
# thttpd –g –d /var/tmp –u root –p 81
```

6) As per step three, the investigator utilises the web browser to view and acquire the binary configuration data from flash memory. In this instance the file to be download is represented by *mtd3-original.bin.*



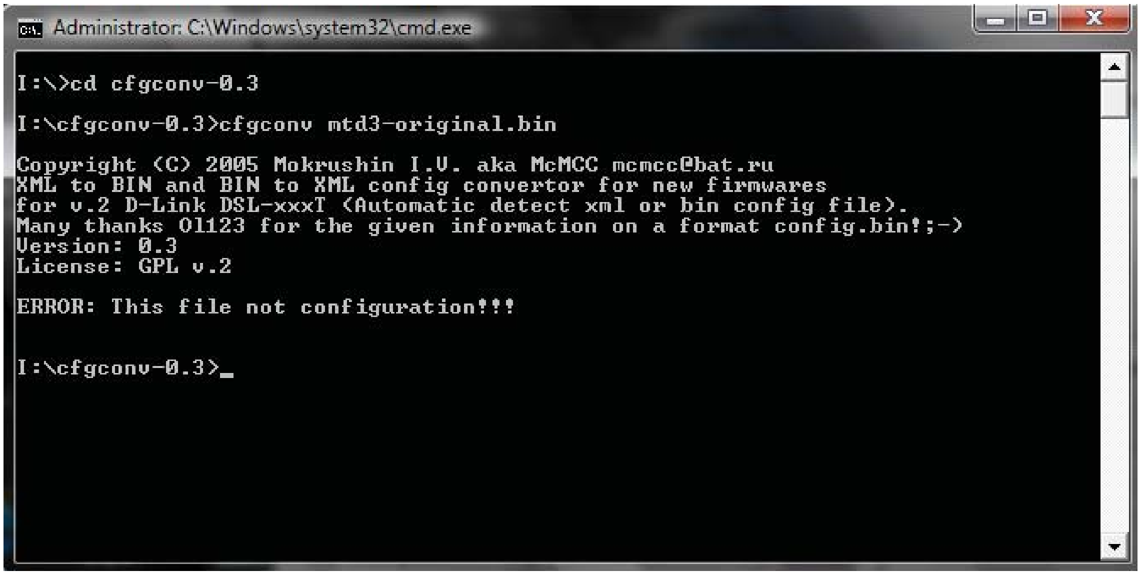
Address http://10.1.1.1:81/

Index of /

```
mode   links  bytes  last-changed   name

dr-x    1          0  Jan  1 12:00  ./
dr-x    1          0  Jan  1 12:00  ../
-r--    1         58  Jan  1 12:00  bfilter_br0_mac
-r--    1         58  Jan  1 12:00  bfilter_br1_mac
-r--    1         58  Jan  1 12:00  bridge_br0_mac
-r--    1         58  Jan  1 12:00  bridge_br1_mac
sr-x    1          0  Jan  1 12:00  cm_diap.ctl=
sr-x    1          0  Jan  1 12:00  cm_logic.ctl=
sr-x    1          0  Jan  1 12:00  cm_pc.ctl=
-r--    1          9  Jan  1 12:00  ddnsd.conf
-r--    1       1014  Jan  1 12:00  env.txt
-r--    1          0  Jan  1 12:00  gateways
-r--    1         44  Jan  1 12:00  hosts
-r--    1         48  Jan  1 20:01  landhcps0.leases
dr-x    1          0  Jan  1 12:00  linux-igd/
-r--    1      65536  Jan  1 19:59  mtd3-original.bin
```

Converting Data of Interest

7) In contrast to acquiring a readable *config.xml* file, the raw configuration data requires a conversion to be of readable form. A utility exists for converting raw binary configuration data into a readable xml based file. As demonstrated in Figure 4 the default raw-data file will not permit an immediate conversion. Through an empirical learning approach it has been discovered that the first 10,240 bytes are not configuration data and must be removed.

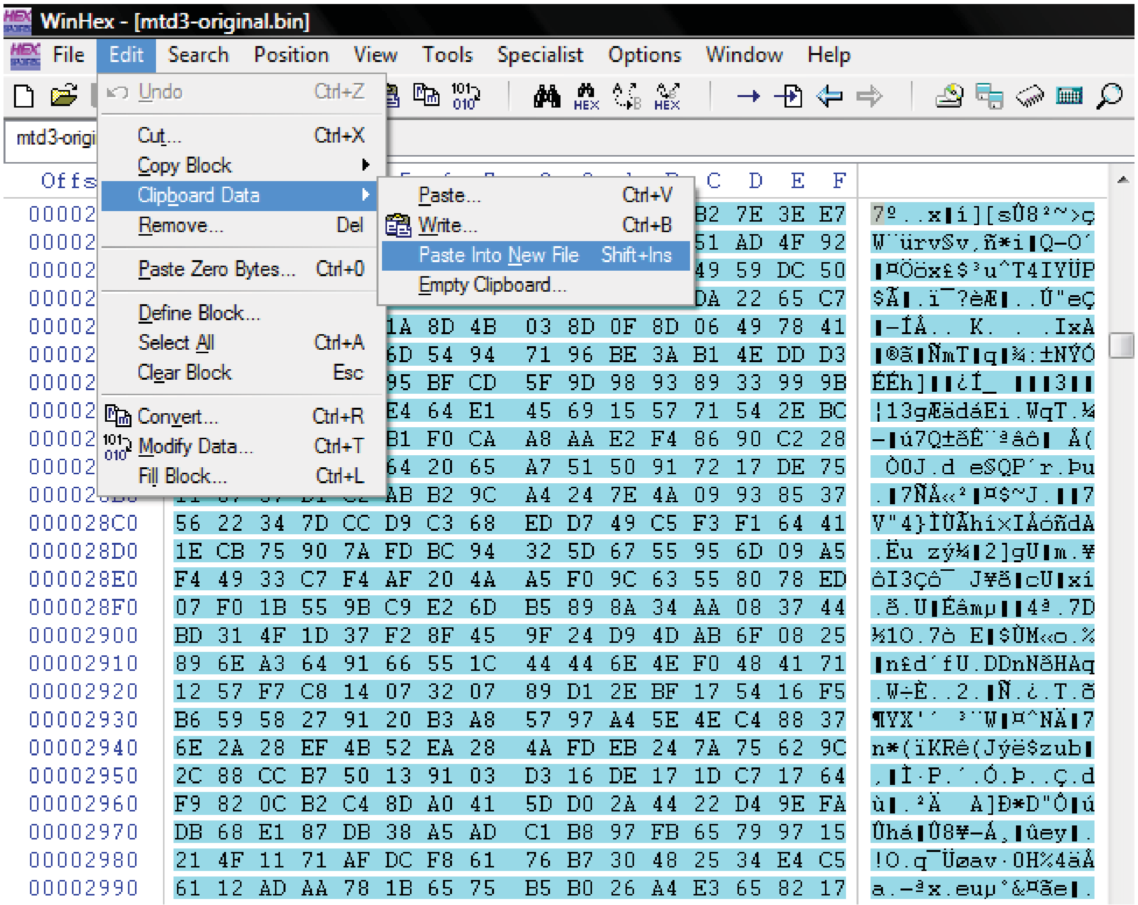


Administrator: C:\Windows\system32\cmd.exe

```
I:\>cd cfgconv-0.3

I:\cfgconv-0.3>cfgconv mtd3-original.bin

Copyright (C) 2005 Mokrushin I.V. aka McMCC mcmcc@bat.ru
XML to BIN and BIN to XML config convertor for new firmwares
for v.2 D-Link DSL-xxxT (Automatic detect xml or bin config file).
Many thanks Ol123 for the given information on a format config.bin!;-)
Version: 0.3
License: GPL v.2

ERROR: This file not configuration!!!


I:\cfgconv-0.3>_
```

Figure 5 Error while converting raw configuration data

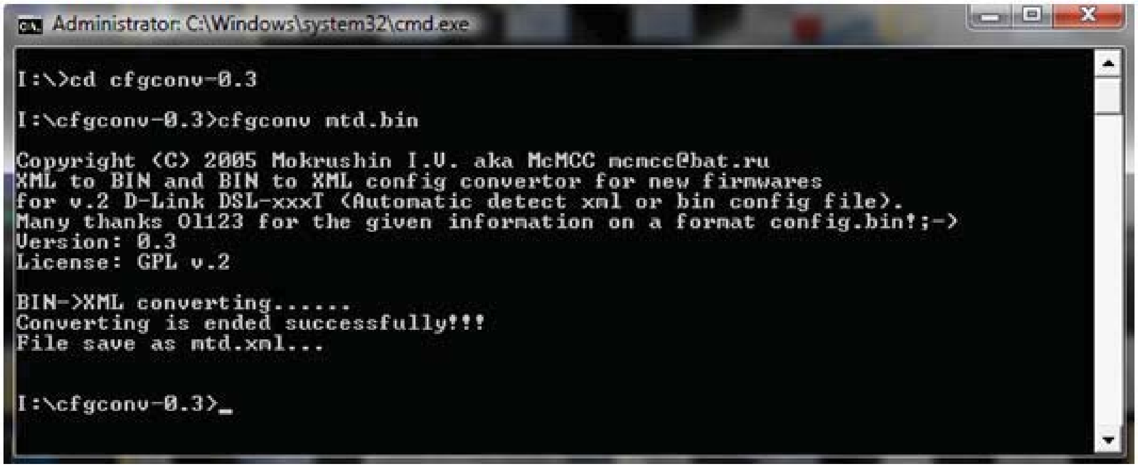
8) In order to effectively transform the raw binary data into a form recognised by the converter utility the first 10,240 bytes which are boot loader configuration variables must be removed. In the experiment demonstrated in Figure 5 the total block size consists of 55,296 bytes.



```
Offs                                    C  D  E  F
00002                                   B2 7E 3E E7   7º..x▌i][sÛ8²~>ç
00002                                   51 AD 4F 92   W¨ürvSv,ñ*i▌Q–O´
00002                                   49 59 DC 50   ▌¤Ööx£$³u^T4IYÜP
00002                                   DA 22 65 C7   $Ã▌.ï¯?èÆ▌..Ú"eÇ
00002                 1A 8D 4B 03 8D 0F 8D 06 49 78 41   ▌–ÎÅ.. K. . .IxA
00002                 6D 54 94 71 96 BE 3A B1 4E DD D3   ▌®ã▌ÑmT▌q▌¾:±NÝÓ
00002                 95 BF CD 5F 9D 98 93 89 33 99 9B   ÉÉh]▌▌¿Í_ ▌▌▌3▌▌
00002                 E4 64 E1 45 69 15 57 71 54 2E BC   ¦13gÆädáEi.WqT.¼
00002                 B1 F0 CA A8 AA E2 F4 86 90 C2 28   –▌ú7Q±ðÊ¨ªâô▌ Â(
00002                 64 20 65 A7 51 50 91 72 17 DE 75    Ò0J.d e§QP´r.Þu
00002                 AB B2 9C A4 24 7E 4A 09 93 85 37   .▌7ÑÅ«²▌¤$~J.▌▌7
000028C0  56 22 34 7D CC D9 C3 68  ED D7 49 C5 F3 F1 64 41   V"4}ÌÙÃhí×IÅóñdA
000028D0  1E CB 75 90 7A FD BC 94  32 5D 67 55 95 6D 09 A5   .Ëu zý¼▌2]gU▌m.¥
000028E0  F4 49 33 C7 F4 AF 20 4A  A5 F0 9C 63 55 80 78 ED   ôI3Çô¯ J¥ð▌cU▌xí
000028F0  07 F0 1B 55 9B C9 E2 6D  B5 89 8A 34 AA 08 37 44   .ð.U▌Éâmµ▌▌4ª.7D
00002900  BD 31 4F 1D 37 F2 8F 45  9F 24 D9 4D AB 6F 08 25   ½1O.7ò E▌$ÙM«o.%
00002910  89 6E A3 64 91 66 55 1C  44 44 6E 4E F0 48 41 71   ▌n£d´fU.DDnNðHAq
00002920  12 57 F7 C8 14 07 32 07  89 D1 2E BF 17 54 16 F5   .W÷È..2.▌Ñ.¿.T.õ
00002930  B6 59 58 27 91 20 B3 A8  57 97 A4 5E 4E C4 88 37   ¶YX'´ ³¨W▌¤^NÄ▌7
00002940  6E 2A 28 EF 4B 52 EA 28  4A FD EB 24 7A 75 62 9C   n*(ïKRê(Jýë$zub▌
00002950  2C 88 CC B7 50 13 91 03  D3 16 DE 17 1D C7 17 64   ,▌Ì·P.´.Ó.Þ..Ç.d
00002960  F9 82 0C B2 C4 8D A0 41  5D D0 2A 44 22 D4 9E FA   ù▌.²Ä  A]Ð*D"Ô▌ú
00002970  DB 68 E1 87 DB 38 A5 AD  C1 B8 97 FB 65 79 97 15   Ûhá▌Û8¥–Á¸▌ûey▌.
00002980  21 4F 11 71 AF DC F8 61  76 B7 30 48 25 34 E4 C5   !O.q¯Üøav·0H%4äÅ
00002990  61 12 AD AA 78 1B 65 75  B5 B0 26 A4 E3 65 82 17   a.–ªx.euµ°&¤ãe▌.
```

Figure 6 WinHex removal of boot loader configuration data

9) Having saved the new binary data file minus the first 10,240 bytes this now permits the configuration program to effectively translate the raw data into a readable xml file.



```
I:\>cd cfgconv-0.3

I:\cfgconv-0.3>cfgconv mtd.bin

Copyright (C) 2005 Mokrushin I.V. aka McMCC mcmcc@bat.ru
XML to BIN and BIN to XML config convertor for new firmwares
for v.2 D-Link DSL-xxxT (Automatic detect xml or bin config file).
Many thanks 01123 for the given information on a format config.bin!;-)
Version: 0.3
License: GPL v.2

BIN->XML converting......
Converting is ended successfully!!!
File save as mtd.xml...

I:\cfgconv-0.3>_
```

RESULTS

In order to determine the feasibility and forensic soundness of the acquisition a series of testing procedures were undertaken. The tests aimed at discovering:

- The default state of configuration data after it has been acquired.
- The state of acquired configuration data after a reboot had been performed.
- The state of acquired and ADSL router configuration data after a manual change was applied.

One of the limiting factors of acquiring data from ADSL routers is the fact that there is no cryptographic hashing software available on the device by default to check the integrity of the data before and after an acquisition has occurred. As a result this limits the soundness of the acquisition and instead relies on the precision of the method followed. To validate the integrity of the data an alternative approach had been taken.

- The ADSL router was reset to known default state.
- The *config.xml* file was acquired (Acquisition #1).
- The ADSL router was rebooted and a subsequent acquisition taken (Acquisition #2)
- Firewall rule sets and wireless settings were modified on the ADSL router and an acquisition was conducted (Acquisition #3).
- The same firewall rule sets and wireless settings were modified to the data of Acquisition #1 and an MD5 digest generated.

Table 3 Comparison of acquisition cryptographic hash digests

Data	**MD5 Digest**
config.xml (1st acquisition)	f6ccb1d8700281cd1f9afe368675bf90
config.xml (2nd acquisition)	f6ccb1d8700281cd1f9afe368675bf90
config.xml (modified on ADSL router)	0f26bb398fdbc1269232ebd4ce2bc780
config.xml (modified on workstation)	0f26bb398fdbc1269232ebd4ce2bc780

As demonstrated by Table 1 acquiring the configuration data after the device has been switched off and rebooted still results in the same md5 digest in both instances. Thus this shows that when acquiring configuration data the device can be switched of and examined at a remote location. Secondly, if a configuration change occurs on the ADSL router and the same manipulation of variables occurs by editing the original configuration data the same unique md5 digest will also be identified.

DISCUSSION

Interpreting the data acquired from the ADSL routers in question is not a trivial task. As it stands there is no elegant manner by which to interpret the data to identify key configuration data sets. Configuration data in ADSL routers function on a principle of utilising *states* to determine if the service or rule set is enabled and/or functioning. While examining the configuration data – Telnet by default is enabled on the majority of ADSL routers. In-turn the state of the service is given a numerical representation of 1. Alternatively, services inclusive of TFTP or SSH are by default disabled. As a result these services in the configuration data are represented by a numerical value of 0 signifying an off or disabled trait.

Utilising the acquisition method and interpreting the data and enabled services may allow the investigator to identify vulnerabilities and/or weaknesses that the device may have been susceptible to. The methods of acquisition described throughout the paper have been used to acquire data of interest from devices purchased on second hand auction sites throughout the Internet. Whilst the analysis of remnant data from ADSL routers still remains in its infancy the approach is beginning to prove and show signs of practicability. In addition, current acquisitions and analyses are demonstrating how consumers are utilising their broadband connection.

In 2009 a series of SoHo based ADSL routers were purchased and acquired in the effort to identify if consumers are in fact removing their identity and network configuration data from these devices. From the series of second hand devices purchased – none of the devices had the Internet Service Provider's account credentials removed. Whilst logging into the ADSL router may initially hide the contents of these confidential items – an examination of the configuration data can quite easily recover these credentials that were used to access the broadband connection. In addition the credentials used to access a broadband connection may also be used to access and modify ISP accounts.

Items of interest include ways in which the devices are being used over the Internet. Whilst downloading of media over the Internet is highly publicised as being illegal an examination of ADSL routers in question is showing that many of the second-hand devices identified had in fact enabled port forwarding of BitTorrent based rule sets. As a result it would be quite plausible to trace the device back to the original owner where further investigations could take place.

CONCLUSION

The research in progress continually demonstrates the importance and feasibility of acquiring data of interest from ADSL routers in a forensically sound manner. This paper has demonstrated two approaches by which data can be acquired with or without the co-operation of the owner of the device in question. This further demonstrates that an investigator can be granted unrestricted access to the configuration data on the device and use this to further analyse and interpret how the device was utilised.

The paper has only demonstrated one aspect of the ADSL router forensic process which could take plan on the device. Whilst other note worthy approaches do exist such as identifying if dynamic malicious processes are occurring – it was the aim of this paper to acquire the configuration data as a wealth of information is contained within these files alone. Future research will analyse the dynamic and volatile aspects of the forensics process and attempt to identify the methods by which the ADSL router could be removed for the scene of the crime whilst still maintaining data integrity.

REFERENCES

AR7 Firmware. (2009). AR7 Based Router Firmware. Retrieved March 13, 20009, from http://ar7.wikispaces.com/Firmware

Baume, T. (2009). Netcomm NB5 Botnet – PSYB0T 2.5L. Retrieved September 10, 2009, from http://users.adam.com.au/bogaurd/PSYB0T.pdf

Breeuwsma, M. (2006). Forensic imaging of embedded systems using JTAG (boundary-scan). *Digital Investigation, 3*(1), 32-42.

Breeuwsma, M., Jongh, M. d., Klaver, C., Knijff, R. v. d., & Roeloffs, M. (2007). Forensic Data Recovery from Flash Memory. *Small Scale Digital Device Forensics Journal, 1*(1), 1-17.

Brown, C. L. T. (2006). *Computer Evidence Collection and Preservation*. Hingham, MA: Charles River Media.

Hunt, S. R. (2009). New worm can infect home modem/routers. Retrieved October 11, 2009, from http://apcmag.com/new-worm-can-infect-home-modemrouters.htm

Magnus, N., & Gassmann, B. (2009). Psyb0t Attacks Linux Routers. Retrieved October 10, 2009, from http://www.linux-magazine.com/Online/News/Psyb0t-Attacks-Linux-Routers-Update

Paul, I. (2009). Nasty New Worm Targets Home Routers, Cable Modems. Retrieved September 22, 2009, from http://www.pcworld.com/article/161941/nasty_new_worm_targets_home_routers_cable_modems.html?tk=rss_main

Sajdak, M. (2009). *Remoterootshellon a SOHO classrouter.* Paper presented at the Confidence 2009, Krakow, Poland.

Szewczyk, P. (2009). ADSL Router Forensics: Methods of Acquisition*, Journal of Network Forensics, 1(1). 16-29.*

Szewczyk, P., & Furnell, S. (2009). *Assessing the online security awareness of Australian Internet users.* Paper presented at the 8th Annual Security Conference, Las Vegas, NV, USA.

COPYRIGHT