UNIVERSITY OF AMSTERDAM
SYSTEM AND NETWORK ENGINEERING

CYBERCRIME AND FORENSICS

Forensic Analysis of Consumer Routers

Students:
Kim van Erkelens
Esan Wit

June 1, 2014
Abstract

Routers are commonplace in modern age and are often the hub of
consumer networks. As a result of this, the router is a veritable treasure
trove of information for those who can read it. But research into routers
and analysis of what information may be recovered depend on model and
vendor. Compared to enterprise routers consumer routers are hardly in-
vested. As such this research examines what kind of traces are left
behind on a consumer router and what may be deduced from this. We in-
vestigate methods of access and extraction and areas of interest. Analysis
of extracted data can offer pointers for further and more specific inves-
tigation. Specific scenarios are tested and the effects on the router are
analysed.
1 Introduction

Nowadays consumer routers feature prominently in any household. These routers process any and all traffic that is send over home networks. As such these routers have access to vast amounts of information. It may be that the suspects’ computer or laptop is unavailable for forensics. In such a situation router analysis may offer clear insights in actions performed by the user when last online.

This research aims to find methods for extracting information from routers for use in forensic analysis. The analysis is a continuation of research into extraction of DHCP state information from home routers by Fiebig [1]. The scope however will be broadened to include other services offered by routers. The project is more in line with research described in [6] because direct terminal access to the router was available.

Research was done on a Netgear WNDR3700v4 home router; it offers 600N wireless and a 4 port Gigabit LAN for connectivity. This router was chosen because of its popularity and its features such as USB device sharing, which can be interesting for forensic analysis. Moreover, no previous forensic analysis of this router was found.

1.1 Research questions

The research question for this project is: How can forensic analysis be conducted on the Netgear WNDR3700 consumer router?

In order to answer this question the following sub-questions are set:

1. What are the different sources of information present on the router and how can they be accessed?

2. What is the volatility of these data?

3. What can be concluded from the data?
2 Related work

Consumer routers are a small topic of research in the area of router forensics. Most of the research has been done on the Cisco IOS system, and on networks and network devices in general.

Different approaches for acquiring data from consumer grade routers are described in previous research. Szewczyk performed multiple studies about this topic. He describes several interfaces through which data can be accessed. The easiest to access interfaces are the management webinterface and telnet or SSH, which can be accessed via the router’s Ethernet port. However, issuing commands on the router may alter its memory. A method for acquiring memory uses the Joint Test Action Group (JTAG) interface [5]. This interface is often present in embedded devices and allows for direct access of the device’s memory. A drawback of JTAG is that it is slow compared to the other methods [6].

Previous research by Fiebig has been done into analysis of DHCP data available on home routers [1]. The research demonstrates a method for imaging the device memory and the extraction of DHCP information. It proposes further research into incorporating additional information from other services provided by the router.

More general research regarding network forensics is conducted by Turnbull [7]. He described the sources of evidence that can be present on a wireless embedded device. He points out that the extraction process varies for each model or firmware revision.
3 Methodology

This section describes the methodologies used for accessing the router, acquiring data, and the scenario’s for generating traces on the router. Also, the type of hardware and firmware versions is mentioned.

3.1 Access

Access to the device was required for the acquisition of data. Various methods of access were available from which three were pursued. The primary access method was the web interface offered by Netgear genie. This console offers insights in the current state of the network as seen by the router. And although it is perfect to view the current state of the network it is useless for historical information. Additionally some services are inaccessible from the web interface and as such cannot be collected. For this reason other avenues were also looked at.

Many Netgear routers come with a telnet service running. This telnet service can be used to gain console access to the device from within the LAN. The telnet service needs first to be unlocked but thankfully Netgear has made the unlock utility available. For this project a Python port of the utility was used\(^1\). All that is required to unlock the telnet service is a connection to the LAN segment and the default username and password. The MAC address of the router is also required but that can be requested via ARP.

The last method investigated for access was via the onboard JTAG interface. The JTAG interface is a programmatic interface to the micro controller that runs the router software. The JTAG method was investigated because it would allow for the processor to be halted before extracting a copy of the memory. Such a copy would be uncontaminated by the acquisition method. Access via the web interface or telnet console would affect the working memory of the router and during the acquisition of state other data might be overwritten by the access method. It is here that the added benefit of the JTAG interface shows.

The router also offers a serial connection for access. However, this was deemed outside the scope due to time constraints. Although the data that were overwritten during acquisition might be a little less compared to telnet access, it would also be less practical with no clear benefit.

3.2 Data acquisition

Using the access methods described it was attempted to extract available information from the device. In particular it was focused on the extraction of DHCP, UPnP (Universal Plug and Play), DNS, and Samba USB file sharing information. This set of services was selected as being most interesting in a forensic investigation, because they may hold information that can be related to a suspect.

\(^1\)http://code.google.com/p/netgear-telnetenable/
The focus on DHCP was due to the importance of being able to place a system at a location. Being able to say something about who was likely doing what is important in any forensic investigation.

UPnP was investigated because it allows applications to request forwards from the router without user consent and because some applications don’t function without forwarded ports. This can be used to show that a specific program was likely running at some time even if the program has since been closed and the port released.

The widespread use of DNS by both applications and users made DNS interesting for analysis. If a log of DNS requests can be uncovered this may be used to construct a timeline visualising user activity.

Finally, because the chosen router supports sharing USB storage devices over the network it was investigated what traces remain behind from this action. Specifically it was researched if the attached USB device could be identified and if a user access log of sorts was available.

### 3.3 Scenario’s

Various scenario’s were setup in order to generate traces on the router. Before and after each step a memory dump and a copy of the file system was made. Then, differences and traces were searched for. The steps performed for each scenario are displayed in Table 1.

<table>
<thead>
<tr>
<th>Service</th>
<th>Steps performed</th>
</tr>
</thead>
</table>
| telnet  | 1. Setup a telnet connection via the LAN interface  
               2. Disconnect the telnet connection          |
| DHCP    | 1. Connect two computers to the LAN interfaces  
               2. Reboot the router while connected        |
| UPnP    | 1a. Start a peer-to-peer file synchronisation program  
               1b. Start a Torrent download               |
| DNS     | 1. Generate DNS requests by browsing the Internet |
| Samba   | 1. Attach a USB flash drive  
               2. Make a connection to the SMB share  
               3. Create a directory and text file  
               4. Disconnect from the SMB share and detach the USB flash drive |

Table 1: Performed scenario’s for each Internet service

For the telnet scenario, a firewall on the computer was configured that allowed telnet and netcat traffic only. This was done to prevent other traffic to end up in the devices memory.
3.4 Hardware and software

The Netgear WNDR3700v4 has an Atheros AR9344 CPU and 128MiB RAM. The router runs on a custom OpenWRT/busybox fork with Netgear genie. Although not advertised many Netgear devices come with a Telnet option available to gain access to the device. The firmware on the WNDR3700v4 used for the forensic analysis was 1.0.1.42. This version uses OpenWrt KAMIKAZE (bleeding edge, r18571) with version 1.4.2 of BusyBox.
4 Results

This section reports the findings of this research. First, obtaining access to the console interface is explained. Next, the results for memory acquisition via JTAG and telnet are described. Data acquisition for the different types of Internet services is outlined after that. Lastly, validation of these data is clarified.

4.1 Console access

The Netgear WNDR3700 has built-in telnet access, but requires a specially formatted packet to be sent to the telnet daemon before logging in is allowed. Several implementations and executables are available for this activation. Netgear provides a program for Windows only, which is called “telnetenable”. For Unix-based systems a C and Python implementation of the program can be found on-line. For this study the Python implementation was used because of its simplicity. Telnet access on the Netgear WNDR3700 could be enabled with a default username and password in the following way [2]:

```bash
python telnetenable.py <IP> <MAC> <Username> <Password>
```

```bash
python telnetenable.py 192.168.1.1 08BD43AAB071 Gearguy Geardog
```

4.2 Memory acquisition

Research was done into memory acquisition via JTAG and telnet. The results are described in the following sections.

4.2.1 JTAG

The preferred method for a memory acquisition of embedded devices utilises the JTAG interface. This method doesn’t alter the memory, but is more complex compared to other methods. It requires the device to be opened, the correct type of JTAG connector, and software that is specific for the architecture of the router.

OpenOCD\(^2\) was chosen as the debugger that would interface via JTAG with the device because of the wide range of supported devices and capabilities\[^4\]. However during the attempts of extracting information it was found that there was a missing configuration for the chip. As such OpenOCD could not properly connect and communicate with the JTAG controller. Other debuggers which did support the chip but were not found. As a result we were unable to extract a memory image using this technique.

\(^{2}\)http://openocd.sourceforge.net/
4.2.2 Telnet

Because acquisition via the JTAG interface wasn’t successful, another approach was taken in order to have a memory image for further analysis. In this approach the router was accessed via telnet. With the dd utility, which is provided by the operating system of the router, an exact byte copy of /dev/mem was made. This copy was sent over the network to the forensic computer using netcat:

root@WNDR3700v4:/# dd if=/dev/mem | nc 192.168.1.3 9000

Although the resulting binary file is 128MB, it cannot be said whether this contains the full contents of the physical memory. Also, part of the memory will be overwritten with this approach.

<table>
<thead>
<tr>
<th>mem1</th>
<th>mem2</th>
<th>mem3</th>
<th>mem4</th>
<th>mem5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3553334</td>
<td>3749386</td>
<td>3790576</td>
<td>3837833</td>
</tr>
<tr>
<td>3758036</td>
<td>2258036</td>
<td>2636854</td>
<td>2823923</td>
<td>2511119</td>
</tr>
<tr>
<td>0</td>
<td>2280038</td>
<td>2280038</td>
<td>2183060</td>
<td>2183060</td>
</tr>
</tbody>
</table>

Table 2: Differences in bytes between sequential memory dumps measuring the impact of connecting, acquiring and disconnecting from the router via telnet

In order to analyse how much memory was overwritten during acquisition, a separate test was run in which memory was copied multiple times from the running image and the difference between these images was analysed. Table 2 shows the differences in bytes between various memory dumps. These results illustrate that there is a small impact of connecting and extracting data from the router. The collected data shows this is between 2 and 4MB. However it should be noted that there is also some memory reuse going on. This can be seen from the fact that only 0.1 to 0.4MB more is different when looking between various memory images where one is skipped.

It should be noted that in this example it was impossible to measure the impact of enabling the telnet service since it’s impossible to extract a copy of the memory without first connecting. Although network activity was minimised during these tests running processes on the router might have added to the difference between memory images unrelated to the acquisition method; however this can not currently be confirmed.

4.3 Data acquisition

The interesting information is stored on a tmpfs file system. Tmpfs is a temporary file storage that appears as a mounted file system, but is stored entirely in memory. As a result one of the properties is that the tmpfs is ephemeral. Compared to normal devices this has the downside that any logs contained on
the router are lost when the device is shut down or rebooted. On boot a new
tmpfs is created and mounted.

Due to this it is possible to extract the information from the memory image. However no tool was found that could extract the file system from the memory image as extracted from the device. Due to this it was required to use live forensics to extract log files of interest.

In contrast to [6] data of interest was acquired using netcat instead of thttpd. This allows for less interaction with the router, and thereby assuming that less of the memory will be overwritten.

4.3.1 DHCP data extraction

DHCP log entries can be found in /var/log/log-message and /var/log/messages. The latter is more useful in a timeline analysis, because it contains absolute timestamps instead of relative ones. Both files contain IP addresses and MAC addresses for the DHCP leases. The two computers connected to the router resulted in the following entries in /var/log/messages:

```
```

The DHCP information in the log files wasn’t consistent. When the router was rebooted while the devices were connected no log entry was created. On lease renewal the first entry mentioning the DHCP lease was created in the logs. When a new device was introduced the lease was assigned and logged accordingly.

Changing the MAC address of an already attached device does not result in an update in the log files. The connectivity was maintained but the action was not logged anywhere.

4.3.2 DNS data extraction

During the research specific attention was paid to the possibility of extracting cached DNS queries from the router. The router uses dnsmasq 2.39 as a caching resolver and holds a cache of 150 items by default. This cache resides in a circular buffer within the program memory. Attempts were made to extract this data based on carving the structure used by dnsmasq however the structure used by dnsmasq is decoupled. That is to say the cache entry and the hostname and IP information are all separate blocks which are stored somewhere within the application memory of dnsmasq. The domain name is stored in a separate struct which is allocated at request time. The cache entry itself is recycled and merely uses pointers to both the domain name and the response information.

As a result we were unable to devise a method to extract this information. One can find domain names in the memory image but no method was found to link these domain names to their source.
It should be noted that even if the full cache can be reconstructed the software does not track when or who the request initially send in the cache. Using the cached TTL an approximate request time may be determined.

4.3.3 UPnP data extraction

Two sources for UPnP data were found. A log entry is added to /var/log/messages and log-message. This entry only mentions a new NAT rule is added, but doesn’t give any further specification. The source host is identified by its IP address only.

UPnP event in /var/log/messages:

```
[UPnP set event: add_nat_rule] from source
  192.168.1.5, Wednesday, May 14, 2014 10:08:20
```

When such a log message is found, the corresponding firewall rule can be shown with iptables:

```
ACCEPT udp -- anywhere 192.168.1.5 udp dpt:5353
ACCEPT tcp -- anywhere 192.168.1.5 tcp dpt:35000
ACCEPT udp -- anywhere 192.168.1.5 udp dpt:35000
```

This information was found on the router only for one of the two programs tested, namely the file synchronisation program. The Torrent download didn’t leave any traces behind on the file system.

4.3.4 USB drive data extraction

When entries about USB devices are found in /var/log/messages, further investigation can be done in other log files. The messages log file contains an entry when a USB device is attached and detached. The Samba share that is started is logged into /var/log/samba/log.nmbd. More interesting though, are log entries about the computer that was connected to the share. Computers that connected to and disconnected from the share are logged to /var/log/samba/log.smbd. In this log file the host is identified by its hostname and IP address. The serial number of the USB flash drive was not found on the file system, but it can be printed with the dmesg command.

USB attached and detached (/var/log/messages):

```
[USB device attached] The USB storage device SanDisk SanDisk Cruzer (C is attached to the router,
  Tuesday, May 13, 2014 14:50:03
[USB device detached] The USB storage device SanDisk SanDisk Cruzer ( is removed from the router,
  Tuesday, May 13, 2014 15:15:12
```
Information about the computer that was connected to the share (/var/log/samba/log.smbd):

[2014/05/13 15:06:20, 1] smbd/service.c:
    make_connection_snum(950)
    desktop-07 (192.168.1.3) connect to service
    USB_Storage initially as user root (uid=0, gid=0)
    (pid 2985)

[2014/05/13 15:14:56, 1] smbd/service.c: close_cnum
    (1150)
    desktop-07 (192.168.1.3) closed connection to
    service USB_Storage

USB flash drive serial number (dmesg):

usb 1-1: new high speed USB device using ath-ehci and address 2
usb 1-1: New USB device found, idVendor=0781, idProduct=5530
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1: Product: SanDisk Cruzer
usb 1-1: Manufacturer: SanDisk
usb 1-1: SerialNumber: 194211116F124A55

4.4 Data validation

The log files described in the previous sections can’t be correlated entirely. The fact that log information for DHCP leases is incomplete, makes it hard to relate it to other log entries. But when DHCP information is present, it is highly likely that IP addresses are for one specific host that can be uniquely identified by the MAC address in this entry. Then, log entries containing this specific IP address can be linked to one MAC address. Since MAC addresses can be spoofed, this is only an indication for the presence of a device.

For checking the integrity of the data, md5sum is available on the router. Unfortunately no shasum variant is present. That means that checking the file integrity before and after acquiring data can be done with MD5 only.
5 Conclusion

The central research question reads: *How can forensic analysis be conducted on the Netgear WNDR3700 consumer router?* Three sub-questions are set in order to answer this question.

**What are the different sources of information present on the router and how can they be accessed?**

The Netgear WNDR3700 runs a Unix based operating system. Therefore log files are stored in the well-known directory `/var/log`. The log files contain DHCP, UPnP, and USB events among other system events. A separate log file for Samba shares exist. Furthermore, several Unix-tools are available for acquiring additional data of interest. Among those are `dmesg` for finding information about connected USB devices and `iptables` for UPnP NAT rules.

Because the Netgear router has a JTAG interface, this is the preferred method for forensic analysis [1]. However, this would require tooling to support the chip configuration which was lacking at the time of this research.

Some of the information however has to be accessed via the console because it cannot be easily extracted from memory. Telnet access can be enabled for that purpose. Then, access to `/dev/mem` and the files on the system is available. Using telnet to extract an image of the memory is possible but overwrites at least 2.5MB of the memory possibly more for the activation of telnet.

**What is the volatility of these data?**

As much from the data is recovered from the log files stored in the tmpfs directories this data is stable as long as the device remains powered on. However, other data such as traces from the attached file system appear to reside in the unallocated memory space and as such may be overwritten at any moment.

Port mappings requested via UPnP remain in the `iptables` for up to 40 minutes after request or renewal. As such, these will be accessible for, at most, 40 minutes after the requesting application stops.

CAM tables are updated when the link state changes and their history is not recorded. This means that unless the system used one of the other services provided by the router it will not be possible to say if the device was ever connected to this router, at least not from the router’s perspective.

**What can be concluded from the data?**

The data available do not allow an accurate timeline to be constructed. It is possible that connections are missed and actions are not logged when using the services provided by the router. However, when traces are found in either logs or memory it is a strong indicator that the device was at some point connected to the network, but since MAC addresses can be spoofed this is only an indicator. If a device is connected to the router and accesses the attached USB device it can’t be concluded if this is the device auto-connecting to a network share or an explicit action taken by the user.
As shown various options are available to extract specific information from the system. Information ranging from connectivity to traces of traffic send over the router. Using router forensics opens possibilities for investigators to construct more accurate timelines. The recovered information may be used to extend or offer new insights when correlated with data of other devices in the network.
6 Further research

This research has shown various information which is available on consumer routers. Only limited support is available in extracting this information in a clear and concise manner. As such tools need to be developed which can accurately extract specific information. An example could be a tool which carves the memory of the router in order to extract the DNS cache maintained. This cache can then be used to establish a profile of domains visited even if traces were removed from the device visiting them. Although taking into account that there is a 150 default limit it might only be useful in retracing steps for a few hours at most before the entire cache is overwritten.

Other avenues exist in scraping the memory for wifi beacon frames and using these as an indicator of device presence. Retention and volatility would have to be investigated for this device.

Additionally it may be possible to extract much of the information whilst bypassing the live system. This would possibly require further research into both the extraction of memory via JTAG bypassing the processing chip and more detailed memory forensics. Extraction of the tmpfs filesystem should be possible from a memory image thus negating the need for live forensics.

Currently the architecture of the router is not supported by volatility which was thus not used for analysis. Improvement of automated tools could enable to extract more information from routers which may prove useful for analysis and correlation. The performed research was executed manually and geared to certain services. Using an extensive library as provided by a forensic toolkit would allow for broader results which may be used to create more insights.

\footnote{https://code.google.com/p/volatility/}
References


Contributions

<table>
<thead>
<tr>
<th>Section</th>
<th>Kim</th>
<th>Esan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>1 Introduction</td>
<td>Adjusted</td>
<td>Initial</td>
</tr>
<tr>
<td>1.1 Research questions</td>
<td>Initial</td>
<td>Initial</td>
</tr>
<tr>
<td>2 Related work</td>
<td>Initial</td>
<td>Adjusted</td>
</tr>
<tr>
<td>3.1 Access</td>
<td>Adjusted</td>
<td>Initial</td>
</tr>
<tr>
<td>3.2 Data acquisition</td>
<td>Initial</td>
<td>Rewritten</td>
</tr>
<tr>
<td>3.3 Scenarios</td>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>3.4 Hard/software</td>
<td>Initial</td>
<td>Extended</td>
</tr>
<tr>
<td>4.1 Console access</td>
<td>Initial</td>
<td>Initial</td>
</tr>
<tr>
<td>4.2.1 JTAG</td>
<td>Adjusted</td>
<td>Initial</td>
</tr>
<tr>
<td>4.2.2 Telnet</td>
<td>Initial</td>
<td>Rewritten &amp; Extended</td>
</tr>
<tr>
<td>4.3 Data acquisition</td>
<td>Initial</td>
<td>Extended</td>
</tr>
<tr>
<td>4.3.1 DHCP</td>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>4.3.2 DNS</td>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>4.3.3 UPnP</td>
<td>Initial</td>
<td>Extended</td>
</tr>
<tr>
<td>4.3.4 USB</td>
<td>Initial</td>
<td>Adjusted</td>
</tr>
<tr>
<td>4.4 Data validation</td>
<td>Initial</td>
<td></td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>Initial</td>
<td>Adjusted &amp; Extended</td>
</tr>
<tr>
<td>6 Further research</td>
<td>Initial</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Contributions per member