Project Proposal B546

This is a project proposal for the study of access routers and various security threats faced by this device. The team of students working on this project includes Jayendra Khandare (jkhandar@indiana.edu) and Govind Mishra (shremish@indiana.edu). This proposal is submitted as a course requirement for CSCI - B546 Malware: Threat and Defense conducted by Prof. Xiaofeng Wang.

1. INTRODUCTION

A SOHO router is a Small office/Home Office router which is used by many people. It is connected to two or more data lines. The basic functionality of this device is to forward data packets to the required destination. When a data packet is sent from a system, the router reads the network address information in the packet to determine the ultimate destination. Then, using information in its routing table, it forwards the packet to the next network. They don't have to worry about too many calculations, hence they lack in other resources.

The main purpose of a router is to connect intra-network to the Internet, and forward data packets destined for a device. These mechanism and other important functions are supported and governed by the firmware. The firmware is also responsible for routing table construction, packet validation, packet lifetime control, etc. This study is catered to find security flaws in this device and suggest improvements for the same. As firmware is the most critical component in router, we'll also work with various open-sourced version of it. Finally, we want to provide with some firmware update and other measures which could be used to make this device more secure.

2. MOTIVATION

Most of DDoS attacks slow down communication which can be taken as a sign. SYN flood won't allow the connection to be established. In short, most of the attacks show some sign. But, if the connection is working as it should and still someone is able to access the data, there is no way of knowing that.

Also, attacks on routers have become very frequent now. For example, "Flytrap" is a custom-made firmware used by government agencies to modify router firmware and sniff on sensitive user data. Also, a project code-named as "Cherry Blossom" can hack into routers and make changes despite having strong passwords.

They also usually have a running instance of "Busy Box" mechanism all the time to support various commands required for functioning.

We can only apply preventive measures in a case like this. A study revealed nearly 79% of American home networks have default passwords, which is an alarming number. The people don't know this threat yet and I can't see it happening anytime soon. The best option here will be to provide them with firmware update.

3. WHAT IS ROUTER FIRMWARE AND HOW SECURE IS IT?

Routers are provided with a default firmware by the ISPs. The firmware within these devices provides routing, control and monitoring functionality coupled with mechanisms to ensure a secure and reliable network. End-users are typically reliant on manufacturers to provide timely firmware updates to mitigate known vulnerabilities.

There are many router providers available nowadays which allow their own router firmware as well as custom-made firmware to be installed on the device. The major players here are Cisco, Linksys, etc. which maintain their firmware, but not as frequent as they should. Although 10 bugs were discovered last year in 25 different Linksys router models by IOActive's senior security consultant Tao Sauvage and researcher Antide Petit, they haven't released a significant update which takes care of these issues. These bugs could make the device an accomplice in

DDoS attacks. It also allows a leak of data from the router which can be used by adversaries.

An investigation was undertaken to identify the underlying software components used in the firmware of currently available, SoHo network devices. Firmware from 37 devices was deconstructed to identify potential security issues; in each instance, the firmware images were found to include vulnerabilities, obsolete software and out-of-date operating system components. 95% of the deconstructed firmware was based on Linux. The Linux kernels identified were typically discontinued and are no longer actively maintained.

The firmware can also be manipulated and installed on devices without any obstruction as confirmed by Wikileaks' Valut7 documents where they provide proof that government agencies are working on "Cherry Blossom" project which changes the router firmware without letting the user know to serve snooping activities.

Weak router security can lead to various attacks such as DDoS, SYN Flood, etc. The always active "Busy Box" can cause attacks like "Shellshock" in which an adversary can download malicious content without the knowledge of the user which leads to further attacks. It also allows access to install various rootkits, the most recent being "Umbreon" which creates a backdoor in the system and allows the adversary to use it whenever he likes and for whatever purpose he wants.

Our contribution through this study will be finding a way which helps avoid these attacks or at least detect and postpone them. This study will also work as a stepping stone for the possibility of finding better ways to tackle this problem in future.

4. SECURING THE DEVICE

For the major concerns mentioned before, we have come up with 3 unique ideas which can put a stop to these exploitations without disturbing the flow of day-to-day working of a non-technical user.

To apply these improvements to the routers currently in work, all we need to do is provide this mechanism as a firmware update / patch. In case of routers, the 'Automatic Firmware Update' option is (most of the times) set to 'ON'. Hence, it won't be a hassle for the end user.

A) Dual Layer Protection

As mentioned earlier, it is very easy to get access to router setting page and 79% users don't change their passwords. This makes it easy for the adversary to hack into such a system.
Whenever an adversary tries to install his version of custom-made firmware in a router, we want to deter that process by adding one more layer of verification. Most of the times, the users are not that interested in installing some custom-made firmware if their system is working just fine. Hence, it is safe to assume that only users who have a technical background might be interested into this and that user base is limited.
In this case, we can ask the ISP to take responsibility of this task or we can use some other mechanism for this step. Once the second verification is done, only then anybody can do significant modifications.
After performing a thorough search on the web and on the homepage of some Internet Service Provider we found no ISPs provide a functionality like this. Also, this technique was not suggested by anyone before. A possible reason for this could be the difficulty to implement or cost of maintenance.

B) Router Broadcasting

The other approach to tackle the problem at hand is to notify user or ISP about malicious activity. We are not sure about the feasibility of this approach at this stage, but we would like to dig deep into it and confirm the possibility of this mechanism for further research. If possible, we'll also try to send the access log to a specified ISP server location.

C) Restricting Busy Box functionalities

"Busy-Box" is the tool being used in most of the routers which is a compilation of many basic commands required for performing multiple actions. An instance of this tool is always running on routers. Most of the rootkit attacks happen because of this

mechanism. We are hoping to restrict it whenever it is performing any malicious activity. Also, keeping a running instance alive all the time is not such a good idea. A scenario is possible where we are ending the instance of this mechanism, only allowing it when it is required and make it function only for that requirement.

5. IMPLEMENTATION STRATEGY

The first task for us is to find a suitable and easy to understand third-party firmware which is freely available to reverse engineer the process of how a device can be infected. For that, we have decided to focus on open-source projects like OpenWrt, DebWRT, HyperWRT, etc. The source code is freely available at their project repositories.

After deciding on the type of firmware to study, we would use some open-source emulators like Qemu, Dynamips or network simulators like GNS3 to simulate router and network architecture which will let us study various aspects of router processing and tasks handled by a router.

At the same time, we will research about Busy Box and commands supported by it. Once we identify the commands which a user doesn't frequently needs and are used by adversaries mostly, we'll put these commands in an encapsulated setting. The access to this setting will be conditional and will require user permission.

Later, we will only focus on finding and implementing a robust and implementable dual authentication feature. This is a critical component of the study because it directly affects the custom-made firmware installation tries. If this works as predicted, most of the attacks can be identified and deterred. At this stage, we are not sure about how ISPs will respond to this method. Hence, for the sake of the study, we will limit ourselves to hard-coding a specific access code in the firmware. As our knowledge of the firmware source code and mechanism is limited, this should suffice. Meanwhile, if we find a better alternative, we will go for that.

If possible, we will also try to install the malware (modified by us for this study) in an actual device to verify whether the findings of this study can be considered as a generic solution or model-specific. As far as I know, most of the devices support third party malwares and have almost 95% of Linux based components. Hence, it is possible to turn out to be a generic solution. But without verifying it with multiple models, we cannot guarantee this claim.

6. EVALUATION PLAN

After the successful modifications in the router firmware, we will either simulate or install the firmware on an actual router which supports custom-made firmware. As most of the devices in use don't have strong passwords, we will imitate the same. This way we can correctly provide environment required for possible router attacks.

The next step will be trying to perform rootkit attacks on the device. Rootkit attacks don't have too many requirements; hence they are easy to conduct. This also helps us determine which commands can be used for malicious purpose.

The final stage will be trying to install a malicious firmware on the device which mistreats or drops the data packets. If this installation is successful, it basically means that our custom-made firmware lacks something which cannot stop such an attack. Here, we will try to modify it further till it rejects the new installation or ask the person for specific access code. When the person trying to install fails to provide a correct access code, the router either contacts ISP (or other specified mechanism) or blocks the access by restarting or other method.

7. TIMELINE

Time	Task
03/09	Proposal Submission
03/10 - 03/16	Firmware Study
03/17 – 03/23	Device simulations
03/24 – 03/30	Restricting Busy Box
03/31 – 04/20	Working on dual authentication
04/21 – 04/25	Clean-up and documentation
04/25	Project Submission and talk