Your Router, Plausible Home to a Stealth Rootkit?

William Pitcock
NodeRebellion
August 2006

With later updates in March 2009:
William Pitcock
Dereferenced Technologies, Ltd.
http://dereferenced.org/

Copyright Notice

Terminology

- **NAT** – Network Address Translation, a form of transparently proxying connections from the Internet to a private LAN.

- **DMZ** – Demilitarized Zone, a segment of a LAN which is not behind NAT. Hosts in the DMZ should have extra security precautions applied.

- **Router** – In the scope of this paper, a router is actually a NAT appliance. NAT appliances allow multiple computers in a small business or home to share a single broadband connection by using Network Address Translation methods. Many NAT appliances run Linux, which this paper centers around.

- **Dropper** – A dropper is a small piece of executable code which downloads a larger piece of executable code from a distribution site and executes it on the local machine. It may also optionally install the executable code. Most malware is installed through droppers.

Introduction

This paper describes a feasible method of attacking a NAT appliance ('router') through a two-staged attack vector. The first attack vector may be any methodology you choose. We assume you already have root (or equivalent) access to a PC.

This paper will outline an attack strategy for gaining a shell on a router and installing malware, as well as explain why installing malware on the router is the most ideal exploit strategy.

Why Attack The Router?

Attacking the router will enable you to monitor network activity with a much higher level of stealth. As most people think the router is a dumb device which simply does NAT translation, it will not be considered a device with a high security risk. Most intrusion analysts at this time will not even consider the router as the place where the malware is hiding.

Further, most people will keep the router on 24/7. Most people shut down their PCs in the evening before they go to bed, or when they leave the office. Also, since the malware has ultimately wound up on the router, you can monitor all network traffic with a decreased likelyhood of setting off any intrusion alarms. Infact, the highest level of risk is getting the dropper onto a host PC to launch your attack against the router with.

Basic Plan

The basic attack plan is something along these lines:

1. Lure the user into installing the stage1 malware dropper. Usually this can be done through cute flash banner ads. Exploitation of pop culture is an excellent strategy here, consider offering your dropper as a "porn download accelerator" and distributing it on various peer-to-peer networks.

2. Once the stage1 malware dropper is on the PC, it should download a more complicated executable from a distribution site. This is considered to be the stage1 malware. The dropper should execute the stage1 malware, but NOT install it or make any changes to the system.

3. Once the stage1 malware is running in RAM, one could take several strategies. The first, would be to outright bruteforce the router, trying many different shellcodes and passwords to get access. The second would be to outright ask for the router username and password if you are disguising your malware as a "network accelerator" tool.

4. If the stage1 malware succeds in getting a shell on the router, it should use the router to download the stage2 malware from the distribution site. The stage2 malware, being the real malware, would do whatever you wanted it to do. Ideally, it would install itself so that reinfection is not necessary. At this point, the stage1 malware should terminate itself.

Attack Components

Lets look at the attack components here, so we understand the attack.

- **Stage 1 Dropper** – the stage1 dropper is a small file offered through P2P or on a website. It should look enticing, to make the user want to download and run it.

- **Stage 1 Malware / Stage 2 Dropper** – to make things clear, the stage1 malware is actually the dropper for the stage2 malware. It's intended to go away once the stage2 malware is installed. The stage1 malware may ask information about the network setup, such as the router administrator username and password. This would be presented under the pretense of "network optimization". The stage1 malware may stay around though to ensure that the router remains infected.

- **Stage 2 Malware** – the stage2 malware is the actual malware installed into the router appliance. The whole point of this attack is to get malware onto the router without the user noticing.

- **Shellcodes** – Shellcodes are used to access a shell on the remote system. Shellcodes exploit security holes in components of the router's firmware to get a root shell on the device. Some routers provide SSH and Telnet interfaces and do not require this, however.

Problems With This Form of Attack

Now that you understand the components of this attack, lets discuss the problems with it.

- **Ramdisks** – Most router appliances use an in-memory filesystem for the runtime session. This means that reinfection may be necessary to ensure the malware remains on the system. Ideally you would want to get rid of the stage1 malware once the stage2 malware is installed. However, you may be able to write the malware to flash with some effort.

- **WAN Traffic Monitoring** – While the LAN might not be able to detect when your malware calls home, the WAN will still be able to. This means that ISPs could take action to notify the owner of the router that it is infected.

- **Disassembly of the stage1 dropper / malware** – Since the stage1 dropper / malware are on a PC, somebody who is curious as to why the malware does not actually optimize his Internet connection may disassemble your software, possibly exposing your plans.

- **Firmware Upgrades** – If a user notices a problem with his router, he may upgrade his firmware. This would cause your malware to be erased. As a result, your malware should not cause any harm to the user's experience, or it may be exposed (although the user may not realize it.)

Conclusion

Gaining access to a router without the end user noticing is an easy task. Getting configuration information from end users who have been duped into running your malware is also an easy task. As long as all of the problems discussed are handled, a rootkit on a NAT appliance could remain permanent, and be fully undetected.

This attack can be detected by monitoring the traffic between the WAN and the router, however most people do not possess the capability of doing this.

ISPs may be able to help defeat this form of attack by inspecting traffic from their users, but this may result in public backlash from privacy advocates. Additionally, traffic inspection may be illegal in some jurisdictions.

I feel (both in 2006 and now) that this form of attack has advantages over the current methods, and will be used by blackhat hackers in the future to steal personally identifying information. As a result, I think that router manufacturers should improve the security in their firmware to make this a harder process. Right now, most Linux-based routers ship unnecessary tools in their firmware such as wget, the busybox ftpget applet, telnet and tftp. These programs are unnecessary for proper operation of a router and simply make it possible to enable malware like the one described in this paper. If routers had properly locked down firmware, then this attack would be much harder than it presently is.

As a result, I have published this paper (twice now) with the intention of exposing this problem so that router manufacturers take a more proactive approach to security. As of 2009, a successful “router bluepill” has been discovered by the DroneBL team and an earlier, more limited test version was discovered in December 2008 by Terry Baume. In fact, in 2009 it has gotten *worse*: *some routers and modems themselves can be exploited from the WAN*, not just the LAN. This could cause giant worms and botnet armies of routers being formed exclusively for the purpose of harvesting personally identifying information.

We are presently in a security crisis, that must be solved in order to maintain a secure Internet for the purpose of e-commerce and maintained privacy. The consequences of failing to do so are grave.