**Case Analysis of Firmware Vulnerabilities and Exploitation**

Joshua Main-Smith

Table of Contents

**Introduction3**

**Case Description3**

**Case Investigation4**

**Recommendation5**

**References7**

**1. Introduction**

Firmware is a topic most people aren’t familiar with, even though it plays a vital role in running your device. Firmware is a fundamental, lower-level device powering anything from your WiFi router, television, laptop computer, mobile phone, printer, and much more [1]. It is also equally important in making sure that it stays up to date and knowing the risks associated with accessing the firmware using unsecure interfaces, such as web enabled firmware (a practice that’s beginning to become more common) [3]. The consequence of neglecting important firmware updates can be seen in the following case example where A. Cui, M. Costello and S. Stolfo exploit a vulnerability found in the firmware of a LaserJet computer. Then, we’ll go over the method used in studying this case example. We will then finish it off by discussing a few examples of firmware vulnerabilities being used in the real-world and some recommendations in updating and configuring firmware to be more secure.

**2. Case Description**

For the first case study, A. Cui, M. Costello and S. Stolfo chose the HP LaserJet printer for their device, which sold over 10 million units in just the year 2010 making the HP family of computers a popular option among consumers. The authors found that by analyzing a standard update dump file for the HP LaserJet (model P2055DN), they were able to reverse engineer the embedded system and find critical design flaws. Using these design flaws, the authors were able to send executable requests to the printer unhindered in the span of a minute and a half. This is significant because executable files can be used to take administrative control of a system when granted permission, bypassing the firewall. With this being accomplished in just a couple of minutes the attack can go unnoticed by most users [2].

**3. Case investigation**

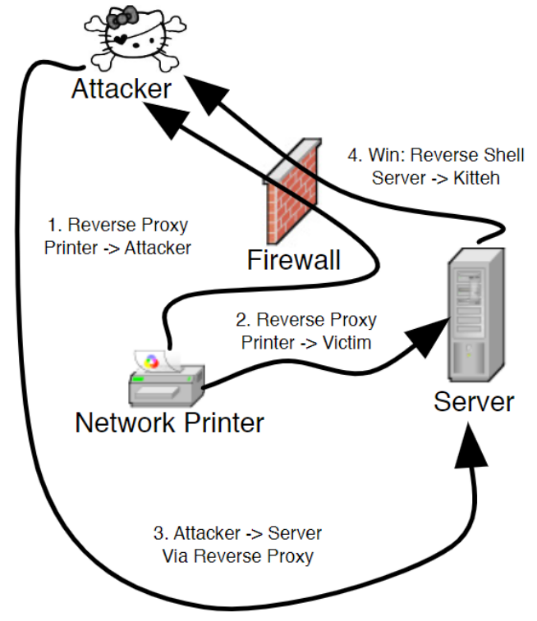
The authors A. Cui, M. Costello and S. Stolfo began by analyzing the binary file provided from the dump file discussed above. From the binary file, the authors were able to determine that the file is either encrypted or compressed (because there were no indicating factors from within the file that it executes any typical printing functions). It was determined that it was compressed due to the way the data was structured. From there, the authors were able to read the header files revealing the structure of the way the firmware is setup. This information can aid in understanding the structural components of the hardware, which the authors used when analyzing the hardware components of the printer such as the breadboard and the chips and wires attached. One chip that the team found had publicly available datasheets, which the teams used in determining the boot order of the system, which revealed several known executable vulnerabilities. When exploiting these known vulnerabilities, the authors were able to successfully gain administrative privileges of the system, for which they could then pivot into other critical components, such as a server with personal documents. As can be seen in figure 1 [2], an attacker is able to use a compromised computer as a tool in gathering information about the victim, bypassing the firewall by delivering a malicious package to the device the attacker has in interest.

Figure 1. Method a potential attacker uses in gaining unauthorized entry.

An important note in this process is that the methods described here can be applied to many other systems, as the vulnerabilities found for the HP LaserJet printer are the same vulnerabilities found in many other embedded devices making it platform independent. This stems from the security fundamentally being an issue of system design [2].

**4. Recommendations**

The above case demonstrates that it’s important for, not only the consumer, but for the industry in designing lower-level embedded systems to be more secure. This is not only a domestic issue, but an international issue as well with China being found in inserting small chips to the firmware for major corporations as a means of espionage and Russia found delivering a malicious payload to government run embedded systems that’s able to survive operation resets [4][5]. It’s understandable that consumers and large corporations neglect critical firmware updates considering the process can be very time-consuming and cumbersome [6], but here are a couple ways that you can be more diligent in protecting your system:

When looking for updates for your laptop, for example, you can go to the manufacturer’s webpage and there will usually be an option to look for different system updates. For example, when going to HP’s website you can select drivers, the device you’re using (printer, laptop, etc), select for HP to detect your device to determine the model number, then you’ll be redirected to a page showing all the updates available for your system. Scrolling down shows a “BIOS” option, which can be downloaded.

After downloading the firmware file, Google offers a tool called VirusTotal that can be used to scan the file in search of any known malicious files [1]. Updating the BIOS usually requires a reboot.

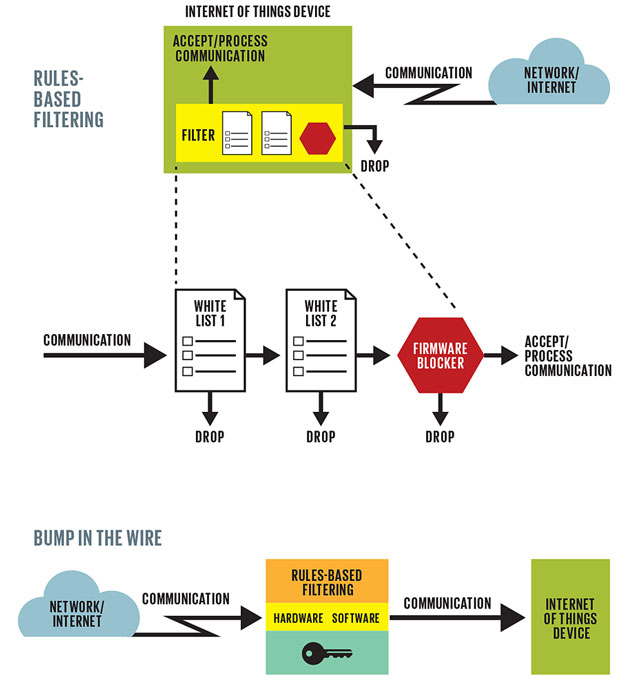
Further, as can be seen in Figure 2 [7], setting up a system of rules with the firewall and a protection system is an effective way in preventing attackers from manipulating a system (a Rules-Based Filtering approach). For instance, one could create a white list in the firewall and protective system to only accept updates from a selected number of systems (systems that are known to be good), and to block all other communication [7].

Figure 2. Rules-Based Filtering of communication with the internet.

**5. Resources**

|  |  |
| --- | --- |
| [1] | D. Murphy, “Google’s VirusTotal Can Now Scan Firmware Updates,” *PC Magazine*, 2016.  https://www.pcmag.com/news/341576/googles-virustotal-can-now-scan-firmware-updates |
| [2] | A. Cui, M. Costello and S. Stolfo, “When Firmware Modifications Attack: A Case Study of Embedded Exploitation,” presented at Net. & Dis. System Security Symposium, Columbia University, NY, 2013.  https://academiccommons.columbia.edu/doi/10.7916/D8P55NKB |
| [3] | A. Costin, A. Zarras and A. Francillon, “Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces,” presented at ASIACCS, Xi’an, China, 2016.  https://dl.acm.org/citation.cfm?id=2897900 |
| [4] | M. Kan, “Does Your Motherboard Have a Secret Chinese Spy Chip?,” *PC Magazine*, 2018.  https://www.pcmag.com/news/364262/does-your-motherboard-have-a-secret-chinese-spy-chip |
| [5] | M. Kan, “Russian Hackers Use Malware That Can Survive OS Reinstalls,” *PC Magazine*, 2018.  https://www.pcmag.com/news/364046/russian-hackers-use-malware-that-can-survive-os-reinstalls |
| [6] | A. Bellissimo, J. Burgess and K. Fu, “Secure Software Updates: Disappointments and New Challenges,” USENIX, MA, 2006.  https://www.usenix.org/conference/hotsec-06/secure-software-updates-disappointments-and-new-challenges |
| [7] | A. Grau, “How to Build a Safer Internet of Things,” *IEEE Spectrum*, Feb. 2015.  https://spectrum.ieee.org/telecom/security/how-to-build-a-safer-internet-of-things |