UNIVERSITY OF AMSTERDAM

Printer Security

Marcel den Reijer, Alexander Blauwgeers, and Vincent van Dongen

December 15, 2016

Abstract

In almost every cooperate environment printers are used to print documents. These documents can consist of personal or sensitive information. This project investigated possible floors in the network traffic that were generated during a print job. the conclusion of the project is that the print job is encoded. Therefore, it's possible to eavesdrop and extract the printed document. This security floor also allows an attacker to conduct a Man-In-The-Middle attack. In order to solve the security floor, we advise to encrypt the traffic between the client and the printer.

Contents

1 Introduction

Printer are used in every corporate environment. The printed documents may consist different types of information, which can be classified for example in personal, sensitive or open. Therefore, printers are a very interesting target for attackers that want to steal personal or sensitive documents. This project aims to find and investigate possible floors in the network traffic that were generated during a print job.
In order to find and investigate possible floors we have formulated a research question. The main question for this research is:

Is it possible to extract a printed document that was sent to a network printer?

The research question can be divided into multiple sub-questions:

1. Is it possible conduct a Man-In-The-Middle-Attack on a print job?
2. Is it possible to extract a document from a recorded PCAP?
3. Is it possible to capture the print job and replay the network capture?

2 Methods

In order to answer the research question, we have defined different methods. Every method has a different approach. In the subsection below are the different methods discussed.

2.1 submethod 1: Scan

The first method we used to investigate the possibilities was a portscan on the printer. This scan is used to see which ports are opened on the printer. Well-known ports (0-1024) are specified in RFC 1700. For the other ports there are various other sources to find out where they might be used for. This method aims to determine which services are running and accesable.

2.2 submethod 2: Passive analysis

The second method we used was a passive traffic analysis. The idea of this method was to capture all the trafic between the printer and the network. By collecting all the networkdata between the client and printer device, we are able to passively analyse the network traffic. The goal is to extract the document from the networkdata.

Most printers use a Print Job Langaue (PJL) structured file to tranfer information. The commands used in this structure are discripted as Printer Command Language(PCL) and can be also used for additiol information like printing quality. For Hewlet Packard(HP) printers this is described in a Technical reference manual. 1). This information can be send in various ways like XML, SOAP or directly as payload of the HTTP protocol, but it depends on the protocol that was used (eg. Internet Printing Protocol (IPP) or JetDirect). By passively analysing the network traffic, we tend to determine the types of protocols and methods that were used. This information will be used to eventually extract the document from the network traffic.

2.3 submethod 3: Replay Attack

Another methode that will be investigated in this report is the possibility of reprinting a caputered print job, which is also known as a ”replay attack”. A replay attack occurs when an attacker copies a stream of messages between two parties and replays the stream or a part of the stream to one or more of the parties. Replay attacks are replays of messages back to a sender as if they came from the receiver as the reply. In order to conduct a replay attack a print job has to be recorded. This will result in a file that contains the data exchange between the two parties (e.g. PCAP file). Next, relavent information can be extracted from the captured file to craft a new file that can be used for the replay attack. Finally, a varierty of tools can be used to resend the print job.

2.4 submethod 4: Vulnerabilities scan

Finally, there are many printers that you can buy in a store that contains bugs, errors, flaws, failures or other faults which enables an attacker to retrieve a printed document [2][3]. Therefore, we are going to scan the printer to determine that the printer doesn't contain suchs errors that enables an attacker to retrieve or reprint a previously printed document. The method is to reflect a list of security vulnerabilities and exposures that aims to provide common names for publicly known problems. An information security ”vulnerability” is a mistake in software that can be directly used by a hacker to gain access to a system or network. An information security exposure is a mistake in software that allows access to information or capabilities that can be used by a hacker as a stepping-stone into a system or network [4]. Common printer Vulnerabilities as well as exposures will be reflected against the printer.

3 Experiment

In this chapter we discuss for every method a different experiment. In the subsection below are the different expermiments discussed.

3.1 Test Environment

For the Proof of Concept a HP deskjet 3630 is bought to perform tests. This printer support wireless networking, which means a local network can be setup to perform tests. This was one of the requirements of the printer from the point of view of ethics. A schematic example is given in figure X. We used the environment experiment 1, 2 and 3.

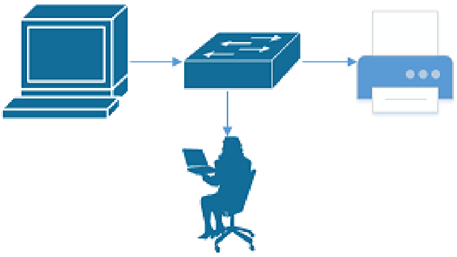

Figure X: Test Environment

3.2 Experiment 1: Scan

For the portscan we used NMAP to scan all ports from 0-65000 of the printer. When we found an open port we looked up the portnumber in RFC 1700. Then we tried to confirm this information by setting up a raw session using PUTTY. We prepared a python script below for the unkown services we found.
SCRIPT

3.3 submExperiment 2: Passive analysis

Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment, Experiment,

3.4 Experimentx 3: Replay Attack

There are many possibilities to perform a replay attack. However, they all have one thing thing in common. A recorded network capture of the print job is required. Wireshark is used to record a network capture. Wireshark is a network packet analyzer that is able to capture and interactively browse the traffic running on a computer network. For this experiment, the network interace of the client will be captured. Next, three different tools will be used to replay the capture against the printer device. The following tools will be used in the experiment: TCPrepaly, Curl and Cups [6][7][8]. Before we are able to resend the caputered data to the printer device, we have to modify and alter parameters inside the packet.

3.5 Experimentx 4: Vulnerabilities scan

A very common tool to test a device for vulnerabilities and Exploits is Metasploit. Metasploit is a popular penetration testing software. Furthermore, Metasploit contains a Vulnerability and Exploit Database, which is a curated repository of vetted computer software exploits and exploitable vulnerabilities[9]. For the experiment we are going to use the printer modules to scan for Vulnerabilities and Exploits. See appendix 1 for the list of CVE's that were used during the scan. After selecting appropriate list of CVEs we tried to attacked or targeted the printer. The results of the attack were that no known CVE was succesfull.

4 Results

Results, Results,

4.1 sResults 1: Scan

trgfd;gkfdlgkjfdgjk

4.2 Results 2: Passive analysis

We discovered that a print job consists of three stages. In the first stage the client announched a new print job in XML format. The printer responsed with a HTTP 200 OK, which indicated that the printer device had succesfully received the XML data. The second stage, the client sends a XML file with information related to the print job. Also, a large non readable data block is send to the printer. Based on the header of the non-readable block, we determined that is Header information related to HP PJL language. Printer Job Language (PJL) was developed by Hewlett-Packard to provide a method for switching printer languages at the job level, and for status readback between the printer and the host computer. [10] This means that the non readable data block contains the document. Yet again, The printer responsed with a HTTP 200 OK. In the final stage, the client sends another XML file indicating that the print job is done.

4.3 subResults 3: Replay Attack

After passive analysis of the network data, we tested the replay attack with the three different tools. The TCPreplay tool didn't work due to the fact that TCPreplay literally resends all the data. Therefore, crucial parameters inside the header had to be modified. Parameters like MAC-addresses, Three-way-handshake, sequence/acknowledge number had to be altered in order to resend the network data. TCPreplay doesn't support Three-way-handshakes. Consequently, we had to setup our own three-way handshake. However, we weren't able to establish a Three-way-handshake for unclear reasons. This resulted in not being able to conduct a replay attack with TCPreplay.

The second tool that we used for the replay attack was Curl. The difference between TCP replay and Curl is that TCP replay sends the complete capture. With Curl, a complete different header is send to the printer and only copies the body part (data part) of a packet. The data part of the packet was extracted from the network capture (PCAP). Therefore, this approach dodges the Three-way-handshake problem that we encounterd with TCPreplay. Unfortunately, only the first stage succeeded. When we send the packet that belongs to the second stage, the printer responded with a HTTP 500 Error even before the entire packet was send to the printer. Therefore, still methode didn't succeed.

The final tool that was used for the replay attack was CUPS. The approach for this tool is different compare to the other tools. instead of replay the entire network capture or only the body (data) part, we are only extracting the PJL part in that is located in the second stage. We'll save the PJL in ASCI coding, Hexadeciamal and RAW data format. Next, we tried to printed the PJL file with the CUPS application that contains a generic driver for printing. fortunately, this methode was successfully.

4.4 subResults 4: Vulnerabilities scan

Finally, we used a vulnerabilities scan to determine possible vulnerabilities and exloit in order to retrieve previous printed documents. Unfortuanelly, no common vulnerabilities or Exploits are present in the printer.

5 Discussion

The Vulnerability scan shows that the printer isn't Vulnerable for the CVE listed in appendix 1. Also, the port scan resulted that no port are open that enable an attacker to access the printer or retrieve usefull data.
However, the experiments shows that it's possible to retrieve the document out of the print job due to the fact that the print job is encoded. Therefore, it's also possible to repint documents and conduct a Man-In-The-Attack to retrieve a document in realtime.
Due to the fact that the print job is send in encoded data, it possible to decode the print job. This kind of attack could be prevented by encrypting the print job between the client and the printer.

6 Conclusion

Based on the experiment, it's possible to extract a document from a recorded PCAP due to the fact that the print job is only encoded. Furthermore, it's possible to extract the PJL data from the network capture. A new print job can be created with the extracted PJL data. This allows an attacker to resend a captured print job. Finnaly, it's possible to conduct a MiTM attack with ARP-poisining.

7 Future

The methods, which are used to capture files were semi-active. This mean some operations were manually performed. After when packets were captured, they were manually exported to a raw data format and saved to a PLC format. A nice future related work is to capture printed files a save it real time, without manually interactions. bla[1]

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.

[3] Knuth: Computers and Typesetting,
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`