POLYTECHNIC OF TURIN - STOCKHOLM'S KTH

Faculty of Engineering

Master of Science in Computer Engineering

Master Thesis

MPTCP Security Evaluation

Analysing and fixing critical MPTCP vulnerabilities, contributing to the Linux kernel implementation of the protocol

Advisors:
prof. Antonio Lioy
prof. Peter Sjödin

Candidate:
Fabrizio Demaria

Company tutors
Intel Corporation Inc
Henrik Svensson
Joakim Nordell
Shujuan Chen

March 2016

Acknowledgements

Thanks to...

Summary

Abstract goes here...

Contents

Chapter 1

Introduction

The introductory part explains what MPTCP is and why it is important to study it. Then, the problem statement for this thesis is introduced, to give a good idea of the topics encountered in the paper. Lastly, the methodology followed to solve the problems is presented in the section "Methodology", where the structure of the sections in the paper is also explained.

1.1 Motivation

This section would start with a general introduction of the interconnected world of today, discussing how hardware and software communication has changed in the last decade. The focus of this part is to bring up the multihoming and multipath reality of the infrastructures of today and how this led almost naturally to the MultipathTCP concept. It would be good to cite similar technologies developed before MPTCP (for example SCTP), explaining in which aspects MPTCP is supposed to be a better option. This should include an overview of the real benefits that can be achieved by adopting MPTCP in common appliances (smartphones for example) as well as modern datacenters. It would be good to explain the fact that MPTCP was designed to be as retrocompatible as possible with current infrastructure (lower layer) and applications (higher layer).

1.2 Problem Statement

This part should introduce the content and the main focus of this paper, as well as presenting the objectives of the thesis:

- studying the security implications of adopting MPTCP on current infrastructures;
- listing the known vulnerabilities affecting the current version of the protocol;
- fixing the ADD_ADDR vulnerability of the protocol;
- developing effective and powerful simulation scenarios in order to test MPTCP;
- contributing to the upstreaming of the protocol into the Linux kernel by developing patches and improving official RFC documentation.

1.3 Methodology

This section should contain a short road map with the various step taken to fix the addressed problems and the general methodology adopted for the thesis' work. Perhaps, it is possible to cite here the working environment and the parties involved. This section might end up with an explanation about the structure of the paper.

Chapter 2

Multipath TCP

From now on the discussion becomes more technical. This chapter is all about how MPTCP works from a technical perspective. In this chapter there is no reference about the work carried out during the stage but only information taken from external documentation.

2.1 Transmission Control Protocol (TCP)

This is an explanation on how plain TCP protocol works. Even if there is no need to go into too much details here, this is a necessary starting point from where the MPTCP extension discussion can start (in the next section).

2.2 MPTCP Design

How MPTCP is added on top of TCP (with all the related design aspects) is reported here. This is the first portion of the thesis containing a more in-depth description of how MPTCP works in the networking stack. This specific section might follow closely the introductory portions of the RFC documents regarding MPTCP.

2.2.1 Control Plane

All the MPTCP options used to manage MPTCP sessions are reported and explained here, including all the details on how to set a new session and add/remove subflows.

2.2.2 Data Plane

This part concerns all the MPTCP options used to manage the data flow in a MPTCP session, including how the byte stream is subdivided into different subflow and how the original order of the packets is provided at the receiver.

2.3 MPTCP Deployment

After having explained all the technicalities about the protocol, it is now possible to talk about its deployment status and the problems encountered by pairing MPTCP with current infrastructures. This might seem a bit outside the scope of the thesis, but it is worth mentioning that the *deployment status* and *implementations* are a good indicator of how much MPTCP is important in the Internet community and they are good topics to motivate the thesis work. Also *middleboxes compatibility* is indeed a fundamental part of the MPTCP security evaluation, being monitoring and detection equipment part of these middleboxes.

2.3.1 Middleboxes Compatibility

This section will be quite technical and it is supposed to list the most important middleboxes and their impact/effect on a MPTCP connection. These boxes include NATs, proxies and firewalls. This part should clearly state why MPTCP widespread adoption is a big challenge.

2.3.2 Implementations

Despite the previously described problematics, MPTCP is a big bet in the IETF community and many implementations have been developed for the most common OSes, listed in this section (with some history notions).

2.3.3 Deployment Status

It should be interesting for the reader to go through some examples of real world's scenarios in which MPTCP is used successfully. Here it is possible to cite some important achievements related to MPTCP (for example the highest throughput ever reached with the new protocol). If available, it would be also good to show some graphs about MPTCP adoption rate or similar.

Chapter 3

MPTCP Security

This chapter starts with a general overview and it later introduces the theory behind the residual threats that affect MPTCP, according to the most recent documents and research. In this chapter there are still no references to the original work carried out during the stage.

3.1 Threat Analysis

A general introduction about the security requirements for MPTCP is reported here. This part is also supposed to present some categorizations related to general networking attacks, in order to give a good idea of the possible threats and their effects on an ongoing connection (not only for the MPTCP case). These notions are later mentioned again when listing the various attacks to which MPTCP is currently vulnerable.

3.2 ADD_ADDR Attack

The most important attack is the ADD_ADDR attack. It is the most dangerous and in the end it is the main topic of the whole work carried out for this thesis. This section explains in details the theory behind the attack as well as the steps to be followed in order to carry out the attack, as reported in RFC 7430. No simulation is cited here, since an entire chapter is dedicated to that later on.

3.2.1 Concept

3.2.2 Procedure

3.2.3 Requirements

3.3 Additional Threats

Even if the paper focuses on ADD_ADDR attack, it is a good point to present here the other residual threats that are reported in RFC 7430. These are considered minor threats.

3.3.1 DoS Attack on MP_JOIN

3.3.2 Keys Eavesdrop

3.3.3 SYN/ACK Attack

Chapter 4

ADD_ADDR Attack Simulation

Since the first part of the thesis work has been devoted to build the attacking tool to reproduce the exploitation of the ADD_ADDR vulnerability, an entire chapter is dedicated to this topic. This work can be a valuable reference for future simulation setups, involving MPTCP or any other networking protocol.

4.1 Environment Setup

Here it will be explained what UML virtual machines are and why they were good candidates for the simulation tests. The setup procedure is also reported here, with graphs to visually show the simulation's network scenario. The Scapy tool is also presented here. This was a great program to manipulate and forge packets.

4.2 Attack Script

This part is supposed to explain how the Scapy attacking script actually replicates all the steps needed to execute the ADD_ADDR attack. The steps were explained from a theoretical point of view in the previous chapter. Some code snippets of the attacking script can be present in this section. Some details on the problems encountered in building the script and how they have been solved can be valuable material, too.

4.3 Reproducing the Attack

This section is a step-by-step tutorial on how to call the attack script and reproduce the attack. It mainly contains the README file in the GitHub repository where the script can be retrieved. Screenshots and/or Whireshark output can be added to show the actual behaviour of the attack over a netcat connection.

4.4 Conclusions

A final evaluation of the experiment is needed. First, it is important to emphasize the limitations of the adopted tool and how this simulation differs with respect to real scenarios. Nevertheless, it is crucial to explain the value of such experiment, that indeed proves the feasibility of the ADD_ADDR attack and better justify the development of a fix, which is the main topic of the next chapter.

Chapter 5

Fixing ADD_ADDR

This chapter is related to the second and most important part of the work performed during the Master Thesis work at Intel. The ADD_ADDR2 option is developed and added to the current MPTCP implementation for the Linux kernel in order to fix the vulnerability.

5.1 The ADD_ADDR2 format

The actual new format and the reasons why it fixes the vulnerability of ADD_ADDR are reported here. Various discussions can be added to explain why this is believed to be the best way to fix the problem. This part doesn't yet include any implementation details and/or code snippets.

5.2 Implementing ADD_ADDR2

An introductory section that shows the main architectural aspects of how MPTCP has been merged into the TCP code and the TCP modules inside the kernel. Here it starts the part with all the details about the implementation of ADD_ADDR2 in the kernel, as part of the work developed during the stage. Code snippets have to be added here. The following subsections are the side issues and side features that have been elaborated during the thesis work.

5.2.1 Retro-compatibility

Version control mechanism was not present but it is needed to negotiate which format to use in a MPTCP session: ADD_ADDR or ADD_ADDR2.

5.2.2 Port Advertisement

Port advertisement in ADD_ADDR is possible according to RFC specifications but it was not part of the implementation at the beginning of the thesis work, so it has been added.

5.2.3 IPv6 Considerations

Longer addresses brought some issues related to TCP option fields limitations.

5.2.4 Crypto-API in MPTCP

A major problem was how to deal with the new hashing requirements introduced by ADD_ADDR2. Extending the current MPTCP hashing function to deal with input messages of arbitrary size is a first point to explain. The second part has to deal with the whole analysis related to adopting the kernel CRYPTO APIs to calculate the HMAC values in MPTCP and why this is not advisable.

5.3 RFC Contributions

Another minor part of the thesis work on MPTCP is related to some small contributions to the official RFC documentation.

5.4 Experimental Evaluation

This part should include performance analysis regarding the new format introduced with ADD_ADDR2. A discussion on how the new format (and all the other modifications introduced with the patches) could impact any aspect of the protocol should be present in this section.

Chapter 6

Conclusions

6.1 Related Work

References to all the related work, including all the efforts to make MPTCP secure and stable, can be reported here.

6.2 Future Work

Here goes the list of the next steps to be taken care of in terms of MPTCP security, in order to facilitate the protocol's upstream and its widespread deployment. This section can also include a more specific discussion on the aspects to be still analysed regarding the new format ADD_ADDR2.

6.3 Final Thoughts

Some final conclusion.

Appendix A

An appendix

Appendix content goes here...