06-29637 *Network Security (Extended)*
Spring Semester 2018
Submission by **Marius Darius Hente**

The University of Birmingham
School of Computer Science
March 10, 2018

# Assignment 2: TCP Analysis

## 1 Introduction

This assignment for the Network Security module revolves around configuring an IPSec implementation. It is split into two parts:

1. Configuring IPSec to work with a shared secret. The key is previously known by the participants, having been securely relayed from one to another.

2. Configuring IPSec to work with certificates. In this case the participants rely on public / private key encryption and certificate authorities instead of a shared secret.

For clarity purposes, the environment is defined: two virtual machines running Debian with the LXDE desktop environment, with one acting as the server (running apache) and the other acting as the client. For testing purposes, a third machine is set up to run as a router between the other two, which will allow us to see all the packets being transmitted.

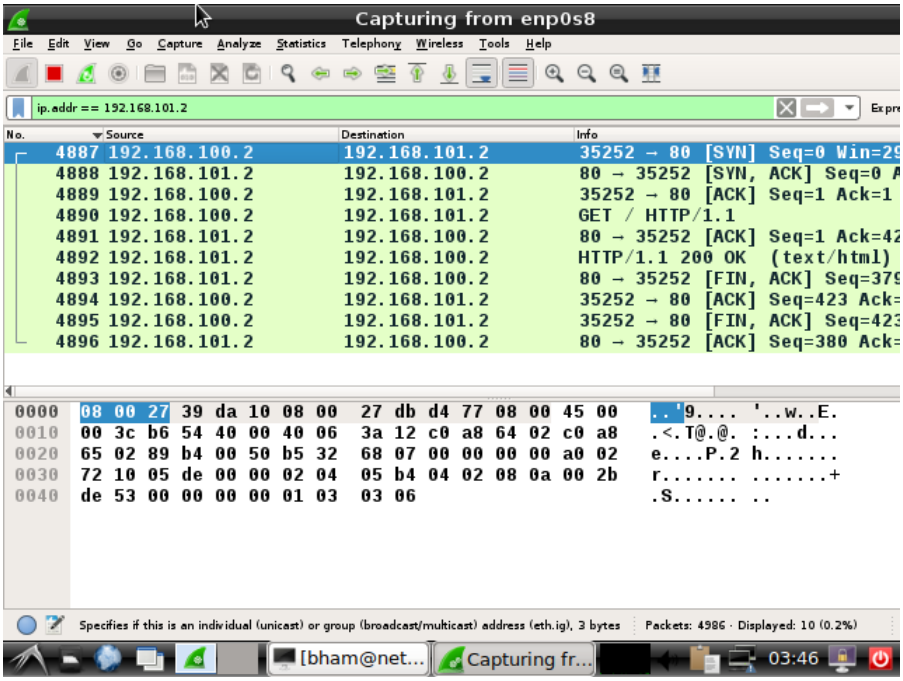
The Strongswan IPSec implementation was chosen, with the "charon" IKE daemon. The machines also have other common internet tools installed such as netcat, wireshark, nmap, etc. The **ipsec** command was used to set IPSec up, in favor of other tools such as vici or the swanctl command, as it allows running simple configurations very easily.

## 2 Part 1: Configuring IPSec with a shared secret

For the first part of the assignment, IPSec will be configured using a shared secret. The the two files that need to be modified are **/etc/ipsec.conf** and **/etc/ipsec.secrets**. The conf file will store the configuration for the connection, while the the secrets file will store the shared secret, and later the certificates.

A configuration file was developed specifying the connection over static IP, and a shared key was placed into the secrets file (the key being "example"). Now to test our configuration, we should do a simple before-and-after test of the connection.

We can attempt to fetch the page before IPSec and see what the WireShark packets look like:



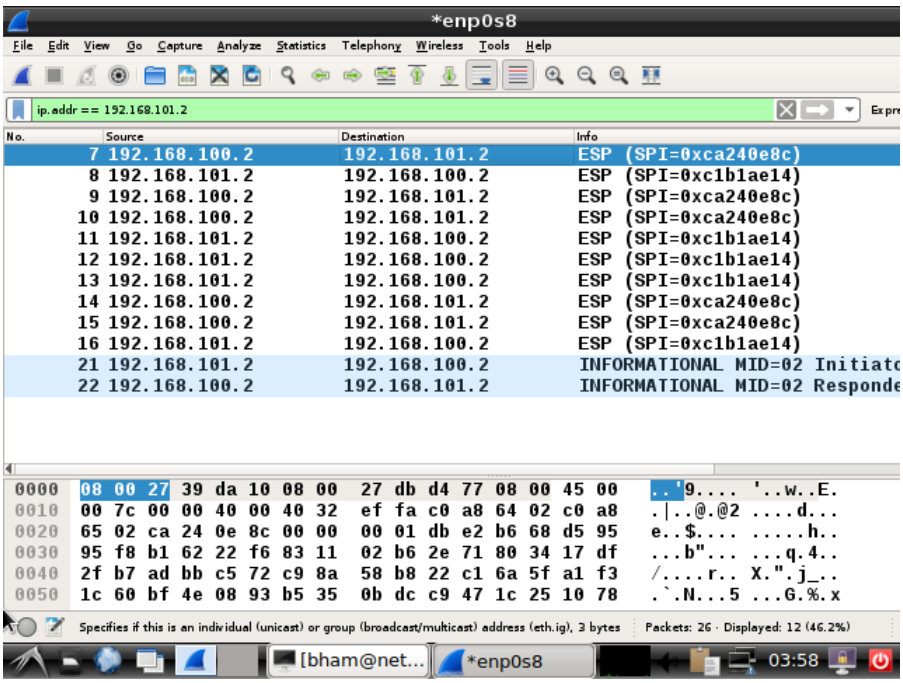
We can clearly see the packets are unencrypted, as we can see the individual TCP headers, the GET message, and the HTTP response. If we were to inspect the packets' payload, we would also see the HTML code that is being sent on the line. An attacker could easily do damage in this case, for example with a man in the middle attack. IPSec is used to provide authentication of the parties (in this part via the shared secret) and encryption of the packets.

Now that we have seen what the base case looks like, we can start IPSec and see how the results differ. The files have already been configured (and will be included with this report), and all must do now is to start the service. To that, we run the following command on both ends of the connection:

sudo ipsec start

Note that if we wanted to debug the connection we could run the command with the –nofork parameter, making it output the debug messages into the console. Since the configuration files are correct, though, we should not need this feature.

Having run the command on the two machines, we can test the connection again by attempting to fetch the web page from the server, and tracking the packets with WireShark:




Success! Our connection is now encrypted. We can no longer see the actual data transmitted, and the data is sent as ESP packets. Furthermore, due to the key used (i.e. the shared secret) we can be confident that the party on the other side of the connection knows our secret, and is therefore authenticated. We have successfully implemented shared-secret IPSec.

# 3 Part 2: Configuring IPSec with certificates

Shared-secret IPSec is generally useful in cases where we can afford to securely relay a key from our location to the other party's location, i.e. we have a good budget that we can invest in security. This is, however, not realistic in many of today's scenarios. The solution is to run IPSec with an alternative authentication method, namely certificates. For the second part of this assignment, the task is to implement IPSec with certificate-based authentication.

# 4 Conclusion

To conclude this report, we have tested all the necessary functionality that our firewall, implemented via iptables, should have. The system behaves as expected in all cases, with the provided rules. It is blocking what it should be blocking, and forwarding only the packets that it should be forwarding. Thank you very much for taking the time to read this report.