Communication Networks 2

SS 2021

Assignment 1

Group 06

Name	Mat.Nummer
Paul Kloker	12034928
Juan Aramis Oposich	11701238

May 8, 2021

1 Task and Protocol Description

The subject of the first task is to observe email client's network traffic and to extract the username and password from the client. Furthermore this assignment offers some reflections on the security aspects of protocols using plaintext on network traffic.

1.1 Internet Message Access Protocol

The IMAP protocol is used by email clients to retrieve email messages from a mail server over a TCP connection. The protocol has been defined in the RFC 3501. Usually an IMAP server typically listens on port number 143, like our first task did. The more secure way is to run IMAP over SSL/TLS, which is assigned the port number 993.

1.1.1 Authentication

As this report will show, the IMAP protocol contains an authentication message, which leaks the vulnerability. The user must authenticate himself before he can gain access to mail. This is done by logging in with his username and password. The password is transmitted in plain text at the IMAP protocol level. Mail servers can therefore forbid clients from transmitting the password if no encrypted session has been established beforehand.

Furthermore IMAP has an extension for **PLAIN Simple Authentication and Security Layer (SASL)**[**zeilenga2006plain**] which was designed to have a wider interoperability with other systems. The main drawback is shown in the first assignment, capturing the username and password of the client.

2 Method(Procedure?)

This section describes the different steps that are needed to recover the forgotten email password from the network traffic between the email client and the email server.

2.1 Traffic capture with WireShark

First all incoming and outgoing traffic needs to be captured by using WireShark. To reduce the number of captured packets the capture filter "`not port 22`" was used be-

cause this prevents the SSH traffic from being captured as well. To get the IMAP or POP authentication message Thunderbird was launched right after the capturing process was started, which was terminated again after about 20 seconds. Scrolling through the capture showed quickly that IMAP was used to check for new messages and not POP.

2.2 Filter captured traffic

In order to show only the required IMAP messages the display filter "`imap`" was used. A screen shot of WireShark with this filter applied can be seen in figure 1.

Figure 1: Captured IMAP traffic

2.3 Decode messages and find password

Base64 Decoding:

```
base64 -di <<<AGNuXzA2QGV4MS5jbjJsYWIuY24udHV3aWVuLmFjLmF0AFBlZ3Vxb3Rhc2Uy
cn_06@ex1.cn2lab.cn.tuwien.ac.atPeguqotase2

base64 -di <<<AGNuXzA2QGV4MS5jbjJsYWIuY24udHV3aWVuLmFjLmF0AFBlZ3Vxb3Rhc2Uy | hexdump -c
0000000  \0   c   n   _   0   6   @   e   x   1   .   c   n   2   l   a
0000010   b   .   c   n   .   t   u   w   i   e   n   .   a   c   .   a
0000020   t  \0   P   e   g   u   q   o   t   a   s   e   2
000002d
```

As can be seen in table 1, tables can also be useful.

3 Conclusion

This shows how easy it is for attackers to retrieve passwords from plain text email traffic. Therefore, it is very important to make sure that all connections to the email server are TLS/SSL encrypted. With IMAP this can be done explicitly over port 143, by using STARTTLS, or implicitly over port 993 which enforces a encrypted connection.

Figure 2: Don't forget to find a fitting caption for your graphics.

Table 1: Routing table for network A

router	destination	via
r1	10.1.2.0/24	10.3.2.1
r1	10.2.1.0/24	10.3.2.1
r1	10.5.3.0/24	10.0.2.1
r2	10.0.3.0/24	10.5.2.1
r3	10.3.0.0/24	10.3.4.1

With vspace, you can add vertical empty space for formatting purposes (which should only be used as a last resort):

The following text is now shifted vertically for one centimeter

- Please use the *itemize* environment for better and clear representation of
- your results

1. Also you can use the *enumerate* environment for

2. representing the sub-example