

ACME_32_a3_report

Group number: 32

Students:

- Lilit Hasan: 1958055
- Matteo Piermartini: 1802597
- Aurora Polifemo: 1802485

Initial Brainstorming

We have to use Squid, a caching proxy for web supporting HTTP, HTTPS and FTP, to create a forward proxy so that each host in the network uses it to navigate the web. The authentication mechanism we decided to use is *htpasswd*, because it's a text file used in Linux that allows us to configure the web server apache. It's a hidden file, and once you send your username and password, you can access the encrypted passwords. So in Squid we created the file *passwords*. For the reverse proxy we need to use Apache2, we need to make it possible to redirect HTTP to HTTPS, in order to secure the connection.

1. Forward Proxy configuration

We need to install and set up Squid in the Proxy Server (100.100.6.3).

For the installation we use the following commands:

```
sudo apt-get update
sudo apt-get install squid
sudo apt-get install apache2-utils
```

Then we need to create the different users we're going to use:

```
sudo touch /etc/squid/passwords
sudo chmod 777 /etc/squid/passwords
sudo htpasswd /etc/squid/passwords Nina → Password: ninaacme32
sudo htpasswd /etc/squid/passwords Pinta → Password: pintaacme32
sudo htpasswd /etc/squid/passwords Maria → Password: mariaacme32
```

We have to modify the *squid.conf* file so to allow an authenticated access by using the password file we set up above:

```
root@proxyserver:/etc/squid# cat squid.conf
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwords
auth_param basic realm proxy
acl authenticated proxy_auth REQUIRED
http_access allow authenticated
http_port 3128

acl SSL_ports port 443 563
acl Safe_ports port 443 563
acl CONNECT method CONNECT
http_access deny !Safe_ports
http_access deny CONNECT !SSL Ports
```

Now, the kali host (100.100.2.100) in the CLIENTS network has to use this Squid service. In order to be able to do that we have to modify its `/etc/environment` file:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin
COMMAND_NOT_FOUND_INSTALL_PROMPT=1
POWERSHELL_TELEMETRY_OPTOUT=1

http_proxy="http://nina:ninaacme32@100.100.6.3:3128"
http_proxy="http://pinta:pintaacme32@100.100.6.3:3128"
http_proxy="http://maria:mariaacme32@100.100.6.3:3128"

https_proxy="http://nina:ninaacme32@100.100.6.3:3128"
https_proxy="http://pinta:pintaacme32@100.100.6.3:3128"
https_proxy="http://maria:mariaacme32@100.100.6.3:3128"

ftp_proxy="http://nina:ninaacme32@100.100.6.3:3128"
ftp_proxy="http://pinta:pintaAcme32@100.100.6.3:3128"
ftp_proxy="http://maria:mariaacme32@100.100.6.3:3128"

no_proxy="localhost,127.0.0.1, ::1"
```

To conclude, on the Proxy Server we need to start the service:

```
systemctl start squid
```

2. Reverse Proxy configuration

To configure the Reverse Proxy, we need to set up the Certificate Authority to distribute the self-signed digital certificate between the known parties, the hosts in the network and the fantasticcoffee host.

On the Opnsense interface, on the Main Router, the first thing to do is to go on System>Trust>Authorities and create two CA, the first is used to sign the second one:

System: Trust: Authorities

Descriptive name:

Method:

Internal Certificate Authority

Key length (bits):

Digest Algorithm:

Lifetime (days):

Distinguished name

Country Code:

State or Province:

City:

Organization:

Email Address:

Common Name:

System: Trust: Authorities

Descriptive name:

Method:

Internal Certificate Authority

Signing Certificate Authority:

Key length (bits):

Digest Algorithm:

Lifetime (days):

Distinguished name

Country Code:

State or Province:

City:

Organization:

Email Address:

Common Name:

| System: Trust: Authorities | | | | | Add | |
|----------------------------|----------|-------------|--------------|---|---------------------|--|
| Name | Internal | Issuer | Certificates | Distinguished Name | | |
| webserver | YES | self-signed | 1 | emailAddress=admin@admin.com, ST=RM, O=acme32, L=RM, CN=internal-webserver, C=IT Valid From: Thu, 01 Jul 2021 13:28:14 +0000 Valid Until: Thu, 28 Mar 2024 13:28:14 +0000 | | |
| webserver-intermediate | YES | webserver | 1 | emailAddress=admin@admin.com, ST=RM, O=acme32, L=RM, CN=webserver-internal, C=IT Valid From: Thu, 01 Jul 2021 13:32:41 +0000 Valid Until: Thu, 28 Mar 2024 13:32:41 +0000 | | |

Then on System>Trust>Certificates we set the server certificate, our fantasticcoffee machine, that will be signed by the intermediate CA:

System: Trust: Certificates

full help [CD](#)

Method

Create an internal Certificate

Descriptive name

fantasticcoffee

Internal Certificate

Certificate authority

webserver-intermediate

Type

Server Certificate

Key length (bits)

2048

Digest Algorithm

SHA256

Lifetime (days)

1001

Distinguished name

Country Code :

IT

State or Province :

RM

City :

RM

Organization :

acme32

Email Address :

admin@admin.com

Common Name :

fantasticcoffee

Alternative Names

Type

Value

URI






https://fantasticcoffee-zentyal.org

-

+

Save

In order to download the p12 certificate that is generated automatically, we can go under System>Trust>Certificates and click on the second-last button. We need this certificate to create the key.

| System: Trust: Certificates | | | |
|---|------------------------|---|---|
| | | | |
| Name | Issuer | Distinguished Name | In Use |
|  fantasticcoffee | webserver-intermediate | subjectAltName=URI:https://fantasticcoffee-zentyal.org, emailAddress=admin@admin.com, ST=RM, O=acme32, L=RM, CN=fantasticcoffee, C=IT |     |
| CA: No, Server: No | | Valid From: Fri, 02 Jul 2021 16:33:15 +0000 Valid Until: Fri, 29 Mar 2024 16:33:15 +0000 | |

On the web server, we need to get the files with the .key and .crt extension.

The first step is to transfer the *fantasticcoffee.p12* file on it:

Web server: `nc -l -p 1234 > fantasticcoffee.p12`

Our terminal: `nc -w 3 100.100.6.2 1234 < fantasticcoffee.p12`

Once we have the file, we can run these commands:

```
openssl pkcs12 -in fantasticcoffee.p12 -out fantasticcoffee.key -nodes -nocerts
openssl pkcs12 -in fantasticcoffee.p12 -out fantasticcoffee.crt -nokeys
```

Then, to use those files, we need to modify the */etc/apache2/sites-enabled/000-default.conf* file:

```
#port:80
<VirtualHost *:80>
    ServerName fantasticcoffe-zentyal.org
    ServerAlias www.fantasticcoffee-zentyal.org
    RedirectPermanent / https://fantasticcoffee-zentyal.org/
</VirtualHost>

#port:443
<VirtualHost *:443>
    ServerName fantasticcoffee-zentyal.org
    ServerAlias www.fantasticcoffe-zentyal.org
    ProxyPreserveHost On
    ProxyPass / http://100.100.4.10:80/
    ProxyPassReverse / http://100.100.4.10:80/
    ErrorLog ${APACHE_LOG_DIR}/Log_error.log
    SSLEngine On
    SSLProxyEngine On
    SSLCertificateFile /etc/apache2/sites-enabled/fantasticcoffee.crt
    SSLCertificateKeyFile /etc/apache2/sites-enabled/fantasticcoffee.key
    LogLevel warn
</VirtualHost>
```

3. Test of the configuration

For the forward proxy, we made two test on Squid:

1. We tested the authorization of the users we created:

```
curl -X Maria:MariaAcme21@100.100.6.3:3128 -I http://cybersecurity.uniroma1.it
```

```
user@kali:~$ sudo curl -x maria:mariaacme32@100.100.6.3:3128 -I http://cybersecurity.uniroma1.it
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Date: Fri, 02 Jul 2021 17:22:05 GMT
Server: Apache/2.4.10 (Debian)
X-Powered-By: PHP/7.0.17
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
Content-Language: en
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 7 (https://www.drupal.org)
Link: <http://cybersecurity.uniroma1.it/>; rel="canonical",<http://cybersecurity.uniroma1.it/>; rel="shortlink"
X-Kong-Upstream-Latency: 276
X-Kong-Proxy-Latency: 0
X-Cache: MISS from proxyserver
X-Cache-Lookup: HIT from proxyserver:3128
Via: kong/1.4.2, 1.1 proxyserver (squid/3.5.27)
Connection: keep-alive
```

2. We tested HTTPS:

```
curl -I https://pypi.org
```

```
user@kali:~$ sudo curl -I https://pypi.org
HTTP/2 200
content-security-policy: base-uri 'self'; block-all-mixed-content; connect-src 'self' https://api.github.com/repos/ *.fastly-insights.com sentry.io https://api.pwnedpasswords.com https://2p66nmm
ycsj3.statuspage.io; default-src 'none'; font-src 'self' fonts.gstatic.com; form-action 'self'; frame-ancestors 'none'; frame-src 'none'; img-src 'self' https://warehouse-camo.ingress.cmh1.psfho
sted.org/ www.google-analytics.com *.fastly-insights.com; script-src 'self' www.googletagmanager.com www.google-analytics.com *.fastly-insights.com https://cdn.ravenjs.com; style-src 'self' font
s.googleapis.com; worker-src *.fastly-insights.com
content-type: text/html; charset=UTF-8
etag: "PL7DPqB+HRSvms+VsYWbQ"
referrer-policy: origin-when-cross-origin
server: nginx/1.13.9
accept-ranges: bytes
date: Fri, 02 Jul 2021 17:29:08 GMT
x-served-by: cache-bwi5166-BWI, cache-mxp6936-MXP
x-cache: HIT, MISS
x-cache-hits: 1, 0
x-timer: S1625246948.917890,VS0,VE109
vary: Accept-Encoding, Cookie, Accept-Encoding
strict-transport-security: max-age=31536000; includeSubDomains; preload
x-frame-options: deny
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-permitted-cross-domain-policies: none
content-length: 26947
```

For the reverse proxy, we tried to connect to 100.100.6.2 (web server) on the browser and we are getting redirected on the site <https://fantasticcoffee-zentyal.org>

4. Final Remarks

Through `/var/log/apache2/Log_error.log` we managed to see some errors when we tried to restart the service using this command: `systemctl restart apache2`.

It was necessary to run these commands in the web server:

```
sudo a2enmod proxy
sudo a2enmod ssl
```

It was necessary to install *modsecurity* too on the web server:

```
apt install libapache2-mod-security2
```

Then on the `/etc/modsecurity/modsecurity.conf` we changed this line:

```
SecRuleEngine On
```