Final Report: Requirements and System Design

Fredrik Krantz
fkra@kth.se
940322-9355

Ruizhi Yang
ruizhi@kth.se
980722-1610

Xin Liu
xinl2@kth.se
951019-9384

March 13, 2020

1 Introduction

We have built a new secure network infrastructure for ACME in order to seamlessly connect its new London branch to the Stockholm headquarters. We used different network tools such as OpenSSL, OpenVPN to finish it.

2 Requirement

We need to setup two separate networks, one in Stockholm and one in London. And the communication between these networks should be in a secure manner, which means no information is lost or exposed to outsiders, and no fake information can be inserted by outsiders. The aspects we need to consider to achieve such secure communication are as follow.

- Authentication
  A digital certificate issued by CA should be assigned to their employees/users for authentication. And also a mobile device for two-factor authentication as a proof of possession.
- Secure connectivity
  Only computers with IP addresses from Stockholm headquarters or the London branch should be able to access to the internal network of ACME. Communication request outside these two network should be denied. But at the same time, employees should be able to access the network from their home network.
- Confidentiality
  Information and data exchanged between the server and a user be confidential and authenticated. Only the networks or devices of the ACME employees can access the main web server which has the core data of the company.
- File exchange
  The file exchange process should be confidential, complete and authenticated. And the transmissions are not only between the internal hosts, but also between the personal devices.
- Wireless access
  Employees shall be able to connect their laptop computers to Stockholm using a Wi-Fi connection from the London branch. And that the connection should be authorized and authenticated through that wireless connection.
- Other
  We wants to be able to monitor the network and network traffic in order to identify suspicious activity and to alert the company of incoming attacks.

3 System design

3.1 External security

As mentioned before, since the company is distributed in different countries, different locations and there is a need for frequent communication, we use the Internet as the communication carrier between the various private networks of the company, that is, VPN, which can achieve secure connectivity with the server. We configured the Stockholm router as the VPN server. The tool we used here is OpenVPN. Employees only need to install OpenVPN on their devices, and use the configure file obtain from the server in order to use it to connect to Stockholm network. We applied for a dynamic DNS(DDNS) hostname at noip.com.Thus, whenever we change the IP-address of the router, employees are still able to connect to the VPN router using the same configure file.

To access the secure data of the web server, users should be authenticated using two factors. An employee is required to install their ACME certificate on devices they use to access the server. The second factor is an one-time 6-digit pin-codes(OTP). We used Google Authenticator to achieve this, which is based on TOTP and should be install on user's mobile devices. Its pin-codes will be refreshed every 30 seconds.

When the communications between the different nodes of the private network need pass through the public Internet and want to be kept secret, then all the data transmitted through the Internet must be encrypted [2].The confidentiality of the data is naturally guaranteed when the transmission of datagram is only within an internal network without passing through the Internet.

3.2 Internal security

Because ACME is an enterprise and it wants to keep its internal resources and communication safe and confidential, it needs both their server and employees to be authenticated. Thus, we use mutual SSL authentication, that is, both server and client need to get their certificates issued by a trusted certificate authority. So, once the verification is successful, the authenticated server and the authenticated client can transmit the encrypted data through a safe tunnel. As mentioned above, we assigned a digital certificate for every employee. When communications are restricted to LANs, certificates can be issued without applying to third parties[3]. SO, we can simulate CA by ourselves in this situation. The tool we use for this is OpenSSL, which is a full-featured toolkit for the TLS and SSL protocols, and is widely used in internet server, including most HTTPS servers[1]. The Root CA credential is kept secure. And a Signing CA with a certificate signed by Root CA is used to issue certificates to all employees. Also, certificates are issued with a 2048-bits RSA key pair and SHA256 hashed message authentication code(HMAC), which is highly secure. Certificates Revocation Lists(CRL) will be regenerated everyday. But if a certificate is revoked, the CRL will be regenerated and distributed immediately.

As for the ACME web server, the company web pages, internal online system and some vital resources are in it. We used Apache HTTP web server, which is a open-source and free web server software It can provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. Also, the web server has configured with SSL/TLS, so people can only access it using HTTPS, which can achieve the requirement of authentication and encryption.

To make sure that the server only accept communication requests from certain IP addresses, we placed a firewall for the server and set the rules of the firewall in order to drop all packets from networks that we don't trust while we'll use authentication against IP spoofing. And the firewall is installed in the web-server, which will prevent unauthorized outsiders to get access to the internal network and sensitive data. The default setting is to drop all packages from external network and only let the users who belongs to the network of Stockholm branch access it. We used Iptables to achieve this.

In order to secure Wi-Fi connections and to make sure that the connected users are authorized and authenticated we will implement a solution with RADIUS. Specifically the freeRADIUS server which has a lot of available services which includes a way to authenticate network connections. This is done by running the freeRADIUS server on the netgear r6100 router. So it will act as a server and as a client. To install FreeRADIUS we ssh into the router and run the opkg command to install several packages which can be found in [4]. After installation and making sure that the radius server is working correctly we configure the wireless config file in "/etc/config/" to reassemble the config shown in appendices. In order to make radius work we will need to uninstall wpad-mini and install wpad on the router using opkg.

After this we only need to add users to the
authorize file in ”/etc/freeradius3/mods-config/files/” on the router. And then starting radius with the command ”LD_LIBRARY_PATH=/usr/lib/freeradius3 radiusd -X”. After this the wireless service on the router can be started/restarted and it should be working as intended.

To detect intrusions and other malicious activities we have chosen to implement an IDS. The IDS we chose is called SNORT which is open source and a popular choice. The IDS will be placed on the company LAN, this is behind the firewall. But to get hold of the network traffic we have to mirror the traffic from the router. Since IDSs does a deeper inspection of packages it will be able to detect problematic packages that have passed the firewall. In order to implement this solution we first setup a virtual server using Oracle VM VirtualBox. We use an Ubuntu Server 18.04.4 host to run the snort service on. It should be set to bridged in the VirtualBox settings in order to let it have an ip on the same lan as the router. Then run:

```
”sudo apt-get update -y”
”sudo apt-get install -y snort”
```

To install snort. After that, go to /etc/snort/snort.conf in order to set up the correct configurations.
Set ”ipvar HOME_NET” to match the local network prefix.
Uncomment ”alert_syslog: LOG_AUTH LOG_ALERT” and ”log_tcpdump: tcpdump.log”

Then start snort using the bash command ”sudo snort -v -c /etc/snort/snort.conf” in order to debug the startup and check that it is capturing packets. After that it is time to mirror the traffic from the provided netgear r6100 router with openWRT. We ssh into the router from a laptop in order to get terminal access. From the terminal we then run the following commands:

```
”opkg update”
”opkg list | grep -i tee”
”opkg install iptables-mod-tee”
```

Then reboot the router and run the following commands:

```
”iptables -t mangle -A PREROUTING -j TEE –gateway [snort-ip]”
”iptables -t mangle -A POSTROUTING -j TEE –gateway [snort-ip]”
```

Then we check the firewall status page on the router web interface in order to confirm the mangle table entries. In order to apply these iptables on startup we create a startup file in /etc/init.d/ on the router. The script for that can be seen in appendices.

Now we can see that snort captures a whole lot more packages. Run snort without -v in order to not see captures in the terminal.

3.3 File exchange

When connection to the company's internal network by VPN or by cable/WiFi onsite the employees should have access to the companys' fileserver. The fileserver in question is a nextcloud service hosted on a ubuntu 18.04.4 server virtual machine with a bridged connection. nextcloud is implemented by running the following commands.

```
”sudo snap install nextcloud”
”sudo nextcloud.manual-install [AdminUserName] [AdminPassword]”
```

And then to add the ability to connect to the site we add the ubuntu hosts IP to nextclouds trusted domains with:

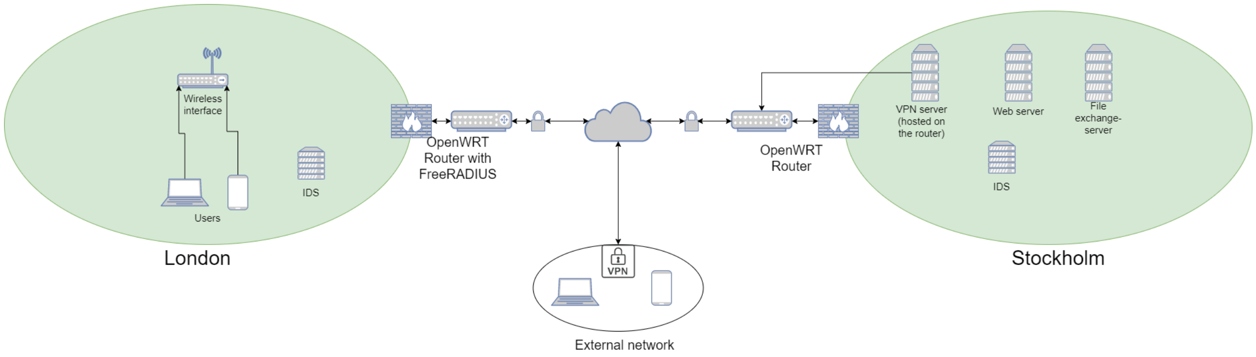
```
”sudo nextcloud.occ config:system:set trusted_domains 1 –value=[IP ADDRESS]”
```

Since we haven't set up a domain name for the server we just add a self-signed certificate for encrypted ssl connections. This is set up with the commands:

```
”sudo nextcloud.enable-https self-signed”
”sudo ufw allow 80,443/tcp”
```

After this one can connect to the nextcloud website with the admin username and password. Then we enable two-factor authentication by entering settings -¿ administration -¿ security and check enforce two-factor authentication. Then enter the apps menu and go to security and install ”Two-Factor TOTP Provider”. This will allow users to use the google authenticator in order to log into the website. Then when adding users one only have to log in and do the TOTP setup with the users company phone.

3.4 Network Topology



3.5 Virtual Machine

We need six VMs in total, they represent four different parts of ACME according to our project, and the OS(operating system) we use is Ubuntu18.04:

1. Internal hosts of Stockholm
2. Internal hosts of London branch
3. Web server in Stockholm
4. VPN server in Stockholm (hosted on router)
5. freeRADIUS server in London (hosted on router)
6. File exchange server in Stockholm

References

[1] What is OpenSSL,Margaret Rouse, https://whatis.techtarget.com/definition/OpenSSL/

[2] Computer Network (version 7), Xiren Xie

[3] Using OpenSSL simulating CA and the issuance of CA certificates, https://www.jianshu.com/p/dcffb46f7a11

[4] https://openwrt.org/docs/guide-user/network/wifi/freeradius

[5] Implementing Mutual SSL Authentication, https://blog.cloudboost.io/implementing-mutual-ssl-authentication-fc20ab2392b3

A Iptables startup script

```
#!/bin/sh /etc/rc.common
START=99
start()
sleep 30 # Make sure FW and iptables have loaded
echo '### STARTING PORT MIRRORING TO [snort-ip] ###'
iptables -t mangle -A POSTROUTING -j TEE –gateway [snort-ip]
iptables -t mangle -A PREROUTING -j TEE –gateway [snort-ip]

run commands:
chmod +x /etc/init.d/script
/etc/init.d/script enable
```

B Wireless config for radius

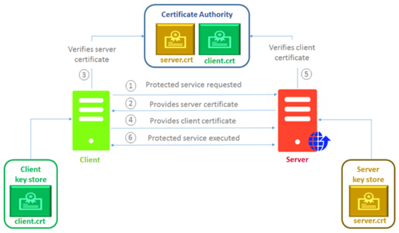
```
config wifi-iface 'default_radio0'
option device 'radio0'
option network 'lan'
option mode 'ap'
option ssid 'OpenWrt'
option encryption 'wpa2+ccmp'
option auth_server '127.0.0.1'
# server IP will be different if your router does not serve RADIUS authentication itself
option auth_secret '**********'
# the auth_secret will be taken from the 'localhost' client in '/etc/freeradius3/clients.conf' file
```

C Web Server

We use Apache web server and configure the Iptables to add only the network address of Stockholm branch to the white list, that is, only the host or devices in the network of the ACME Stockholm can access the web server.
We add the config of SSL/TLS and use OpenSSL for generating the private-key and certification to support HTTPS.

D Authentication using OpenSSL

D.1 Steps of Mutual Authentication[5]


Certificate Authority
server.crt
client.crt
Verifies server certificate
Verifies client certificate
③
⑤
① Protected service requested
② Provides server certificate
④ Provides client certificate
⑥ Protected service executed
Client
Server
Client key store
client.crt
Server key store
server.crt

1. A protected service present on a server is requested by client
2. Server sends its certificate to the client
3. The client verifies the server's certificate
4. Client then sends its certificate to the server
5. The server verifies the client's certificate
6. Once the verification is successful, client is able to execute the protected service

D.2 Implementation Details

1. On CA:
   - (a) Install OpenSSL
   - (b) Generate private key (private key/public key pair) using OpenSSL
   - (c) Generate a self-sign certificate using OpenSSL
   - (d) If there is a CSR, sign it using CA's private key and return the signed certificate using OpenSSL
2. On server:
   - (a) Enable SSL module on Apache2
   - (b) Install OpenSSL
   - (c) Generate private key using OpenSSL
   - (d) Submit a CSR to CA using OpenSSL
   - (e) After receive the the signed certificate from CA, copy the certificate and private key file to the corresponding directory
   - (f) Configure SSL certificate and open the client verification
   - (g) Configure the forced jump of HTTPS
3. On client:
   - (a) Install OpenSSL
   - (b) Generate private key using OpenSSL
   - (c) Submit a CSR to CA using OpenSSL
   - (d) Receive the signed certificate from CA

E OpenVPN setup manual

E.1 Linux

Step 1: Install OpenVPN using command: apt-get install OpenVPN,
Step 2: Download the configure file provided by ACME and unzip it,
Step 3: Start OpenVPN on this configure file with the follwing command: openvpn client.ovpn .

E.2 Windows

Step 1: Download OpenVPN https://openvpn.net/community-downloads/, then install it,
Step 2: Download the configure file provided by ACME and unzip it,
Step 3: Open the configure file *client.ovpn* with Openvpn
Step 4: Start connection

E.3 Mobile devices

Step 1: Download OpenVPN from Playstore(Android) or Appstore(IOS),
Step 2: Download the configure file provided by ACME and unzip it,
Step 3: Open the configure file *client.ovpn* with Openvpn
Step 4: Start connection.