ACME Network Project

GROUP 6

External Security & Web Servers (Keyao Huang)

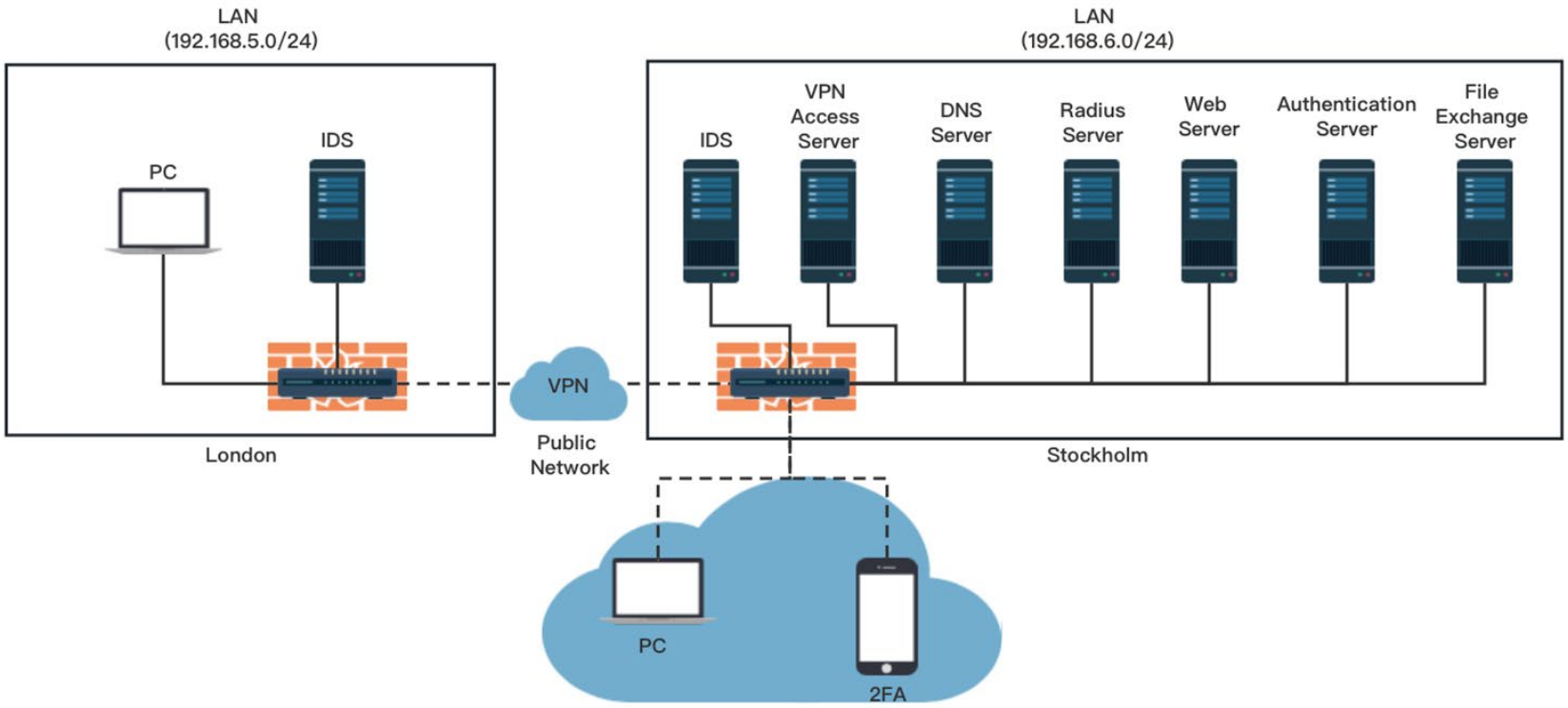
DNS & IDS (Zhiyuan Lin)

Wireless Security & File Exchange (Enze Wang)

CA & OpenSSL (Ziyang Song)

Presentation Date: March 15, 2023

Network Topology



External Security - VPN & 2FA

1. Router VPN

Stockholm Router -> OpenVPN Server

Only Accept One Client (London Router)

2. VPN Access Server

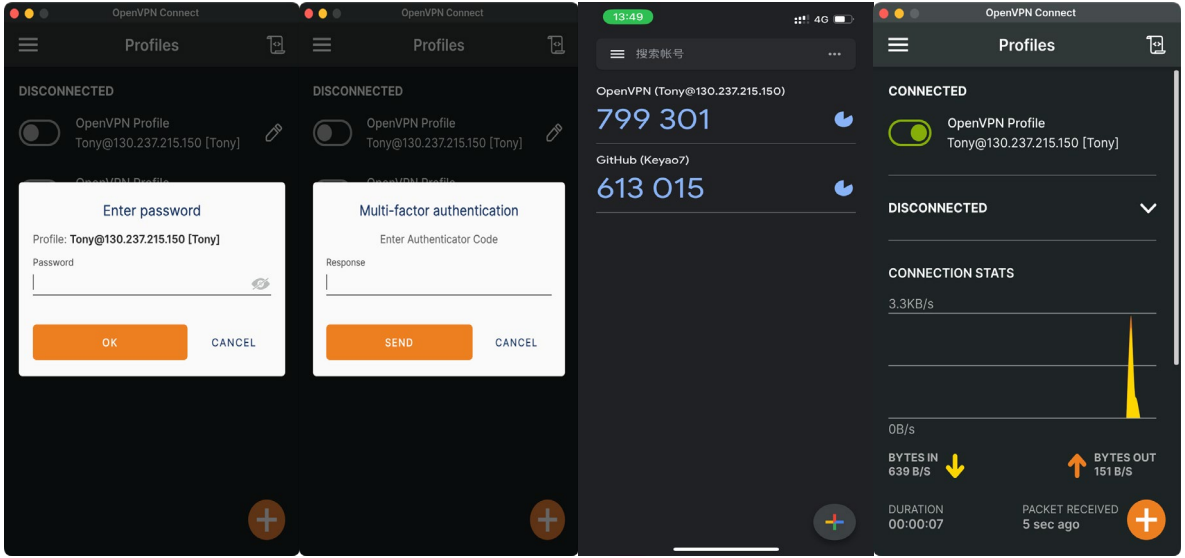
Install OpenVPN-as in VM

Accept External Client

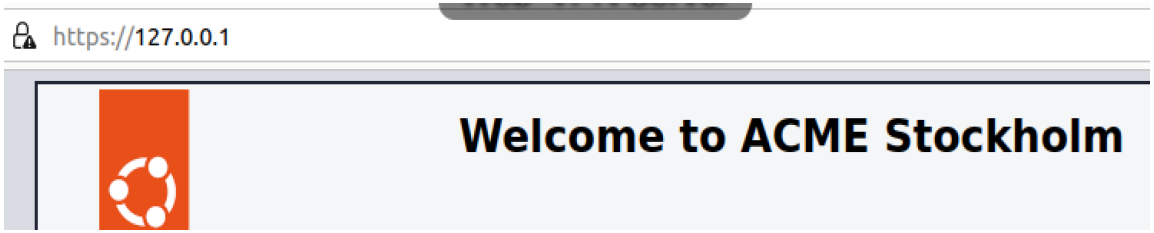
3. VPN 2FA

One Time 6 Digit Pin-Codes

With Google Authenticator



Web Servers

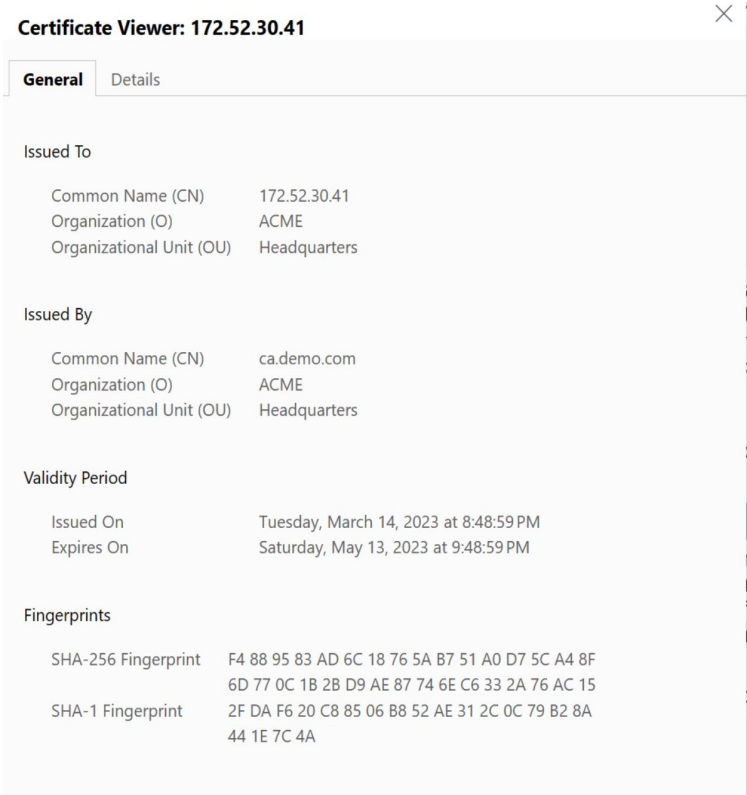


1. Normal Web Server

- Can be access by In-company users and external users
- Using HTTPS

2. Critical Web Server

- Not allow external users to access
- Using iptables to block external users




```
ubuntu@ubuntu:~$ sudo iptables -L
[sudo] password for ubuntu:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  192.168.9.0/24       192.168.6.113        tcp dpt:http-alt
```

Internal Security—IDS

- Snort as an NIDS, with mirrored traffic from router to get more coverage
- Configure the local rules file or download the rules from SNORT website.
- Set up snort.conf file, test and run

Table: Mangle

Chain *PREROUTING* (Policy: *ACCEPT*, 53435 Packets, 27.04 MB Traffic)

Pkts.	Traffic	Target	Prot.	In	Out	Source	Destination	Options	Comment
53.94 K	27.34 MB	TEE	all	*	*	0.0.0.0/0	0.0.0.0/0	TEE gw:192.168.6.113	-

Chain *POSTROUTING* (Policy: *ACCEPT*, 164503 Packets, 88.22 MB Traffic)

Pkts.	Traffic	Target	Prot.	In	Out	Source	Destination	Options	Comment
164.50 K	88.22 MB	TEE	all	*	*	0.0.0.0/0	0.0.0.0/0	TEE gw:192.168.6.113	-

```
"
#-------------
# LOCAL RULES
#-------------
alert icmp any any -> $HOME_NET any (msg:"ICMP Packet Detected";sid:1000001;rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH incoming"; flow:stateless; flags:S+; sid:1000002; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"Potential SSH Brute Force Attack"; flow:to_server; flags:S; threshold:type threshold, track by_src, count 3, seconds 60; classtype:atten
```

DNS

- BIND9 is used for deploying the DNS service for STOCKHOLM network

- Configure db.demo.com and db.168.192 and named.conf.local files

db.168.192
/etc/bind

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     demo.com. root.demo.com. (
                                1         ; Serial
                           604800         ; Refresh
                            86400         ; Retry
                          2419200         ; Expire
                           604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
1.0.0   IN      PTR     localhost.
233.6 IN PTR demo.com   ;
133.6.168.192.in-addr.arpa.  IN   PTR     radius.demo.com
113.6.168.192.in-addr.arpa.  IN   PTR     web.demo.com
184.6.168.192.in-addr.arpa.  IN   PTR     nextcloud.demo.com
```

db.demo.com
/etc/bind

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     demo.com. root.demo.com. (
                                2         ; Serial
                           604800         ; Refresh
                            86400         ; Retry
                          2419200         ; Expire
                           604800        ; Negative Cache TTL
                          )
@       IN      NS      localhost.
@       IN      A       192.168.6.233
web     IN      A       192.168.6.113
radius  IN      A       192.168.6.133
nextcloud  IN   A       192.168.6.184
;@      IN      AAAA    ::1
* IN  A  192.168.6.233   ;
; Add the DNSKEY records
demo.com.  IN  DNSKEY  257 3 8 ( AwEAAbSFZZlOofeKT2k6fqCx5DAR/R4i1FPSs8jSHUVs/VV2Vj5WT7Zn
TIW6pKljI8SNVQruC0e8WXjswlskG9Pxp4WgiIXuPSgrW5TwRKme4SIE 1XlDgD6B/ycyovG/nAp/
ycA6PaL8ahMBxVcGNSYthoHK8PlfCEP6Svvi 460rta/LL2OUYweDi0G7Y1ak7Cnc1V4Txs6COKMLKLg+3drXS1fvz06c
i9Ev/2if7nEdYS819dTw6jGhPy+h2jgNfNeX85kWJM/R7OL4vQ/wJiQ8
ZPvQ1nS6D+CrTJNHNlzImi0jgSzEHveUhrXjNCWQ1ttRFBPABVFYcDL1 iZa6jZ0rA00= )
demo.com.  IN  DNSKEY  256 3 8 ( AwEAAdPT7PjlsrgNIV+weyrXOtRqvJexjgOd5VNHvb05RVzkJRXZbU/u
VxgtlQLeWIgRgOZ9TfkS3b3DON15eKa29K17NBG00lX/8NgU8V4OxzVH
8Kpj+zcBsaprAIv+WtUzP9wYCPTKacDBqEnrHhdxuTC4PEr4t1wm21P3 nozFviEObCOMQh/ZnpH0D93l/
uI7xg8G6DqXhLaB0xIIaRDehNH17zSB ZcP3C3v4SkotbnQ5tMNL1ql5ODK3qFDAd5VNm9CyTYK9IRJUP1N9eRZR
DTe5JkNXpyujJjO9YeJpBWBdTgX7i7pmjK79XJtRokTMLak2I+RD2aIA GYoUhh4+b7U= )
```

Wireless Security: Radius Server

Main configuration for freeradius :

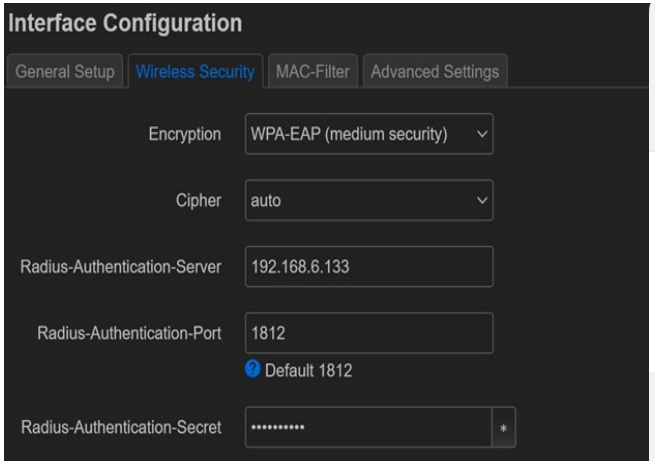
1) at /freeradius/3.0/clients.conf set two routers as the NAS

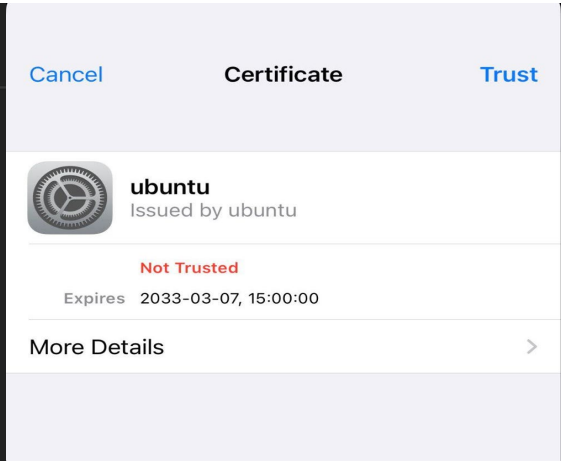
2) edit /freeradius/3.0/users file which includes employees' user/password information

3) under /freeradius/3.0/certs/client.cnf, modify the cnf file and run "make" generate user certificates

Set WPA-EAP as the encryption

for router and server on OpenWRT :





Wireless Security: Radius Server

Start the debug mode : “freeradius -X”, and “ready to process request” is shown in the terminal.

```
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tunnel
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 40307
Listening on proxy address :: port 47073
Ready to process requests
```

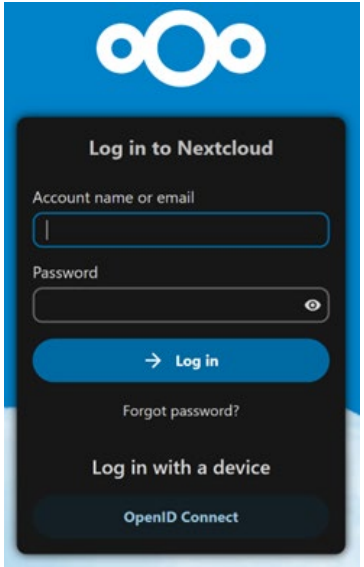
Now, employees can access the wireless internet through radius authentication:

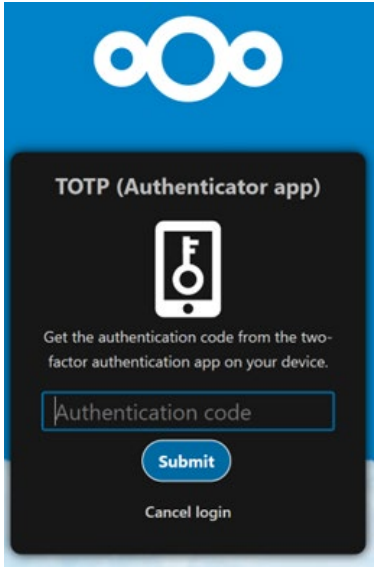
1) A user connects to the Network Access Server (router) and initiates a login.
2) The NAS (router) communicates with the RADIUS server using a shared secret mechanism that the port 1812 is for authentication.
3) The NAS sends a RADIUS message (an Access-Request) to the server.

File Exchange server: Nextcloud

We run nextcloud server with docker container. So, we run the server using Docker Desktop.

Sever is running, you should be able to login Nextcloud at 192.168.6.184:9006 with 2FA. After successfully logging in, users should be able to chat and shire files. For example, one way to share files is transferring ownership.





Document Structure

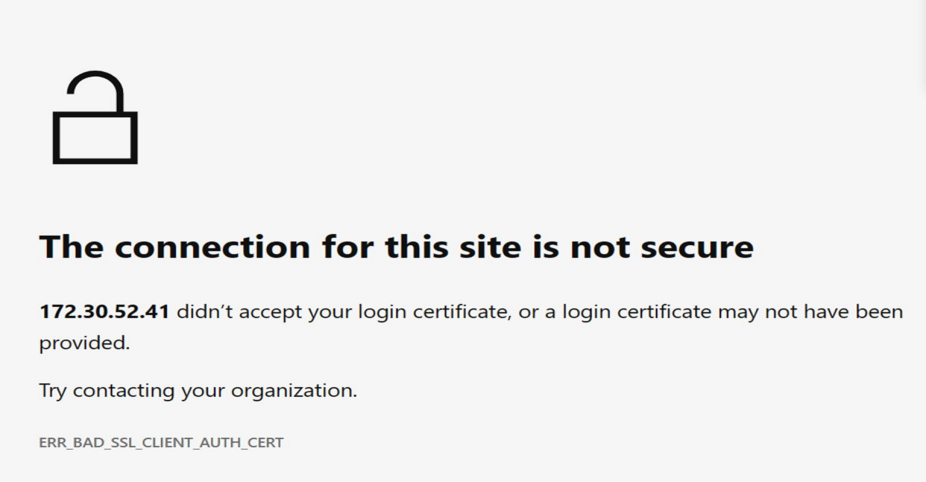
- *.key Private Key
- *.pem Certificate or Certificate chain file
- *.csr Certificate signing request (CSR)
- *.cnf Configuration File
- *.crl Certificate revocation list (CRL)
- *.vrfy Certificate Verification

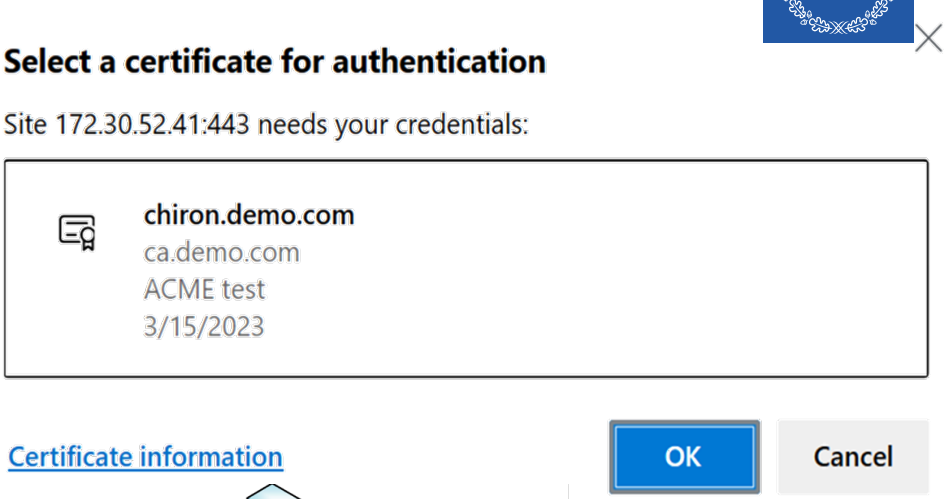
All files use PEM format.

Hierarchy:

1. Root CA
(External, with full chain)
2. Intermediate CA
(Internal, maintaining local CRL)
3. Server or User
(With mutual TLS verification)

Web Server certificate and authentication





General Structure of Trust (Imaginary)