

Unit 9 Lab - Certificate and Key Madness

Setting up Rsyslog with TLS

Complete the lab: <https://killercoda.com/het-tanis/course/Linux-Labs/211-setting-up-rsyslog-with-tls>

```
KILLERCODA 50 min
StreamDriver: "gtls"
StreamDriver: AuthMode: "anon"
}

input {
  type: "tcp"
  port: "6514"
}

Hit esc: wq to write and quit

You know that systems do not take configuration
file changes without a restart of the service, so
restart rsyslog

systemctl restart rsyslog

systemctl status rsyslog --no-pager

Verify that your system is listening on port 6514 for
TCP traffic.

ss -ntulp | grep 6514

You are ready to head to the next part of the lab.

CHECK
```

```
KILLERCODA 50 min
Let's do Linux Security
Setup rsyslog on node01 to sent logs
over TLS

So far you have enabled rsyslog collection on
controlplane on TCP Port 6514.

Now you must configure the node01 server to
securely send logs over the correct port.

Solution
▶ Solution

BACK CHECK

Validation successful
```

```

KLLKCODA  [img alt="GitHub icon"/> [img alt="Discord icon"/> PLUS
v2 /root/.ssh/config
Add this to the bottom of the file.

# Custom TLS Client
=====

global(DefaultHostStreamVerifierFile="/usr/local/share/
action(type="onoff" protocol="tcp" target="control")

*.*@centralline:6514

Restart the service

systemctl restart rsyslog

Create a user

useradd -s somuser

Exit back to controlplane

exit

Verify that the node01 system logs are being
pushed over to controlplane

grep somuser /var/log/*

Where did the logs land for the user that was
created?
```

Review Solving the Bottom Turtle

Review pages 41-48 of <https://spiffe.io/pdf/Solving-the-bottom-turtle-SPIFFE-SPIRE-Book.pdf>

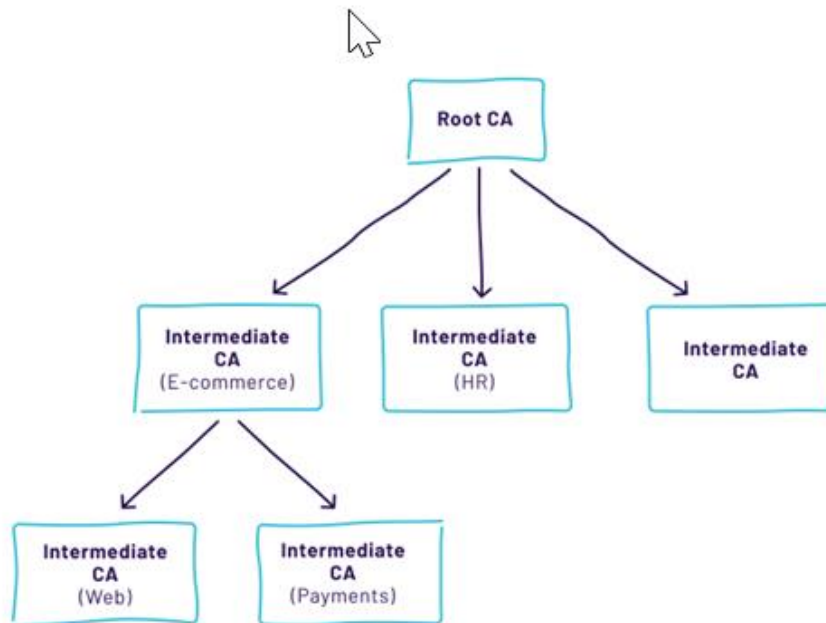


Fig. 3.3: Illustration of intermediate certificate authorities.

Does the diagram on page 44 make sense to you for what you did with a certificate authority in this lab?

SSH – Public and Private key pairs

Complete the lab: <https://killercoda.com/het-tanis/course/Linux-Labs/212-public-private-keys-with-ssh>

```

KLLKCODA  @  PLUS
Your public key has been saved in ProLUG.pub
The key fingerprint is:

Inspect what was generated for permissions and ownership from a Linux
system perspective.

ls -l

Can you tell the public from the private version of the key?

What are the permissions of the two keys?

Test the keys for cryptographic similarity.

ssh-keygen -y -e -f ProLUG
ssh-keygen -l -f ProLUG

ssh-keygen -y -e -f ProLUG.pub
ssh-keygen -l -f ProLUG.pub

Do you see that they have the exact same fingerprint?

Delete the .pub public key and regenerate it from the private key.

rm -rf /root/.ssh/ProLUG.pub

Validate that you do not see the key

ls -l

Generate a new public key from the private one.

ssh-keygen -f ProLUG -y > ProLUG.pub
```

```
controlplane:/keyfs $ ls -l
total 8
-rw-r----- 1 root root 411 Jan 13 04:29 PROLOG
-rw-r----- 1 root root 99 Jan 13 04:29 PROLOG.pub
--BEGIN SSHd PUBLIC KEY -----
controlplane:/keyfs $ ssh-keygen -y -f PROLOG
comment: "256-bit 1025519", converted by root@controlplane from open
SSH
AAAAB3NzaC1yd29lIHR5cGU9IHNpdGUiDjEwMzUxOTQ5bWVudC5hcXNlaTJyYXk/CXNlaH
-----END SSHd PUBLIC KEY-----
controlplane:/keyfs $ ssh-keygen -l -f PROLOG
256 SHA256:Zc1Z7tR1tESAAWMKkdnoDSc8upmH/H4XISq/bacTYijP/cXNlaH
controlplane:/keyfs $ ssh-keygen -l -f PROLOG.pub
256 SHA256:Zc1Z7tR1tESAAWMKkdnoDSc8upmH/H4XISq/bacTYijP/cXNlaH
controlplane:/keyfs $ ssh-keygen -y -f PROLOG.pub
--BEGIN SSHd PUBLIC KEY -----
controlplane:/keyfs $ ssh-keygen -y -f PROLOG.pub
comment: "256-bit 1025519", converted by root@controlplane from open
SSH
AAAAB3NzaC1yd29lIHR5cGU9IHNpdGUiDjEwMzUxOTQ5bWVudC5hcXNlaTJyYXk/CXNlaH
-----END SSHd PUBLIC KEY-----
controlplane:/keyfs $ ssh-keygen -l -f PROLOG
256 SHA256:Zc1Z7tR1tESAAWMKkdnoDSc8upmH/H4XISq/bacTYijP/cXNlaH
controlplane:/keyfs $
```

```
ctrl@kali:~/keys$ rm -rf /root/.keys/prodLug.pub
ctrl@kali:~/keys$ ls -l
total 4
-rw-r----- 1 root root 411 Jun 13 04:29 ProDUg
ctrl@kali:~/keys$ ssh-keygen -f ProDUg.y > prodLug.pub
ctrl@kali:~/keys$ ls -l
total 8
-rw-r----- 1 root root 411 Jun 13 04:29 ProDUg
-rw-r----- 1 root root 609 Jun 13 04:32 prodLug.pub
ctrl@kali:~/keys$ cat prodLug.pub
--- BEGIN SSH PUBLIC KEY ---
Comment : "256-bit ECDHE521, converted by root@kali:~/keys from Open
AAAC3WzCk1ZDITEHSAANIDbkPDSvCBspium+JHkQc3IibGtYjYpbC/CxS4B
E="
--- END SSH PUBLIC KEY ---
256 SHA256:EPVDT07HXKULJHxwK4GSe6qU4GDXX0Xnctng ctrl@kali:~/control
lines (ECDHE521)
```

