PKI - Root-CA | Come fare # TEST-001

PKI | ROOT CA OFFLINE

VM rca001 · Debian 13 Trixie · OpenSSL

Serie PKI — Parte 1 di 3 · Prerequisito: nessuno · Parte 2: Intermediate CA · Parte 3: Certificati e distribuzione

Indice

Prefazione

La PKI del virtual lab trk è strutturata su due livelli distinti:

- Root CA offline (rca001) — l'autorità radice. Non è mai connessa alla rete dopo l'installazione. Firma esclusivamente la Intermediate CA.
- Intermediate CA online (ica001, Step-CA) — emette i certificati per tutti i servizi del lab: OPNsense, Webmin, NethServer, Nextcloud e altri.

La Root CA è offline per proteggere la chiave privata più critica dell'infrastruttura. Se venisse compromessa, tutta la PKI andrebbe ricostruita da zero. Tenerla offline e spegnere la VM dopo ogni utilizzo riduce il rischio al minimo.

In questo lab utilizziamo Debian 13 Trixie con OpenSSL. In produzione la Root CA vive su un HSM, su un laptop air-gapped conservato in cassaforte, o su una VM con disco cifrato LUKS su storage dedicato e isolato dalla rete.

Requisiti

Download ISO Debian

```
ISO netinstall : https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/
SHA256SUMS : stessa directory — scaricare insieme all'ISO
$ sha256sum -c SHA256SUMS --ignore-missing
Verifica che l'ISO scaricata corrisponda al checksum ufficiale. Output atteso: debian-XX-amd64-netinst.iso: OK. Eseguire nella stessa cartella dove si trovano ISO e SHA256SUMS.
```

Alternativa grafica: GtkHash — aprire l'ISO, incollare il valore SHA256 dal file SHA256SUMS, cliccare Verifica.

Configurazione VM

```
Hostname : rca001
OS : Debian 13 Trixie (64-bit)
RAM : 4096 MB (4 GB)
vCPU : 1
Disco : 16 GB
Video Memory : 32 MB — necessario per la GUI LXDE
Firmware : UEFI
Audio : disabilitato
Rete : Rete interna — LAB-TRK-D — da rimuovere in Fase 07
IP : 10.1.5.205/24 | GW: 10.1.5.254 | DNS: 10.1.5.253
```

Installazione Debian — punto critico

Durante la partizione disco selezionare: Guidato — usa intero disco e imposta LVM cifrato. Impostare la passphrase LUKS quando richiesto — salvarla in KeePassXC su cet001 (voce: rca001 — LUKS Disk). Senza questa passphrase la VM non si avvia.

In tasksel non selezionare nessun desktop — installare solo 'standard system utilities'. Il desktop LXDE si installa manualmente dopo.

Software da installare dopo l'avvio iniziale

Dopo la prima installazione Debian (senza desktop in tasksel), da root:

```
# apt install -y lxde-core mousepad
Installa il desktop LXDE minimale e l'editor di testo Mousepad. lxde-core contiene solo il necessario — nessun gioco, nessun player multimediale.

# systemctl enable lightdm && systemctl set-default graphical.target
```

Abilita il display manager e imposta il target grafico come default al boot.

```
# reboot
```

Al riavvio LXDE parte automaticamente con la schermata di login grafica.

KeePassXC — password da creare prima di iniziare

```
Applicazione : KeePassXC installata su cet001
Gruppo : vlab.trk.lab / PKI
Voce da creare : rca001 — Root CA Key
Password : generare con KeePassXC — minimo 20 caratteri
 : il generatore integrato garantisce entropia sufficiente
```

La passphrase Root CA Key protegge la chiave privata ca.key.pem. Senza di essa non è possibile firmare nessun certificato. Perderla significa ricostruire tutta la PKI da zero.

Accesso e trasferimento file

```
Accesso : console VirtualBox con desktop LXDE — nessun SSH
File I/O : share VirtualBox montata automaticamente su /media/sf_share/
Clipboard bidirezionale per copiare testo tra host e VM
Share : cartella sul host ~/Scaricati/Share/ — crearla se non esiste
In produzione: USB cifrata LUKS — nel lab la share è sufficiente
```

[Fase 01]

Guest Additions e cartella condivisa

I VirtualBox Guest Additions abilitano due funzionalità indispensabili: il clipboard bidirezionale (copia/incolla tra host e VM) e le cartelle condivise. Senza Guest Additions il clipboard non funziona anche se abilitato nelle impostazioni VirtualBox.

Con rca001 avviata e LXDE in esecuzione — inserire il CD dei Guest Additions:

- Menu VirtualBox → Dispositivi → Inserisci immagine CD Guest Additions
- LXDE monta il CD automaticamente — compare come icona sul desktop o in /media/cdrom0

Aprire il terminale LXDE e diventare root:

```
$ su -
```
Accede alla shell root. Necessario per installare i moduli kernel dei Guest Additions.

```
# sh /media/cdrom0/VBoxLinuxAdditions.run
```
Avvia l'installazione dei Guest Additions. Compila il modulo kernel vboxsf per le cartelle condivise e vboxvideo per il clipboard. L'operazione richiede qualche minuto.

```
# usermod -aG vboxsf th3r0ck
```
Aggiunge l'utente th3r0ck al gruppo vboxsf. Necessario per accedere alla cartella condivisa senza permesso negato. Effettivo al prossimo login.

```
# reboot
```
Riavvia per caricare i nuovi moduli kernel e applicare il cambio di gruppo.

Con rca001 spenta — configurare le impostazioni e la cartella condivisa in VirtualBox:

- VirtualBox → Impostazioni rca001 → Generale → Avanzate → Clipboard → Bidirezionale

Il clipboard bidirezionale permette di copiare la passphrase da KeePassXC su cet001 e incollarla nel terminale di rca001.

- Creare la cartella sul host: mkdir -p ~/Scaricati/Share/
- VirtualBox → Impostazioni rca001 → Cartelle condivise → Aggiungi (+)
- Percorso cartella: /home/th3r0ck/Scaricati/Share/
- Nome condivisione: share
- Punto di mount: /media/sf_share

- Spuntare: Montaggio automatico — Permanente — OK
- Avviare rca001

Dopo l'avvio — verificare da terminale (come th3r0ck):

```
$ ls /media/sf_share/
```
Deve mostrare il contenuto di ~/Scaricati/Share/ sul host. Se dà permesso negato: verificare di aver fatto il logout dopo usermod e che il gruppo vboxsf sia attivo.

La rete è ancora attiva in questa fase — verrà rimossa definitivamente nella Fase 07.

[Fase 02]

Struttura directory CA

OpenSSL richiede una struttura precisa di cartelle e file di stato per gestire la CA. Crearla manualmente garantisce il pieno controllo su dove vengono salvati chiavi, certificati e log.

Aprire il terminale LXDE e diventare root:

```
$ su -
```
Tutte le operazioni della CA vengono eseguite come root — la chiave privata è in /root/ca/private/ accessibile solo a root.

```
# mkdir -p /root/ca/{certs,crl,newcerts,private}
```
Crea le quattro cartelle necessarie in un solo comando tramite brace expansion. certs = certificati emessi, crl = liste di revoca, newcerts = copia di ogni certificato firmato (richiesta da OpenSSL), private = chiave privata.

```
# chmod 700 /root/ca/private
```
Restringe l'accesso alla cartella della chiave privata al solo utente root. Nessun altro processo può leggere il contenuto.

```
# touch /root/ca/index.txt
```
Crea il database dei certificati emessi. OpenSSL lo aggiorna automaticamente ad ogni firma — è il registro ufficiale della CA.

```
# echo 1000 > /root/ca/serial
```
Imposta il numero seriale iniziale. Ogni certificato firmato riceve un numero progressivo univoco — permette di identificarlo e revocarlo se necessario.

[Fase 03]

File di configurazione openssl.cnf

Il file openssl.cnf definisce il comportamento completo della CA: percorsi, algoritmi, policy di firma, estensioni X.509. Senza questo file ogni operazione richiederebbe decine di parametri da riga di comando.

Aprire Mousepad dal menu LXDE (Accessori → Mousepad). Copiare il contenuto del file qui sotto e incollarlo in Mousepad con Ctrl+V. Salvare con File → Salva come → /home/th3r0ck/Documenti/openssl.cnf

Contenuto completo del file openssl.cnf:

```
[ ca ]
default_ca = CA_default

[ CA_default ]
dir = /root/ca
certs = $dir/certs
crl_dir = $dir/crl
new_certs_dir = $dir/newcerts
database = $dir/index.txt
serial = $dir/serial
RANDFILE = $dir/private/.rand
private_key = $dir/private/ca.key.pem
certificate = $dir/certs/ca.cert.pem
crlnumber = $dir/crlnumber
crl = $dir/crl/ca.crl.pem
crl_extensions = crl_ext
default_crl_days = 180
default_md = sha256
name_opt = ca_default
cert_opt = ca_default
default_days = 3650
preserve = no
policy = policy_strict

[ policy_strict ]
countryName = match
stateOrProvinceName = optional
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

```
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
string_mask = utf8only
default_md = sha256
x509_extensions = v3_ca
prompt = no

[ req_distinguished_name ]
C = IT
O = TRK Lab
CN = TRK Lab Root CA

[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ v3_intermediate_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ crl_ext ]
authorityKeyIdentifier = keyid:always

[ server_cert ]
basicConstraints = CA:FALSE
nsCertType = server
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
```

Dopo aver salvato il file in /home/th3r0ck/Documenti/, copiarlo nella directory CA da terminale root:

```
# cp /home/th3r0ck/Documenti/openssl.cnf /root/ca/openssl.cnf
```

Copia il file nella directory della CA. Da questo momento OpenSSL usa sempre questo file per tutte le operazioni.

```
# cat /root/ca/openssl.cnf
```

Verifica che il file sia arrivato completo. Controllare che siano presenti le sezioni [ca], [req], [v3_ca] e [req_distinguished_name] con C, O, CN.

prompt = no è obbligatorio per OpenSSL 3.x su Debian 13. Senza questa riga il comando di generazione certificato dà errore 'No objects specified in config file'.

policy_strict significa che la CA firma solo certificati con lo stesso Country e Organization. Impedisce di firmare certificati per organizzazioni diverse da TRK Lab.

pathlen:0 in v3_intermediate_ca impedisce alla Intermediate CA di firmare altre CA intermedie — può emettere solo certificati finali. Standard di sicurezza per PKI a due livelli.

[Fase 04]

Generazione chiave privata Root CA — 4096 bit, AES-256

La chiave privata è il cuore della Root CA. Chiunque la possieda può firmare certificati a nome di TRK Lab. Va protetta con una passphrase forte e non deve mai uscire da rca001.

Prima di procedere: aprire KeePassXC su cet001, voce 'rca001 — Root CA Key', copiare la password. Si incollerà nel terminale di rca001 con Ctrl+Shift+V quando richiesto.

```
# openssl genrsa -aes256 -out /root/ca/private/ca.key.pem 4096
```

Genera una chiave RSA a 4096 bit cifrata con AES-256. OpenSSL chiede la passphrase due volte (Enter PEM pass phrase + Verifying). Incollare con Ctrl+Shift+V da KeePassXC. Nessun carattere viene mostrato a schermo durante l'input — è normale.

```
# chmod 400 /root/ca/private/ca.key.pem
```

Imposta i permessi in sola lettura per root. Nessun altro utente o processo può accedere alla chiave.

Verifica che la chiave sia corretta:

```
# openssl rsa -check -noout -in /root/ca/private/ca.key.pem
```

Verifica l'integrità della chiave senza stamparla. Output atteso: RSA key ok. OpenSSL chiede la passphrase — incollare da KeePassXC.

4096 bit è il minimo raccomandato per una Root CA con validità 20 anni. In produzione alcuni standard richiedono 8192 bit per CA con validità superiore.

[Fase 05]

Generazione certificato Root CA — 20 anni

Il certificato Root CA è il documento pubblico che identifica la CA. È auto-firmato: non esiste nessuna autorità superiore che lo certifica — la Root CA è l'autorità massima della PKI.

```
# openssl req -config /root/ca/openssl.cnf -key /root/ca/private/ca.key.pem -new -x509 -days 7300 -sha256 -extensions v3_ca -out /root/ca/certs/ca.cert.pem
```

Genera il certificato auto-firmato. -days 7300 = 20 anni di validità. I dati del certificato (C, O, CN) vengono letti automaticamente da openssl.cnf senza richiesta interattiva. OpenSSL chiede solo la passphrase — incollare da KeePassXC.

```
# chmod 444 /root/ca/certs/ca.cert.pem
```

Il certificato pubblico è leggibile da tutti ma non modificabile. Va distribuito a tutti i client e server del lab.

[Fase 06]

Verifica certificato Root CA

Prima di distribuire il certificato, verificare che contenga esattamente i dati corretti. Un errore in questa fase invaliderebbe tutta la PKI.

```
# openssl x509 -noout -text -in /root/ca/certs/ca.cert.pem
```

Mostra il contenuto completo del certificato in formato leggibile. Verificare i campi elencati di seguito. L'output è lungo — scorrere fino ai campi rilevanti.

Campi da verificare nell'output:

- Issuer e Subject: entrambi C=IT, O=TRK Lab, CN=TRK Lab Root CA (auto-firmato = Issuer uguale a Subject)
- Validity: Not After deve essere circa 20 anni dalla data odierna
- Public Key Algorithm: rsaEncryption — Public-Key: (4096 bit)
- X509v3 Basic Constraints: critical — CA:TRUE
- X509v3 Key Usage: Digital Signature, Certificate Sign, CRL Sign

```
# openssl verify -CAfile /root/ca/certs/ca.cert.pem /root/ca/certs/ca.cert.pem
```

Verifica la validità crittografica del certificato rispetto a se stesso. Output atteso: /root/ca/certs/ca.cert.pem: OK

Se l'output non è OK, non proseguire. Ripetere le Fasi 04 e 05 — qualcosa è andato storto nella generazione.

[Fase 07]

Export certificato pubblico e rimozione rete

Il certificato pubblico (ca.cert.pem) deve essere distribuito a tutti i sistemi del lab perché possano verificare i certificati firmati dalla CA. La chiave privata (ca.key.pem) non esce mai da rca001.

```
# cp /root/ca/certs/ca.cert.pem /media/sf_share/pki/ca.cert.pem
```

Copia il certificato pubblico nella cartella condivisa VirtualBox, sottocartella pki/. Da qui sarà accessibile all'host e alle altre VM del lab. In produzione questo passaggio avviene tramite USB cifrata con LUKS.

```
# cp /root/ca/openssl.cnf /media/sf_share/pki/openssl-rca.cnf
```

Esporta la configurazione OpenSSL nella share — necessaria nella Parte 2 per firmare la Intermediate CA.

Rimozione rete da VirtualBox — spegnere la VM:

```
# poweroff
```

Spegne rca001. La rimozione della NIC avviene con la VM spenta.

Con rca001 spenta — rimuovere la NIC in VirtualBox:

- VirtualBox → Impostazioni rca001 → Rete → Scheda 1 → deselezionare Abilita scheda di rete → OK
- rca001 non si riconnette mai alla rete — si riaccende solo per firmare la Intermediate CA

In produzione: rimozione fisica della scheda di rete, VM su storage cifrato LUKS, accesso fisico ristretto. Nel lab virtuale è sufficiente disabilitare la NIC in VirtualBox.

Informazioni e Documentazione

- Doc. — OpenSSL: https://www.openssl.org/docs/
- Doc. — openssl-req(1): man openssl-req
- Doc. — openssl-genrsa(1): man openssl-genrsa
- Doc. — openssl-x509(1): man openssl-x509
- Blog. — PKI Design (Jamie Nguyen): https://jamielinux.com/docs/openssl-certificate-authority/
- Inf. — X.509 standard: https://en.wikipedia.org/wiki/X.509
- Inf. — Debian 13 Trixie: https://www.debian.org/releases/trixie/

Crediti

Autore: th3r0ck alias "trk"

- Blog — https://www.btrk.it/
- YouTube — https://www.youtube.com/@th3r0ck48

Grazie alla community open source: Debian Project, OpenSSL Project, Oracle VirtualBox.

Software utilizzati per produrre questi contenuti:

- LibreOffice — guide PDF
- OBS Studio — registrazione video
- Okular — visualizzazione PDF durante la registrazione
- GIMP — immagini
- Inkscape — grafica vettoriale
- Shotcut — montaggio video

Sostieni il software libero

I progetti elencati in questa guida sono distribuiti gratuitamente dalla loro comunità di sviluppatori.
Molti offrono la possibilità di fare una donazione direttamente dal loro sito ufficiale.

Se usi questi strumenti nel tuo lavoro o nel tuo studio, considera di contribuire — anche una piccola donazione aiuta a mantenere attivo lo sviluppo e la documentazione.

Disclaimer

Questo video tutorial, la guida e l'articolo nel blog sono prove in laboratorio a scopo didattico.
Il curatore non si assume alcuna responsabilità sull'uso del contenuto.

L'implementazione in produzione è responsabilità di professionisti del settore,
in possesso delle competenze necessarie.

Questo progetto nasce per appassionati e amatori del mondo IT.
Impara, sperimenta, divertiti.