Cisco IOS Certificate Authority Server

Keith Bogart

CCIE #4923

- kbogart@ine.com
- @keithbogart1
- linkedin.com/in/keith-bogart-2a75042

CCIE Routing & Switching

Topic Overview

- Why Use A Cisco IOS Certificate Authority?
- Prerequisites For Creating A Cisco IOS CA
- Cisco IOS CA Minimal Configuration
- Cisco IOS CA Verification
- IOS CA Server Feature Options

IOS CA Server

- Certificate Authorities come in two form-factors:
  - Servers: Companies like Verisign and DigiCert utilize Servers to implement their Certificate Authority functionality
  - Networking Devices: A Cisco IOS Router can be configured as a CA
- Site-To-Site VPNs can utilize Routers as their CA
- All other devices (routers, firewalls, etc) that require Certificates will point to this device as their "Trustpoint"
- Trustpoint: Cisco IOS term for the Certificate Authority

Clearly if the Digital Certificate on your system (let's say a web server) is to be trusted by anyone in the Internet, it must be signed by a known, public Root Authority. Those Root Authorities use Server OS (such as Microsoft, Linux, or others) to run their CA functionality.

-

If Certificates will be installed on routers or Firewalls within your organization, for use with forming VPNs with other devices also owned by your organization…you don't need to utilize public Root Authorities. In this case, you can configure a router as the Root Authority which grants Certs to other devices within your organization.

-

You could also configure a Server within your company to do this job, but many documents state that by utilizing a router for this function, it frees up your servers for other tasks.

Prerequisites For IOS CA Server

- An IOS CA Server needs a self-signed Digital Certificate before it can sign the Identity Certificates of other devices.
- The following features/protocols must be preconfigured prior to creating a self-signed Digital Certificate:
  - Time/Date should be accurate (NTP recommended)
  - IP HTTP Server functionality
  - RSA Keypair
    - Can be manually created or dynamically derived

Look to other videos by INE to describe the process of creating RSA Keypairs and configuration of NTP (Network Time Protocol).

-

Cisco routers (and Firewalls) will enroll their Certificate to the IOS CA Server using SCEP (Simple Certificate Enrollment Protocol). The IP HTTP Server functionality must be enabled on the IOS CA Server so that it can process these SCEP requests because SCEP utilizes HTTP messages.

IOS CA Server

- Cisco IOS CA Server Configuration tasks (minimum configuration):

```
ntp server x.x.x.x
ip http server
!
crypto pki server <keypair name>
 grant auto
 no shut
```


RSA Keypair will be dynamically generated, using the name of whatever you specified in the command, “crypto pki server <keypair name>”.

-

The RSA keypair will NOT be exportable. This is typically a good thing, unless you’re doing HSRP or some other FHRP and need one router to serve as a Backup CA to another router, in which case they’d both need the same CA Server configuration AND the same keypairs. The only way to do that is to ensure the keypair is exportable so you can copy it from one router…and import it into another router.

-

See the next slide for other default behaviors as a result of this command.

IOS CA Server Verification

```
Router-1#show crypto pki server
Certificate Server R1-Key:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=R1-Key
    CA cert fingerprint: C6DBFE28 70C7A72B 170EF22A 3CD7A272
    Granting mode is: auto
    Last certificate issued serial number (hex): 1
    CA certificate expiration timer: 09:04:46 UTC Oct 21 2021
    CRL NextUpdate timer: 15:04:46 UTC Oct 22 2018
    Current primary storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```


Notice the defaults:

-

The CA's self-signed Root Cert defaults to a lifetime of about 3-years.

-

When Identity Certificates are granted, they are stored in NVRAM. You may want to change this to another location if you expect this CA to grant a lot of Certs.

-

Database level is "minimum": This means that the only things stored in the Database are items critical for the CA to function as a CA (such as it's own Certificate, CRL and public/private keys). It will not store anything about any Identity Certificates that it issues at this level. If you wish the IOS CA to store that information, you need to change the "database level" command to either "names" or "complete".

IOS CA Server Options

```
Router-1(config)#crypto pki server R1-Key
Router-1(cs-server)#?
CA Server configuration commands:
  auto-rollover  Rollover the CA key and certificate
  cdp-url        CRL Distribution Point to be included in the issued certs
  crl            server crl
  database       Certificate Server database config parameters
  default        Set a command to its defaults
  eku            Configure EKU parameters
  exit           Exit from Certificate Server entry mode
  grant          Certificate granting options
  hash           Hash algorithm
  issuer-name    Issuer name
  lifetime       Lifetime parameters
  mode           Mode
  no             Negate a command or set its defaults
  redundancy     sync this server to the standby
  serial-number  Serial Number of Last Certificate Issued
  show           Show this certificate server configuration
  shutdown       Shutdown the Certificate Server
```


In the remainder of this video I'm going to talk about some of these other options you can configure to give yourself finer control over the Cisco IOS CA Server operation. However I'm not going to exhaustively cover each and every one.

Database Options

```
Router-1(cs-server)#database ?
  archive   Backup Certificate Server Signing Certificate and Keys
  level     Level of data stored in database
  url       URL the Certificate Server database information will be written to
  username  Database username to access the primary network storage
```

- Archive: Allows you to create a backup file containing CA Server's signing Certificate and keys as either .pem or PKCS#12 format.
  - Format you select will be dependent on what type of platform will be importing this information.
  - You will also need to specify an 8-character (minimum) password.
- Level: Controls what information will be stored within the Database.
  - Minimum= Default. No issued client certs are stored.
  - Names = Subject-Name and expiration of issued client certs are stored.
  - Complete = Complete information from issued client certs are stored.

Different platforms and devices require SSL certificates to be converted to different formats. For example, a Windows server exports and imports .pfx files while an Apache server uses individual PEM (.crt, .cer) files.

-

PKCS12 creates files with a .pfx extension. PEM format is the most common.

-

"URL" keyword allows you to have control over the location of where various items within the database will be stored (defaults to storing everything within NVRAM).

EKU Options

```
Router(cs-server)#eku ?
  client-auth         Client Auth
  code-signing        Code Signing
  email-protection    E-Mail Protection
  ipsec-end-system    IPSEC End System
  ipsec-tunnel        IPSEC Tunnel
  ipsec-user          IPSEC User
  ocsp-signing        OCSP Signing
  server-auth         Server Auth
  time-stamping       Time Stamping
```

- EKU = Extended Key Usage
  - Optional extension to an Identity Certificate
  - Clients must request EKU Options within their original CSR

Auto-Rollover

```
Router-1(cs-server)#auto-rollover ?
  <0-1825>  Overlap time between CA certificates during rollover, in days
  <cr>
```

- After the lifetime of the CA Server expires, the CA will no longer issue Clients Identity Certificates.
- Auto-Rollover allows CA Server to dynamically issue a new, self-signed Root Certificate as well as a new RSA Keypair.
- Clients should be configured to auto-enroll so that;
  - When the client's own identity cert is about to expire, it auto-enrolls for a new certificate.
  - When the client detects that the Root CA's cert is about to expire, it automatically re-enrolls for a new Identity Cert using the Root CA's new certificate for signature.

Auto-rollover is actually three values <days><hours><minutes>

Hash

```
Router-1(cs-server)#hash ?
  md5     use md5 hash algorithm
  sha1    use sha1 hash algorithm
  sha256  use sha256 hash algorithm
  sha384  use sha384 hash algorithm
  sha512  use sha512 hash algorithm
```

- This command controls which Hash algorithm is used by the IOS CA Server in order to hash the contents of its own, Self-Signed Root Certificate
  - Default is MD5 (considered weak)
  - Recommendation is to use at least SHA256

While the IOS CA Server will use MD5 by default to create its own, self-signed signature…it uses SHA1 with RSA Encryption by default when hashing-and-signing Client Identity Certificates. This conforms to the current standard.