



Certificate Authorities

Peter J. Welcher

Introduction

This article is pouring forth into my laptop computer following all the hurricane-induced rains the East Coast has experienced. My best wishes to those in Florida and elsewhere who sustained harm from the storms and wind. This article may be occasionally breezy but hopefully not windy.

The Cisco IOS code now contains a Certificate Authority server. There is a link to the configuration notes page [below](#), along with some other good links. I thought it might be interesting to take a closer look at how this works.

When I was travelling around earlier this year presenting on WLAN Security, I asked audiences about **Certificate Authorities (CA's)** and **Public Key Infrastructure (PKI)**. I found that almost nobody had them, or planned to implement them. Admittedly, the Microsoft administrators in their organization might have already implemented or be planning to implement CA / PKI and the network staff not known about it (or cared). So does that mean here's lack of knowledge about or fear of CA / PKI, or perhaps people aren't interested because they don't see potential value to them? After all, if you're reading this, you're probably a networking person, hence way too busy to inquire into things you don't need to deploy soon.

So it seems useful to talk about what a CA is, what PKI is, and why you and I might care. And that's the topic for this article.

What are CA's and PKI?

If you like reading standards, see ITU standard X.509 and the related RFC 2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. See also <http://www.ietf.org/rfc/rfc2459.txt>. There are a lot of RFC's and drafts listed on the IETF PKIX Working Group web page, at <http://www.ietf.org/html.charters/pkix-charter.html>.

Have you ever used a credit card to buy something on the Internet? Then you've probably used both a CA and PKI, without even knowing it. The PKI comes in because you really need somebody to confirm that the site you're on is the site you think it is. That is, say the site claims to be amazon.com. How do you know it really IS the one and only amazon.com? Otherwise you might be on a phishing site typing your card info in for the convenience of somebody who'd like nothing better than spending your hard-earned money. You trust the CA to verify that the site is who it claims to be. Ah, but how do you know to trust the CA? Well, your browser knows to trust it because it has a copy of its public key installed with the browser. How can you trust your browser? Well, you did get it from a reputable source, didn't you? This may seem a bit paranoid, but that's the first rule in thinking about security and trust.

There are links below for Thawte and Verisign, two of the commercial CA's on the Internet. Commercial or government sites pay for managed PKI services, basically a healthy fee for each computer that is authenticated. Thawte and Verisign vouch for the identity of computers within those organizations. The computers in question might just be e-commerce web servers, or it might be all users (or now, Cisco IP phones) within the organization.

Thawte -- Commercial CA	http://www.thawte.com/
Verisign -- Commercial CA	http://www.verisign.com/

Under the hood, here's what's going on. Each computer has to be issued a certificate, an electronic file digitally signed by the CA saying in effect "by the authority vested in me, you can believe this is John Doe's computer and its public key". The digital signature ties together identity and a public encryption key. This scheme also uses hash coding of the signature, and encryption, to detect alterations. The online CA can then be used to confirm that the digital signature is valid, that it did in fact issue the certificate. If somebody obtains the computer's certificate and puts it on another computer, then they might be able to have another computer pretend to be John Doe's computer. This is somewhat like having your driver's license or ID card copied.

So a Certificate Authority (CA) is just the entity that issues and vouches for digital certificates. Public Key Infrastructure (PKI) is the technology and processes that do the work. You create a public and a private key, and a certificate containing identity info and your public key. You submit it to the PKI. The PKI verifies your identity, stores the identity info and public key (think directory or phone book), digitally signs the certificate, and returns it to you. That CA digital signature confirms that your identity is associated with the public key in the certificate. The certificate gives you a way to digitally sign things by using your private key (proving they really come from you and not someone masquerading as you). The public and private keys also give you a convenient way to automatically and securely negotiate a 3DES or AES key over public networks. But let's not get seduced into the marvels of public/private key crypto right now -- see some of my other articles for more on that topic (IKE, IPSec).

You trust the CA's signature since you have the CA's public key from a trusted source, e.g. your browser vendor. Since you also trust the CA, you can then trust the certificates it has signed. Stolen or lost certificates are in essence handled somewhat like stolen credit cards, using a "certificate revocation list". There are lots of intricate protocol details lurking here, but let's not get sidetracked into them either. The Microsoft article referenced below provides a gentle introduction in a little more depth than this article can attempt.

How Secure Is the CA?

There are differing degrees for need and security when interacting with a CA. This is like someone who you trust in some matters but not others.

If you want to work in your lab, perhaps to gain familiarity with PKI, you might use the free [SimpleCA](#) on Windows or the free software at the "[Set up your own](#)" link below. Anybody else using certificates you issue is trusting you, your identity verification process, and the physical and network security controls you have instituted over your lab computer. The certificates probably cannot be verified online (unless you set up a server for that), so web or email users may have to trust your signing authority or your certificate (say, based on calling to check with you).

Set up your own Certification Authority using free software	http://slwww.epfl.ch/SIC/SL/CA/
SimpleCA (One of apparently two freeware packages with the same name)	http://users.skynet.be/ballet/joris/SimpleCA/

What this lacks is convenient large-scale administration. It's very manual. You have to create certificates. You have to distribute them. You have to confirm that you issued them. It's also not well enough integrated that non-technical people can really be expected to use it.

Microsoft's implementation has more automation, and appears adequate for most corporate purposes. It ties into Active Directory, which may well be the corporate repository of employee info. It can use IIS to digitally sign certificates, if you're willing to trust Microsoft user logins / web authentication as adequate verification of identity. Probably good enough for most corporate purposes, but how much money would you be willing to pay out based on such a certificate? Convenient, fairly secure. A company might use Microsoft CA to confirm identities within the company. Should outsiders have that need, then external access would be needed for checking certificates. The Microsoft CA would probably be in a server room. Some provision for redundancy and high availability might be made. That would make the CA fairly trustworthy, e.g. for internal email and encryption.

If you're allergic to Microsoft in some fashion, there is ongoing work in the open source and university communities to create open source, free tools for a PKI that scales to a good size. Google turned up the following (and more):

- 1 <http://ospkibook.sourceforge.net/docs/OSPki-2.4.7/OSPki-html/ospki-book.htm>
- 1 <http://www.mozilla.org/projects/security/pki/>

So there are alternatives for those who like getting their hands into the software. One can make a case that auditing the code improves ability to trust in the PKI. Web certificate signing based on web logins (student ID, password) is usually

considered good enough for universities, particularly due to the costs of doing identity verification more rigorously.

If you're thinking money and e-commerce, that's where Thawte and Verisign come in. They're well-known and fairly trusted. They claim to have put in a lot of due diligence securing their servers, people, processes, etc.

If you're talking banking and large amounts of money, then you're into another whole world, where the procedures used to set up a CA in a vault, etc., matter greatly and have to be properly managed, with proper records. One key element is making sure no one was alone in position to potentially compromise the security of the CA server. You may well issue certificates offline (by courier?) in a situation like this. You want very good verification of identity, so you don't wire-transfer millions to an offshore bank account. The U.S. Federal Reserve checks banks that set up CA's, and doing it right costs at least \$1-2 million. The consultancy Betrusted (www.betrusted.com) started with that as one special area of expertise. They were spun off from PricewaterhouseCoopers and bought by One Equity Partners, private equity arm of Bank One.

What's the point here? The PKI is only as good as the security. If somebody can tap into the CA server, it's like somebody stealing a list of people, credit cards, and expiration dates. They can then pretend to be you (or worse, amazon.com, or yet worse, some bank). So tight physical and logical access controls, audit trails, etc. relate to how secure the server is. When you get to the banking level, you need to think about process, because perhaps the person installing the OS on the server left a back door, so they could get in from outside 5 or 10 years later.

Some Terminology

Technically, a certificate is a digitally signed statement binding together identity information and public key, signed by the issuer, the Certificate Authority (CA).

Enrollment is the process whereby a CA verifies identity, stores certificate information, and signs the certificate.

PKCS-10 is a standard for the certificate request message.

X.509 is the most prevalent standard for certificates. PGP and GPG can also issue certificates, which may not meet the X.509 standard.

Certificate hierarchy is a chain of CA's from root to issuing CA. For instance, your corporate CA might have certificate from Verisign. In effect, Verisign vouches that ca.corp.com (or whatever) is Corp Co's corporate certificate authority, and then that CA may vouch for Corp Co's employees, internal web server, etc. This is generally simpler than having to decide individually if you trust ca.a.com and ca.b.com, etc. The top CA in this hierarchy is of course the root CA. Generally, having only a few root CA's makes life simpler.

Why Do PKI in a Router?

From the Cisco New Feature notes:

The Cisco IOS Certificate Server feature embeds a simple certificate server, with limited certification authority (CA) functionality, into the Cisco IOS software. Thus, the following benefits are provided to the user:

1. *Easier public key infrastructure (PKI) deployment by defining default behavior. The user interface is simpler because default behavior are predefined. That is, you can leverage the scaling advantages of PKI without all of the certificate extensions that a CA provides, thereby allowing you to easily enable a basic PKI-secured network.*
1. *Direct integration with Cisco IOS software.*

Reading a little into this, more and more Cisco devices need to authenticate themselves to each other, or send encrypted traffic. Shared keys and CiscoSecure ACS can be used for this (e.g. as with WDS for WLAN and WLSE) but it starts getting painful as it scales up. It could also be more secure. For devices such as the Cisco IP phones, the scaling could become an issue. So a clean scalable solution is needed.

What I think I see is that dependency on external PKI is a barrier to using certificates in a couple of ways. If your organization still hasn't moved to Active Directory, well, upgrades to future Microsoft OS's may start getting painful -- but that's an issue for the server folks. But it sure means you can't use certificates yet. The second side to this is something I see in Network Management in large shops. The network staff find that getting DNS and other changes done quickly and correctly by the systems administrators is hard. So they much prefer to have their own network device DNS server. Since

DNS is so important lately, I personally like redundant DNS and DHCP servers. But the reality is that local admin control is sometimes needed, since otherwise changes you needed yesterday just don't happen.

So a simple PKI in the Cisco IOS gives control of certificates for networking device use back to network staff, should that be deemed appropriate or necessary. If you have Active Directory and want to use it, or some other CA, fine. But if you want something to facilitate quick PKI deployment for network device security, this is a nice alternative. And since all the program code is from Cisco, one can hope it'll be solidly interoperable!

The Cisco CA code uses SCEP, Simple Certificate Enrollment Protocol, over HTTP. This does require enabling the Cisco IOS web server. It provides a uniform automated way for other devices to enroll with the CA. You can always disable the Cisco IOS web server after a large-scale rollout, and use manual certificate enrollment, if you wish to tighten security.

There are two particular concerns when setting up a CA. The identity / public key bindings take effort to create. Deploying PKI to user PC's is well-known to be labor intense. So having a solid database to store the info in is important. Securely and reliably backing up the data is also important. Both are mildly challenging for a router, since it has no hard drive to store such information on. We'll see how Cisco resolved this in the next article.

Summary

Chapters of the following two books may be useful if you want to read more about public key cryptography, PKI, CA's, hashing schemes, etc. Caution: Stallings (as usual) gets rather technical, but the book is a fine reference.

Merike Kaeo, <i>Designing Network Security</i>	http://www.amazon.com/exec/obidos/tg/detail/-/1587051176/
William Stallings, <i>Cryptography and Network Security</i>	http://www.amazon.com/exec/obidos/tg/detail/-/0130914290/

The following links may be useful or interesting:

IOS New Feature Notes: Cisco IOS Certificate Server	http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_4/gt_ioscs.htm
IOS Configuration Guide: Configuring Certification Authority Interoperability	http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fipsenc/scfinter.htm
Verisign White Paper: Managed Public-Key Infrastructure -- Securing Your Business Applications	http://www.verisign.com/static/005303.pdf
Microsoft Windows 2000 Public Key Infrastructure	http://www.microsoft.com/technet/prodtechnol/windows2000serv/evaluate/featfunc/2000pk.mspx
Microsoft PKI Seminar (voice + rather vanilla PPT bullet slides)	http://www.microsoft.com/seminar/shared/asp/view.asp?url=/Seminar/en/20020531tnt1-48/manifest.xml
Certificate Authorities: How Valuable Are They?	http://www.networkcomputing.com/806/806f1.html
Nifty Network Computing magazine "example of certificate application" diagram	http://img.cmpnet.com/nc/806/graphics/certificate.pdf

I do hope this article was interesting and useful. Ideas, comments, questions are always welcome. Send to the email address below.

As noted above, I hope to show lab work with IOS-based Certificate Authority in the next article. Stay tuned!

Dr. Peter J. Welcher (CCIE #1773, CCSI #94014, CCIP) is a Senior Consultant with Chesapeake NetCraftsmen.

NetCraftsmen is a high-end consulting firm and Cisco Premier Partner dedicated to quality consulting and knowledge transfer. NetCraftsmen has nine CCIE's, with expertise including large network high-availability routing/switching and design, VoIP, QoS, MPLS, IPSec VPN, wireless LAN and bridging, network management, security, IP multicast, and other areas. See <http://www.netcraftsmen.net> for more information about NetCraftsmen. Pete's links start at <http://www.netcraftsmen.net/welcher> . New articles will be posted under the Articles link. Questions, suggestions for articles, etc. can be sent to [pjw <at> netcraftsmen <dot> net](mailto:pjw@netcraftsmen.net) (formatted this way to fool email harvesting software).

10/5/2004, updated 12/17/2004.
Copyright (C) 2004 Peter J. Welcher