;login:
THE MAGAZINE OF USENIX & SAGE
December 2001 • Volume 26 • Number 8

inside:

SECURITY

Certification Revocation

By Paco Hope

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

certificate revocation

Why You Should Do It and Why You Don't

by Paco Hope

Paco Hope has a M.CS. from the University of Virginia where he worked as the head system administrator in the Department of Computer Science. Hope joined Tovaris, Inc. in 2000 and is the director of product development.

paco@tovaris.com

Many companies implement X.509-compliant public key cryptography systems to identify parties in a variety of secure or authenticated communications. Public key certificates are used to secure email, grant access to the corporate intranet, verify secure Web sites, and perform a variety of other authentication and encryption duties. When public key certificates are presented, many integrity checks are performed on them to ensure their validity. Yet it is possible for all these checks to succeed even when the certificate is actually unfit for use.

Despite the fact that a certificate's signature is correct and the certificate has not yet expired, it may have been revoked. Certificates may be revoked for a variety of reasons and X.509 provides mechanisms for revoking certificates and learning about their revocation. Both Microsoft and Sun Microsystems have had to revoke certificates in recent years due to security breaches. Those revocations are supposed to provide information that prevents the end user from trusting the bad certificates. In practice, however, revocation mechanisms are rarely or poorly implemented. This article will introduce the concept of certificate revocation: what it means, how it happens, and how it fits into an overall public key infrastructure. We will explore why an organization wants to implement revocation, what solutions are out there, and some of the limitations to current practices.

Revocation Explained

X.509 certificates consist of various pieces of identifying information such as a name, email address, and serial number. They also contain the cryptographic key material used in the mathematics of cryptography. Their purpose typically is to bind a particular real-world entity (a person, machine, or company) to these mathematical bits for the purposes of some kind of secure transaction.

Early on in a secure transaction, an entity presents their certificate. To verify the authenticity of the certificate the other party performs several mathematical consistency checks on it. These involve checking the expiration date, verifying the digital signature that was applied by the issuing CA, and semantically interpreting various bits that indicate approved uses of the certificate. Those consistency checks, however, are not the whole picture. Certificates can pass all those checks and still be inappropriate to use. That's where the process of revocation fits in.

Revocation in X.509 infrastructures is solely the purview of the Certificate Authority (CA). Unlike PGP, where the end user usually revokes her own certificate, in X.509 only the CA who issued (signed) a certificate can revoke it. Revocation does not alter the public key certificate. After all, copies of the certificate might be stored in a browser cache, an email program's address book, or in various other convenient places. Instead, the fact that a certificate has been revoked is published in some other way. Applications which wish to check revocation status must first obtain the certificate, and then separately attempt to determine if the certificate has been revoked.

The first, and most well-known mechanism for publishing revocation is the Certificate Revocation List (CRL).[1] In their simplest form they are lists issued by CAs containing

the serial numbers of all certificates that the CA has revoked. A flexible and real-time protocol has also been defined and is offered by some major vendors. Other revocation methods, several of them patented, have also been defined and made available commercially. After first understanding the simplest CRL, we can explain the variations on that theme and explore other novel approaches.

Certificate Revocation Lists

Every certificate is issued by a CA, and that CA assigns it a unique serial number which is encoded in the certificate. The issuing CA and the certificate's serial number are paired to form an identifier for the certificate.

A Certificate Revocation List, then, is very much what its name implies. It is issued by a specific CA and lists the serial numbers of all certificates that CA has ever revoked. For each certificate it also lists the date the revocation occurred, and optionally a reason why the certificate was revoked. The entire list is signed by the CA's private key and presumably published widely. CRLs also include a publication date and a "next publication" date to indicate when the current CRL was issued and when the next CRL should be expected. By consulting the appropriate CRL, an X.509-compliant application can definitively determine whether or not a given certificate has been revoked.

In their original conception, CRLs would be published regularly by the CA, and made available through some public interface like FTP or HTTP. Finding CRLs this way would have to be supported by every X.509-compliant application – either directly or via the underlying operating system – from email software to Web browsers to VPN access software. PKI software, such as that which provides CA services, must also regularly update the CRLs in order to keep them current. Each application would initiate a connection each time it considers a certificate for which it has no verification information. Since CRLs regularly expire and are regularly updated, the application or operating system would have to monitor its revocation information and, if it became stale, refresh it.

The load imposed by HTTP or FTP connections from millions of desktop systems is one major disadvantage to this approach. If, for instance, secure email became a standard practice, and every recipient of every message were verified for revocation information, the burden of these connections would be a significant problem. Since CA certificates can live for very long times (some have expiration dates 20 years from now), CRLs can conceivably grow very long. Thus it could be especially inefficient to download and refresh a list of all revoked serial numbers, when the existence or nonexistence of a single number is the only information needed.

Improvements to CRLs

Several incremental improvements to the CRL process have been proposed and published; some have become commercial products. These improvements tend to focus on making CRLs easier to find, smaller to download, and/or easier to manage.

Entrust created and patented CRL Distribution Points (CDPs). A CDP is a reference (typically in the form of a URL) indicating the location of the CRL. This reference is included in the content of the certificate itself. Inclusion in the certificate makes it immediately available to any application which needs to find the relevant CRL. Entrust's Web site says they offer a royalty-free license to use CDP technology. Despite that, CDPs are rarely found in public key certificates. Even licensees like Microsoft and Verisign rarely – if ever – include CDPs in the certificates they issue to end users.

Delta CRLs are intended to make smaller, supplementary CRLs available more frequently, while making bigger complete CRLs available less often. In such a scenario the CA issues a base CRL at relatively long intervals, such as monthly or weekly. The CA additionally issues incremental CRLs that are supplements to the base CRLs. Those supplements (called "deltas") are issued on relatively short intervals, like daily or hourly. To definitively determine revocation, an application must have a sufficiently recent base CRL and the corresponding most recent delta CRLs. Delta CRLs offer modest improvements in performance, given their smaller downloads. The burden of many TCP connections to a central distribution location and the burden of making all applications or operating systems maintain CRL information remain substantially the same.

Still another variation allows partitioning of CRLs by indicating which CRLs contain which serial numbers. It allows the CA to manage the size of CRLs and allows clients to download smaller pieces of the overall CRL information.

Complete Departures from CRLs

A few other approaches to disseminating revocation information have been proposed. Certificate Revocation Trees (CRTs) form the basis of a Valicert product. They consist of complex binary hash trees computed by the CA. One primary advantage is that the revocation or lack of revocation can be represented by data substantially smaller than an entire CRL. Fractions of the overall revocation information can be sent to verify whether or not a certificate has been revoked. Like CDPs, CRTs are patented so there is only one product on the market that makes use of them, and it is not your Web browser or email software.

The IETF has standardized a real-time verification protocol for getting up-to-the-minute revocation information. The Online Certificate Status Protocol (OCSP) is defined in RFC 2560.[2] An OCSP "responder" is a system which listens on a network for revocation queries. Querying systems send a query identifying a certificate. That query may need to be digitally signed before the OCSP responder will honor it. The OCSP responder determines the status of the certificate in question and replies with that status, or indicates that the status is unknown. The OCSP responder must sign all responses using a special key and certificate issued by the CA.

The primary disadvantage to OCSP is the cryptographic demands. Every response must be signed, and the signatures on requests might have to be verified as well. This disadvantage can be overcome by sufficient hardware and network engineering. Like CRLs, however, if the certificate itself does not indicate the relevant OCSP responder, an application has no means of determining that an OCSP responder exists. OCSP responders have no way to "refer" a query, either. If a responder does not know the answer to a query, it has no means of indicating another server which might have the desired answer. Thus an application could issue queries to many different OCSP responders, but ultimately receive no valid revocation information.

OCSP is gaining momentum among large commercial PKI vendors as a more efficient method of determining revocation status. Widely available applications (Web browser, email software, operating systems) are slowly incorporating support for it. There are close to a dozen major vendors offering OCSP-based products, such as Digital Signature Trust Company, CertCo, and RSA Security. Making productive use of an OCSP responder, however, requires consistent and functional support in each and every X.509 application. Uniform and reliable OCSP performance will take time to mature.

Why You Care

Digital certificates and public key cryptography capabilities are rapidly appearing in all sorts of network-enabled applications. Every major email software program has some support for digital certificates. Most vendors of online chat and instant messaging services are adding cryptography to make secure chat possible. Internet e-commerce relies substantially on the integrity of certain CA certificates. Many VPN and firewall access systems utilize X.509 public key certificates in one way or another. What happens if a certificate is compromised?

We need only look to January 2001 to see a textbook example of why revocation is important.[3] Verisign erroneously issued two code-signing certificates to an unknown person who somehow persuaded them that he represented Microsoft. In a routine audit of January's activities, Microsoft noticed the erroneous issuance and the certificates were immediately revoked. This event was very well covered in security newsletters and publications. The fact is that someone somewhere has a certificate officially issued by Verisign that states unequivocally that it belongs to "Microsoft Corporation." Yet, the actual truth is that it neither belongs to nor represents Microsoft in any way. If you have not applied Microsoft's patch (which installs a partial CRL that covers the erroneous certificates), your Web browser or email software might automatically execute malicious code, because it has a seemingly legitimate, Verisign-issued certificate. The expiration date on those erroneous certificates is January 31, 2002. The malicious use of these certificates poses a threat long after that date.

Why You Don't Check Revocation

The simple answer is: you can't. CRLs are extremely difficult – if not actually impossible – to find. Verisign and Thawte publish their CRLs reasonably well. In this author's experience only Verisign's CRLs are so readily available that they are usable by applications that have only rudimentary X.509 support. There are many other CAs that I trust, however, and few if any publish CRLs that can be found by end users.

My Netscape browser (venerable at version 4.76) includes about 60 trusted root CA certificates. There is absolutely no association between any of them and their CRLs. The Web site I visit today may have a certificate that was originally issued and subsequently revoked by the "TC TrustCenter Class 2 CA." I will never know. Netscape (or Internet Explorer, or any number of other applications who might trust this CA) has no way of divining the location of the CRL for this CA. If the certificate I receive from the Web site happens to have a CRL Distribution Point in it, and the CDP points to a valid CRL, then I stand a chance. Otherwise, there is no way to know. If the certificate authorities will not publish their revocation information regularly and obviously, the end user has no hope of using it.

The usability of CRLs is so bad that email lists are needed to supplement them. When Sun Microsystems had to revoke two certificates in October 2000, they used email security bulletins to spread the word.[4] There was no CRL to rely upon. A telling detail is that the advisories identified the two revoked certificates, but they do not identify any mechanism of checking a CRL or any other regularly updated revocation information. Worse yet, Sun's security bulletin recommends deleting the certificate from browsers that have mistakenly trusted it. If the browser encounters the certificate again, it will prompt the user. The user will see "Sun Microsystems" and a valid signature, and will probably choose to trust the certificate. Deletion does not prevent the

The usability of CRLs is so bad that email lists are needed to supplement them.

REFERENCES

1. R. Housley, W. Ford, W. Polk, and D. Solo, RFC 2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999. Status: PROPOSED STANDARD.

2. M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, RFC 2560, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP," June 1999.

3. Kathleen Murphy, "Verisign Gets Duped in Security Attack on Microsoft," *Internet World*, March 2001, *http://www.internetworld.com/news/archive/03262001b.html*.

4. Sun Microsystems, Security Bulletin 00198, October 2000, *http://sunsolve.Sun.COM/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/198*.

browser from trusting the certificate. Only the presence of revocation information truly represents a barrier to trust.

Can you enable your existing enterprise applications or your operating system to make use of OCSP? The answer is probably yes. If you invest in a commercial OCSP offering of some kind, or if Microsoft or Apple or Sun integrate OCSP into their operating system you might get real-time revocation information. As it is, there are many concerns about OCSP's ability to scale, given its high demands for cryptography on every query and response. Various proposals are circulating to allow caching of answers, pre-signing of answers, and other mitigating techniques. Few, if any, of those improvements are readily available in both server and client implementations.

Conclusion

Revocation is a dirty little secret in the PKI world. Those who understand its role in secure transactions realize that current technologies simply fail to offer realistic solutions for the teeming millions. Some system administrators, CTOs, and decision makers are just getting their feet wet in PKI technology. They are probably already struggling to grasp the essentials — not to mention such subtle issues as how a certificate that appears valid is, in fact, invalid.

If you are considering investing in X.509 PKI technology, or if you have already invested the equivalent of a developing nation's economy in your PKI technology, ask some hard questions about revocation. Are your end users using some kind of technology (CDPs, OCSP, CRLs) to learn about revocation of your own certificates? Are your collaborators able to get up-to-date revocation information about your certificates? Are you able to get up-to-date revocation information about your collaborators' certificates? What if your collaborators bought from a different PKI vendor?

Consider all the different applications where you might be applying PKI technology. There are already many vendors hawking PKI-enabled email, Web browsing, B2B e-commerce, instant messaging, and VPN technology. How can revocation information be made available to each and every one of those applications?

Ultimately, security boils down to risk assessment and risk mitigation. Your organization will have to assess the risk of trusting a revoked certificate, and decide what amount of mitigation is required. The cost of that mitigation follows naturally from that analysis. Expect to make compromises. The ideal is to maintain and manage revocation information in all the places you use public key certificates. That ideal will probably not be realistic or affordable without some fundamental change in the way PKI companies and technologies operate.