The Key to Smarter Enterprise Security

Making the case for cloud-based code signing in Enterprise

The challenge of effective key management

Code signing is at a crossroads. In many enterprises, the traditional approach remains standard procedure: you purchase a code signing certificate, download it, and deploy it locally for all your code-signing needs. When carefully managed, this approach can still be very effective in guarding against malware. But recent events have shown that many companies are finding it a challenge to manage their code signing certificate deployments—and failures in certificate management can have serious consequences.

In perhaps the highest-profile incident of the last few years, the Stuxnet worm relied on stolen certificates to spy on—and subvert—industrial systems around the globe. New research by Symantec suggests that the worm, first identified in June 2010, may have been active as early as 2005[1]. The threat was severe enough that in March 2013, a team of NATO researchers took the extraordinary step of declaring the use of Stuxnet an "act of force" against Iran[2]. This highly sophisticated malware specimen continues to find high-level global targets, largely in the Middle East[3].

Although Stuxnet was powerful enough to gain a great deal of attention on the international diplomatic stage, other less dramatic attacks continue to impact the enterprise on a day-to-day basis. In December 2012, Mozilla, Microsoft, and Google all updated their browser blacklists to include a list of fraudulent SSL certificates issued for common URLs such as mail.google.com, login.live.com, and login. yahoo.com5[4]. Sometimes these issues stem from certificate authorities (CAs) with insufficiently strong authentication and security practices. But far more frequently, malware spreads because once-legitimate code-signing keys have placed their signatures on malicious code.

The story is often the same: One person in an organization buys a certificate and distributes the keys to a few colleagues. Those colleagues distribute it further. Maybe someone puts it on a server or recklessly emails it, perhaps even transmitting it beyond the firewall. Before long, the keys are scattered haphazardly throughout the organization. Under those circumstances, it's common for a certificate to be used fraudulently for months before anyone knows about it—at least until the issuing CA discovers the fraud, issues a widescale revocation, and notifies the enterprise that there's a problem. In many cases, the revocation will wipe out a great deal of legacy code, leaving IT organizations unaware of what's protected and what's not.

As an example, in late 2015 Symantec identified suspicious activity involving malware signed with a valid code-signing certificate. The investigation led to the discovery of Suckfly, an advanced threat group conducting targeted attacks using multiple stolen certificates from South Korean companies. The most likely scenario was that the certificates had been stolen with the help of malware that has the ability to search for and extract certificates from within the organization[5].

Some might call this one of the risks of operating in a modern-day enterprise, but it's not an inevitable risk. In this paper, we discuss several alternative approaches to code signing in the enterprise—alternatives that, in one way or another, minimize the local mismanagement of keys. After a careful consideration of the choices currently on the market, we argue that cloudbased code signing may be the most costeffective, secure approach for protecting your SSL certificates without overburdening your information technology (IT) department.

Mobile code-signing model

Cloud-based code signing is not a new technology. In fact, it's been available to mobile developers since 2003 as part of a safe and effective process for secure key management. Developers simply upload their content to the cloud for the code-signing provider to sign. After that, the provider generates and maintains the keys, enabling developers to spend more time creating apps and less time administering certificates.

The value of this approach is clear. Instead of issuing one certificate and using it over and over—thus exposing developers to wide-scale fraud—cloud-based code signing relies on short-lived certificates that are generated, signed, and immediately destroyed. A sophisticated auditing and time-stamping system helps developers keep track of when the code was signed and by whom.

It should come as little surprise that cloud-based code signing is now standard procedure in many parts of the mobile world. Developers who write apps for Apple® iPhone®, Symbian, or Microsoft mobile devices can rely on a cloudbased system for the secure signing of app code, whether the infrastructure is

Cloud-based code signing is the backbone of many new mobile devices

In some of the newest and most innovative mobile devices, a cloud-based code-signing apparatus forms a critical part of the development process on the back end.

Developers use a service that looks like the mobile device manufacturer's user interface (UI) built on DigiCert APIs. When they sign in with the mobile device manufacturer, the manufacturer pushes the enrollment process to DigiCert. All submitted apps are then signed by DigiCert in a process that is, from the developer's point of view, entirely seamless.

It's important to note that in addition to signing apps, the DigiCert API also vets the developers themselves. This vetting process is backed by a particularly sophisticated validation apparatus, informed by DigiCert's deep experience as the world's largest CA, and built on a worldwide operations network with multilanguage support.

What about regulatory requirements for financial services companies?

The financial services industry has always been subject to stricter regulations than almost anyone else, and code signing is no different. In fact, under new federal regulations, many financial institutions are required to know exactly what code they signed and who has access to the keys.

Fortunately, compliance in this regard is fairly easy to achieve. DigiCert reporting capabilities can produce all of that information on demand, making it easier for financial services companies to meet the latest regulatory requirements.

selfmaintained (as in the case of Apple) or hosted by a third party (as in the case of Microsoft devices).

This technology doesn't need to remain limited to mobile devices. In fact, it's long past the time that such a service should be made more widely available to developers in the enterprise, allowing them to upload files and apps to the cloud, then entrusting key maintenance—including detailed auditing and reporting—to the code-signing vendor.

Creating an in-house key management infrastructure: pros and cons

Some large enterprises prefer to follow the Apple model, building and managing their own infrastructures for secure key management. But this approach raises several concerns. First, there are the common security concerns that arise from transmitting sensitive data to another entity in the cloud. Additionally, some organizations worry that they generate so much content, and store that content in such large files, that no cloud-based entity would be large enough and scalable enough to support their high level of production.

However, these concerns can be countered with relative ease. Anxieties around secure transmission are moot: a cloud-based code-signing vendor wouldn't be receiving code but simply storing compiled files—that is, files that would be distributed over the Web under any circumstances. Besides, managing your own infrastructure hardly guarantees security. As many enterprises have discovered to their great dismay, a self-designed security infrastructure often introduces just as many—if not more—variables and vulnerabilities as an infrastructure designed and operated by another entity.

With regard to scalability, a cloud vendor is far better prepared to absorb unexpected increases in demand than a standalone enterprise infrastructure. That's because an enterprise-class cloud service is predicated around scalability and will offer, as part of its essential design, a capacity far larger than the anticipated demand from any single enterprise customer.

Other considerations make the development of an in-house infrastructure downright impractical. For instance, most organizations would face prohibitive costs to build and maintain a state-of-the-art code-signing apparatus, keeping up with the latest standards from Microsoft, Java, Android, and other vendors, all while developing features that would secure and track every possible type of transaction. Meanwhile, effective key protection requires enterprises to purchase and operate a hardware security module—which is a significant capital investment.

All of these factors beg a simple question: with so many IT organizations challenged to make the most of increasingly limited resources, is it really practical to maintain a complex code-signing infrastructure while other more strategic tasks go under-addressed?

DigiCert Secure App Service: World's first cloud-based codesigning solution

DigiCert now offers a revolutionary approach to code signing for enterprises everywhere. The DigiCert Secure App Service solution gives corporations and individuals the ability to sign apps and files in the cloud, generate reports on all signing activity, and keep track of engineering output in an integrated Web-based portal. As an added benefit, this solution also includes DigiCert Extended Validation (EV) Code Signing in the cloud, freeing organizations from the burden of managing physical EV tokens. That means you can get all the benefits of EV Code Signing—working with Microsoft's latest and greatest, including the SmartScreen® filter—along with the broader benefits of cloud-based code signing, all in one secure solution.

Other features of the DigiCert Secure App Service solution include the following:

- **Role-based access.** DigiCert offers nested, granular controls so you can specify access and privileges for any individual in your organization. In the event of personnel changes, you can adjust access levels quickly and securely from a centralized control panel.

- **Broad range of signing types.** Sign all major types of code—including content for Microsoft, Java, and Android—with a single, comprehensive service.

- **Developer vetting services.** Validate the reputation of any developer, anywhere in the world, with a robust authentication system supported by a global operations network.

- **Full reporting and auditing.** In-depth reports and logs help you keep track of signed code and activity, with all reports easily exportable to Microsoft® Excel®, CSV file, or PDF.

- **Automated time stamping.** Upon signing your code, DigiCert provides an automatic time-stamping service using Microsoft Authenticode,® in keeping with standards specified in the RFC 3161 Public Key Infrastructure Time-Stamp Protocol (TSP).

- **Restricted authentication by IP address.** Specify an IP address range for authenticating access, helping ensure that all code-signing activity remains safely within your organization.

- **Administrative approval of signing requests.** If you want to intensify security, simply create an extra level of assurance by requiring administrative approval of all authentication requests.

- **Test integration.** Request testing and approval prior to signing, thus establishing another safeguard for maintaining the integrity of your code.

- **Access through API or Web portal.** Take advantage of a full set of APIs for integration into on-premise systems and workflows, or simply use our out-of-the-box Web portal for comprehensive key management.

- **Bulk upload capabilities.** Cut down administrative costs—and minimize human error— by uploading batch files of apps.

- **Private or public CAs.** Use private roots, or simply chain to a trusted root— whichever approach works best for your organization and your infrastructure.

In addition to these features and benefits, DigiCert offers yet another level of service: a threat intelligence group that constantly monitors and scans files used across the Internet. This group typically catches malware soon after launch. If malware is inadvertently signed by a customer, the group escalates the issue, and DigiCert works with the customer to discover what happened and revoke the certificate.

Few other CAs have access to that level of data. Fewer still monitor the emergence of malware so aggressively.

Conclusion

The traditional approach to code signing is under attack. When you deploy a certificate locally, you open your organization to the possibility of mismanaged keys—and the threat of malware spreading through once-legitimate code. Now there's an alternative. For years, mobile app developers have relied on cloud-based code signing to verify the authenticity of code through a centrally managed infrastructure. This approach—now available to the enterprise for the first time— is simply a safe, cost-effective option for protecting your code signing certificates without overburdening your IT organization.

To discover how the DigiCert Secure App Service solution can help you manage your codesigning efforts and provide enterprise-level support, all at a predictable cost, visit https://www.websecurity.symantec.com/code-signing/secure-app today.

[1] "Stuxnet 0.5: The Missing Link" Symantec (February 2013).
[2] "NATO Researchers Say Stuxnet Attack Was an 'Act of Force" Mashable (March 2013).
[3] "Six-year-old patched Stuxnet hole still the web's biggest killer" The Register, May 2016)
[4] "How to Avoid Fraudulent SSL" Symantec (December 2012).
[5] "Suckfly: Revealing the secret life of your code signing certificates" Symantec (March 2016).

For more information, contact an IoT expert
1.801.701.9695 or iot@digicert.com

Lehi

2801 North Thanksgiving Way Suite 500
Lehi, UT 84043
USA

Mountain View

487 E. Middlefield
Buildings K & J
Mountain View, CA 94043
USA

UK

88 Wood Street, Suite 1001 & 1002
London EC2V 7RS England

Switzerland

Balexert Tower, 18 Avenue Louis-Casai
Unites 01 and 30CH-1209
Geneva, Switzerland

Cape Town

Gateway Bldg. (3rd, 4th, & 5th floors)
Century Blvd & Century Way 1
Century City, Cape Town 7441
South Africa

Australia

437 St. Kilda Road
Level 3, Unit 4.01
Melbourne VIC 3004
Australia

China

23F/Taikang Financial Tower
38 East Third Ring Road
Chaoyang District, Beijing, 100026
China

Japan

Ginza 3-Chome
5F Okura Bekkan
3-4-1 Ginza Chuo-ku
Tokyo 104-0061
Japan

India

10th Floor-RMZ Eco World, Sarjapur,
Marathalli Outer Ring Road
Devarabeesanahalli Village
Bangalore, India 560103

digicert®
Symantec™ Website Security