Global Information Assurance Certification Paper

PKI Trust Models: Whom do you trust?

GIAC (GSNA) Gold Certification

Author: Blaine Hein, blaine.hein@telenet.be

Advisor: Stephen Northcutt

Accepted: July 28, 2013

Abstract

There has been a substantial amount of attention in the media recently regarding Public Key Infrastructures (PKI). Most often, secure web server exploits and signed malware have generated this attention and have led to the erosion of trust in PKI. Despite this negative media attention, there has been very little detailed discussion of the topic of PKI Trust proliferation and control. PKI is an integral part of our daily lives even though, for the most part, we never notice it. Europe is several years ahead of North America in the ubiquitous deployment of PKI to its citizens, but North America has begun to catch up. This paper covers four major areas including the definition of trust and trust models, implementation of trust, auditing of trust, and managing trust. The paper provides proof of concept tools to allow administrators to understand their current level of PKI trust and techniques manage trust.

1. Introduction to Public Key Infrastructure Trust Models

1.1. Definition

Trust is a concept that infants learn at a very early age in life, even before they can speak. It is a concept that is critical to wellbeing in many ways, but it is too broad to consider in a computing environment. When thinking about trust in CIS systems, a more concise definition is required. The Merriam-Webster definition of Trust is the "assured reliance on the character, ability, strength, or truth of someone or something." (Trust - Definition and more from the free Merriam-Webster dictionary, n.d.) In the context of this paper, trust is not a belief that one operating system is better or more secure than another operating system. This paper quantifies trust with auditable and verifiable concepts.

1.2. Scope

This paper requires an intermediate level of understanding of the concepts employed within public key cryptography and of the X.509 standards and IETF RFC standards. Recommended reading includes "Implementing Public Key Infrastructure (PKI) Using Microsoft Windows Server 2012 Certificate Services" (Naish, 2014) and "Digital Certificate Revocation" (Vandeven, 2014). These PKI concepts including Certificate revocation and asymmetric cryptography form the baseline level of knowledge required for this paper.

The tools and techniques presented in this paper are adaptable to any enterprise based operating system, browser or application currently on the market. The proof of concepts scripts included with this paper concentrate on the Microsoft Windows operating system version 7 or newer, and Server 2008 R2 and newer. Earlier versions of PowerShell may not support sufficient cryptographic methods required to run these scripts. Similar tools and command line options apply across multiple operating systems with minor adaptations required to address the differing cryptographic libraries.

Author Name, email@address

1.3. Why do we need trust?

This is not the question. As soon as you touched the power switch on your computer, you already trusted someone. The real question is "How do you know?" The tools and techniques presented in this paper will change a qualitative gut feeling of trust into a quantitative and auditable set of trust relationships.

At the most basic level, trust in a certificate does not equate to assurance that the certificate is good. Enumerating the assurance level of certificates within a system requires a different scope of activities.

The starting point to auditing PKI trust requires a knowledge of all loaded certificates, and their function within the system. To achieve a higher level of process maturity requires an ability to manage PKI trust within the system.

1.4. Trust anchors

Trust anchors are a core concept within public key infrastructures. They are certificates that we believe in without needing to find further evidence (Housley, Ashmore, & Wallace, 2010). To use a building analogy, they are the cornerstone. Every other stone comes after the corner stone. Compromising the integrity of the cornerstone can cause a building collapse.

1.5. Building paths to Trust Anchors

In some buildings, the cornerstone is out of sight in the depths of the basement. When looking at a stone on the third story, there is no direct information about the cornerstone. A search following the pillars and walls downward is required to find the cornerstone. Building paths to trust anchors involves the same techniques. The terms commonly used here are certificate chains and path validation.

1.5.1. Certificate Fields

There are several certificate fields and extensions involved with path validation. These extensions define the identity of the certificate and create linkages between certificates. Table 1 provides the definitions of the specific certificate fields required for

path validation (Cooper, et al., 2008). Certificate stores include other data fields associated with each certificate. The Certificate "Alias" or "Friendly Name" provides a short human readable tag for each certificate. The user or application that installs the certificate chooses the value for this field. The certificate thumbprint is calculated by the certificate store as the certificate is installed, based on the SHA-a hash of the entire certificate.

Field / Extension Name	Purpose
Issuer Distinguished Name (DN)[1]	The issuer field contains the identity of the Authority that signed the certificate.
Subject DN	The Subject field contains the identity of the certificate holder. In the case of self-signed certificates, it is the same as the issuer.
Public Key	The public key is the part of the asymmetric key pair shared with the entire community or world.
Authority Key Identifier (AKI)	The AKI is the SHA-1 hash of the public key held by the signer of the certificate.
Subject Key Identifier (SKI)	The SKI is the SHA-1 hash of the public key included within this certificate. In the case of self-signed certificates, it is the same as the AKI

Table 1 Description of certificate field and extension purposes

1.5.2. Unique References to Certificates

Understanding certificate chains requires an understanding of the information that is unique and unambiguous within a certificate. Table 2 explains the scope of the

[1] Distinguished Names, (DN) defined in the ITU-T X.501 standard denote a structured naming convention used within X.509 certificates.

certificate information and extensions utilized within this paper. The certificate thumbprint is the only unique information. However, the thumbprint does not assist with path validation.

Name	Uniqueness
Friendly Name / Alias	Uniquely defines certificate, only within certificate store. Arbitrary descriptor
Thumbprint	Uniquely defines certificate. Any change to certificate changes the thumbprint
Subject DN	Not unique: DN may be common to several certificates.
Issuing DN	Not unique: DN may be common to several CA certificates.
Public Key	Should be unique, but uniqueness is not easily enforced.
Authority Key Identifier (AKI)	Uniquely defines the public key that signed the certificate. (See Public Key)
Subject Key Identifier (SKI)	Uniquely defines the public key of the certificate. (See Public Key)

Table 2 Uniqueness of Certificate Information

1.5.3. Certificate Chains

Unlike buildings, which start from the bottom up, building PKI certificates is a top down process. First, the self-signed Root CA certificate is established. Next, the Root CA signs a subordinate CA certificate. This subordinate CA may in turn create an additional subordinate CA. The lowest layers of subordinate CAs issue certificates to people, applications, or devices. The minimum number of Certificate Authorities to establish a chain is one. While there is no theoretical maximum, the average certificate chains have between two and three CAs in the hierarchy.

Validating certificate chains starts from the bottom upwards. The process relies on the ability to search for the next certificate in the chain based on information stored in the current certificate. Several protocols including Transport Layer Security (TLS) enhance the ability to build certificate chains by including all of the certificates as part of the protocol exchange.

To overcome the lack of unique information described above, path validation uses the Issuing DN and AKI certificate extensions. The Issuing DN provides an efficient

search mechanism to find the potential parent certificate. From the collection of parent certificates returned by the search, the AKI narrows the results to the certificate with the correct asymmetric key pair used to sign the certificate.

1.5.4. Path Validation

Path validation is the process of verifying the integrity of the certificate chain up to a trusted root CA (Cooper, et al., 2008). Revocation status, which is out of scope of this paper, is a critical component of the integrity verification.

Figure 1 depicts the linkages between certificates in a chain. The subject of the higher-level certificate becomes the issuer of the next certificate downward in the chain. Likewise, the SKI becomes the AKI for the next certificate. The client searches in various locations depending on the Operating system to find a certificate that matches the issuer DN in its own certificate. While the DN facilitates the discovery of certificates, the AKI and SKI values determine whether the certificate is the correct one. If a Certificate Authority has generated a new asymmetric key pair, the SKI values within that certificate should change. During a rekey, there is no requirement for the CA to change its DN. The SKI and AKI values ensure the selection of the correct certificate from the CA to build the chain.

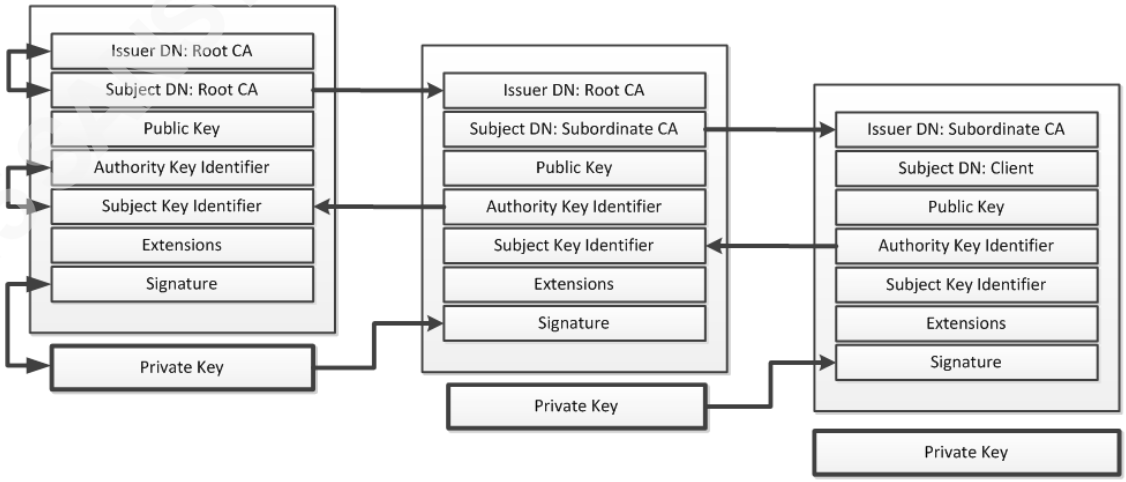

Figure 1 Certificate chain and linkages between certificates

A second special case for path validation exists with Cross Certification. In this instance, there are two certificates with the correct subject DN and AKI values included. The client must select the one that completes the path. The next paragraph provides more details on cross certification.

There are many more fields within certificates used to validate certificate chains. CRLs, OCSP, and blacklisting are revocation tools already in place within browsers and operating system.

1.5.5. Cross Certification

Cross certification is a method of inter-connecting two Public key infrastructures to allow for building certificate chains across a larger community (Cooper, et al., 2008). The two CAs implementing cross certification sign each other's CA certificate. With the addition of the two new certificates, client certificates perceive the cross-certified infrastructure as subordinate to their own Root CA. Figure 2 visualizes this scenario. The red client at the bottom right builds a certificate chain from its own certificate up to its trusted root certificate. To validate the blue client, the process starts in the same manner. When the client reaches the top of the chain, it must choose between two certificates that both appear to have issued the subordinate CA certificate. The presence of both a blue link and a red link in the diagram indicates this choice. The blue link points to a certificate that the red client categorizes as an unknown root CA. The red link points to the red client's own trusted root CA. Selection of the red link completes the path validation between the two clients and confirms the trust relationship between the two.

Author Name, email@address

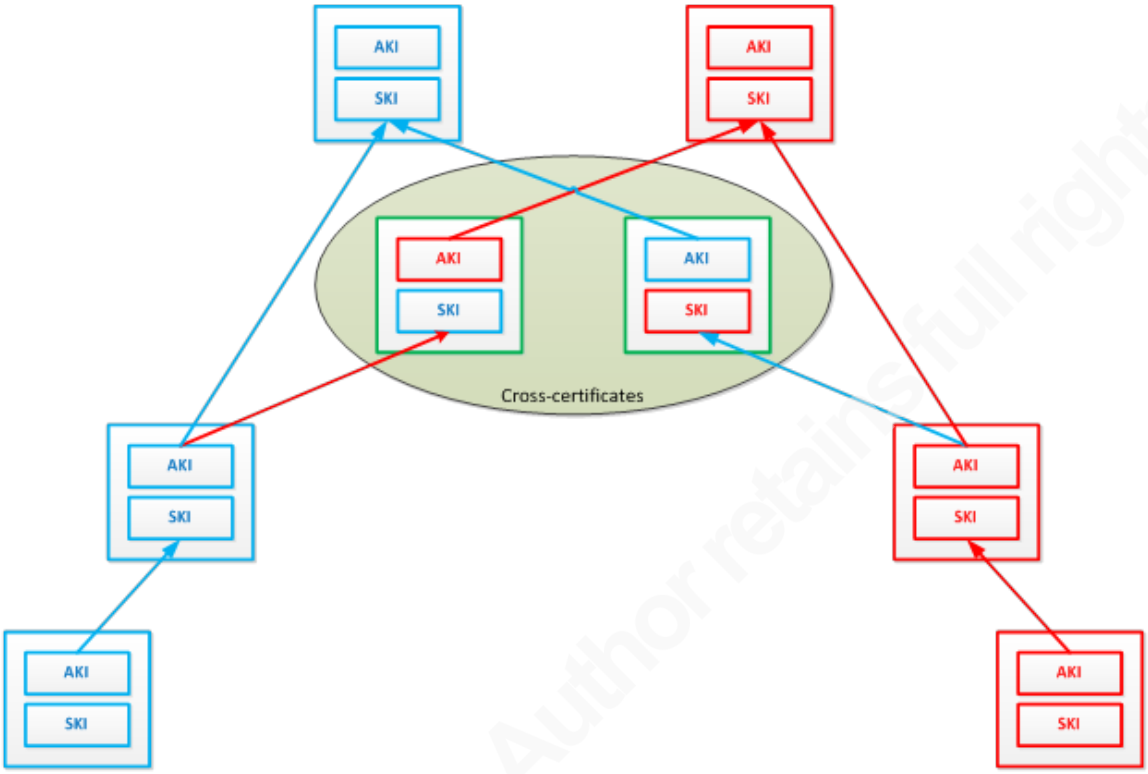

Figure 2 Cross-certification

While the generation of the cross-certificates is quite easily accomplished, the processes and procedures required to establish trust between two organizations is much more complex.

1.5.6. Web Browser trust model.

Another method of building a larger community of trusted certificates involves simply declaring trust in a group of top level (Root) Certificate authorities. Web browsers and the Microsoft family of operating systems are the most common implementers of this trust model.

1.5.7. Web of trust

The web of trust model is an alternative PKI trust architecture from the hierarchical PKI models discussed so far. Instead of a pyramid-based trust inheritance from a root CA in the enterprise, the web of trust model implements all CAs as root CAs.

Author Name, email@address

Programs such as Pretty Good Privacy (PGP) are the utmost implementation of Web of trust. Decentralization extends to the end users without the requirement for a CA. Users manage trust at the level of individual keys. Decentralized control of each key pair is the primary difference from the hierarchical model. This last model is not relevant to a managed PKI deployment within an enterprise as there are no Certificate Authorities to install into a trusted root store.

2. Certificate Stores

2.1. Microsoft Certificate Store

The Microsoft family of operating systems includes built in certificate stores to hold trust anchors. Microsoft uses the Windows Update service to publish selected root certificates into the Trusted Root CA Store based on requests from the operators of these root CAs. The Microsoft Root CA Program governs eligibility for direct publication of root certificates. (Microsoft, 2015).

Internet Explorer and the Windows compatible versions of Chrome and Safari rely upon the Microsoft certificate stores for identifying the trusted root certificates. Figure 3 shows a subset of the available certificate stores via a snap-in for the Microsoft Management Console.

Both GUI and command line tools support Installation and management of enterprise root CA certificates. For example, the following commands export the thumbprint and subject of all certificates from the Root CA certificate store belonging to LocalMachine.

```
set-location -path cert:\LocalMachine\Root

get-childitem >rootcerts.txt
```

2.2. Firefox and other Mozilla based browsers

Mozilla includes a PKCS#11 loadable module as part of the core to Firefox and other Mozilla based browsers. This module contains the certificates registered in the Mozilla trusted root certificate program (Included CAs, 2015). A user database also exists, that holds additional trusted root CA Certificates loaded by the user. Users can also modify the permissions of the preloaded root CA certificates, by adding entries into the user database.

The pre-loaded CA certificates are included in the following files:

- **Windows**: libnssckbi.dll
- **Unix, Linux, and other *nix variants**: libnssckbi.so
- **Mac OS X**: libnssckbi.dynlib

To ensure that the audit achieves accurate results, it is important to correlate the contents of the user certificate database with the contents of the preloaded PKCS#11 module as the user database takes precedence (CA:FAQ, 2015).

It is not possible to remove certificates from the loadable module. To manage the root certificates, the NSS Certutil.exe application is required. Although this application shares the same name with the Microsoft Windows command, the usage syntax is completely different as can be seen from the command snippet taken from the Mozilla

```
        list = PK11_GetAllTokens(CK_INVALID_MECH,PR_FALSE,PR_FALSE,
&pwdata);
                if (list) for (le = list->head; le; le->next) {
                   rv = PK11_CertsInSlot(le->slot, <your_callback>, <your_params>);
                }
```

web site.

Author Name, email@address

2.3. MAC OSX and Safari

Apple implements a certificate store (keychain) in MAC OSX. Unlike Windows, the Apple certificate store is a combination of a password manager and certificate store. By default, there are two keychains within the system referred to as the login and system keychains. The user may create more keychains. Similar to the other products, both GUI and command line interfaces are available (OS X Man Pages, 2012).

```
security> export -k login.keychain -t certs -o /tmp/certs.pem
security> find-certificate -a -p > allcerts.pem
```

Exports all certificates from all keychains into a pem file called allcerts.pem.

Additional care is required when auditing the OSX keychain as it handles much more than certificates. Respect the privacy of the other data objects including passwords and other encrypted data types if they are not in scope of the audit.

Safari running in a Microsoft Windows environment utilizes the standard Microsoft certificate stores.

2.4. OpenSSL

OpenSSL stores CA Certificates in unencrypted PEM files by default. File system security is of utmost importance to protect these files. Software applications that use the OpenSSL cryptographic API will sometimes implement a dedicated certificate store as part of the application. The number of systems reported to be vulnerable to the Heartbleed attack provides clear evidence to the common usage of the OpenSSL cryptographic API.

Author Name, email@address

2.5. JAVA

A substantial number of deployed applications and devices rely on JAVA. JAVA key stores are contained within encrypted files. The trusted root CA certificates for JAVA are stored in the file (path to JAVA)/lib/security/cacerts. Unlike the Mozilla trusted certificate store, the user can update this store. This file only stores CA certificates without keys. Certificates with keys are stored in an additional key store having the default extension of .jks. In order to audit JAVA Key stores, the auditor requires both of the key store passwords. The default passwords on many java-based devices are 'changeit' for the cacerts file and 'testpassword' for the user key store (keytool - Key and Certificate Management Tool, n.d.). Application developers rarely change these passwords and simply rely on file system security on the device to protect the key stores. In many cases, the hard coded default passwords remain permanently in the device software.

Apache Tomcat, VMWare products, and of course the JAVA Runtime are example applications which implement cryptography based on JAVA.

3. Certificate Usage

In basic terms, certificates support the implementation of authentication, integrity, and confidentiality. When enhanced by other processes and applications, certificates also support authorization and non-repudiation. Commonly, these PKI security mechanisms combine to provide two distinct usage cases; secure tunnels and secure data structures. In the scope of this paper, business cases with external entities provide the largest area for analysis.

3.1. Secure Tunnels

Secure tunnels most commonly include SSL, TLS, IPSEC, and SSH. There are many other examples of secure tunnels, but most of them use SSL or TLS as their transport mechanism.

Author Name, email@address

One special case of secure tunnels shown in Figure 3 is an SSL interception proxy. Within the Enterprise, interception proxies allow for the analysis of encrypted traffic using the SSL or TLS protocols. SSL proxies behave like certificate authorities, in

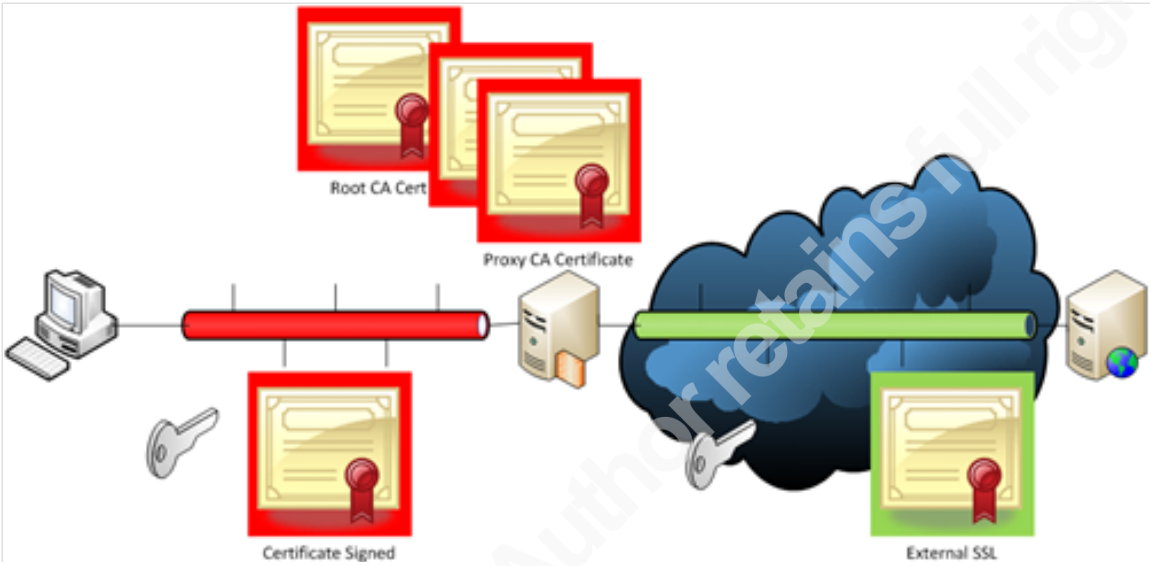

Figure 3 SSL Interception Proxy Configuration

that they issue SSL certificates for all of the secured sites that internal users visit. The proxy intercepts HTTPS requests and pretends to be the destination web site. At the same time, the proxy establishes another connection to the real web server.

Unauthorized SSL proxy certificates such as Superfish (Superfish Joins the MITM Club, 2015) will also appear within the audit results.

3.2. Secure Data Structures

For the purpose of this paper, secure data structures refer to digitally signed and optionally encrypted objects. Example data structures that support digital signatures include S/MIME (Ramsdell & Turner, 2010), digitally signed documents, and signed code. S/MIME implements the PKCS#7 signed data structure as described in (kaliski, 1998). Microsoft's Windows Authenticode Portable Execution Signature Format also leverages the PKCS#7 standard (Windows Authenticode Portable Executable Signature Format, 2008).

The quality of the code is outside the scope of this audit task. The quality of the code permitted to execute within the operating system is critical to system security, but requires a different scope of analysis to this paper. The primary intent of auditing certificate trust is to understand who signed the data object, and whether we have a business need to trust that signature. A good source to learn more about digitally signed malware is the SANS InfoSec Handlers Diary Blog (Zeltser, 2013).

Commonly signed executable code includes DLLs, executables MSI (installer) files, scripts, XML, Authenticode, ActiveX, and macros.

4. Whom do you trust?

4.1. Gathering raw data

Building the answer to this question requires a series of queries and scripts. At the top end are the Root CA certificates. At the bottom end are the end user certificates used for day-to-day business. In between are an unknown number of intermediate certificates. Certificate stores have a range of formats, locations, and permissions.

Appendix A provides proof of concept scripts. The first script sequentially parses through the identified certificate stores and writes each certificate into a directory structure corresponding to the certificate chain where it belongs. The first script catalogs

```
Get-ChildItem -File -Name -Recurse >Certlist.txt
```

the certificate thumbprint, subject, Issuer, subject key identifier, and authority key identifier. Next, this script checks the validity of the non-root certificates. A single line script writes a hierarchical directory listing to a file to allow for additional post processing in Microsoft Excel.

4.2. Post Processing

The second script scans the file system looking for signed code. When found, the script verifies the digital signature of the file. If the signature is valid, the script writes the

filename and the thumbprint of the signing certificate to a second csv file. The signing certificate is stored to the same directory structure as the first script.

At this stage, there are potentially a large number of certificates with no clear linkages between them. The post processing uses the thumbprint, SKI, AKI, and subject information within certificates to link the end user certificates to their appropriate Root CA certificates. The establishment of these chains will determine which root CA Certificates are required by the operating system to function.

Microsoft provides a list of required certificates for its family of operating systems (Trusted root certificates that are required by Windows , n.d.). Verification of the Microsoft list against the results of the audit analysis will provide an additional level of validation of the scripts in use. If the audit process recommends the deletion of a certificate on the Microsoft list, (or any other browser or operating system vendor list,) conduct additional investigation before implementing trust pruning on an operational network.

4.3. Tools

There are a number of tools and scripts referenced within this paper. Select tools for an audit based on their efficiency, effectiveness, cost, and availability within the Enterprise. Many of these tools automatically install with the relevant application or operating system. Others are available free for download.

There are commercial tools available that perform several of the data gathering tasks described in this paper. Two of these products are Entrust Discovery and Venafi TrustAuthority. As the goal of the paper centers on the process to determine the final set of trusted root certificates to retain, commercial tools remain out of scope of the analysis.

5. Putting it all together

5.1. Understanding the big picture

The first step of the certificate audit is to enumerate the complete list of certificates installed within the local computing environment. While auditing the trusted root certificates is the primary goal, the full chains of certificates are required to determine which root certificates are critical to the business needs of the organization. When removing a trusted root CA certificate, remove the entire certificate chain at the same time.

The greatest effort required to succeed during a certificate trust audit is to understand the business needs and business processes of the organization. The framework of the 20 Critical Security controls (Critical Security Controls, n.d.) provides a clear organizational structure for this analysis. An organization with a mature implementation of these controls has the tools and business processes in place to succeed with the audit. Figure 4 depicts the Root CA trust management process.

5.1.1. Inventory of Devices (Control 1)

The inventory of allowed devices provides a starting point to enumerate trusted root certificates within those devices by identifying the type of cryptographic libraries and by extension, the types of certificate stores installed on the device. Devices may use

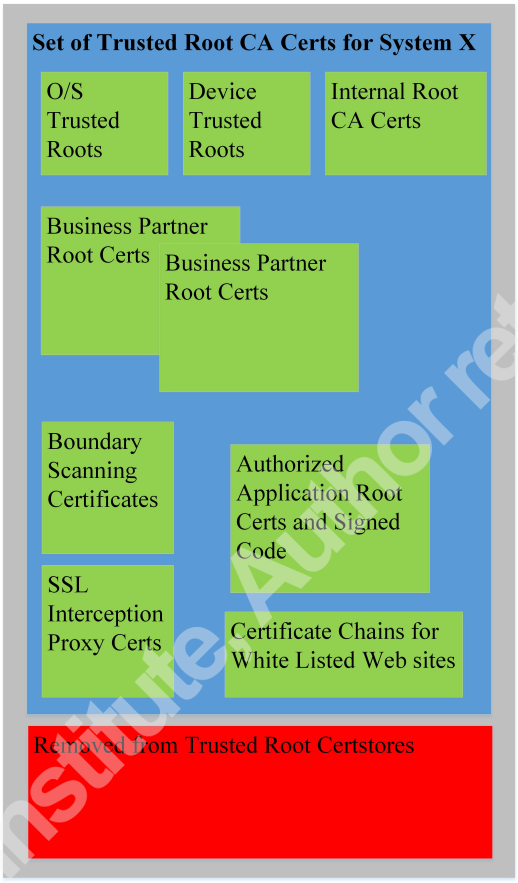

Figure 4 Root CA Trust Management

externally provided certificates to secure the management interface.

5.1.2. Inventory of Software (Control 2)

The inventory of supported Operating System types and versions provides a baseline to determine the minimum root CA certificates required by the operating system. An enumeration of the software application inventory verifies the complete identification of all signed code described earlier.

5.1.3. Secure Configurations for IT devices (Control 3)

The output of the audit feeds back into this control. Record the required root CA certificates and subordinate certificates within each operating system and application profile. Configuration control on profiles is critical. If a new code-signing certificate signed an application patch, the entire certificate chain may need updating at the same time. This would occur only in circumstances where the certificate provider for the software vendor changes. This could alternatively be an indication of malicious intent on the part of a third party.

5.1.4. Application Software Security (Control 6)

Organizations that implement SSL interception proxies for outgoing HTTPS connections are able to remove a larger number of certificates from internal systems. The root CA issuing credentials to the SSL proxy needs to be included into the internal trusted Root CA certificate store only for those devices or workstations with a business need for communications through the SSL proxy. Capture sites that bypass the SSL Proxy, including the required certificate chains, as a required setting within the appropriate system audit files. Auditing certificate chains at the SSL proxy or gateway device provides a treasure trove of useful information, whether or not the organization implements SSL session interception and decryption at the gateway. There are two obvious ways to gather SSL Certificate chains. The first way is to log all HTTPS connection request headers at the gateway. A script processes this data, connecting to every unique web server and downloading the certificate chain (McCabe, 2014). The second approach involves passively monitoring the stream of HTTPS traffic as it arrives inbound to the network and capturing the certificate chains in near real time (Amann, Vallentin, Hall, & Sommer, 2012).

Organizations that implement Secure E-mail gateways need to document business partners relationships including the relevant address space and certificate hierarchies.

5.1.5. Secure Configuration for Network (Control 10)

Firewalls, Virtual Private Network (VPN) appliances, and mail gateways often include their own trusted root certificate stores. These certificate chains are also within the scope of the audit. Additionally, the secure traffic passing through these devices falls within the scope.

5.1.6. Data Protection (Control 17)

Document configuration information and audit logs from Mail and Web Gateways along with details of any secure channels in place with business partners. If an SSL interception proxy is in place, the Proxy CA certificate chain is required, along with the list of sites that bypass the proxy.

5.2. Example Root Certificate vetting

Microsoft Office is on the list of authorized applications. In Table 4, the files in the directory, C:\Program Files \Microsoft Office15\root\integration have valid digital signatures, signed with the certificate thumbprint 67B1757863E3EFF760EA9EBB02849AF07D3A8080.

Table 5, the hierarchy of Code Signing Certificates also contains a reference to the code-signing certificate with the same thumbprint. A Root Certificate authority with a thumbprint of cdd4eeae6000ac7f40c3802c171e30148030c072 issued this certificate.

From Table 3, the DN for this certificate is CN=Microsoft ROOT Certificate Authority, DC=Microsoft, DC=COM

As this application is on the authorized application list, the Root CA Certificate CN=Microsoft ROOT Certificate Authority, DC=Microsoft, DC=COM must be retained in the Trusted Root CA certificate store.

Author Name, email@address

5.3. Pruning Trust

There are more trusted root CA certificates loaded into most systems than are actually required for normal business activities. Plan the selection process for pruning trust carefully and subject it to the same configuration management processes already in place within the organization. Self-denial of service incidents will quickly derail the management of trust in an organization.

6. Conclusions

So far, the scripts have run on a limited number of personal desktop machines running Windows 7 plus one test machine running Windows 10. The Windows 10 machine does not yet include the full Microsoft root certificate collection, as it is still a prerelease version at the time of writing this document. All machines tested contained well over 100,000 signed files. There have typically been just over 100 code-signing certificates and less than 100 Root CA certificates at the top of the certificate chains.

When completed manually, the vetting process described in 5.2 feels quite awkward and labor intensive. When augmented with the Excel table lookup formula, the process becomes quite intuitive. Implementing the last steps as a pivot table in Excel would provide a visual process that even senior management could understand.

Author Name, email@address

7. References

Amann, J., Vallentin, M., Hall, S., & Sommer, R. (2012, November). *Extracting Certificates from Live Traffic: A Near Real-Time SSL Notary Service.* Retrieved from International Computer Science Institute: http://www.icir.org/johanna/papers/icsi12extractingcertificates.pdf

CA:FAQ. (2015, February 13). Retrieved July 21, 2015, from Mozzilla wiki: https://wiki.mozilla.org/CA:FAQ

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008, May). *RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Retrieved from http://www.ietf.org/rfc/rfc5280.txt

Critical Security Controls. (n.d.). Retrieved from http://www.sans.org/critical-security-controls/

Entrust Inc. (2014, June). *Entrust Discovery 2.4 Administration Guide Issue 3.0.* Retrieved from Entrust Trusted Care: https://secure.entrust.com/trustedcare/documentation/proxy2.cfm/external/24828/discovery_2_4_admin_guide_issue3.pdf

Housley, R., Ashmore, S., & Wallace, C. (2010, June). *RFC 4949 - Internet Security Glossary, Version 2*. Retrieved July 21, 2015, from Internet Engineering Task Force (IETF): https://tools.ietf.org/html/rfc5914

Included CAs. (2015, March 02). Retrieved July 21, 2015, from Mozzilla wiki: https://wiki.mozilla.org/CA:IncludedCAs

kaliski, B. (1998, March). *PKCS #7: Cryptographic Message Syntax.* Retrieved from Internet Engineering Task Force (IETF): https://tools.ietf.org/html/rfc2315

keytool - Key and Certificate Management Tool. (n.d.). Retrieved July 21, 2015, from Oracle Java SE Documentation: http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html

Author Name, email@address

Manage Trusted Root Certificates. (n.d.). Retrieved from Windows Server: https://technet.microsoft.com/en-us/library/cc754841.aspx

McCabe, J. (2014, June 26). *Reading a Certificate off a remote SSL Server for Troubleshooting with Powershell!* Retrieved from Parallel Universe - MS Tech Blog: http://blogs.technet.com/b/parallel_universe_-_ms_tech_blog/archive/2014/06/26/reading-a-certificate-off-a-remote-ssl-server-for-troubleshooting-with-powershell.aspx

Microsoft. (2015, March 24). *Introduction to The Microsoft Root Certificate Program*. Retrieved April 30 , 2015, from Microsoft Technet: http://social.technet.microsoft.com/wiki/contents/articles/3281.introduction-to-the-microsoft-root-certificate-program.aspx

Naish, M. (2014, September 13). *Infosec reading room*. Retrieved from SANS Institute: http://www.sans.org/reading-room/whitepapers/certificates/implementing-public-key-infrastructure-pki-microsoft-windows-server-2012-certificate-servic-35427

OS X Man Pages. (2012, March 1). Retrieved from MAC Developer Library: https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/security.1.html

Ramsdell, B., & Turner, S. (2010, January). *RFC 5751 - Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification*. Retrieved from Internet Engineering Task Force (IETF): https://tools.ietf.org/html/rfc5751

Superfish Joins the MITM Club. (2015, February 20). Retrieved from Entrust IDentity ON: http://www.entrust.com/superfish-joins-mitm-club/

Trust - Definition and more from the free Merriam-Webster dictionary. (n.d.). Retrieved from free Merriam-Webster dictionary: http://www.merriam-webster.com/dictionary/trust

Trusted root certificates that are required by Windows . (n.d.). Retrieved from Microsoft Support: https://support.microsoft.com/en-us/kb/293781

Author Name, email@address

Vandeven, S. (2014, July 14). *Infosec reading room*. Retrieved from SANS Institute: https://www.sans.org/reading-room/whitepapers/certificates/digital-certificate-revocation-35292

Windows Authenticode Portable Executable Signature Format. (2008, August 29). Retrieved from Windows Hardware Dev Center : https://msdn.microsoft.com/en-us/library/windows/hardware/gg463180.aspx

Zeltser, L. (2013, May 2011). *Extracting Digital Signatures from Signed Malware*. Retrieved from InfoSec Handlers Diary Blog: https://isc.sans.edu/diary/Extracting+Digital+Signatures+from+Signed+Malware/15779

Author Name, email@address

Appendix A

Microsoft Certificate Stores

Navigating certificate stores with PowerShell is much like browsing directories on

```
set-location -path cert:\LocalMachine\Root

get-childitem >rootcerts.txt
```

Thumbprint	Subject
----------	-------
CDD4EEAE6000AC7F40C3802C171E30148030C072	CN=Microsoft Root Certificate Authority, DC=microsoft, DC=com
BE36A4562FB2EE05DBB3D32323ADF445084ED656	CN=Thawte Timestamping CA, OU=Thawte Certification, O=Thawte, L=Durbanvill...
A43489159A520F0D93D032CCAF37E7FE20A8B419	CN=Microsoft Root Authority, OU=Microsoft Corporation, OU=Copyright (c) 19...
8F43288AD272F3103B6FB1428485EA3014C0BCFE	CN=Microsoft Root Certificate Authority 2011, O=Microsoft Corporation, L=R...
7F88CD7223F3C813818C994614A89C99FA3B5247	CN=Microsoft Authenticode(tm) Root Authority, O=MSFT, C=US
3B1EFD3A66EA28B16697394703A72CA340A05BD5	CN=Microsoft Root Certificate Authority 2010, O=Microsoft Corporation, L=R...
245C97DF7514E7CF2DF8BE72AE957B9E04741E85	OU=Copyright (c) 1997 Microsoft Corp., OU=Microsoft Time Stamping Service ...
18F7C1FCC3090203FD5BAA2F861A754976C8DD25	OU="NO LIABILITY ACCEPTED, (c)97 VeriSign, Inc.", OU=VeriSign Time Stampin...
F18B538D1BE903B6A6F056435B171589CAF36BF2	CN=thawte Primary Root CA - G3, OU="(c) 2008 thawte, Inc. - For authorized...
E12DFB4B41D7D9C32B30514BAC1D81D8385E2D46	CN=UTN-USERFirst-Object, OU=http://www.usertrust.com, O=The USERTRUST Netw...
DE28F4A4FFE5B92FA3C503D1A349A7F9962A8212	CN=GeoTrust Global CA, O=GeoTrust Inc., C=US
D69B561148F01C77C54578C10926DF5B856976AD	CN=GlobalSign, O=GlobalSign, OU=GlobalSign Root CA - R3
D4DE20D05E66FC53FE1A50882C78DB2852CAE474	CN=Baltimore CyberTrust Root, OU=CyberTrust, O=Baltimore, C=IE
D23209AD23D314232174E40D7F9D62139786633A	OU=Equifax Secure Certificate Authority, O=Equifax, C=US
CA3AFBCF1240364B44B216208880483919937CF7	CN=QuoVadis Root CA 2, O=QuoVadis Limited, C=BM
B51C067CEE2B0C3DF855AB2D92F4FE39D4E70F0E	CN=Starfield Root Certificate Authority - G2, O="Starfield Technologies, I...
B31EB1B740E36C8402DADC37D44DF5D4674952F9	CN=Entrust Root Certification Authority, OU="(c) 2006 Entrust, Inc.", OU=w...
B1BC968BD4F49D622AA89A81F2150152A41D829C	CN=GlobalSign Root CA, OU=Root CA, O=GlobalSign nv-sa, C=BE
AD7E1C28B064EF8F6003402014C3D0E3370EB58A	OU=Starfield Class 2 Certification Authority, O="Starfield Technologies, I...
A8985D3A65E5E5C4B2D7D66D40C6DD2FB19C5436	CN=DigiCert Global Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US
99A69BE61AFE886B4D2B82007CB854FC317E1539	CN=Entrust.net Secure Server Certification Authority, OU=(c) 1999 Entrust....
97817950D81C9670CC34D809CF794431367EF474	CN=GTE CyberTrust Global Root, OU="GTE CyberTrust Solutions, Inc.", O=GTE ...
91C6D6EE3E8AC86384E548C299295C756C817B81	CN=thawte Primary Root CA, OU="(c) 2006 thawte, Inc. - For authorized use ...
8F43288AD272F3103B6FB1428485EA3014C0BCFE	CN=Microsoft Root Certificate Authority 2011, O=Microsoft Corporation, L=R...
8CF427FD790C3AD166068DE81E57EFBB932272D4	CN=Entrust Root Certification Authority - G2, OU="(c) 2009 Entrust, Inc. -...
8782C6C304353BCFD29692D2593E7D44D934FF11	CN=SecureTrust CA, O=SecureTrust Corporation, C=US
85371CA6E550143DCE2803471BDE3A09E8F8770F	OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized us...
75E0ABB6138512271C04F85FDDDE38E4B7242EFE	CN=GlobalSign, O=GlobalSign, OU=GlobalSign Root CA - R2
742C3192E607E424EB4549542BE1BBC53E6174E2	OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US
74207441729CDD92EC7931D823108DC28192E2BB	CN=Class 2 Primary CA, O=Certplus, C=FR
627F8D7827656399D27D7F9044C9FEB3F33EFA9A	E=premium-server@thawte.com, CN=Thawte Premium Server CA, OU=Certification...
5FB7EE0633E259DBAD0C4C9AE6D38F1A61C7DC25	CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc...
58119F0E128287EA50FDD987456F4F78DCFAD6D4	CN=UTN - DATACorp SGC, OU=http://www.usertrust.com, O=The USERTRUST Networ...
503006091D97D4F5AE39F7CBE7927D7D652D3431	CN=Entrust.net Certification Authority (2048), OU=(c) 1999 Entrust.net Lim...
4EB6D578499B1CCF5F581EAD56BE3D9B6744A5E5	CN=VeriSign Class 3 Public Primary Certification Authority - G5, OU="(c) 2...
47BEABC922EAE80E78783462A79F45C254FDE68B	CN=Go Daddy Root Certificate Authority - G2, O="GoDaddy.com, Inc.", L=Scot...
3E2BF7F2031B96F38CE6C4D8A85D3E2D58476A0F	CN=StartCom Certification Authority, OU=Secure Digital Certificate Signing...
3921C115C15D0ECA5CCB5BC4F07D21D8050B566A	CN=America Online Root Certification Authority 1, O=America Online Inc., C=US
3679CA35668772304D30A5FB873B0FA77BB70D54	CN=VeriSign Universal Root Certification Authority, OU="(c) 2008 VeriSign,...
2796BAE63F1801E277261BA0D77770028F20EEE4	OU=Go Daddy Class 2 Certification Authority, O="The Go Daddy Group, Inc.",...

Table 3 Microsoft root certificate store

Author Name, email@address

a hard drive. Table 3 shows the export of the Microsoft Windows 7 root certificate store.

Validate Authenticode signatures in PowerShell

Validating the collection of signed code provides visibility to the auditor showing clearly, what has changed since the previous audit. The auditor should validate the transition from the previous baseline, validating all managed configuration changes, and should arrive at the current audit configuration without discrepancies.

```
get-child-item –name –recuse –force | foreach-object {get-
authenticodesignature $_} >output.txt
```

	Directory:	C:\ProgramFiles\MicrosoftOffice15\root\Integration
SignerCertificate	Status	Path
----------	------	----
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	C2RInt.msi
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	C2RIntLoc.en-us.msi
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	C2RIntLoc.fr-fr.msi
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	integrator.exe
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	onedrivesetup.exe
6474839AF67AB79C91007FF62FE08E2ACF016B83	HashMismatch	OneDriveSetup.exe.bak
	NotSigned	QFE31927.msp
	NotSigned	QFE31928.msp
	NotSigned	QFE31932.msp
108E2BA23632620C427C570B6D9DB51AC31387FE	Valid	SkyDriveSetup.exe.bak
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	SPPRedist.msi
67B1757863E3EFF760EA9EBB02849AF07D3A8080	Valid	SPPRedist64.msi
	Directory:	C:\ProgramFiles\MicrosoftOffice15\root\Licenses
SignerCertificate	Status	Path
----------	------	----
	Directory:	C:\ProgramFiles\MicrosoftOffice15\root\loc
SignerCertificate	Status	Path
----------	------	----
	Directory:	C:\ProgramFiles\MicrosoftOffice15\root\mcxml
SignerCertificate	Status	Path
----------	------	----
AC1FD0922A4A2A6E5779ACDD628747C28394B0B9	Valid	AppVIsvSubsystems32.dll
AC1FD0922A4A2A6E5779ACDD628747C28394B0B9	Valid	AppVIsvSubsystems64.dll

Table 4 Validated Authenticode Signatures

Author Name, email@address

Author Name, email@address

Hierarchy of Code Signing Certificates

SKI of Root Certificate	SKI of Sub CA (Plus Root Certificate)	SKI of Code Signing Certs (Plus Sub CA certificate)	Code Signing Certificates.
0eac826040562797e52513fc2ae10a539559e4a4	cdd4eeae6000ac7f40c3802c171e30148030c072.cer		
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	D57FAC60F1A8D34877AEB350E83F46F6EFC9E5F1.cer	
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	fa06fc4b85ca532d997fc82652d59750c2084ec7	E0C9681409908329BC40551075EA179C6F1D1DAE.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	cb781223d895dfc272ce2b3d63dcf37d337dbd6a	009EDD1DB65137F4E33EBCACC5C4128EEF5BE42A.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	df0689bf85a2043a8c3dbb32dd6975ae5f2975c4	0480C9444EA631C9BBE497D86A3D27AA940A06F0.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	5971a65a334dda980780ff841ebe87f9723241f2	108E2BA23632620C427C570B6D9DB51AC31387FE.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	14c2b0a5082f5458485cd4774abb9c8fcb1df6b6	10CA4C317BB85198AC3D1EA07B07E1A32675FF68.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	1b520ef3112ab7c089c311813dd49658b9a3536e	19F8F76F4655074509769C20349FFAECCECD217D.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	6bc8c65120f0b42fd3a0b6ae7f5e26b2b8875229	282D9806C3DF7345929F64F5895EF2EA4AC29302.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	e7e4ad81c6064a14d20f21e87daff9b8b0819b6f	2DC6287C7005CC05E1B4C1E698A4426144153DA6.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	4da6dd6e22abb45b305f656d0960a386924f0f50	47DFD0F96F7E0EFB6199DEC35C7B8EFEB93040A4.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	eed96ba97553cd4fee1b4e19061ea39cabcf94fd	564E01066387F26C912010D06BD78D3CF1E845AB.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	b66fa30f40eed3e91e5eca1ea001767cf1f9a8e9	5893C42A688EBEAEED2B7F90B15B654ED33248DB.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	1f5ee25d508d5686be4a3ccf04e8a787b5cbbf83	67B1757863E3EFF760EA9EBB02849AF07D3A8080.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	1e5d4be4a6a94c089da23126963a1eb8ff175a38	6EBDFB3CF5D0A9691F8700F96BB16D62D344F229.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	23d2e96db9630f664108eda164c2b0946880ef48	6FFAD4A3B15F6A2C71D43C8E551DCECAB3A5183C.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	149e5f8161edc9d98cf72766e9299f5737ff904e	70DA0D1A49F1F74886EC597D3596B8F9A84F77D6.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	e7e4ad81c6064a14d20f21e87daff9b8b0819b6f	78F1C3CC14F772D253B53858A428475C120A2F6F.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	dc7edeba8cdafa98f2bc1e52dad818a978db6735	7C0D1182AEE7777FD76B44CBF133DB6325D30929.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	265b3e5b5d965fe2f777887f5e455358a82e5bb8	8363887511B4835B79C383ECF06FC055B5839255.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	0003a6e5a2c471a282c38f76cd2c0d12e29d4a28	8849D1C0F147A3C8327B4038783AEC3E06C76F5B.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	d972d4cb0c625fdda3749f5f0e9841a85b134326	93859EBF98AFDEB488CCFA263899640E81BC49F1.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	38780573c81b329b5f928655af89bac699b1748e	9617094A1CFB59AE7C1F7DFDB6739E4E7C40508F.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	e46f5fca89f953bef070570650aac2790dac569b	9BF69D5E8D01A92F413B60A4BE003E323CB52F7F.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	23d1732a4cbdfbe521fa711e9915d117f9c4a68a	9E95C625D81B2BA9C72FD70275C3699613AF61E3.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	de51dc1bf21e4a053f121cc1bcc3b3652c910598	AC1FD0922A4A2A6E5779ACDD628747C28394B0B9.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	2c5a7b5abc063c62f62b587713700213 1bbfdfc6	BC0B6D0D7398035FCFBE8CC1AD8724A23A3A89DB.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	da26f8c95623a1b4bcd25d0a4a5536ed69132808	C7C2A56AE0B3C3D85E595A35C4BF74C8C6B28E55.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	8781b7dfeefa77fa59056337040ad7e9daa0e0e0	D468FAEB5190BF9DECD9827AF470F799C41A769C.cer
0eac826040562797e52513fc2ae10a539559e4a4	f321408e7c51f8544b98e517d76a8334052e26e8	149e5f8161edc9d98cf72766e9299f5737ff904e	D5A80B750063C72A76ED19BB46A32D212362EF08.cer

Table 5 Hierarchy of Exported Code Signing Certificates

Author Name, email@address