NIST SPECIAL PUBLICATION 1800-16B

Securing Web Transactions

TLS Server Certificate Management

Volume B:
Security Risks and Recommended Best Practices

Murugiah Souppaya
Computer Security Division
Information Technology Laboratory

William Haag
Applied Cybersecurity Division
Information Technology Laboratory

Paul Turner
Venafi
Salt Lake City, UT

William C. Barker
Dakota Consulting
Silver Spring, MD

November 2018

DRAFT

This publication is available free of charge from:
https://www.nccoe.nist.gov/projects/building-blocks/tls-server-certificate-management

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

NCCoE
NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

DISCLAIMER

National Institute of Standards and Technology Special Publication 1800-16B, Natl. Inst. Stand. Technol. Spec. Publ. 1800-16B, 44 pages, (November 2018), CODEN: NSPUE2

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to tls-cert-mgmt-nccoe@nist.gov.

Public comment period: November 29, 2018 through December 31, 2018.

All comments are subject to release under the Freedom of Information Act (FOIA).

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in IT security—the NCCoE applies standards and best practices to develop modular, easily adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to recreate the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Md.

To learn more about the NCCoE, visit https://www.nccoe.nist.gov/. To learn more about NIST, visit https://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

This NIST Cybersecurity Practice Guide consists of the following volumes:

- **Volume A:** an executive-level summary describing the challenge that the TLS Server Certificate Management Project addresses, and a high-level description of the recommended solution (posted for public comment concurrent with Volume B)
- **Volume B:** recommended best practices for large-scale TLS server certificate management (posted for public comment concurrent with Volume A, *Executive Summary*)
- **Volume C:** a description of an example automated TLS certificate management solution for preventing, detecting, and recovering from certificate-related incidents, and a mapping of the example solution's capabilities to the recommended best practices and to NIST security guidelines and frameworks **(planned for 2019 release)**

- **Volume D:** a description of how to build this example solution **(planned for 2019 release)**

The solutions and architectures presented in this practice guide are built upon standards-based, commercially available and open-source products. These solutions can be used by any organization managing TLS server certificates. Interoperable solutions are provided that are available from different types of sources (e.g., both commercial and open-source products).

KEYWORDS

Authentication; certificate; cryptography; identity; key; key management; PKI; private key; public key; public key infrastructure; server; signature; TLS; Transport Layer Security

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
Dean Coclin	DigiCert
Tim Hollebeek	DigiCert
Clint Wilson	DigiCert
Robert Smith	F5
Rob Clatterbuck	SafeNet AT
Jane Gilbert	SafeNet AT
Seamus Hartmann	Symantec
Alexandros Kapasouris	Symantec
Mehwish Akram	The MITRE Corporation
Brett Pleasant	The MITRE Corporation
Susan Prince	The MITRE Corporation

Name	Organization
Susan Symington	The MITRE Corporation
Aaron Aubrecht	Venafi
Justin Hansen	Venafi

Contents

List of Figures

1 Introduction

Organizations risk losing revenue, customers, and reputation, and exposing internal or customer data to attackers if they do not properly manage Transport Layer Security (TLS) server certificates. TLS is the most widely used security protocol to secure web transactions and other communications on the internet and internal networks. TLS server certificates are central to the security and operation of internet-facing and internal web services. Improper TLS server certificate management results in significant outages to web applications and services—such as government services, online banking, flight operations, and mission-critical services within an organization—and the risk of security breaches. Organizations must ensure that TLS server certificates are properly managed to avoid these issues.

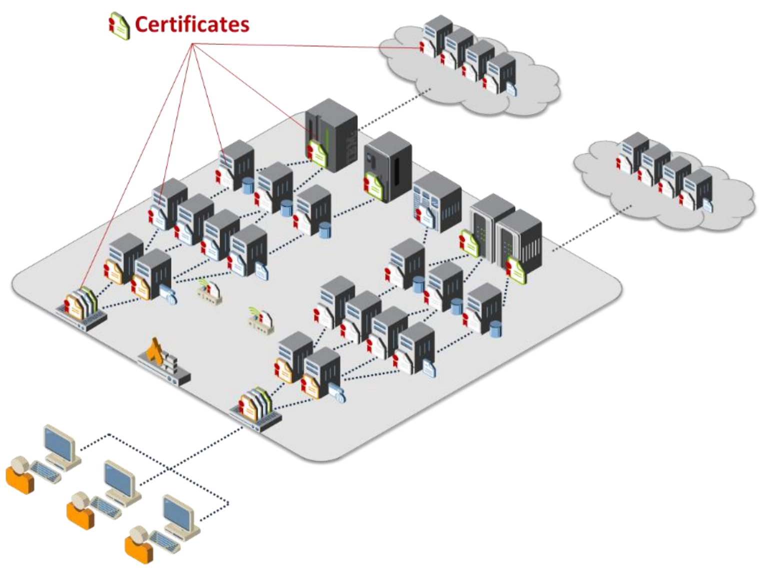
The broad distribution of TLS server certificates across multiple groups and technologies within an enterprise requires that organizations establish formal management programs that include clear policies and responsibilities, a central Certificate Service, and education. Successful implementation of a certificate management program relies on executive sponsorship, clear objectives, an action plan, and regular progress reviews.

2 TLS Server Certificate Background

TLS is the security protocol used to authenticate and protect internet and internal network communications for a broad number of other protocols—including Hypertext Transfer Protocol (http) for web servers; Lightweight Directory Access Protocol (LDAP) for directory servers; and Simple Mail Transfer Protocol, Post Office Protocol, and Internet Message Access Protocol for email.

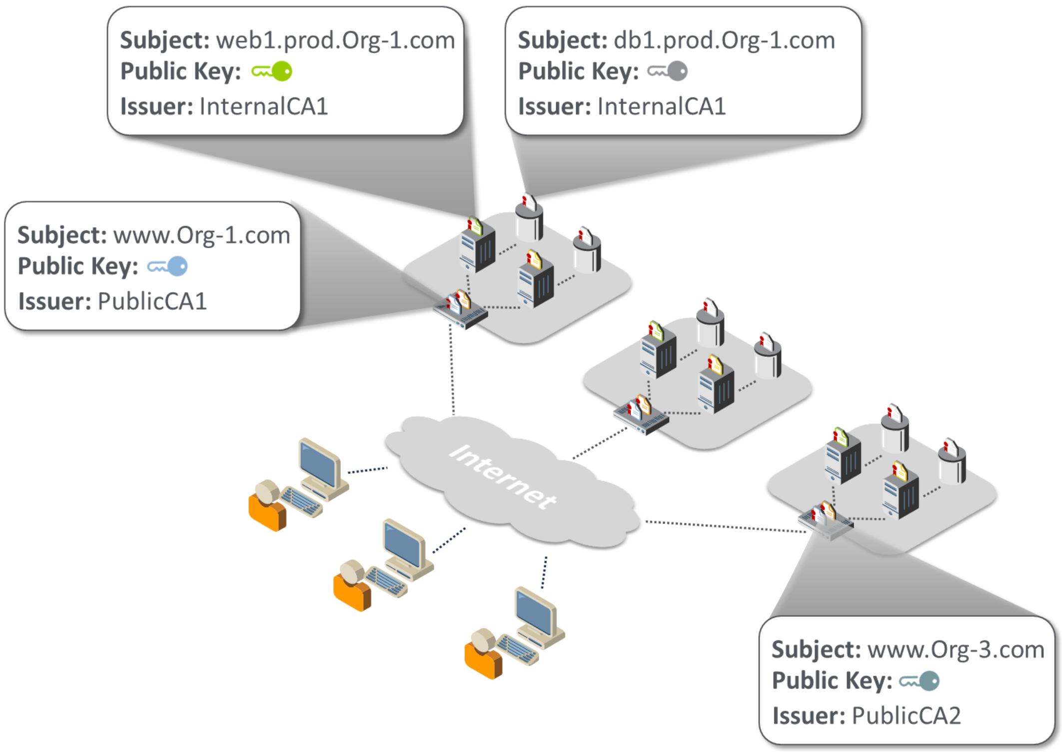
TLS server certificates serve as machine identities that enable clients to authenticate servers via cryptographic means. For example, when a bank customer connects across the internet to an online banking website, the customer's browser (i.e., the TLS client) will present an error message if the server does not provide a valid certificate that matches the address that the user entered in the browser. Further, TLS server certificates are used extensively inside corporate and government networks to establish trust between machines — servers, applications, devices, micro-services, etc. Most enterprises have thousands of certificates, each identifying a specific server in their environment. (Note: Web browsers play the role of clients to web servers. As such, they contain functionality to automatically establish TLS connections on behalf of users, evaluate certificates received during the TLS handshake process, and present errors when unexpected certificate issues are encountered.) Figure 2-1 illustrates the pervasive use of certificates within organizations.

Figure 2-1 TLS Certificates Are Broadly Used for Communications in Organizations



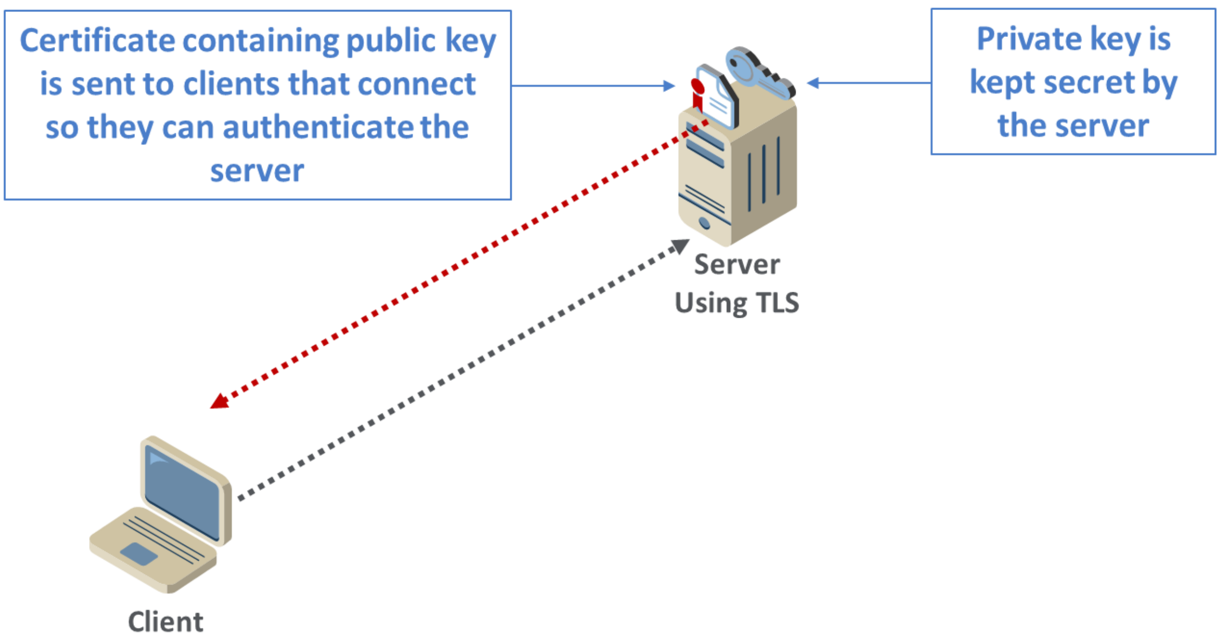
Each TLS server certificate contains the address of the server that it identifies (e.g., www.organization1.com) and a cryptographic key, called a public key, that is unique to the server and used by clients to securely authenticate to the server (see Figure 2-2).

Figure 2-2 Server Address, Public Key, and Issuer Information on Four of the Organization's TLS Server Certificates



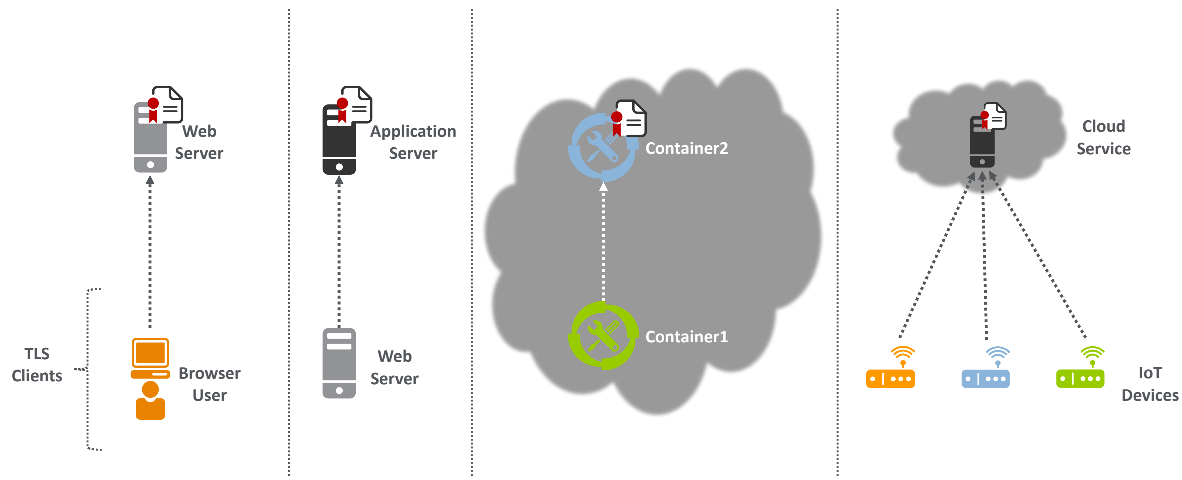
As shown in Figure 2-3, each server holds a private key that corresponds to the public key in the certificate so that each server can prove that it is the holder of the certificate. While the certificate is shared with any client that connects to the server, the private key must be kept secure and secret so that it cannot be obtained by an attacker and used to impersonate the server. Many private keys used with TLS are stored in plaintext files on TLS servers. Alternatively, private keys can be stored in files encrypted with a password; however, the passwords are generally stored in plaintext configuration files so that they are accessible by the TLS server software when it is started. These common practices make it possible for private keys to be viewed and copied by system administrators or malicious actors.

Figure 2-3 Upon Connecting to the Server, the Client Receives the Server's TLS Certificate, Which Includes the Server's Public Key



In addition to users with browsers connecting to servers that have TLS server certificates, automated processes also connect as clients to TLS servers and must trust TLS server certificates. Examples of automated processes acting as TLS clients include a web server making requests to an application server, one cloud container connecting to another, or an Internet of Things (IoT) device connecting to a cloud service. (See Figure 2-4.)

Figure 2-4 Browsers and Various Automated Processes (Web Servers, Containers, and IoT Devices) Connect as Clients to TLS Servers

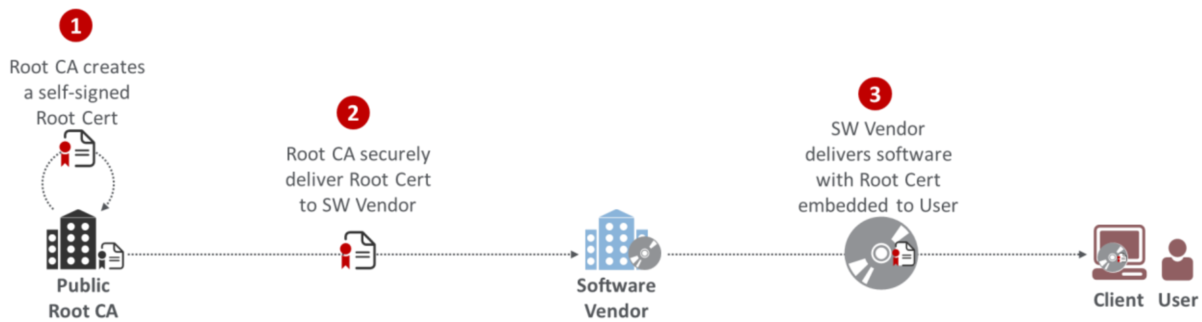


2.1 Certificate Authorities

TLS server certificates are issued by entities called certificate authorities (CAs). CAs digitally sign certificates so that their authenticity can be validated — to prevent attackers from easily impersonating servers. Clients (e.g., browsers, devices, applications, services) validate certificates by using a CA's certificate to verify the signature. Clients, such as browsers, are configured to trust specific CAs (called root CAs). This is done by installing a CA's certificate, commonly called a root certificate, on the client.

Some CAs arrange for their root certificate to get installed by software manufacturers in their software (e.g., browser, application, or operating system) so that the certificates issued by the CAs are trusted broadly. These CAs are commonly called public root CAs. (See Figure 2-5.)

Figure 2-5 A Public Root CA's Root Certificate Is Delivered to the User, Installed on a Software Vendor's Software

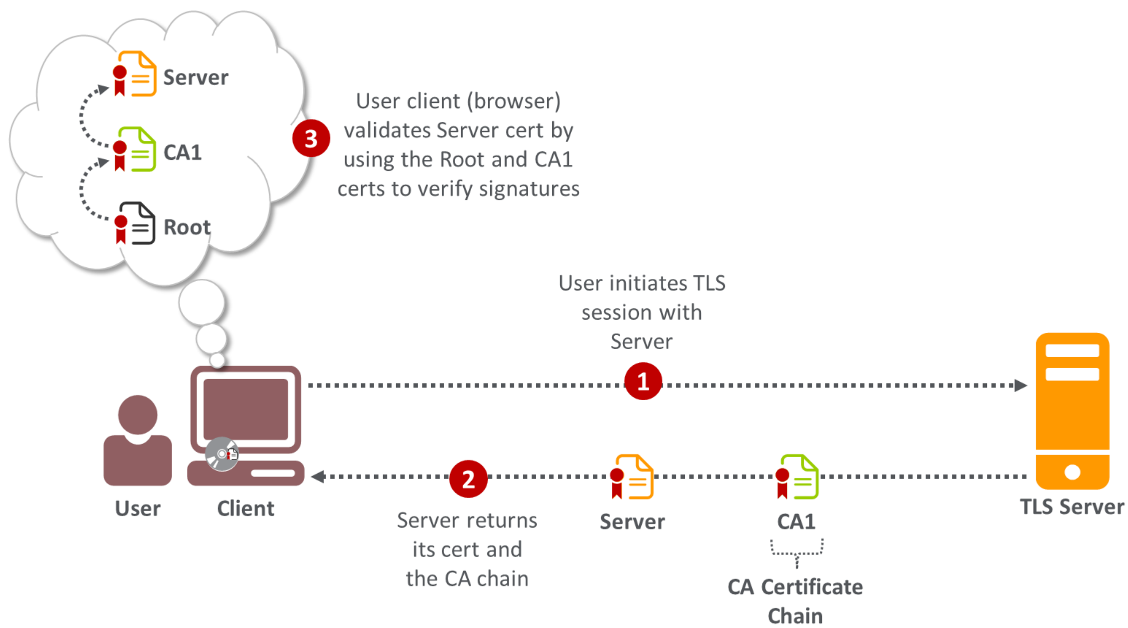


To protect them from attacks, root CAs are generally not connected to the internet and do not issue TLS server certificates directly. Root CAs certify other CAs, generally called intermediate or issuing CAs, which issue TLS server certificates. (See Figure 2-6.)

Figure 2-6 A Root CA Issues a Certificate to an Intermediate/Issuing CA, Which Issues TLS Server Certificates

As shown in Figure 2-7, when a client, such as a browser, connects to a TLS server, the server will return its certificate as well as the certificate for the CA that issued its certificate (called the CA certificate chain).

Figure 2-7 Upon Connecting to the Server, the Client Receives Both the Server's TLS Certificate and Its CA Certificate Chain



Public CAs are regularly audited to ensure that they operate in compliance with the [CA/Browser Forum Baseline Requirements](), which are standards intended to minimize the possibility of CA compromises and fraudulent certificates. When CAs have been found to violate the requirements, their root certificates have been removed from and/or distrusted by browsers, requiring customers of those CAs to rapidly replace their TLS server certificates.

There are three different types of certificates issued by public CAs (as specified by the CA/Browser Forum, which defines standards for public CAs), each with a different level of validation required by the CA to confirm the identity of the requester and its authority to receive a certificate for the domain in question:

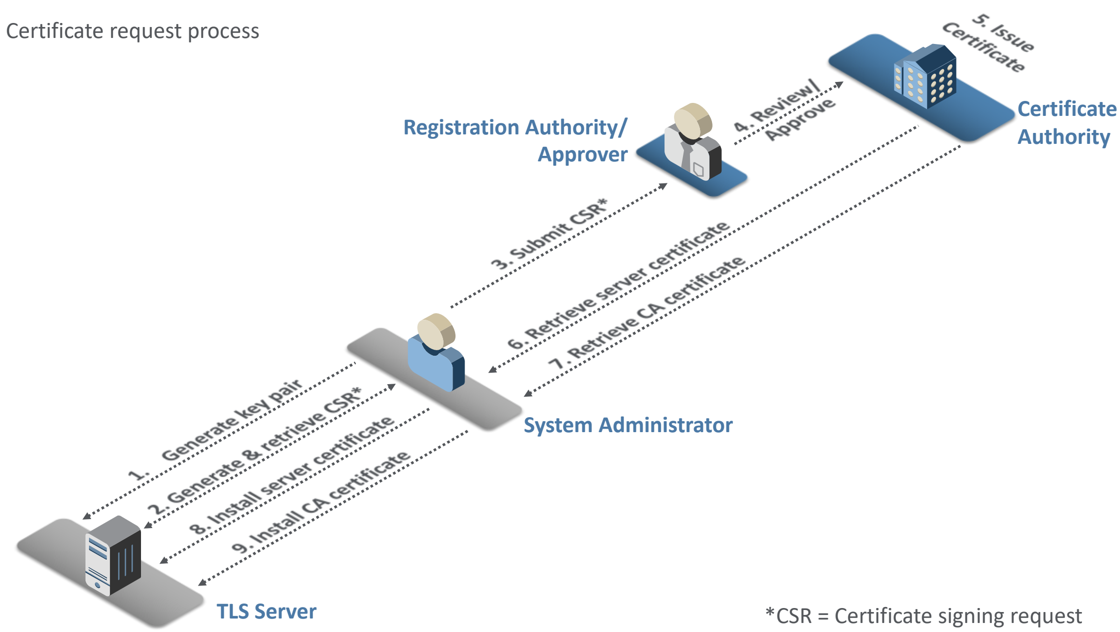
- Domain Validated (DV): The CA validates that the requester is the owner of the domain, by verifying that the requester can reply to an email address associated with the domain, has operational control of the website at the domain address, or is able to make modifications to the Domain Name System (DNS) record for the domain
- Organization Validated (OV): In addition to the checks for DV certificates, the CA conducts additional vetting of the requester's organization
- Extended Validation (EV): EV certificates undergo the most rigorous checks, including verifying the identity and the legal, physical, and operational existence of the entity requesting the certificate, by using official records

Organizations that wish to issue certificates to their internal TLS servers can establish their own CAs, commonly called internal CAs. Organizations using internal CAs must ensure that all clients connecting to their servers trust the internal CAs by installing the internal CAs' root certificates on each system acting as a client (e.g., browsers, operating systems, applications, appliances).

2.2 Certificate Request and Installation Process

The following steps, shown in Figure 2-8 and detailed below, are typically followed by a system administrator to get a TLS certificate for a server that he or she manages.

Figure 2-8 Certificate Issuance Process



1. The system administrator for the TLS server uses utilities on the server to generate a cryptographic key pair (a public key and a private key).
2. The system administrator enters the address of the server (e.g., www.organization1.com). The utilities create a request for a certificate, called a certificate signing request (CSR), which contains the address of the server and the public key. The system administrator retrieves a copy of the CSR (which is contained in a file) from the server.
3. The system administrator submits the CSR to the registration authority (RA), who acts as a reviewer and approver of the certificate request.

4. The RA/approver reviews the CSR, performs necessary checks to confirm the validity of the request and the authority of the requester, and then sends an approval to the CA.
5. The CA issues the certificate.
6. The CA notifies the system administrator that the certificate is ready, either by emailing a copy of the certificate or providing a link from which it can be downloaded. The system administrator retrieves the server certificate.
7. The system administrator retrieves the CA certificate chain from the CA.
8. The system administrator installs the server certificate on the server.
9. The system administrator installs the CA certificate chain on the server.

The CA certificate chain is used by TLS clients to validate the signature on the server certificate. When a client connects to a TLS server, the server returns its certificate and the CA certificate chain, which can contain one or more CA certificates. The client starts with one of its locally trusted root CA certificates and successively validates the signatures on certificates in the CA certificate chain until it reaches the server certificate.

The system administrator must note the expiration date in the certificate and ensure that a new certificate is requested and installed before the existing certificate expires.

3 TLS Server Certificate Risks

When TLS server certificates are not properly managed, organizations risk negative impacts to their revenue, customers, and reputation. There are four primary types of negative incidents that result from certificate mismanagement: outages to important business applications, caused by expired certificates; security breaches resulting from server impersonation; outages or security breaches resulting from a lack of crypto-agility; and increased vulnerability to attack via encrypted threats.

3.1 Outages Caused by Expired Certificates

TLS server certificates contain an expiration date to ensure that the cryptographic keys are changed regularly; this reduces the possibility of a security breach caused by a compromised private key. If a server certificate is not changed before its expiration date, then clients should generate an error message and stop the connection process to the server. This causes the application supported by the server with the expired certificate to become unavailable.

Application outages can also be caused by the mismanagement of CA certificate chains that results in expired intermediate CA certificates. The TLS server is responsible for providing the client with the intermediate CA certificates (CA certificate chain) necessary for the client to link the server’s end-entity certificate with the root CA certificate that is trusted by the client. The absence or expiration of an intermediate certificate means that the client will not trust the server, even though the server may have a perfectly trustworthy end-entity certificate. Intermediate CA certificates are typically renewed every few years, and it is possible for a TLS server to fail to use the most current version. As a result, although the server certificate has been updated, the installed intermediate CA certificate may expire, resulting in an outage due to expiration. Such outages are often difficult to diagnose because the focus of investigation is typically on the server certificate, which is still valid and not the cause of the outage.

Nearly every enterprise has experienced an application outage due to an expired certificate, including outages to major applications such as online banking, stock trading, health records access, and flight operations. Organizations' increased use of TLS server certificates to secure the organizations' applications increases the likelihood of outages because there are more certificates to track and more certificates per business application that can impact operations.

Various scenarios result in a certificate expiring while still in use, causing an outage, including these:

- The system administrator forgets about the certificate
- The system administrator ignores notifications that the certificate will soon expire
- The system administrator does not properly install or update the CA certificate chain
- The system administrator is reassigned, and nobody else receives expiry notifications
- The system administrator enrolls for a new certificate but does not install it on the server(s) in time or installs it incorrectly
- The application relies on multiple load-balanced servers, and the certificate is not updated on all of them

Troubleshooting an incident where an application is unavailable due to an expired certificate can be complex and often requires hours to discover the source of the problem. If the server on which an expired certificate is deployed is being accessed by people using browsers, then each of those people will receive an error message, making it clear that the cause of the issue is an expired certificate. If, on the other hand, the server with the expired certificate is an application server receiving requests from a web server, then the web server stops its operations and may log a message, but that message may not be immediately discovered in the log file, increasing the amount of time required to identify the root cause of the outage and fix it.

3.2 Server Impersonation

An attacker may be able to impersonate a legitimate TLS server (e.g., a banking website) if the attacker is able to get a fraudulent certificate containing the address of the server and the attacker's own public key by tricking a trusted CA into issuing the certificate to the attacker or by compromising the CA and issuing the certificate. A client connecting to the attacker's server will accept the certificate because the certificate contains the address to which the client intended to connect and because the certificate has been issued by a trusted CA. Because the certificate contains the attacker's public key (and the attacker also holds the private key corresponding to this public key), the attacker can decrypt the communications from the client (including passwords intended for login to the legitimate server). Alternatively, if the attacker can access a copy of the legitimate server's private key, then the attacker can also impersonate that server by using the legitimate server's certificate. To successfully perform these attacks, the attacker must redirect traffic destined for the legitimate server to a system that the attacker is operating (e.g., using Border Gateway Protocol [BGP] hijacking or DNS compromise). (Note: The BGP is used to communicate optimal routes between internet service providers on the internet. It is possible for an attacker to hijack traffic by falsely advertising that the fastest route to one or more internet protocol [IP] addresses is via systems that the attacker is operating, thereby causing traffic to be rerouted through the attacker's systems. The DNS provides translation between human-readable addresses [e.g., www.company123.com] and IP addresses. If an attacker can compromise an organization's DNS account, then the attacker can change the IP address to which traffic that is intended for that organization will be sent.)

Most private keys used on TLS servers are stored in files. The private keys are directly managed and handled by system administrators, who can make copies of the private keys. In addition, many TLS servers are clustered (for load balancing); therefore, the TLS server certificate and the private key must be copied to each server in the cluster. The manual handling and copying of private keys significantly increase the possibility of a key compromise.

3.3 Lack of Crypto-Agility

There are several types of incidents that have required organizations to replace large numbers of TLS certificates and private keys, including the following incident types:

- **CA compromise:** If a CA is breached by an attacker, then the attacker can cause that CA to issue fraudulent certificates. After the CA breach is discovered and forensics are performed, it may be concluded that certificates issued by the CA cannot be trusted and that new certificates must be installed on all servers with certificates from the compromised CA
- **Vulnerable algorithm:** Cryptographic algorithms are constantly evaluated for vulnerabilities, by parties with both positive and negative intent. When an algorithm is found to be vulnerable (e.g., Secure Hash Algorithm 1 [SHA-1] for signature generation), TLS server certificates that are dependent on the algorithm must be replaced. Ongoing advancements in quantum computing require that organizations establish the ability to rapidly replace all existing certificates and keys and be prepared for implementation of post-quantum algorithms.
- **Cryptographic library bug:** Because cryptographic operations are quite complex, a few groups have specialized in developing cryptographic libraries that are used by TLS servers and other systems. If a bug is found with the key-generation functions of a cryptographic library, then all keys generated since the bug was introduced must be replaced. (Note: In 2008, a key-generation bug in the cryptographic libraries in Debian Linux was discovered. That bug was introduced in 2006. In 2017, a key-generation bug was discovered in the Infineon cryptographic libraries used in smart cards and trusted platform module chips.)

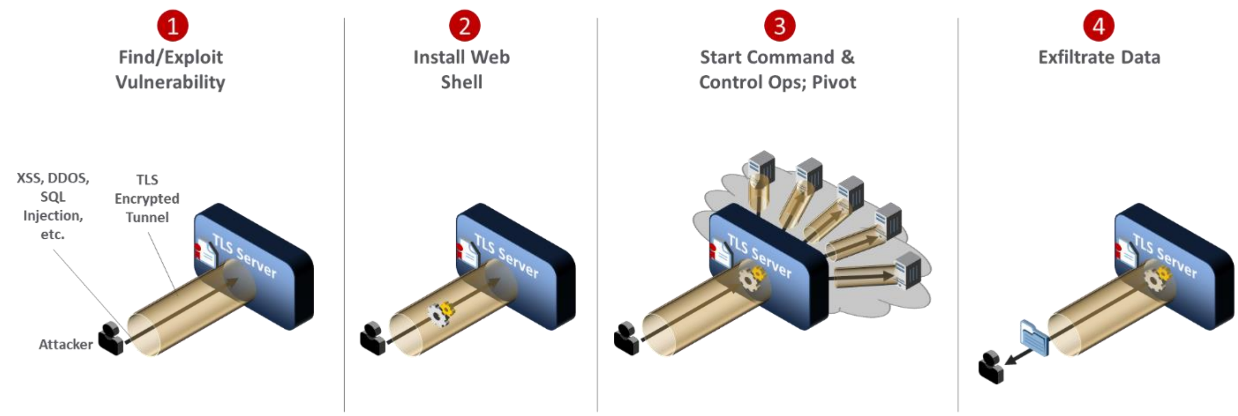
Most enterprises are not prepared to respond to the large-scale cryptographic failure that results from these types of incidents. Many organizations do not have comprehensive inventories of their TLS server certificates. In addition, they cannot contact the certificate owners because they do not have up-to-date information about the certificate owners responsible for each certificate. Finally, many organizations rely on manual processes to manage certificates and do not have processes for tracking the progress in replacing large numbers of certificates — leaving the organizations to guess how many systems have been updated. All of these factors can result in organizations requiring several weeks or months to replace all affected certificates, during which time business applications can be unavailable or risk security breaches.

3.4 Encrypted Threats

Many organizations are working to encrypt all communications by using TLS server certificates to prevent interception of plaintext credentials and eavesdropping on communications. While TLS server certificates enable confidentiality for legitimate communications, they can also allow attackers to hide their malicious activities within encrypted TLS connections. When a TLS server certificate is installed and enabled on a server, all users who connect (including attackers) can establish an encrypted connection to the server. An attacker who establishes an encrypted connection can then begin to probe the server for vulnerabilities within that encrypted connection.

The following steps, shown in Figure 3-1 and detailed below, describe how an attacker can leverage encrypted connections in his or her attacks.

Figure 3-1 How an Attacker Leverages Encrypted Connections to Hide Attacks



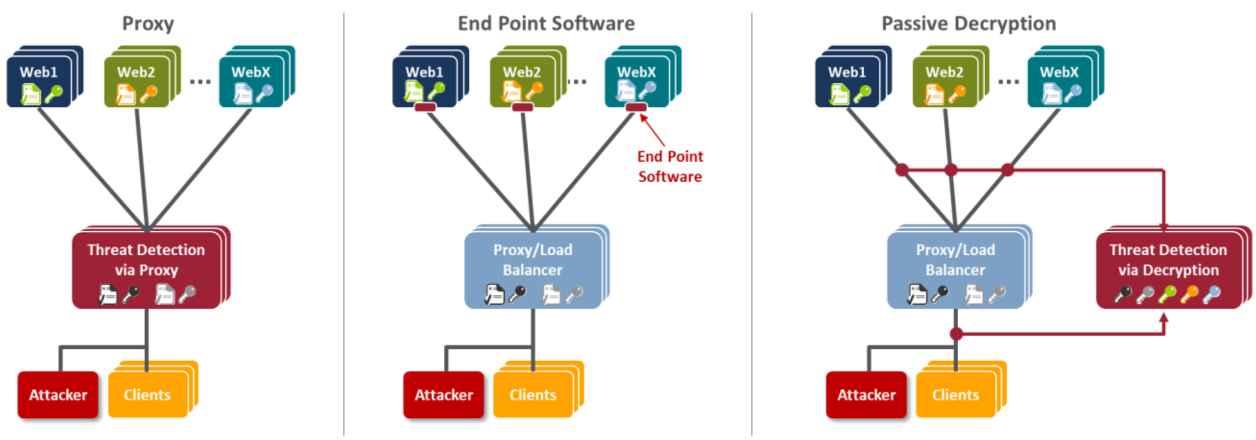
1. The attacker begins by connecting to a server and establishing an encrypted TLS session. Within that encrypted session, the attacker can probe for vulnerabilities that exist on the server and its software
2. If the attacker discovers a vulnerability and sufficiently elevates his or her privileges, then the attacker can load malware, generally called a “web shell,” onto the server

3. With this web shell loaded, the attacker can send commands over TLS connections (i.e., encrypted connections facilitated by the server's certificate). The attacker can then work to pivot to other systems by probing for vulnerabilities in servers accessible from the compromised system. The increased use of encryption enables an attacker who has compromised one system to pivot and attack other systems via encrypted connections, without detection. without being detected
4. Once the attacker has successfully reached data that he or she desires, the attacker is able to use the web shell to exfiltrate data. Because the attacker is establishing TLS connections by using the server's certificate to connect to the web shell, all of the exfiltrated data is encrypted while in transit

There are several methods for organizations to gain visibility into encrypted communications so that they can monitor and detect malicious activity. Some examples are listed below and are illustrated in Figure 3-2.

1. placing a threat detection system in front of servers that acts as a reverse proxy
2. installing end point software on each server to monitor communications
3. passively decrypting communications

Figure 3-2 Methods for Gaining Visibility into Encrypted Communications



The use of threat detection proxies is ideal at the perimeters of organizations for monitoring inbound internet communications for attacks. The threat detection proxy is connected in-line, requiring all inbound traffic to pass through it before moving on to the next device. The threat detection proxy terminates the TLS connection. It decrypts and examines incoming traffic. If the traffic is determined to be malicious, then the proxy drops it. Because the threat detection proxy is terminating all TLS connections, it must have a certificate for each server to which clients are attempting to connect. After the threat detection proxy decrypts and examines the traffic, it can establish a TLS session with the appropriate server behind it and send the traffic to that server in an encrypted TLS session.

While a threat detection proxy is ideal for use at the perimeter of an organization, many organizations also want to inspect their internal TLS traffic. Many enterprise applications include multiple tiers of servers and services (e.g., load balancers, web servers, application servers, databases, identity services) that communicate with each other internally via encrypted TLS sessions, making it impractical to place threat detection proxies between all systems on internal networks.

End point software can be installed on each server to monitor communications, alleviating the need to install proxies, but may impose additional processing requirements on servers that are already under a high load. In addition, because of the diversity of TLS server systems, it may be difficult to find an end point solution that operates on all platforms and provides comprehensive and consistent visibility and monitoring of all communications.

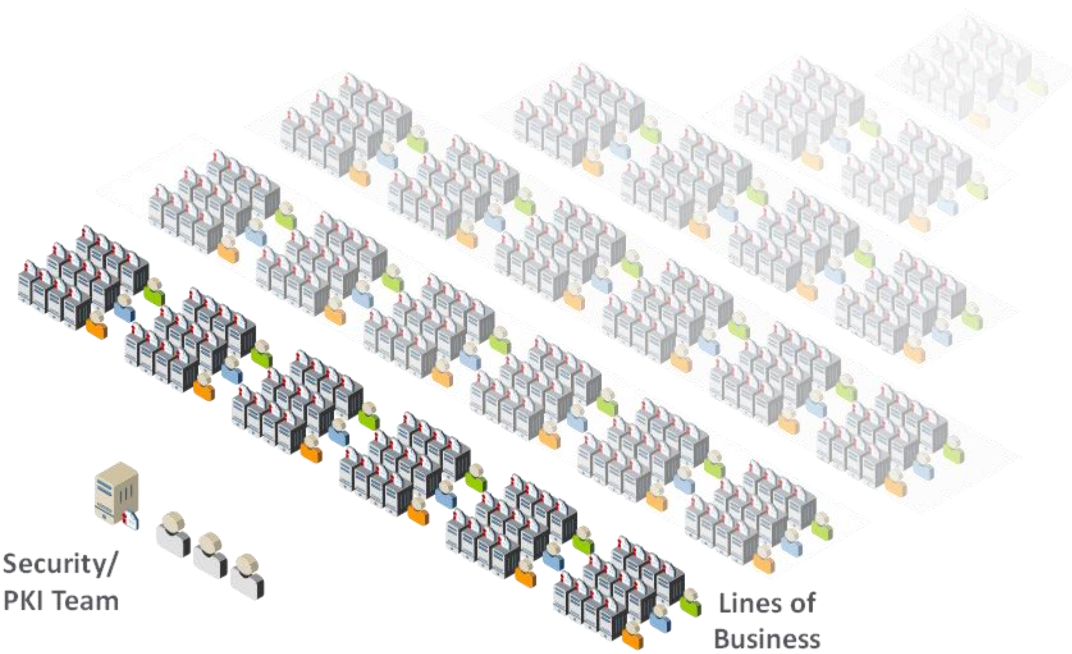
Passive, out-of-band decryption and threat analysis are performed by using devices that decrypt TLS-encrypted communications but that do not terminate TLS connections. The TLS connection is established between the client and the server. The passive decryption device listens to the TLS traffic without affecting it and decrypts it. Threat analysis is performed either by the passive decryption device or via other systems to which decrypted traffic is forwarded. Security-focused passive decryption devices can detect malicious traffic that has been sent on TLS connections, but these devices do not react in real time to block this traffic. Passive decryption does not require a change in network architecture or loading additional software on TLS servers. However, passive decryption poses a TLS server certificate management challenge because private keys must be copied to decryption devices from each TLS server whose communications will be monitored. The transfer of private keys must be done securely to avoid a key compromise and rapidly to avoid blind spots in monitoring for attacks. Automation can significantly aid in securely transferring private keys from TLS servers to the decryption device and keeping keys up-to-date when certificates are replaced.

4 Organizational Challenges

Despite the mission-critical nature of TLS server certificates, many organizations do not have clear policies, processes, and roles and responsibilities defined to ensure effective certificate management. Moreover, many organizations do not leverage available technology and automation to effectively manage the large and growing number of TLS server certificates. As a result, many organizations continue to experience significant incidents related to TLS server certificates.

As illustrated by Figure 4-1, the management of TLS server certificates is challenging due to the broad distribution of certificates across enterprise environments and groups, the complex processes needed to manage certificates, the multiple roles involved in certificate management and issuance, and the speed at which new TLS servers are being deployed. TLS server certificates are typically issued by a Certificate Services team (often called the public key infrastructure team). However, the certificates are commonly installed and managed by the certificate owners — the groups and the system administrators responsible for individual web servers, application servers, network appliances, and other devices for which certificates are used.

Figure 4-1 TLS Certificates Are Distributed Broadly Across Enterprise Environments and Groups



4.1 Certificate Owners

The term “certificate owner” is used to denote a group responsible for systems where certificates are deployed. Typically, there are several roles within a certificate owner group, including executives who have ultimate accountability for ensuring that certificate-related responsibilities are addressed, system administrators who are responsible for managing individual systems and the certificates on them, and application owners who can review and approve certificate requests from system administrators to ensure that only authorized certificates are issued. The certificate owners typically are not knowledgable about the risks associated with certificates or the best practices for effectively managing certificates.

With the advent of virtualization, the development and operations (DevOps) teams provision systems and software through programmatic means. This introduces a new type of certificate owner and new TLS server certificate challenges for organizations. As organizations push for more rapid and efficient deployment of business applications, many DevOps teams deploy certificates without coordination with the Certificate Services team. This can result in certificates for mission-critical applications not being tracked. This can be particularly problematic if bugs in DevOps programs/scripts cause certificates to be improperly deployed or updated. In addition, as DevOps teams adopt newer frameworks and tools, it is important to continue to monitor certificates and applications deployed and maintained by older DevOps frameworks and tools.

4.2 Certificate Services Team

The Certificate Services team is typically the group that has been given responsibility for managing relationships with public CAs and for the internal CAs. The Certificate Services team typically comprises one to three people. Though the team members have good knowledge and expertise about TLS server certificates, they do not have the resources or access required to directly manage certificates on the extensive number of systems where certificates are deployed. However, the Certificate Services team is often blamed when TLS certificate incidents, such as outages, occur.

5 Recommended Best Practices

To effectively address the risks and organizational challenges related to TLS server certificates and to ensure that they are a security asset instead of a liability, organizations must establish a formal TLS certificate management program with executive leadership, guidance, and support. The formal TLS certificate management program must include clearly defined policies, processes, and roles and responsibilities for the certificate owners and the Certificate Services team, as well as a central Certificate Service. The program should be driven by the Certificate Services team but must include active participation by the certificate owners — whether the certificate owners are responsible for traditional servers, appliances, virtual machines, cloud-based applications, DevOps, or other systems acting as TLS servers.

5.1 Establishing TLS Server Certificate Policies

As previously mentioned, most certificate owners are typically not knowledgable about the best practices for effectively managing TLS server certificates. Because certificate owners are responsible for the systems where certificates are deployed, it is imperative that they be provided with clear requirements and that those requirements be enforced as policies. This section provides recommended TLS server certificate policies. It also includes recommended responsibilities for the certificate owners and the Certificate Services team to successfully meet those requirements and policies.

5.1.1 Inventory

To address TLS server certificate risks, organizations must establish and maintain clear visibility across all TLS server certificates in their environment so that they can perform the following actions:

- detect potential vulnerabilities (e.g., the use of weak algorithms, such as SHA-1)
- identify certificates that are nearing expiration and replace them

- respond to large-scale cryptographic incidents, such a CA compromise, vulnerable algorithms, and cryptographic library bugs
- ensure compliance with regulatory guidelines and established organizational policy

This visibility is achieved by maintaining an inventory of all TLS server certificates. A single central inventory is recommended as it minimizes the possibility of overlooking critical TLS server certificates.

Recommended Requirement:

- An up-to-date inventory of all deployed certificates (end-entity certificates and CA certificate chain certificates) MUST be maintained. For each certificate, the inventory should include the following components:
    - Subject Distinguished Name (DN)
    - Subject Alternative Names (SANs)
    - issue date (i.e., notBefore date)
    - expiration date (i.e., notAfter date)
    - issuing Certificate Authority
    - key length
    - key algorithm (e.g., Rivest, Shamir, & Adleman [RSA]; Elliptic Curve Digital Signature Algorithm [ECDSA])
    - signing algorithm
    - validity period (i.e., from the notBefore date/time to the notAfter date/time)
    - installed location(s) of certificate (e.g., IP or DNS address and file path)
    - certificate owner (i.e., the group responsible for the certificate)
    - contacts (i.e., the group of individuals that should be notified of issues)
    - approver(s) (i.e., the parties responsible for reviewing issuance and renewal requests)
    - type of system (e.g., web, email, directory server, appliance, virtual machine, container)
    - business application (i.e., the application using the certificate)
    - applicable regulations (e.g., Payment Card Industry Data Security Standard [PCI-DSS], Health Insurance Portability and Accountability Act [HIPAA])
    - key-usage flags
    - extended key-usage flags

Recommended Responsibilities:

- Certificate Services team: provide a central system for certificate owners to establish and maintain their inventories
- certificate owners: establish and maintain an inventory of all certificates and keys on their systems

5.1.2 Ownership

To rapidly respond to issues with TLS server certificates, it is necessary to know who is responsible for each certificate. This information must be kept up-to-date as people are reassigned or terminated. Because reassignments can happen frequently, and because there may be a lag in updating ownership information, it is recommended that ownership be assigned to functional groups (e.g., an Active Directory [AD] group) that contain multiple individuals instead of assigning ownership to individuals. In cases where DevOps technologies are used to deploy TLS server certificates, the group responsible for the technology must be tracked, in addition to the application owner, so that they can be contacted when incidents arise.

Recommended Requirement:

- Contact information for certificate owners MUST be assigned to functional groups (e.g., AD groups) and must be updated within <30> business days of a role reassignment or termination. (Note: Here and elsewhere in this practice guide, when specific time frames, such as "<30> business days," are recommended, these values are often placed within brackets ("<>") to indicate that they are being provided only as suggestions. Each organization should determine the time frames to be instituted within its own enterprise, based on its needs. If it is possible for organizations to require compliance within shorter time frames, then that would be preferable.)

Recommended Responsibilities:

- Certificate Services team: provide a system to track ownership as part of the inventory
- certificate owners: keep ownership information up-to-date

5.1.3 Approved CAs

CAs are trusted issuers of certificates. If organizations do not control the CAs that are used to issue certificates in their environments, then they will face several potential risks:

- **Increased costs:** If multiple groups are individually purchasing certificates from CAs, then the cost per certificate can be significantly higher because organizations are not taking advantage of volume discounts
- **Trust issues:** Each CA that is used to issue TLS certificates to servers in an organization must be trusted by the clients connecting to those servers via a root certificate. If a large number of CAs (internal and external) is used, then the organization must take on the extra burden of

maintaining multiple trusted CA certificates on clients to avoid cases where the necessary CA is not trusted, which can result in outages

- **Security risk:** A certificate owner may decide to set up his or her own CA on a system that does not have the necessary security controls and to configure the system to trust that CA. This increases the possibility of an attacker impersonating a server if the attacker compromises that CA and issues fraudulent certificates
- **Unexpected CA incidents:** If one of the untracked CAs used in the organization's environment encounters an issue, such as a CA compromise or suddenly being untrusted by browser vendors, then the organization may have to scramble to respond to avoid security or operational issues for core applications

To ensure that they can rapidly respond to a CA compromise or another incident when using public CAs, organizations should maintain contractual relationships with more than one public CA. By doing this, organizations will not have to scramble to negotiate a contract (which may take days or weeks) while attempting to respond to an urgent situation. Organizations must also maintain at least one backup internal CA so that they can respond to an internal CA compromise or incident.

Recommended Requirements:

- Certificates must be issued only by the following CAs:
    - <External CA1>
    - <External CA2>
    - <Internal CA1>
    - <Internal CA2>
    - <…>
- Contractual relationships with at least two public CAs that conform to the CA/Browser Forum Baseline Requirements should be maintained at all times
- Internal CAs must be securely operated. Backup internal CAs must be maintained to support a rapid response to incidents, such as CA compromise

Recommended Responsibilities:

- Certificate Services team: manage business relationships with approved external CAs, and operate or outsource the operation of approved internal CAs
- certificate owners: ensure that only certificates from approved CAs are used

5.1.4 Validity Periods

The validity period for a certificate defines the time that it is valid, from the first date/time (notBefore) to the last date/time (notAfter) that it can be used. It is important to note that the validity period of a certificate is different than the cryptoperiod of the public key contained in the certificate and the corresponding private key. It is possible to renew a certificate with the same public and private keys (i.e., not rekeying during the renewal process). However, this is only recommended when the private key is contained with a hardware security module (HSM) validated to Federal Information Processing Standards (FIPS) Publication 140-2 Level 2 or above.

One of the greatest risks of private-key compromise is from administrators who have direct access to plaintext private keys (including the ability to make a copy) and who are then reassigned or terminated. Although certificates would ideally be changed (rekeyed) each time that an administrator with access to private keys is reassigned, this is often not practical. Therefore, ensuring that certificates and their corresponding private keys are changed regularly is important, as shorter validity periods reduce the time that a compromised private key can be used for malicious purposes. However, validity periods that are too short may increase the risk of outages. Organizations must determine the ideal validity period that balances security and operational risks for their organization. In general, due to the regular reassignment of administrative staff, it is recommended that validity periods be one year or less. The automated management of certificates can enable a more frequent renewal of certificates.

Recommended Requirement:

- The maximum validity period (i.e., from the notBefore date to the notAfter date for certificates must be <one year or less>

Recommended Responsibilities:

- Certificate Services team: ensure that CAs are available to certificate owners to issue certificates with approved validity periods
- certificate owners: ensure that certificates are renewed and replaced before their expiration

5.1.5 Key Length

Each certificate contains a public key that is mathematically matched to a private key (which should be kept secret). To prevent an attacker from guessing the value of the private key, it is necessary to randomly pick the value of the private key from a large set of possible values. For example, it is more difficult for someone to guess a number selected between zero and 1,000,000 than a number selected between zero and 100. The key length effectively defines the size of the range of numbers from which private and public key values are selected. A longer key length is considered more secure. However, longer key lengths require more processing power and time, as well as more storage. Consequently, a balance must be struck between security risk and resource requirements. The National Institute of Standards and Technology (NIST) monitors the industry to continually assess the potential crypto-analytical capabilities of potential attackers and their ability to guess the values of private keys, and sets recommended minimum key lengths. It is recommended that organizations require the use of keys with key lengths equal to or greater than the NIST recommendations.

Recommended Requirement:

- All certificates must use key lengths that comply with NIST Special Publication (SP) 800-131A, which are currently equal to or greater than the following key lengths:
    - RSA: <2,048>
    - ECDSA: <224>

Recommended Responsibilities:

- Certificate Services team: provide dashboards, reports, and alerts that enable the rapid detection of unauthorized key lengths, and provide automation technologies that enable rapid remediation
- certificate owners: use only TLS certificate public and private keys whose key lengths meet or exceed the organization’s key-length policy, monitor their inventory, and replace certificates that do not comply with the policy

5.1.6 Signing Algorithms

Certificates are digitally signed by CAs so that their authenticity can be verified. Signatures are generated by using digital signature algorithms (e.g., RSA, ECDSA) and hash algorithms (e.g., Secure Hash Algorithm 256 [SHA-256]). If certificates are signed by using a signing algorithm with an insufficient key length or by using vulnerable hash algorithms (e.g., SHA-1), then attackers can forge certificates and impersonate TLS servers. Consequently, organizations must ensure that all certificates are signed by using cryptographic algorithms that conform to approved standards.

Recommended Requirement:

- All certificates must be signed with an approved signature algorithm and key length and with an approved hash algorithm (e.g., SHA-256), as defined in NIST SP 800-131A and FIPS Publication 180-4

Recommended Responsibilities:

- Certificate Services team: ensure the availability of CAs that use approved signing algorithms, and provide reporting and alerting tools to enable the rapid identification of noncompliant certificates
- certificate owners: use only certificates signed with an approved signature algorithm and key length and with an approved hash algorithm, and identify and replace certificates signed with unapproved algorithms or key lengths

5.1.7 Subject DN and SAN Contents

Each certificate contains a unique identifier, called a subject DN, for the TLS server to which the certificate is issued. This identifier is in the form of an X.500 DN, which can include information such as the country, state, city/locality, organization, organizational unit (e.g., department), and a common name (CN). The CN contains the DNS address of the TLS server. For publicly trusted certificates, the contents of the Subject DN are governed by the public CA that issues them. For internal certificates, the contents of the Subject DN fields, such as the organizational unit, can help identify the group responsible for certificates when reporting centrally.

Public CAs will often perform checks to validate that an organization owns a top-level domain (e.g., www.company123.com) and will then allow the organization to request a certificate with Subject DNs and with SANs containing domains subordinate to that domain (e.g., www.company123.com, www.server1.company123.com). Consequently, it is critical that organizations implement approval processes that ensure that the Subject DNs and SANs in all certificate requests are thoroughly reviewed and vetted before they are sent to the CA.

Recommended Requirements:

- Names used in Subject DNs must conform to the following requirements:
    - The Organization (O) attribute in the Subject DN must be one of the following values:
        - <e.g., Company, Inc.>
    - The Organizational Unit attribute in the Subject DN must conform to the following categorization:
        - <specify whether department, location, or another categorization should be used>
    - The Locale (City), State (Province), and Country codes must be set to the following location:
        - <City, State, Country of organization identified in O = headquarters offices>
    - The CNs and the SANs may not include wildcards (e.g., *.company123.com).
- The CNs in all Subject DNs and SANs must be reviewed and approved by an individual who is knowledgable about the application or system for which the certificate is being requested and who can confirm that the requester is authorized to make the request.

Recommended Responsibilities:

- Certificate Services team: provide technology solutions to automatically detect and prevent Subject DN and SAN policy violations
- certificate owners: ensure that the Subject DNs and SANs in all certificates comply with policy

5.1.8 Certificate Request Reviews – Registration Authority (RA)

To prevent the issuance of rogue certificates that can be used maliciously to impersonate legitimate servers, all certificate requests must be vetted to ensure that they are issued only for valid systems and requested only by authorized parties. For certificates that are requested by individuals, it is important that the reviewer/approver has sufficient knowledge about the need for the certificate and about the personnel authorized to request certificates for the specific DNS address of the servers. It is generally impossible for a central team to be aware of all new applications and the people authorized to request certificates for those applications. Consequently, it is necessary to have certificate requests reviewed by local application owners who have this knowledge. For certificates that are requested by automated processes, such as DevOps frameworks, the necessary automated controls must be put in place to ensure that requesting applications are authenticated and that the DNS addresses for which they request certificates match specific patterns.

Recommended Requirements:

- All manual certificate requests for first issuance or renewal MUST be reviewed and approved by the business or application owner, who will confirm that the following statements are true:
    - A certificate is required for the application/system. The certificate CN (when included) and/or SANs of the certificate match the addresses of the application/system in question
    - The requester is authorized to make the request
- When certificates are being issued by automated processes, the automated process must be reviewed by the business or application owner prior to implementation, who will confirm that the following statements are true:
    - The automated process is capable of requesting certificates for specific CNs and/or SANs
    - There is consideration for the automation of the entire certificate life cycle, including renewal and revocation, built into the automated processes
    - A system for auditing and reviewing all certificates issued by the automated processes is in place

Recommended Responsibilities:

- Certificate Services team: provide a central system for assigning approvers, alerting approvers when certificate requests need approval, and enabling approvers to review and approve/reject requests
- certificate owners: assign review/approval responsibility to individuals who have knowledge of the systems (addresses) required for applications and of the individuals authorized to request certificates for those systems, and approve certificate requests in a timely manner

5.1.9 Private Key Security

Each TLS server certificate has a corresponding private key that must be kept secret. Often, the private keys used with TLS server certificates are stored in plaintext files, which may be accessible by administrators if not properly secured. Even when the files where private keys are stored are encrypted with passwords, the passwords are stored in plaintext configuration files so that TLS servers can gain access to the private keys when they are started. It is possible to protect TLS private keys in HSMs; however, due to the large number of TLS servers where private keys would be required, many organizations have not used HSMs to protect private keys. Organizations must assess the criticality and risk of each TLS server and determine the appropriate level of protection that is required for private keys. Further, organizations must ensure that only authorized personnel have access to private keys and that the authorized personnel are trained in the processes necessary to keep the private keys secure.

Recommended Requirements:

- Access to TLS server private keys stored in plaintext files MUST be limited to authorized personnel. For mission-critical systems, TLS private keys should be stored in an HSM
- Individuals granted access to private keys must complete training on procedures and practices for keeping private keys secure

Recommended Responsibilities:

- Certificate Services team: provide training on the proper procedures for keeping private keys secure, and provide automation to simplify the management of TLS private keys stored in HSMs
- certificate owners: ensure that only authorized personnel are granted access to private keys, regularly review who is granted access to private keys, and ensure that the authorized personnel receive training on the proper procedures for keeping private keys secure

5.1.10 Rekey/Rotation upon Reassignment/Terminations

Most private keys associated with TLS server certificates are stored in plaintext files. System administrators who manually manage TLS server certificates and associated private keys on their systems can make copies of the private-key files. Consequently, if a system administrator is reassigned or terminated, then the private key and certificate must be replaced (renewed) with a new key pair and certificate, and the previous certificate must be revoked, to prevent any malicious activities with the original private key and certificate. If automation is used for the management of certificates and private keys and if direct access by system administrators is limited (via limited-access controls and audit logging on any access), then certificate owners can avoid replacing certificates when a system administrator is reassigned or terminated.

Recommended Requirement:

- Private keys, and the associated certificates, that have the capability of being directly accessed by an administrator MUST be replaced within <30> days of reassignment or <5> days of termination of that administrator

Recommended Responsibilities:

- Certificate Services team: provide automated certificate and key management services that remove the need for administrators to manually access private keys, alleviating the need to rotate certificates and private keys when a system administrator is reassigned or terminated
- certificate owners: ensure that manually managed certificates and private keys are replaced when a system administrator with access is reassigned or terminated

5.1.11 Proactive Certificate Renewal

When a certificate is nearing expiration, it must be replaced. The replacement of certificates involves multiple steps, including reviewing and approving requests and testing the newly installed certificate(s) to ensure that the application they secure is operating properly after replacement. If an unexpected issue is encountered with the new certificate and the associated private key, the previous certificate and private key can be restored and used if the certificate has not yet expired. If certificate owners are not proactive and instead wait until the last minute before requesting, obtaining, and installing a new certificate, this procrastination can cause unplanned, urgent work by multiple teams (including the Certificate Services team) and risk unplanned downtime for the application. Certificate owners must plan, initiate, and complete the certificate renewal, installation, and testing process several weeks ahead of certificate expiration to ensure that unexpected issues and circumstances can be addressed and to avoid unnecessary “fire drills” for supporting teams (e.g., the Certificate Services team).

Recommended Requirement:

- Certificates MUST be renewed, installed, and tested at least <30> days prior to expiration of the currently installed certificate
    - If the validity period (total lifetime) of a certificate is shorter than <60> days (e.g., 20-day certificates used in short-lived/automated applications), then the certificate should be renewed before <80 percent> of the total validity period has elapsed

Recommended Responsibilities:

- Certificate Services team: provide automated services for monitoring certificate expiration dates, send reports to certificate owners showing certificates that are expiring in the next <60–90> days, send alerts and escalations to certificate owners for certificates expiring in <30> days or fewer, and send alerts to executives for certificates expiring in <30> days or fewer
- certificate owners: track upcoming expiration dates for their certificates, schedule replacement (in change windows where necessary), and ensure that certificate renewal and installation (of the new certificate) are completed prior to the minimum renewal windows

5.1.12 Crypto-Agility

There are several incidents that can require organizations to rapidly replace large numbers of certificates and private keys, including CA compromise or distrust, vulnerable algorithms, or bugs in cryptographic libraries. There have been multiple examples of these incidents in recent years, including the CA compromise of DigiNotar, the distrust of Symantec certificates by browser vendors, the deprecation of SHA-1 for signature generation, and cryptographic library bugs in Debian and Infineon. In 2006, NIST first recommended that organizations stop using SHA-1 for signatures. However, many organizations were still struggling to eradicate the use of certificates signed with SHA-1 in 2017, when their use was forcibly stopped by browser vendors.

An unexpected cryptographic incident can require an organization to rapidly respond to ensure that its operations and services to customers are not interrupted for an extended period. In addition, the industry is preparing for a transition to quantum-resistant algorithms, which will require organizations to replace large numbers of certificates and private keys.

Recommended Requirements:

- System owners MUST maintain the ability to replace all certificates on their systems within <2> days to respond to security incidents such as CA compromise, vulnerable algorithms, or cryptographic library bugs
- System owners MUST maintain the ability to track the replacement of certificates so that it is clear which systems are updated and which are not
- Select and establish contracts with backup CAs for public and internal certificates to enable rapid transition in response to a CA compromise

Recommended Responsibilities:

- Certificate Services team: document effective processes for replacing large numbers of certificates and private keys; train all certificate owners on certificate replacement processes; provide services, such as automation, that enable the rapid replacement of large numbers of certificates and private keys; actively track the occurrence of cryptographic incidents that require replacement of certificates and private keys, and communicate clearly to certificate owners when such an event occurs; and ensure that contracts with backup CAs for both public certificates and internal certificates (if applicable) are in place
- certificate owners: proactively support crypto-agility by maintaining an inventory of all certificates and owners, making sure that certificate replacement processes are as efficient as possible and that personnel are trained; and appropriately prioritize replacement of certificates and private keys when cryptographic incidents occur

5.1.13 Revocation

If the private key associated with a TLS server certificate is compromised, then the certificate can be revoked by the CA so that potential relying parties are alerted and do not trust the certificate. Certificate owners must understand their responsibility in revoking certificates and must proactively revoke certificates when an incident occurs. In addition, because certificates are ideally replaced several days or weeks before they expire, it is important to revoke the replaced certificate once it has been confirmed that the new certificate and private key are operating properly. This will prevent the old certificate and private key (which are still valid until they expire or are revoked) from being used for malicious purposes. In addition, an inadvertent or malicious revocation of a certificate can cause downtime for the application that it secures; therefore, organizations must ensure that they have processes to prevent unauthorized revocation.

Recommended Requirements:

- TLS server certificates must be revoked if the associated private key has been or is suspected of being compromised
- When a certificate is renewed, the old certificate must be revoked within <5> days after the new certificate has been installed, tested, and set into operation
- Revocation of a TLS server certificate outside the renewal/replacement process can be initiated only by a certificate owner or identified security personnel and should be approved by the Certificate Services team or a designated security approver

Recommended Responsibilities:

- Certificate Services team: provide the infrastructure and services to ensure that certificates can be rapidly and securely revoked when necessary and to ensure that certificates cannot be revoked without proper approval
- certificate owners: request revocation of old certificates that have been replaced but that are still valid, and request revocation of certificates when a private key is compromised or suspected to be compromised

5.1.14 Continuous Monitoring

Because of the broad use of TLS server certificates in all critical communications, operational or security failures related to TLS server certificates can significantly impact the business operations of organizations. TLS certificates must be continuously monitored to prevent outages and security vulnerabilities. The certificates should be monitored for impending expiration; for situations in which they are not operating, are not configured properly, or are vulnerable; and for situations in which they are not consistent with policy.

Recommended Requirements:

- The expiration dates of certificates must be continuously monitored. Notifications must be automatically sent to certificate contacts <90, 60, and 30> days prior to expiration. If a certificate is not successfully renewed and replaced <30> days prior to expiration, then escalation notifications must be sent to the certificate owner management and incident response teams
- The operation and configuration of certificates must be periodically checked to identify any issues or vulnerabilities
- Certificates must be periodically checked to ensure that they are consistent with policy

Recommended Responsibilities:

- Certificate Services team: provide systems and services for continuously monitoring TLS server certificates, and support certificate owners in implementing TLS server certificate continuous monitoring and in keeping it operational
- certificate owners: ensure that continuous monitoring processes are in place and operational for all of their TLS server certificates

5.1.15 Logging TLS Server Certificate Management Operations

TLS server certificates serve as trusted credentials that authenticate servers for mission-critical applications. Just as logging data access is required for forensics and other purposes, logging all certificate and private-key management operations is critical. Organizations must ensure that they have a complete chain of custody for private keys and certificates that includes a log of all operations, including key-pair generation, certificate requests, request approval, certificate and key installation, the copying of certificates and keys (e.g., for load-balanced applications), certificate and key replacement, and certificate revocation. Logs must be collected and stored in a central location so that the complete chain of events for certificates and private keys can be reviewed when necessary.

Recommended Requirement:

- A complete automated log MUST be maintained of all TLS certificate and private-key management operations (from creation to installation to revocation) that includes a description of the operation performed, any relevant metadata about the event (e.g., the location of files), the identity of the person/application performing the operation, and the date/time that it was performed

Recommended Responsibilities:

- Certificate Services team: provide a system for collecting all logged events, and provide tools that automatically log certificate and private-key management operations

- certificate owners: ensure that all tools used for certificate and private-key management operations log events in a central log

5.1.16 TLS Traffic Monitoring

While providing authentication and confidentiality for legitimate communications and operations, TLS can also be used by attackers to hide their operations, such as scanning for vulnerabilities, leveraging vulnerabilities for privilege escalation, denial-of-service operations, and data exfiltration. In addition to monitoring the content of TLS communications for external-facing systems, organizations must monitor TLS communications for internal systems to help detect attackers who are attempting to pivot between internal systems (to gain access to critical data) or are exfiltrating compromised data. This monitoring may be accomplished in a variety of ways, including via proxy, end point software, or passive decryption.

Recommended Requirement:

- Communications passed through TLS will be monitored for unauthorized operations and data exfiltration via proxy, end point software, passive decryption, or another method

Recommended Responsibilities:

- Certificate Services team: provide a secure method for transporting TLS private keys between TLS servers and passive decryption devices when passive decryption is used for TLS traffic monitoring
- certificate owners: ensure that all communications protected by TLS are monitored for unauthorized operations and data exfiltration

5.1.17 Certificate Authority Authorization

An attacker can impersonate a server if the attacker is able to get a certificate issued that includes the name of the server and his or her own public key. To mitigate this type of attack, organizations can populate Certificate Authority Authorization (CAA) records for the DNS domains of their servers, with the names of one or more CAs that are authorized to issue certificates for that server. When a CA receives a certificate request for a domain, it must check the domain in the DNS to see if a CAA record is defined. If a CAA record is defined, then, before issuing a certificate, the CA must ensure that the CA's name is listed in a CAA record for the domain. CAA records can be specified for second-level domains (e.g., www.organization1.com), which will apply to all subordinate domains and to individual domains (e.g., www.alpha.organization1.com). Because an attacker can attempt to request a certificate for a domain from one of the CAs listed in the CAA record, the organization should ensure that the listed CAs accept certificate requests only from parties authorized by the organization.

Recommended Requirement:

- CAA records MUST be populated with authorized CAs for all domains for which public certificates may be issued

Recommended Responsibilities:

- Certificate Services team: ensure that CAA records are defined with approved CAs for all second-level domains owned by an organization
- certificate owners: ensure that the Certificate Services team is aware of all second-level domains for which the certificate owner is requesting certificates

5.1.18 Certificate Transparency

Certificate Transparency (CT) provides a publicly searchable log of issued certificates. CT is primarily focused on certificates issued by public CAs. Some browsers require that certificates issued by public CAs be published to a publicly available CT log; otherwise, the browser will display a warning to the user. The availability of CT logs enables organizations to confirm that unauthorized certificates have not been issued for their domains.

Recommended Requirement:

- CT logs MUST be regularly monitored to ensure that unauthorized certificates have not been issued for any domains owned by the organization

Recommended Responsibility:

- Certificate Services team: establish an automated process for monitoring CT logs

5.1.19 CA Trust by Relying Parties

Clients that connect to TLS servers verify the validity of those servers' certificates by using CA certificates or root certificates that they store locally in their systems. Many operating systems and applications (e.g., browsers) are preloaded with certificates from public CAs that have met the requirements of standards organizations, such as the CA/Browser Forum. Some applications, such as browsers, may include more than 100 trusted CAs. To reduce their exposure to CA compromise incidents, organizations should minimize the CAs that their clients trust to only those they are likely to need to trust. For example, if certain systems acting as TLS clients are used only for internal operations, then they should trust only the certificate(s) from the internal CA(s). Furthermore, if certain TLS clients communicate with TLS servers from select partners, then certificates from only the CAs expected to be used by those partners should be trusted. Organizations must maintain an inventory of CA certificates trusted on all of their systems, ensure that only needed CAs are trusted, and maintain the ability to rapidly remove or replace CA certificates that should no longer be trusted.

Recommended Requirement:

- CA certificates trusted by TLS clients MUST be limited to only those required to validate TLS certificates of the servers with which the client communicates. All unneeded CA certificates MUST be removed. The following CAs should never be trusted:

- <e.g., DigiNotar>
- <…>

Recommended Responsibilities:

- Certificate Services team: provide the technology and services for discovering and creating inventories of existing CA certificates and for managing (e.g., adding, removing) CA certificates
- certificate owners: limit CA trust to the minimum needed for each system, and ensure that all other CAs are removed

5.2 Establish a Certificate Service

Manually managing TLS server certificates is infeasible due to the large number of certificates in most enterprises. It is also not feasible for each certificate owner to create their own certificate management system. The most efficient and effective approach is for the Certificate Services team to provide a central Certificate Service that includes technology-based solutions that provide automation and that support certificate owners in effectively managing their certificates. This service should include the technology/services for CAs, certificate discovery, inventory management, reporting, monitoring, enrollment, installation, renewal, revocation, and other certificate management operations.

The central Certificate Service must also provide self-service access for certificate owners so that they are able to configure and operate the services for their areas without requiring significant interaction with the Certificate Services team. Furthermore, the central Certificate Service must be able to integrate with other enterprise systems, including identity and access management systems, ticketing systems, configuration management databases, email, workflow, and logging and auditing.

5.2.1 CAs

Approved CAs must be designated and made available to certificate owners for requesting public and internal certificates. If, as is common, different CAs will be used for issuing public and internal certificates, then instructions should be provided to certificate owners to help them select the correct CA based on the purpose of the server where the certificate will be used. Establish backup CAs for both public and internal certificates, including completing contracts with backup public CAs so that an immediate cutover is possible in case of a CA compromise, for business reasons or because of some other motivation.

5.2.2 Inventory

An up-to-date inventory of deployed TLS server certificates is the foundation of an effective certificate management program. The functionality required by an inventory system generally makes it infeasible for certificate owners to operate and manage their own inventory systems. It is imperative that the Certificate Services team provides a central system that certificate owners can use to maintain an inventory of their certificates. Without a central, up-to-date inventory, the Certificate Services team has no way of proactively monitoring for security and operational risks or supporting certificate owners in minimizing risks.

The central inventory system should provide the following characteristics and functions:

- **Automatic parsing:** Certificates contain multiple fields of information (e.g., subject, issuer, expiration date) that must be monitored. The inventory system should provide automatic parsing of the contents of certificates that are loaded into it so that searches can be performed on individual fields
- **Additional metadata:** It must be possible to associate additional information/metadata with each certificate (e.g., identifiers of the owners and approvers; installed locations; application identifiers; cost center numbers)
- **Organization:** With hundreds or thousands of certificates spread across many certificate owners and geographic locations, the inventory system should support organizing certificates into distinct groups/folders
- **Access controls:** To prevent unauthorized actions, it should be possible to define and enforce access controls that are assigned to groups or individuals
- **Support certificate management:** As the foundation of a certificate management program, the inventory system must integrate with and support all other certificate management functions (e.g., discovery, enrollment portal, approvals, automation)

5.2.3 Discovery and Import

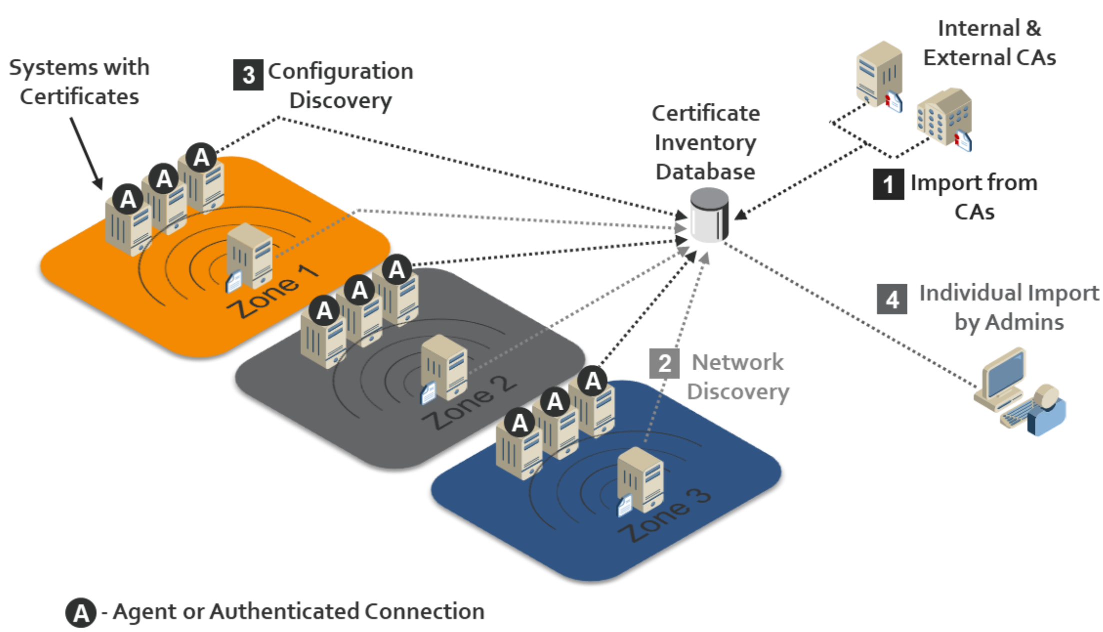
Manually establishing and maintaining an up-to-date and comprehensive inventory is difficult, if not impossible. Because of the complexity of most enterprise environments — which contain firewalls, different security/operations restrictions, etc. — it is often not sufficient to have a single method of automatically populating and maintaining an inventory. The central Certificate Service must provide multiple options for automated discovery and the import of certificates, including the options listed below:

- **CA import:** automated import of certificates from CAs. This is often the fastest way to initially populate the certificate inventory. However, it will only provide an inventory of certificates from known CAs
- **Network discovery:** automated scanning of one or more configurable sets of IP addresses, IP address ranges, and ports for TLS server certificates. This helps provide a comprehensive view of all certificates and their locations. Organizations typically find certificates from unapproved CAs and self-signed certificates (which should likely be replaced with certificates from approved CAs). The network discovery service must also support operation across multiple network zones separated by firewalls

- **Configuration discovery:** Network discovery can find certificates and determine their network location(s); however, it does not allow for collection of configuration information, such as the type of keystore (e.g., Privacy Enhanced Mail, Public Key Cryptography Standards [PKCS] #12, HSM), the storage location on the server, and other information that can be helpful in detecting issues and in setting up automated management for the certificate. The inventory system should provide a means of discovering certificate configuration information via an authenticated connection or agent
- **Bulk import:** In addition to network discovery and CA import, it is beneficial to have the option for administrators to import certificate data. This helps in cases where network discovery and CA import are not possible and in cases where there is additional information/metadata (e.g., contacts, approvers, cost centers) that can be associated with each certificate to help in tracking and management.

Figure 5-1 depicts options for automated discovery and import of certificates.

Figure 5-1 Various Options for Automated Discovery and the Import of Certificates



5.2.4 Management Interfaces

Certificate owners and the Certificate Services team must provide user interfaces to view and manage certificates. The interfaces should be simple enough to support certificate owners who have small numbers of certificates and perform management operations infrequently. The interfaces should also offer more-sophisticated functionality to support the needs of certificate owners with large numbers of certificates and the needs of the Certificate Services team.

The interfaces should provide the following characteristics and functions:

- **Inventory view:** Certificate owners should be able to view their certificates (to which they have been granted access). The Certificate Services team must be able to view the entire inventory
- **Searching and filtering:** Certificate owners with large numbers of certificates, and the Certificate Services team, should be able to search and filter operations so that they can quickly find specific certificates
- **Enrollment and renewal:** The portal should provide a simple method to request new certificates and to renew existing certificates. Having a single interface for enrollment and renewal across all CAs reduces the retraining needed when moving CAs, resulting in better crypto-agility
- **Approvals:** If an external system is not used for reviewing certificate requests, then the portal must provide a method for an approver to perform RA functions to review the relevant details of certificate requests and to approve/reject the requests with comments

5.2.5 Automated Enrollment and Installation

Manually requesting, installing, and managing large numbers of certificates is error-prone and resource-intensive; increases security risk; and does not allow for a rapid response to large-scale incidents, such as CA compromises. In cloud environments, the ability to quickly spin up new instances to support increased loads is critical. Because most enterprises have a range of systems from different vendors with diverse management methods, the central Certificate Service should offer multiple options for automation, including the options listed below:

- **Programmatic automation:** The central Certificate Service should provide a set of application programming interfaces (APIs) (e.g., Representational State Transfer) that enable enrollment, revocation, reporting, etc. The central Certificate Service should support easy integration with and access from DevOps frameworks and other programming tools
- **Standard protocol support:** The central Certificate Service should support standard protocols for requesting certificates, including the Simple Certificate Enrollment Protocol (SCEP), Automated Certificate Management Environment, and Enrollment over Secure Transport
- **Proprietary automation:** Some systems may not support programmatic or standards-based enrollment and installation but may provide other methods (e.g., APIs, command-line utilities) that can be used to automate certificate enrollment and installation. This may be performed with an agent or via a remote authenticated connection
- **Secure key transport:** To enable detection of encrypted threats by using passive decryption devices, the central Certificate Service must provide the ability to securely transport TLS private keys from TLS servers to the decryption devices that enable inspection of encryption communications

Automation must support integration with HSMs when HSMs are used for protection of private keys.

5.2.6 RA/Approvals

Certificate requests must be reviewed and vetted to ensure that unauthorized certificates are not issued or used for malicious purposes. Large enterprises generally have hundreds of different departments, business applications, projects, and systems administrators, making it infeasible for a central group to have the relevant knowledge needed to vet requests. The central Certificate Service should provide the ability to assign individuals (e.g., application owners) to review certificate requests for their respective areas. Once approvers are assigned, the central Certificate Service should automatically route certificate requests to assigned reviewers for approval and enable them to review any relevant data needed to properly vet requests.

5.2.7 Reporting and Analytics

To address TLS server certificate-related risks, certificate owners and the Certificate Services team must have visibility across their inventory and be able to quickly identify TLS server certificate issues or vulnerabilities. The most efficient method of addressing risks is proactive notifications sent by the central Certificate Service, based on configured rules. However, reports and dashboards can help in planning (e.g., an unexpectedly large number of certificate expirations coming in the next few weeks) and identifying anomalies that would otherwise not be caught by the automated rules. The central Certificate Service should support the following reporting and analysis tools:

- **Custom reporting:** Users should be able to create customized reports, including the data to be presented, the filtering criteria for the results, the scheduling of execution, and the selection of report recipients
- **Dashboards:** To help in identifying anomalies or unexpected issues, dashboards should proactively highlight risks, such as certificates with weak keys, vulnerable algorithms, impending expirations, operational errors, and other issues
- **Interfaces to monitoring systems:** Many organizations rely upon automated security incident and event monitoring systems that collect, analyze, and correlate information that is subsequently displayed or used to notify humans of events and the actions required. Anomalies and issues must be delivered to such systems

5.2.8 Passive Decryption Support

If passive decryption devices are used to monitor TLS-encrypted communications for attacks, then those devices must have copies of the private keys from all monitored TLS servers so that the devices are able to decrypt TLS traffic to those servers. Manually transporting private keys from TLS servers to passive decryption devices creates risk of a compromise. Consequently, when passive decryption is used, the central Certificate Service must provide an automated and secure method for transporting private keys from TLS servers to passive decryption devices and for keeping the private keys up-to-date when new keys (and certificates) are deployed.

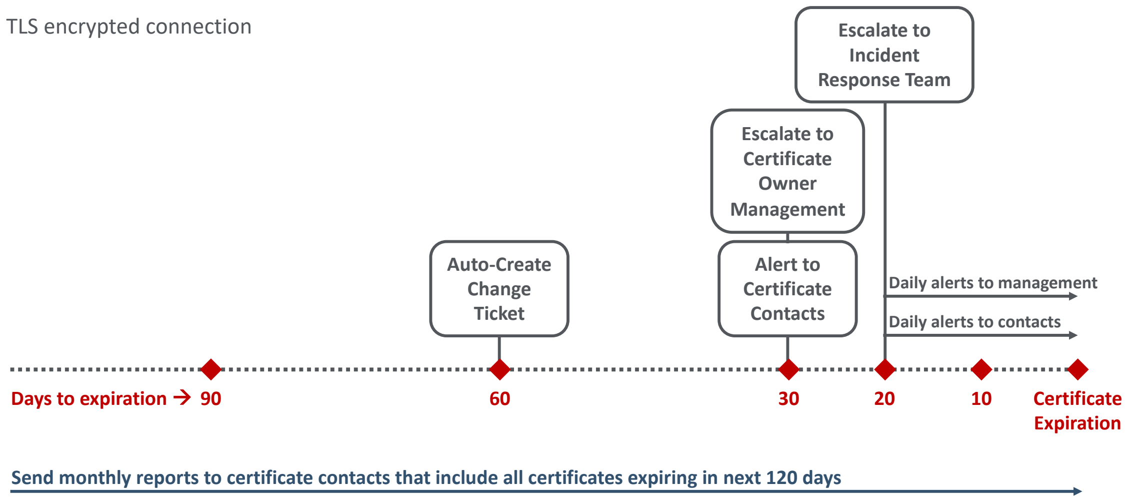
5.2.9 Continuous Monitoring

To prevent operational or security incidents, the certificates must be continuously monitored across the enterprise. Continuous monitoring should include the following types of monitoring:

- **Expiration monitoring:** To prevent outages due to expired certificates, the expiration dates for all certificates should be monitored. It should be possible to configure the time periods when notifications will be sent to certificate contacts prior to expiration (e.g., 90 days, 60 days, 30 days). If timely action is not taken, then it should be possible to escalate and send notifications to managers or a central incident response team
- **Operation/configuration monitoring:** Once a known good state is established (e.g., the location and configuration of certificates), the central Certificate Service should monitor and detect situations in which certificates are not operating, are not configured properly, or are vulnerable
- **Policy compliance:** The central Certificate Service should detect and send alerts when deployed certificates are not consistent with policy

Because certificate expirations are a regular occurrence, especially for certificate owners with large numbers of certificates, it is important to not inundate certificate owners with notifications, as they will likely start to ignore them. An effective strategy is to combine the use of reports, change tickets, and alerts. Sending regular (e.g., monthly) reports containing a list of certificates expiring within a certain number of days (e.g., 120 days) helps certificate owners plan for expirations. Automatically creating change tickets in the organization’s central ticketing system can ensure that certificate renewals and replacements are handled in the same way that other change operations are performed. Sending alerts within 30 days of expiration and escalating to management and incident response teams ensures that certificates that are not replaced in a timely fashion are identified before they expire. Figure 5-2 provides an example schedule for reports, tickets, and alerts.

Figure 5-2 Example Timeline of Processes and Notifications Triggered by Impending Certificate Expiration



5.2.10 Education

Management of TLS server certificates in an enterprise environment is complex, time-consuming, error-prone, and security-sensitive. Most certificate owners are not knowledgable about TLS server certificates, the processes for effectively managing certificates, or their own certificate-related responsibilities. Consequently, the Certificate Services team must provide readily accessible educational materials, preferably online and available on demand. The TLS server certificate educational materials should include the following items:

- a basic introduction to certificates and keys (e.g., when certificates are used, obtaining certificates, protecting keys, certificate changes, revocation)
- risks of improper TLS server certificate management
- an explanation of TLS server certificate policies and certificate owner responsibilities
- step-by-step instructions for managing TLS server certificates, including any of the following steps that are offered via the central Certificate Service:
    - creating an inventory
    - reviewing the inventory and identifying risks/vulnerabilities (e.g., generating reports)

- manually requesting and installing TLS server certificates on each relevant operating system/application (e.g., Apache)
- DevOps/API-based request and installation
- agentless automated installation
- agent-based automated installation
- renewing certificates
- revoking certificates

There are many educational resources available on the internet that can alleviate the need to create new materials. An internal TLS server certificate education website can include links to helpful web pages and websites.

5.2.11 Help Desk

In addition to educational materials, certificate owners must have a central support service that they can contact about questions and that can assist in troubleshooting issues. Many certificate owners may be new to TLS server certificate management or may be responsible for only a small number of certificates (e.g., one to five certificates) and will likely need assistance in successfully performing necessary operations. Any certificate owner calling the help desk should be required to have completed the educational programs that apply to their use cases so that help-desk personnel do not need to explain basic concepts that can be learned prior to the request for help.

TLS server certificates are typically installed or renewed during scheduled maintenance windows, which are often scheduled on weekends and/or in the middle of the night. Issues related to TLS server certificates can often arise during these scheduled maintenance operations; therefore, help-desk personnel should be made available during all times when certificate issues may arise (e.g., 24 hours a day, seven days a week). Help-desk personnel should be knowledgable about and experienced in TLS server certificate management. It is possible to have general help-desk personnel answer and address Level One certificate calls and escalate to more-experienced personnel as needed for Level Two and Level Three calls.

5.3 Terms of Service

It is helpful to define the terms of service for the central Certificate Service to avoid confusion by certificate owners about the services that they will receive and their responsibilities. The terms of service should include those listed below:

- a description of the services provided (e.g., network discovery, monitoring enrollment, automation)

- responsibilities of the certificate owners and the Certificate Services team (e.g., the Certificate Services team will help with network discovery, but a certificate owner is responsible for working with the network team to allow the discovery on their systems)
- expected service levels — stated in service level agreements — with response times

5.4 Auditing

Due to the fundamental role that TLS server certificates play in securing data and systems, periodic reviews of TLS server certificate management practices are essential. Auditors must confirm that TLS server certificate policy requirements are addressed. For example, all certificate owners must be able to demonstrate that they have a certificate inventory and to describe the steps that they have taken to ensure that all certificates are included in the inventory. The Certificate Services team must demonstrate that it is providing the services needed for certificate owners to comply with policy.

TLS server certificate risks can lie latent for long periods of time and then can unexpectedly have significant impact to an organization's operations —due to either operational outages or security issues. Consequently, regular audits of certificate management practices performed by compliance auditors are critical to prevent unanticipated issues.

6 Implementing a Successful Program

The broad distribution of TLS server certificates across distinct groups, networks, and systems can present unique challenges in implementing an effective certificate management program across an enterprise environment. The following resources are helpful for successful implementation:

- **Executive owner:** It is essential to have an executive owner for the certificate management program. This executive owner must be prepared to educate the executives of each group of certificate owners on TLS server certificate risks and the executives' responsibilities
- **Prioritization of risks:** Each organization has different challenges and priorities related to TLS server certificates. Although the best practices detailed in this practice guide are intended to help address all of the risks related to TLS server certificates, it is helpful to prioritize those risks based on historical certificate issues and business needs. This prioritization can help in communications with certificate owners and with setting objectives and prioritizing tasks
- **Objectives:** Establishing clear and achievable objectives provides targets, helps focus efforts, and improves the likelihood of successful implementation. For example, if an organization finds that it does not have an inventory and recognizes that there are two groups that may be difficult to inventory in the near term, then one objective may be to create an inventory of all other groups' TLS server certificates in the next 12 months
- **Action plan:** An action plan with specific tasks, responsibilities, and milestones, geared to achieve the objectives, should be created, communicated, and reviewed by all stakeholders (e.g., certificate owners, Certificate Services team, executive owner). The action plan should be

prioritized to address the most important objectives first. For example, an action plan might include the following objectives:

- 30 days from the start of the project:
  - complete certificate imports from CA1, CA2, and CA3
  - require certificate enrollment through the central Certificate Service portal, and prevent enrollment directly to CAs
- 90 days from the start of the project:
  - complete network discovery across all North American and European data centers
  - complete the assignment of certificate owners for all certificates in inventory
- 180 days from the start of the project:
  - automate certificate enrollment and installation on all load balancers
  - automate certificate enrollment and installation for all e-commerce web servers
  - complete network discovery across all Asia-Pacific data centers

- **Regular executive reviews:** The objectives and action plan should be reviewed with the executive owner at commencement of the project, and regular reviews should be scheduled (e.g., every 90 days) to track progress. During these reviews, the executive owner should note areas where additional action by certificate owners is needed so that the executive owner can proactively communicate with peer executives to ensure that action is taken
- **Periodic audits:** Due to the critical role that TLS server certificates play in the security and operations of organizations, and the risks resulting from improper management, regular audits should confirm that the Certificate Services team and certificate owners are fulfilling their responsibilities in TLS server certificate management.

Appendix A List of Acronyms and Abbreviations

AD	Active Directory
API	Application Programming Interface
BGP	Border Gateway Protocol
CA	Certificate Authority
CAA	Certificate Authority Authorization
CN	Common Name
CSR	Certificate Signing Request
CT	Certificate Transparency
DevOps	Development and Operations
DN	Distinguished Name
DNS	Domain Name System
DV	Domain Validated
ECDSA	Elliptic Curve Digital Signature Algorithm
EV	Extended Validation
FIPS	Federal Information Processing Standards
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
PKCS	Public Key Cryptography Standards
RA	Registration Authority
RSA	Rivest, Shamir, & Adleman (public key encryption technology)

SAN	Subject Alternative Name
SCEP	Simple Certificate Enrollment Protocol
SHA-1	Secure Hash Algorithm 1
SHA-256	Secure Hash Algorithm 256
SP	Special Publication
TLS	Transport Layer Security

Appendix B References

T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) protocol version 1.2,” RFC 5246, Aug. 2008. Available: https://tools.ietf.org/html/rfc5246.

E. Rescorla, “HTTP over TLS,” RFC 2818, May 2000. Available: https://tools.ietf.org/html/rfc2818.

J. Sermersheim, “Lightweight Directory Access Protocol (LDAP): The protocol,” RFC 4511, June 2006. Available: https://www.ietf.org/rfc/rfc4511.txt.

J. Klensin, “Simple Mail Transfer Protocol,” RFC 5321, Oct. 2008. Available: https://tools.ietf.org/html/rfc5321.

J. Myers and M. Rose, “Post Office Protocol – Version 3,” RFC 1725, Nov. 1994. Available: https://tools.ietf.org/html/rfc1725.

M. Crispin, “Internet Message Access Protocol – Version 4rev1,” RFC 3501, Mar. 2003. Available: https://tools.ietf.org/html/rfc3501.

V. Rekhter et al., “A Border Gateway Protocol 4 (BGP-4),” RFC 4271, Jan. 2006. Available: https://tools.ietf.org/html/rfc4271.

P. Mockapetris, “Domain Names – Concepts and Facilities,” RFC 1034, Nov. 1987. Available: https://tools.ietf.org/html/rfc1034.

D. Cooper et al., “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 5280, May 2008. Available: https://tools.ietf.org/html/rfc5280.

E. Barker and A. Roginsky, “Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths,” National Institute of Standards and Technology (NIST) Special Publication (SP) 800-131A Revision 1, Gaithersburg, MD, Nov. 2015. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf.

Information Technology Laboratory, “Secure Hash Standard (SHS),” NIST, Federal Information Processing Standards PUB 180-4, Gaithersburg, MD, Aug. 2015. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

T. Pornin, “Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA),” RFC 6979, Aug. 2013. Available: https://tools.ietf.org/html/rfc6979.

K. Moriarty et al., “PKCS #12: Personal Information Exchange Syntax v1.1,” RFC 7292, July 2014. Available: https://tools.ietf.org/html/rfc7292.

M. Pritikin et al., “Simple Certificate Enrollment Protocol draft-nourse-scep-23,” Internet Draft, Sept. 7, 2011. Available: https://tools.ietf.org/html/draft-nourse-scep-23.

T. Polk et al., "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations," NIST SP 800-52 Revision 1, Gaithersburg, MD, Apr. 2014. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf.

E. Barker, "Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms," NIST SP 800-175B, Gaithersburg, MD, Aug. 2016. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-175B.pdf.