NIST SPECIAL PUBLICATION 1800-16C

Securing Web Transactions

TLS Server Certificate Management

Volume C:
Approach, Architecture, and Security Characteristics

Murugiah Souppaya
NIST

Mehwish Akram
Brian Johnson
Brett Pleasant
Susan Symington
The MITRE Corporation

William C. Barker
Strativia

Paul Turner
Venafi

Clint Wilson
DigiCert

Dung Lam
F5

Alexandros Kapasouris
Symantec

Rob Clatterbuck
Jane Gilbert
Thales Trusted Cyber Technologies

June 2020

This publication is available free of charge from:
http://doi.org/10.6028/NIST.SP.1800-16

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

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

NCCoE
NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified by name or company logo or other insignia in order to acknowledge their participation in this collaboration or to describe an experimental procedure or concept adequately. Such identification is not intended to imply special status or relationship with NIST or recommendation or endorsement by NIST or NCCoE; neither is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1800-16C Natl. Inst. Stand. Technol. Spec. Publ. 1800-16C, 64 pages, (June 2020), CODEN: NSPUE2

FEEDBACK

As a private-public partnership, we are always seeking feedback on our practice guides. We are particularly interested in seeing how businesses apply NCCoE reference designs in the real world. If you have implemented the reference design, or have questions about applying it in your environment, please email us at tls-cert-mgmt-nccoe@nist.gov.

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

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 information technology 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 [4] and details the steps needed for another entity to re-create the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Maryland.

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 1800 series) 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.

ABSTRACT

Transport Layer Security (TLS) [2] [11] server certificates are critical to the security of both internet-facing and private web services. A large- or medium-scale enterprise may have thousands or even tens of thousands of such certificates, each identifying a specific server in its environment. Despite the critical importance of these certificates, many organizations lack a formal TLS certificate management program, and the ability to centrally monitor and manage their certificates. Instead, certificate management tends to be spread across each of the different groups responsible for the various servers and systems in an organization. Central security teams struggle to ensure certificates are being properly managed by each of these disparate groups. Where there is no central certificate management service, the organization is at risk, because once certificates are deployed, it is necessary to maintain current inventories to support regular monitoring and certificate maintenance. Organizations that do not properly manage their certificates face significant risks to their core operations, including:

- application outages caused by expired TLS server certificates
- hidden intrusion, exfiltration, disclosure of sensitive data, or other attacks resulting from encrypted threats or server impersonation
- disaster-recovery risk that requires rapid replacement of large numbers of certificates and private keys in response to either certificate authority compromise or discovery of vulnerabilities in cryptographic algorithms or libraries

Despite the mission-critical nature of TLS server certificates, many organizations have not defined the clear policies, processes, roles, and responsibilities needed for effective certificate management. Moreover, many organizations do not leverage available automation tools to support effective management of the ever-growing numbers of certificates. The consequence is continuing susceptibility to security incidents.

This NIST Cybersecurity Practice Guide shows large and medium enterprises how to employ a formal TLS certificate management program to address certificate-based risks and challenges. It describes the TLS certificate management challenges faced by organizations; provides recommended best practices for large-scale TLS server certificate management; describes an automated proof-of-concept implementation that demonstrates how to prevent, detect, and recover from certificate-related incidents; and provides a mapping of the demonstrated capabilities to the recommended best practices and to NIST security guidelines and frameworks.

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

DOCUMENT CONVENTIONS

The terms “shall” and “shall not” indicate requirements to be followed strictly to conform to the publication and from which no deviation is permitted.

The terms “should” and “should not” indicate that among several possibilities, one is recommended as particularly suitable without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited.

The terms “may” and “need not” indicate a course of action permissible within the limits of the publication.

The terms “can” and “cannot” indicate a possibility and capability, whether material, physical, or causal.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

CALL FOR PATENT CLAIMS

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory [ITL] draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

i) under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

ii) without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to tls-cert-mgmt-nccoe@nist.gov.

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
Dung Lam	F5
Robert Smith	F5
William Polk	NIST
Andrew Regenscheid	NIST
Rob Clatterbuck	Thales TCT
Jane Gilbert	Thales TCT
Alexandros Kapasouris	Symantec
Nancy Correll	The MITRE Corporation
Sarah Kinling	The MITRE Corporation
Bob Masucci	The MITRE Corporation
Mary Raguso	The MITRE Corporation
Aaron Aubrecht	Venafi
Justin Hansen	Venafi

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build this example solution. We worked with:

Technology Partner/Collaborator	Build Involvement
DigiCert	External Certificate Authority and CertCentral console
F5	BIG-IP Local Traffic Manager (load balancer)
Thales TCT	Luna SA 1700 Hardware Security Module
Symantec	SSL Visibility Appliance for TLS interception and inspection
Venafi	Trust Protection Platform (TLS certificate manager, log server, and scanning tool)

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

Contents

List of Figures

List of Tables

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

1 Summary

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) recognizes the need to ensure secure communications between clients and servers. To enhance secure communications, the NCCoE launched a project titled Transport Layer Security (TLS) Server Certificate Management. This project uses commercially available technologies to develop a cybersecurity reference design that can be implemented in enterprise environments to reduce outages, improve security, and enable disaster recovery activities related to TLS certificates.

TLS [2] [11] is a broadly used cryptographic protocol that enables authentication and encryption of communications between clients and servers. TLS requires the use of both a certificate that contains information about the certificate owner, as well as a corresponding private key. A server using TLS must have a certificate (and the corresponding private key) to authenticate itself and to establish symmetric keys for encryption. The ongoing maintenance of TLS certificates is labor-intensive and can produce erroneous conditions if the certificate maintenance is not performed correctly.

This project focuses on management of TLS server certificates in medium and large enterprises that rely on TLS to secure both customer-facing and internal applications. Client certificates may optionally be used in TLS for mutual authentication with a TLS server, but management of client certificates is outside the scope of this project. This project demonstrates how to establish, assign, change, and track an inventory of TLS certificates in a manner designed to reduce outages, improve security, and enable disaster recovery activities. This publicly available NIST Cybersecurity Practice Guide details a set of practical steps for implementing a cybersecurity reference design that addresses this TSL server certificate management challenge.

1.1 Challenge

TLS server certificates and private keys are generally installed and managed by the server's system administrator—others usually do not have the access rights required on the system to manage them. To get a certificate, an administrator executes commands on the system to generate a cryptographic key pair (the public key and the private key), and then requests a certificate from a certificate authority (CA). Because many system administrators are not knowledgeable about certificates and cryptography, this process can be confusing and error prone. Large organizations often have a central group, typically called the public key infrastructure (PKI) team, that manages the CAs, which can include external public CAs and internally operated CAs. Due to its expertise in certificates, the PKI team typically supports the system administrators through the key pair generation and certificate request process. Medium and large organizations have many system administrators but only a handful of people on the PKI team. This distributed management environment for certificates and private keys fosters a variety of risks and challenges [8]:

- **Application Outages:** Nearly every enterprise has experienced application outages due to expired TLS server certificates, causing major disruptions to online banking, reservations systems, and healthcare services, to name a few. The drive to encrypt all communications (internal and external) is expanding the reliance on TLS server certificates, increasing the potential for critical system outages.
- **Security Risks:** TLS server certificates function as trusted machine identities. If an attacker can get a fraudulent certificate or compromise a private key, they can impersonate the server or eavesdrop on communications.
- **Disaster Recovery Risks:** Several certificate-related incidents can require an organization to rapidly change large numbers of TLS server certificates, including a CA compromise, algorithm deprecation, or cryptographic library bug. If an organization is not prepared for rapid replacement, its services could be unavailable for days or weeks.

1.2 Solution

The TLS Server Certificate Management Project addressed the risks and challenges described above by:

- Defining an initial reference design that represents a typical enterprise network and recommended TLS infrastructure.
- Building that reference design by using currently available components. This build is known as an "example solution." In the course of building the example solution, the reference design was enhanced. The example solution is an instantiation of the final reference design.
- Demonstrating how the example solution addresses these risks.

The approach taken to address these issues with life-cycle management of the certificates includes the following phases:

- **Establish Governance:** The project team defined a set of certificate management policies based on the guidance provided in existing NIST documents to establish consistent governance of TLS certificates.
- **Create and Maintain an Inventory:** A PKI team worked with project staff representing lines of business and system administrators to establish a complete inventory of all TLS server certificates through automated discovery. The team leveraged configurable rules to automatically organize discovered certificates and associate them with owners as required to enable automated notifications.
- **Register for and Install Certificates:** Certificates were requested and installed to address cases where new certificates were needed, or existing certificates were nearing expiration and required renewal and replacement. Because enterprise environments are diverse, with different technical and organizational constraints, possible methods for requesting and installing certificates were demonstrated, including:

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

  - **Manual:** Security, operational, or technical requirements/constraints mandate that the server's system administrator manually requests a certificate by using command line tools and a certificate management system portal.
  - **Standardized Automated Certificate Installation:** A TLS server is configured to automatically request and install a certificate by using a protocol, such as the Automatic Certificate Management Environment (ACME) protocol, developed by the Internet Engineering Task Force (IETF).
  - **Installation Using a Proprietary Method:** The certificate management system uses a method that is proprietary to the TLS server to install certificates on one or more systems that do not support a standard automated method for requesting and installing certificates.
  - **Development Operations (DevOps)-Based Installation:** A DevOps framework used to install and configure servers/applications also requests and installs certificates. In the current effort the NCCoE undertook only a limited demonstration. This limited demonstration employed Kubernetes in a cloud environment where DevOps frameworks are commonly used.
  - The majority of private keys used with certificates are stored in files; however, Hardware Security Modules (HSMs) were demonstrated to increase the security of private keys. Where practical, the methods listed above were performed on a system that uses an HSM for private-key protection.

- **Continuously Monitor and Manage:** The inventory of certificates was monitored for expiration, proper operation, and security issues. Notifications and alerts were triggered when anomalies were detected. Management operations were regularly performed to ensure proper operation and security.
- **Detect, Respond, and Recover from Incidents:** Scenarios were demonstrated in which, due to situations such as CA compromise or a broken algorithm (e.g., cryptographic library bug that created weak keys for certificates), a large number of certificates required rapid replacement. The certificate management system orchestrated replacement of all certificates.

1.3 Benefits

The project demonstration and its associated documentation offer the following benefits to organizations that have operational or security requirements to implement TLS:

- **Reduced Overhead and Risks—**Large- and medium-size organizations can reduce labor-intensive overhead and risks associated with TLS certificate maintenance by using an example solution comprising currently available components.
- **Improved Information Technology (IT) Environments—**Descriptions of demonstrated methods for using the example solution can reduce the occurrences of erroneous conditions resulting from improper performance of certificate maintenance.

- **Enhanced Cybersecurity—**The availability of source material that explains how the example solution can satisfy specified security requirements can enhance the maturity of cybersecurity programs throughout systems' life cycles.

2 How to Use This Guide

This NIST Cybersecurity Practice Guide demonstrates a standards-based reference design and provides users with the information they need to replicate security platforms composed of currently available components that can be used by large and medium-size organizations to reduce the labor-intensive overhead associated with maintenance of TLS certificates. This reference design is modular and can be deployed in whole or in part.

This guide contains four volumes:

- NIST SP 1800-16A: *Executive Summary*
- NIST SP 1800-16B: Security Risks and Recommended Best Practices
- NIST SP 1800-16C: *Approach, Architecture, and Security Characteristics*–what we built and why **(you are here)**
- NIST SP 1800-16D: *How-To Guides*–instructions for building the example solution
- Depending on your role in your organization, you might use this guide in different ways:
- **Business decision makers, including chief security and technology officers,** will be interested in the *Executive Summary,* NIST SP 1800-16A, which describes the following topics:
- challenges that enterprises face in managing TLS server certificates
- example solution built at the NCCoE
- benefits of adopting the example solution

Senior information technology and security officers will be informed by NIST SP 1800-16B, *Security Risks and Recommended Best Practices*, which describes the:

- TLS server certificate infrastructure and management processes
- risks associated with mismanagement of certificates
- organizational challenges associated with certificate management
- recommended best practices for server certificate management
- recommendations for implementing a successful certificate management program
- You might share the *Executive Summary,* NIST SP 1800-16A, with your leadership team members to help them understand the importance and benefits of adopting standards-based TLS server certificate management.

- **Technology or security program managers** who are concerned with how to identify, understand, assess, and mitigate risk will be interested in the following sections of the guide, NIST SP 1800-16C*,* which describe what we did and why:
- Section 3.4.1, Threats, Vulnerabilities and Risks
- Section 3.4.3, Security Control Map, maps the security characteristics of this example solution to cybersecurity standards and best practices
- You might share *Security Risks and Recommended Best Practices,* NIST SP 1800-16B*,* with your leadership team members to help them understand the security context for adopting the standards-based TLS server certificate management approach described in this volume.
- **IT professionals** who want to implement an approach like this will find the whole practice guide useful. You can use the how-to portion of the guide, NIST SP 1800-16D, to replicate all or parts of the build created in our lab. The how-to guide provides specific product installation, configuration, and integration instructions for implementing the example solution. We do not recreate the product manufacturers' documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create an example solution.
- This guide assumes that IT professionals have experience implementing security products within the enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. Your organization can adopt this solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing parts of enhanced TLS server certificate management. Your organization's security experts should identify the products that will best integrate with your existing tools and IT system infrastructure. We hope that you will seek products that are congruent with applicable standards and best practices. Section 4.3, Technologies, lists the products we used and maps them to the cybersecurity controls provided by this reference solution.

A NIST Cybersecurity Practice Guide does not describe "the" solution, but a possible solution. This is a draft guide. We seek feedback on its contents and welcome your input. Comments, suggestions, and success stories will improve subsequent versions of this guide. Please contribute your thoughts to tls-cert-mgmt-nccoe@nist.gov.

2.1 Typographic Conventions

The following table presents typographic conventions used in this volume.

Table 2-1 Typographic Conventions

Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For detailed definitions of terms, see the *NCCoE Glossary*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit.**
`Monospace`	command-line input, on-screen computer output, sample code examples, and status codes	`Mkdir`
`Monospace Bold`	command-line user input contrasted with computer output	**`service sshd start`**
blue text	link to other parts of the document, a web URL, or an email address	All publications from NIST's NCCoE are available at https://www.nccoe.nist.gov.

3 Approach

The approach taken to building and demonstrating the TLS server certificate management example solution involved composing demonstration environments that included test, diagnostic, and support elements used in the lab for demonstration and test purposes. The demonstration environment includes 1) components typically residing outside the organizational firewall (e.g., public certificate authorities) and 2) systems typically deployed within organizational network environments (e.g., TLS servers, load balancers, DevOps frameworks, internal certificate authorities, certificate managers, and certificate network scanning tools). The goal of the example solution is to permit stakeholders, such as those in the list that follows, to more effectively manage and maintain TLS server certificates throughout system life cycles:

- people in leadership positions who are responsible for cybersecurity

- people in leadership positions who are responsible for the line of business or application and who will drive the need for certificates to be deployed
- system administrators responsible for managing TLS servers and ensuring the load balancer will be represented
- DevOps developers responsible for programming/configuring and managing the DevOps framework
- individuals responsible for reviewing and approving/rejecting certificate management operations
- individuals responsible for managing certificate management systems and public/internal CAs

The NCCoE team accomplished the project in the following sequence:

- established a set of recommended certificate management policy requirements based on the guidance provided in existing NIST documents to establish consistent governance of TLS certificates
- solicited industry collaborators to provide components, operational experience, and configuration assistance; integrated the components into a demonstration environment; configured the components to provide services
- worked with industry collaborators to refine a notional reference design into a demonstration environment capable of:
    - leveraging configurable rules to establish a complete inventory of all TLS server certificates through automated discovery, and automatically organizing discovered certificates and associate owners to enable automated notifications
    - registering for and installing certificates by using manual and automated methods, including protocols such as ACME, proprietary installation methods, and a DevOps framework
- worked with industry collaborators to integrate HSMs into the demonstration environment for protecting private keys
- documented collaborator contributions
- documented the final architecture of the demonstration environment
- worked with industry collaborators to demonstrate continuous monitoring of the inventory of certificates for expiration, proper operation, and security issues and generation of notifications and alerts when anomalies are detected
- worked with industry collaborators to demonstrate detection, response, and recovery from security incidents
- conducted security and functional testing of the demonstration environment

- conducted and documented the results of a risk assessment and a security characteristics analysis, including mapping the security contributions' demonstrated capabilities to the *Framework for Improving Critical Infrastructure Cybersecurity* (Cybersecurity Framework) [4], NIST Special Publication (SP) 800-53 [7], and the recommended policies in NIST SP 1800-16B
- documented the steps taken to install and configure each component of the demonstration environment
- worked with industry collaborators to suggest future considerations for TLS certificate management in general

3.1 Audience

This guide is intended for individuals responsible for security architecture and strategy, system administration, PKI support, IT systems acquisition, cybersecurity assessments, IT system component development, marketing and support for environments for which TLS is an essential security protocol for providing confidentiality and integrity protection to systems and operations, and implementing security solutions in organizations' IT support activities. The technical components will appeal to system administrators, IT managers, IT security managers, and others directly involved in the secure and safe operation of IT networks.

3.2 Scope

As stated in the Summary above, this project focuses on management of TLS server certificates in medium and large enterprises that rely on TLS to secure both customer-facing and internal applications. This guide shows how to establish and maintain an inventory of TLS certificates; assign and track certificate owners (i.e., custodians), identify issues with and vulnerabilities of the TLS infrastructure, automate enrollment and installation, report, and continuously monitor TLS certificates in the environment described above.

This project limits its scope to TLS server certificates. Client certificates may optionally be used in TLS for mutual authentication, but management of client certificates is outside the scope of this project.

The security and integrity of TLS relies on secure implementation and configuration of TLS servers and effective TLS server certificate management. Guidance regarding the implementation and configuration of TLS servers is outside of the scope of this document. Secure implementation and configuration of TLS servers is addressed in NIST SP 800-52 [14]. Organizations should provide clear instruction to groups and individuals deploying TLS servers in their environments, to read, understand, and follow the guidance provided in NIST SP 800-52.

3.3 Assumptions

This project is guided by the following assumptions:

- The processes for obtaining and maintaining TLS server certificates in medium and large IT enterprises is labor-intensive and error prone.
- The drive to encrypt all communications (internal and external) is expanding reliance on TLS server certificates, thereby increasing the potential for critical system outages due to expired certificates.
- TLS server certificates serve as trusted machine identities; if an attacker can get a fraudulent certificate or compromise a private key, they can impersonate the server or eavesdrop on communications.
- Certificate-related incidents (e.g., a CA compromise, algorithm deprecation, or cryptographic library bug) can require an organization to rapidly change large numbers of TLS server certificates.
- If an organization is not prepared for rapid replacement, then its services could be unavailable for days or weeks.

3.4 Risk Assessment

NIST SP 800-30 Revision 1, *Guide for Conducting Risk Assessments* [5] states that risk is "a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of (i) the adverse impacts that would arise if the circumstance or event occurs and (ii) the likelihood of occurrence." The guide further defines risk assessment as "the process of identifying, estimating, and prioritizing risks to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, resulting from the operation of an information system. Part of risk management incorporates threat and vulnerability analyses, and considers mitigations provided by security controls planned or in place."

The NCCoE recommends that any discussion of risk management, particularly at the enterprise level, begins with a comprehensive review of NIST SP 800-37 Revision 2, *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy* [6]—material that is available to the public. The Risk Management Framework (RMF) [9] guidance, as a whole, was invaluable and gave us a baseline to assess risks, from which we developed the project, the security characteristics of the build, and this guide.

3.4.1 Threats, Vulnerabilities, and Risks

NIST SP 1800-16B, *Security Risks and Recommended Best Practices*, describes the risks associated with management of TLS server certificates. It points out that, despite the mission-critical nature of TLS server certificates, many organizations do not have clear policies, processes, 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. Malicious entities are using encryption to attack organizations at an ever-increasing rate. TLS is being turned against enterprises to:

- deliver malware undetected
- listen in on private conversations
- disrupt secured transactions
- exfiltrate data over encrypted communication channels

Volume B states that certificate owners are typically not knowledgeable about the best practices for effectively managing TLS server certificates. The RMF [9] process described in NIST SP 800-37, together with the Cybersecurity Framework and NIST SP 800-53, informed our risk assessment and subsequent recommendations from which we developed the security characteristics of the build and this guide.

The most serious risks associated with certificate management stem from certificate owners, responsible for the systems where certificates are deployed, not being provided clear certificate management requirements, not understanding their responsibilities in fulfilling those requirements, and those requirements not being enforced as policies. Risks identified in Volume B include:

- outages caused by expired certificates due to:
  - the system administrator forgetting about the certificate
  - the system administrator ignoring notifications that the certificate will soon expire
  - the system administrator not properly installing or updating the CA certificate chain
  - the system administrator being reassigned and nobody else receiving expiry notifications
  - the system administrator enrolling for a new certificate but not installing it on the server(s) in time, installing it incorrectly, or not resetting the application/server, so the newly installed certificate is loaded and used
  - the application relying on multiple load-balanced servers and the certificate not being updated on all of them
- server impersonation (an attacker being able to impersonate a legitimate TLS server)
- the organization not being able to replace certificates and private keys in a timely manner due to inadequate records, knowledge, and processes in instances such as:
  - CA compromise
  - cryptographic algorithm vulnerability
  - cryptographic library bugs
- encrypted threats such as TLS encryption allowing attackers to hide malicious activities within encrypted TLS connections

Also, as pointed out in Volume B, an attacker may be able to masquerade as a server to all clients if:

- the server's private key
  - is weak
  - can be obtained by an attacker
- an attacker can obtain a public key certificate for a public key corresponding to its own private key in the name of the server from a CA trusted by the clients

Aside from the risks of not managing TLS server certificates properly, additional risks often plague TLS implementations themselves. Proper protocol specification does not guarantee the security of implementations. In particular, when integrating into higher level protocols, TLS and its PKI-based authentication are sometimes the source of misunderstandings and implementation shortcuts. An extensive survey of these issues can be found in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*.

3.4.2 Security Categorization and NIST SP 800-53 Controls

Under the RMF, the first step in managing risk is determining the impacts of exploitation of system confidentiality, integrity, and availability vulnerabilities. NIST SP 800-53-controls needed to mitigate system vulnerabilities are keyed to the Federal Information Processing Standards (FIPS) 199 impact levels. Based on the risks identified, and assuming a *Standards for Security Categorization of Federal Information and Information Systems*, FIPS 199 [13] ***moderate*** impact level (exploitation of vulnerabilities would result in serious harm to the system and its mission), a number of NIST SP 800-53 controls are assigned to address TLS server certificate risks: AC-1, AC-5, AC-6, AC-16, AT-2, AU-1, AU-2, AU-3, AU-6, AU-12, AU-13, AU-14, CA-1, CA-2, CA-5, CA-7, CM-2, CM-3, CM-5, CM-6, CM-8, CM-9, CM-12, CP-2, CP-3, CP-4, CP-7, CP-13, IA-3, IA-4, IA-5, IA-9, IR-1, IR-2, IR-3, IR-4, MA-1, MA-6, PL-2, PL-9, PL-10, PM-1, PM-2, PM-4, PM-5, PM-7, PM-9, RA-3, RA-5, RA-7, SA-1, SA-3, SA-4, SA-10, SC-1, SC-6, SC-8, SC-12, SC-17, SC-23, and SI-4. Appendix C of Volume B describes these security controls and their relevance to the best practices identified in Volume B.

3.4.3 Security Control Map

The objective of this project is to demonstrate how the processes for obtaining and maintaining TLS server certificates in medium and large IT enterprises can be made less labor-intensive and error prone, to reduce security and operational risks. This requires adherence to the following principles:

- **Governance and Risk Management:** The project includes clear recommended policies that can be used to educate the lines of business and system administrators to ensure they understand the security risks and their responsibilities in addressing those risks. Organizations are free to copy and use these recommended policies for definition of their own internal TLS certificate management policies.

- **Visibility and Awareness:** Most organizations do not have an inventory of their TLS server certificates and private keys, their installed locations, and their responsible individuals/groups. This project demonstrates how to achieve visibility and awareness of all certificates.
- **Reliable and Efficient Certificate Provisioning:** This project demonstrates effective processes to ensure availability of valid certificates and keys for TLS servers while minimizing overhead and the impact on operations.
- **Certificate Disaster Recovery:** This project demonstrates effective processes for organizations to be prepared for and to respond to large-scale incidents (e.g., CA compromise) that require rapid replacement of large numbers of certificates and keys.
- **Audit Logging:** Many organizations do not generate, store, and review audit logs for their certificates and associated private keys. This project demonstrates how to establish and maintain complete audit trails of certificate and private-key life cycles.
- **Secure Certificate Management Platform:** The certificate management platform in this project is deployed on a hardened system and provides the security attributes required to protect the assets it manages.
- **Private-Key Security:** The project demonstrates automated management, which reduces the requirement for direct administrator access to private keys, and HSM-based private-key protection, which significantly increases private-key security.

Appendix B of Volume B maps the recommended best practices for TLS server certificate management described in volume B to the Cybersecurity Framework Subcategories. The following table lists the security Subcategories of the Cybersecurity Framework that are supported by the example TLS server certificate management example solution described in this volume, and it maps these Cybersecurity Framework Subcategories to other informative security control references.

Table 3-1 Mapping Security Characteristics of the Example Implementation to the Cybersecurity Framework and Informative Security Control References

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
Identify (ID)	ID.AM-2: Software platforms and applications within the organization are inventoried.	• CCS CSC 2 • COBIT 5 BAI09.01, BAI09.02, BAI09.05 • ISA 62443-2-1:2009 4.2.3.4 • ISA 62443-3-3:2013 SR 7.8 • ISO/IEC 27001:2013 A.8.1.1, A.8.1.2 • NIST SP 800-53 Rev. 4 CM-8
	ID.AM-6: Cybersecurity roles and responsibilities for the entire workforce	• COBIT 5 APO01.02, DSS06.03 • ISA 62443-2-1:2009 4.3.2.3.3

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
	and third-party stakeholders (e.g., suppliers, customers, partners) are established.	• ISO/IEC 27001:2013 A.6.1.1 • NIST SP 800-53 Rev. 4 CP-2, PS-7, PM-11
	ID.GV-1: Organizational cybersecurity policy is established and communicated.	• CIS CSC 19 • COBIT 5 APO01.03, APO13.01, EDM01.01, EDM01.02 • ISA 62443-2-1:2009 4.3.2.6 • ISO/IEC 27001:2013 A.5.1.1 • NIST SP 800-53 Rev. 4 -1 controls from all security control families
	ID.GV-2: Cybersecurity roles and responsibilities are coordinated and aligned with internal roles and external partners.	• CIS CSC 19 • COBIT 5 APO01.02, APO10.03, APO13.02, DSS05.04 • ISA 62443-2-1:2009 4.3.2.3.3 • ISO/IEC 27001:2013 A.6.1.1, A.7.2.1, A.15.1.1 • NIST SP 800-53 Rev. 4 PS-7, PM-1, PM-2
	ID.GV-3: Legal and regulatory requirements regarding cybersecurity, including privacy and civil liberties obligations, are understood and managed.	• CIS CSC 19 • COBIT 5 BAI02.01, MEA03.01, MEA03.04 • ISA 62443-2-1:2009 4.4.3.7 • ISO/IEC 27001:2013 A.18.1.1, A.18.1.2, A.18.1.3, A.18.1.4, A.18.1.5 • NIST SP 800-53 Rev. 4 -1 controls from all security control families
	ID.GV-4: Governance and risk management processes address cybersecurity risks.	• COBIT 5 EDM03.02, APO12.02, APO12.05, DSS04.02 • ISA 62443-2-1:2009 4.2.3.1, 4.2.3.3, 4.2.3.8, 4.2.3.9, 4.2.3.11, 4.3.2.4.3, 4.3.2.6.3 • ISO/IEC 27001:2013 Clause 6 • NIST SP 800-53 Rev. 4 SA-2, PM-3, PM-7, PM-9, PM-10, PM-11

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
Protect (PR)	PR.AC-1: Identities and credentials are issued, managed, verified, revoked, and audited for authorized devices, users, and processes.	• CCS CSC 16 • COBIT 5 DSS05.04, DSS06.03 • ISA 62443-2-1:2009 4.3.3.5.1 • ISA 62443-3-3:2013 SR 1.1, SR 1.2, SR 1.3, SR 1.4, SR 1.5, SR 1.7, SR 1.8, SR 1.9 • ISO/IEC 27001:2013 A.9.2.1, A.9.2.2, A.9.2.3, A.9.2.4, A.9.2.6, A.9.3.1, A.9.4.2, A.9.4.3 • NIST SP 800-53 Rev. 4 AC-2, IA-1, IA-2, IA-3, IA-4, IA-5, IA-6, IA-7, IA-8, IA-9, IA-10, IA-11
	PR.AC-3: Remote access is managed.	• COBIT 5 APO13.01, DSS01.04, DSS05.03 • ISA 62443-2-1:2009 4.3.3.6.6 • ISA 62443-3-3:2013 SR 1.13, SR 2.6 • ISO/IEC 27001:2013 A.6.2.2, A.13.1.1, A.13.2.1 • NIST SP 800-53 Rev. 4 AC-17, AC-19, AC-20
	PR.AC-4: Access permissions and authorizations are managed, incorporating the principles of least privilege and separation of duties.	• CIS CSC 3, 5, 12, 14, 15, 16, 18 • COBIT 5 DSS05.04 • ISA 62443-2-1:2009 4.3.3.7.3 • ISA 62443-3-3:2013 SR 2.1 • ISO/IEC 27001:2013 A.6.1.2, A.9.1.2, A.9.2.3, A.9.4.1, A.9.4.4, A.9.4.5 • NIST SP 800-53 Rev. 4 AC-1, AC-2, AC-3, AC-5, AC-6, AC-14, AC-16, AC-24
	PR.AC-6: Identities are proofed and bound to credentials and asserted in interactions.	• CCS CSC 16 • COBIT 5 DSS05.04, DSS05.05, DSS05.07, DSS06.03 • ISA 62443-2-1:2009 4.3.3.2.2, 4.3.3.5.2, 4.3.3.7.2, 4.3.3.7.4 • ISA 62443-3-3:2013 SR 1.1, SR 1.2, SR 1.4, SR 1.5, SR 1.9, SR 2.1 • ISO/IEC 27001:2013 A.7.1.1, A.9.2.1

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
		• NIST SP 800-53 Rev. 4 AC-1, AC-2, AC-3, AC-16, AC-19, AC-24, IA-1, IA-2, IA-4, IA-5, IA-8, PE-2, PS-3
	PR.AC-7: Users, devices, and other assets are authenticated (e.g., single-factor, multi-factor) commensurate with the risk of the transaction (e.g., individuals' security and privacy risks and other organizational risks).	• CCS CSC 1, 12, 15, 16 • COBIT 5 DSS05.04, DSS05.10, DSS06.10 • ISA 62443-2-1:2009 4.3.3.6.1, 4.3.3.6.2, 4.3.3.6.3, 4.3.3.6.4, 4.3.3.6.5, 4.3.3.6.6, 4.3.3.6.7, 4.3.3.6.8, 4.3.3.6.9 • ISA 62443-3-3:2013 SR 1.1, SR 1.2, SR 1.5, SR 1.7, SR 1.8, SR 1.9, SR 1.10 • ISO/IEC 27001:2013 A.9.2.1, A.9.2.4, A.9.3.1, A.9.4.2, A.9.4.3, A.18.1.4 • NIST SP 800-53 Rev. 4 AC-7, AC-8, AC-9, AC-11, AC-12, AC-14, IA-1, IA-2, IA-3, IA-4, IA-5, IA-8, IA-9, IA-10, IA-11
	PR.DS-1: Data at rest is protected.	• CCS CSC 17 • COBIT 5 APO01.06, BAI02.01, BAI06.01, DSS06.06 • ISA 62443-3-3:2013 SR 3.4, SR 4.1 • ISO/IEC 27001:2013 A.8.2.3 • NIST SP 800-53 Rev. 4 SC-28
	PR.DS-2: Data in transit is protected.	• CCS CSC 17 • COBIT 5 APO01.06, DSS06.06 • ISA 62443-3-3:2013 SR 3.1, SR 3.8, SR 4.1, SR 4.2 • ISO/IEC 27001:2013 A.8.2.3, A.13.1.1, A.13.2.1, A.13.2.3, A.14.1.2, A.14.1.3 • NIST SP 800-53 Rev. 4 SC-8
	PR.DS-3: Assets are formally managed throughout removal, transfers, and disposition.	• COBIT 5 BAI09.03 • ISA 62443-2-1:2009 4. 4.3.3.3.9, 4.3.4.4.1 • ISA 62443-3-3:2013 SR 4.2 • ISO/IEC 27001:2013 A.8.2.3, A.8.3.1, A.8.3.2, A.8.3.3, A.11.2.7

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
		• NIST SP 800-53 Rev. 4 CM-8, MP-6, PE-16
	PR.DS-6: Integrity-checking mechanisms are used to verify software, firmware, and information integrity.	• ISA 62443-3-3:2013 SR 3.1, SR 3.3, SR 3.4, SR 3.8 • ISO/IEC 27001:2013 A.12.2.1, A.12.5.1, A.14.1.2, A.14.1.3 • NIST SP 800-53 Rev. 4 SC-16, SI-7
	PR.DS-8: Integrity-checking mechanisms are used to verify hardware integrity.	• COBIT 5 BAI03.05 • ISA 62443-2-1:2009 4.3.4.4.4 • ISO/IEC 27001:2013 A.11.2.4 • NIST SP 800-53 Rev. 4 SA-10, SI-7
	PR.IP-2: A system development life cycle to manage systems is implemented.	• COBIT 5 APO13.01 • ISA 62443-2-1:2009 4.3.4.3.3 • ISO/IEC 27001:2013 A.6.1.5, A.14.1.1, A.14.2.1, A.14.2.5 NIST SP 800-53 Rev. 4 SA-3, SA-4, SA-8, SA10, SA-11, SA-12, SA-15, SA-17, PL-8
	PR.IP-3: Configuration change control processes are in place.	• COBIT 5 BAI01.06, BAI06.01 • ISA 62443-2-1:2009 4.3.4.3.2, 4.3.4.3.3 • ISA 62443-3-3:2013 SR 7.6 • ISO/IEC 27001:2013 A.12.1.2, A.12.5.1, A.12.6.2, A.14.2.2, A.14.2.3, A.14.2.4 • NIST SP 800-53 Rev. 4 CM-3, CM-4, SA-10
	PR.PT-1: Audit/log records are determined, documented, implemented, and reviewed in accordance with policy.	• CCS CSC 14 • COBIT 5 APO11.04 • ISA 62443-2-1:2009 4.3.3.3.9, 4.3.3.5.8, 4.3.4.4.7, 4.4.2.1, 4.4.2.2, 4.4.2.4 • ISA 62443-3-3:2013 SR 2.8, SR 2.9, SR 2.10, SR 2.11, SR 2.12 • ISO/IEC 27001:2013 A.12.4.1, A.12.4.2, A.12.4.3, A.12.4.4, A.12.7.1 • NIST SP 800-53 Rev. 4 AU Family

Cybersecurity Framework Function	Cybersecurity Framework Subcategory	Informative References
	PR.PT-5: Mechanisms (e.g., fail-safe, load balancing, hot swap) are implemented to achieve resilience requirements in normal and adverse situations.	• COBIT 5 BAI04.01, BAI04.02, BAI04.03, BAI04.04, BAI04.05, DSS01.05 • ISA 62443-2-1:2009 4.3.2.5.2 • ISA 62443-3-3:2013 7.1, SR 7.2 • ISO/IEC 27001:2013 A.17.1.2, A.17.2.1 • NIST SP 800-53 Rev. 4 CP-7, CP-8, CP-11, CP-13, PL-8, SA-14, SC-6
	DE.AE-5: Incident alert thresholds are established.	• COBIT 5 APO12.06 • ISA 62443-2-1:2009 4.2.3.10 • NIST SP 800-53 Rev. 4 IR-4, IR-5, IR-8
	DE.CM-1: The network is monitored to detect potential cybersecurity events.	• COBIT 5 APO12.06 • ISA 62443-2-1:2009 4.3.4.5.9 • ISA 62443-3-3:2013 SR 6.1 • ISO/IEC 27001:2013 A.16.1.2 • NIST SP 800-53 Rev. 4 AU-6, CA-2, CA-7, RA-5, SI-4
Respond (RS)	RS.AN-5: Processes are established to receive, analyze, and respond to vulnerabilities disclosed to the organization from internal and external sources (e.g., internal testing, security bulletins, or security researchers).	• CIS CSC 4, 19 • COBIT 5 EDM03.02, DSS05.07 • NIST SP 800-53 Rev. 4 SI-5, PM-15
	RS.MI-2: Incidents are mitigated.	• ISA 62443-2-1:2009 4.3.4.5.6, 4.3.4.5.10 • ISO/IEC 27001:2013 A.12.2.1, A.16.1.5 • NIST SP 800-53 Rev. 4 IR-4
	RS.MI-3: Newly identified vulnerabilities are mitigated or documented as accepted risks.	• ISO/IEC 27001:2013 A.12.6.1 • NIST SP 800-53 Rev. 4 CA-7, RA-3, RA-5

4 Architecture

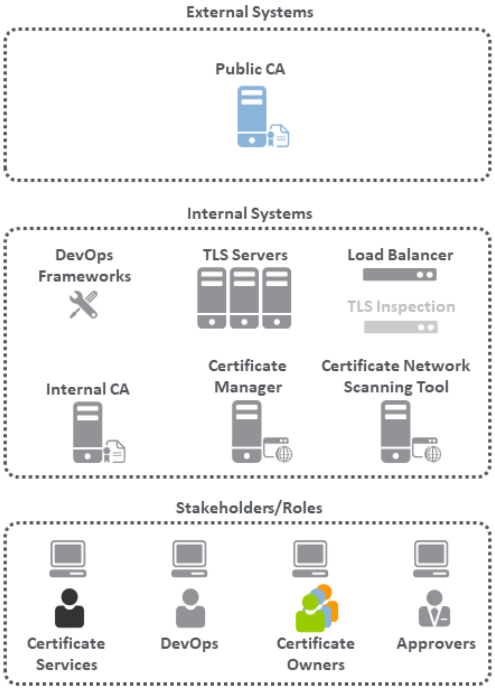
The TLS server certificate management architecture enables medium and large enterprises to manage their TLS server certificates and cryptographic keys efficiently and effectively. The architecture provides the following protections:

- use of a certificate manager and related certificate scanning, monitoring, and storage components to:
    - automate establishment and maintenance of an inventory of TLS server certificates and keys
    - assign and track certificate owners
    - automate enrollment, installation, renewal, and rapid replacement of certificates and keys
    - continuously monitor certificates and keys, report on their status, and automate remediation to enforce compliance with policy [3] and avoid unintended expiration
    - support disaster recovery through rapid, large-scale replacement of certificates
    - log all certificate management operations
- use of a TLS inspection appliance to decrypt network traffic encrypted via TLS, so it can be inspected for malware and other threats
- use of a hardened, tamper-resistant physical appliance that securely generates, stores, manages, and processes cryptographic key pairs for use with TLS certificates; this enables those keys to remain securely within the confines of the secure device while they are used to issue signed TLS certificates

4.1 Logical Architecture

The functions demonstrated in this project require a variety of component systems and configurations. Figure 4-1 depicts the architectural components used in the logical architecture and the roles that support TLS server certificate management.

Figure 4-1 Logical Architecture Components and Roles



4.1.1 External Systems

The architecture includes a CA component that typically resides outside the organizational firewall:

- **Public CA:** A publicly trusted CA issued one or more of the certificates used on the TLS servers in the implementation.

4.1.2 Internal Systems

The architecture includes the following systems that are typically deployed within organizational network environments.

- **TLS Servers:** Multiple systems were configured as TLS servers (e.g., web server, application server, or other service). Certificates are deployed and managed on these systems.
- **Load Balancer:** A load balancer acted as a TLS server with a certificate and facilitated the load balancing of traffic to other TLS servers.
- **DevOps Framework(s):** A DevOps framework (Kubernetes) automated management of containers acting as TLS servers and deployment of certificates on those TLS servers.
- **Internal CA:** An internal CA issued certificates to some TLS servers.

- **Certificate Manager:** A certificate management system was used to inventory and manage TLS server certificates deployed in the environment.
- **Certificate Network Scanning Tool:** A vulnerability scanning tool facilitated discovery of TLS server certificates via network scanning.
- **TLS Inspection Appliance:** This appliance decrypts traffic encrypted via TLS. As a result, traffic is analyzed and inspected for malicious activity, viruses, malware, or other threats. (Figure 4-1 depicts this component by using a faded icon to convey that some organizations, as a matter of policy, may not want to include it in their network architecture.)
- Humans play an important part in the management of TLS server certificates in enterprises. Descriptions of their different roles are explained below:
- **Certificate Owners:** The groups and individuals responsible for the systems where certificates are deployed; they establish and maintain an inventory of all certificates and keys on their systems. Typically, there are several roles within a certificate owner group, including executives who are accountable for ensuring certificate-related responsibilities are addressed; system administrators who manage individual systems and the certificates on them, including requesting and installing certificates; and application owners. The certificate owners typically are not knowledgeable or familiar with the risks associated with certificates or the best practices for effectively managing them. Nonetheless, they must ensure their certificates are compliant by relying on the central certificate service technologies, expertise, and guidance supplied by the Certificate Services team.
- **Certificate Services Team:** This group includes experts that drive and support the organization’s formal certificate management program. They manage relationships with public CAs to manage internal CAs, and provide the central certificate service that certificate owners use to establish and maintain their certificate and key inventories. This team is knowledgeable about TLS server certificates but typically lacks sufficient resources or access required to directly manage certificates on the extensive number of systems where certificates are deployed.
- **DevOps:** This group provisions systems and software through automated programmatic processes and tools known collectively as DevOps. It is a common practice to request and deploy TLS server certificates by using DevOps technologies.
- **Approvers:** Approvers serve as registration authorities within organizations. In this role, they review certificate signing requests, and confirm the validity of the request and the authority of the requester. They also send the approval of the certificate signing request to the certificate service or CA.

The internal and external components described above were integrated to create the TLS server certificate management example solution in the TLS lab. Figure 4-2 depicts the logical architecture of the example solution. The logical architecture shows the network structure and components that enable various types of TLS server certificate management operations. For several reasons, it is not intended to serve as a definitive example for an organization to model its own network design. For starters, it lacks a firewall, intrusion detection system, and other components an organization may use to secure its network. Although some IT professionals may consider these components essential to ensuring network security, they were not part of the logical architecture for the example implementation. The TLS team concluded that these components were not relevant in showcasing the TLS server certificate management functionality.

Figure 4-2 shows the logical architecture of the TLS server certificate management example implementation, which comprises an external CA and an internal network logically organized into three zones. These zones roughly model a defense-in-depth strategy of grouping components on subnetworks that require increasing levels of security as one moves inward from the perimeter of the organization: a demilitarized zone (DMZ) between the internet and the rest of the enterprise; a data center hosting applications and services widely used across the enterprise; and a more secure data center hosting critical security and infrastructure components, including certificate management components.

At the ingress from the internet within the DMZ, a load balancer is deployed to act as a TLS proxy—distributing incoming traffic from external users across three TLS servers behind it that are serving the same application: two Apache servers and one Microsoft Internet Information Services (IIS) server. (Note: To simplify the illustration, the connections between individual components are not shown.) TLS certificate management is used to enroll and provision new certificates to the load balancer and servers in the DMZ, and to perform overall certificate management on these devices, including automatically replacing certificates nearing expiration.

Within the data center zone of the logical architecture sit various types of web servers, application servers, and a DevOps framework—all act as TLS servers. These components are used to demonstrate the ability to automatically enroll and provision a new certificate as well as automatically replace a certificate that is nearing expiration on these systems. Various types of certificate management are also demonstrated, including remote agentless management, the ACME protocol, and a DevOps certificate management plug-in.

Within the DMZ and the data center zone, taps (depicted as white dots) are used on the network connections between the load balancer, the servers behind it, and the network connections between the DMZ servers and the second-tier servers in the data center behind them. These taps send traffic on the encrypted TLS connections to a TLS inspection appliance for passive decryption. In Figure 4-2, this TLS inspection appliance is depicted by using a faded icon to convey that some organizations, as a matter of policy, may not want to include it as part of their network architecture. However, for those organizations that consider passive inspection as part of their security assurance strategy, the certificate manager depicted in the architecture can securely copy private keys from several different TLS servers to the TLS inspection appliance. It can also securely replace expiring keys on those servers and immediately copy them to the inspection appliance before expiration.

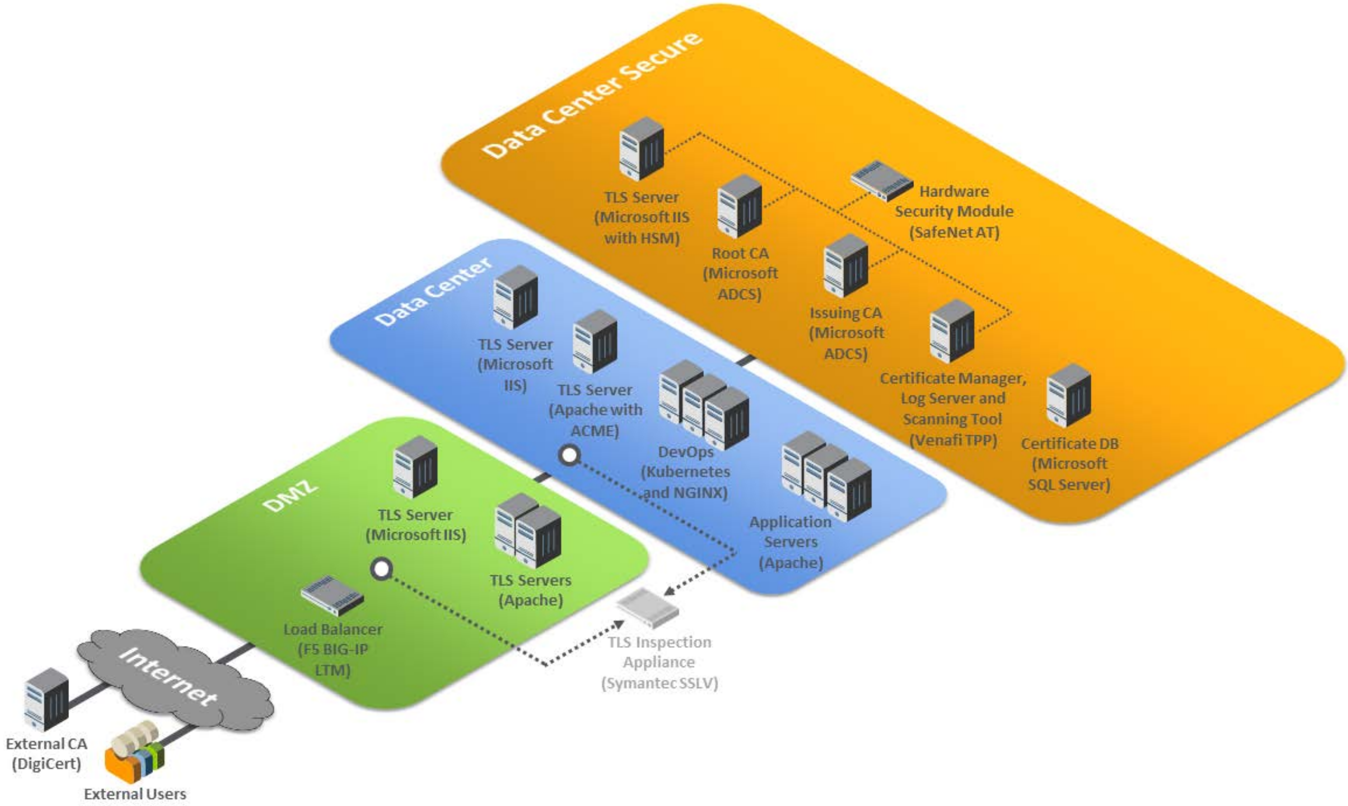
Within the data center secure zone of the logical architecture sit the components that perform TLS server certificate management: internal root and issuing CAs, a certificate manager, a certificate log server, a certificate network scanning tool, a certificate database, and an HSM. For demonstration purposes, a TLS server connected to the HSM is also present in this zone.

The certificate manager, in conjunction with the certificate database and the various types of servers in the rest of the architecture, demonstrates establishment and maintenance of a systematized inventory of certificates (and keys) in use on the network. The certificate manager also monitors the TLS certificates (and keys) managed by the inventory system and responds to any issues. For example, it will send expiration reports and notifications to certificate owners, informing them a certificate is being automatically replaced, is about to expire, or does not conform to policy. It also supports disaster recovery efforts by quickly replacing a large number of certificates located throughout the network architecture.

The certificate manager, in conjunction with the CAs, enrolls and provisions certificates (and keys), stores attributes with those certificates, and discovers the absence of an expected certificate from a machine where it should be installed. The certificate owner or the Certificates Services team can alert a certificate manager when a certificate must be revoked or if the owner associated with a certificate needs to be changed. The certificate scanning tool discovers certificates not currently being managed by the inventory. The certificate log server records all automated certificate and private-key management operations, including certificate creation, installation, and revocation; key pair generation; certificate requests and request approvals; certificate and key copying; and certificate and key replacement.

All components in the data center secure zone, except for the certificate database, are configured to use the HSM to securely generate, store, manage, and process private and symmetric keys. Cryptographic operations are performed within the HSM, ensuring that keys remain safe within its hardened confines rather than risk exposure outside it. The HSM stores and protects the symmetric keys that secure sensitive data in the certificate database. It generates, stores, manages, and performs signing operations with the internal CAs' signing keys and cryptographic operations with the TLS server private key.

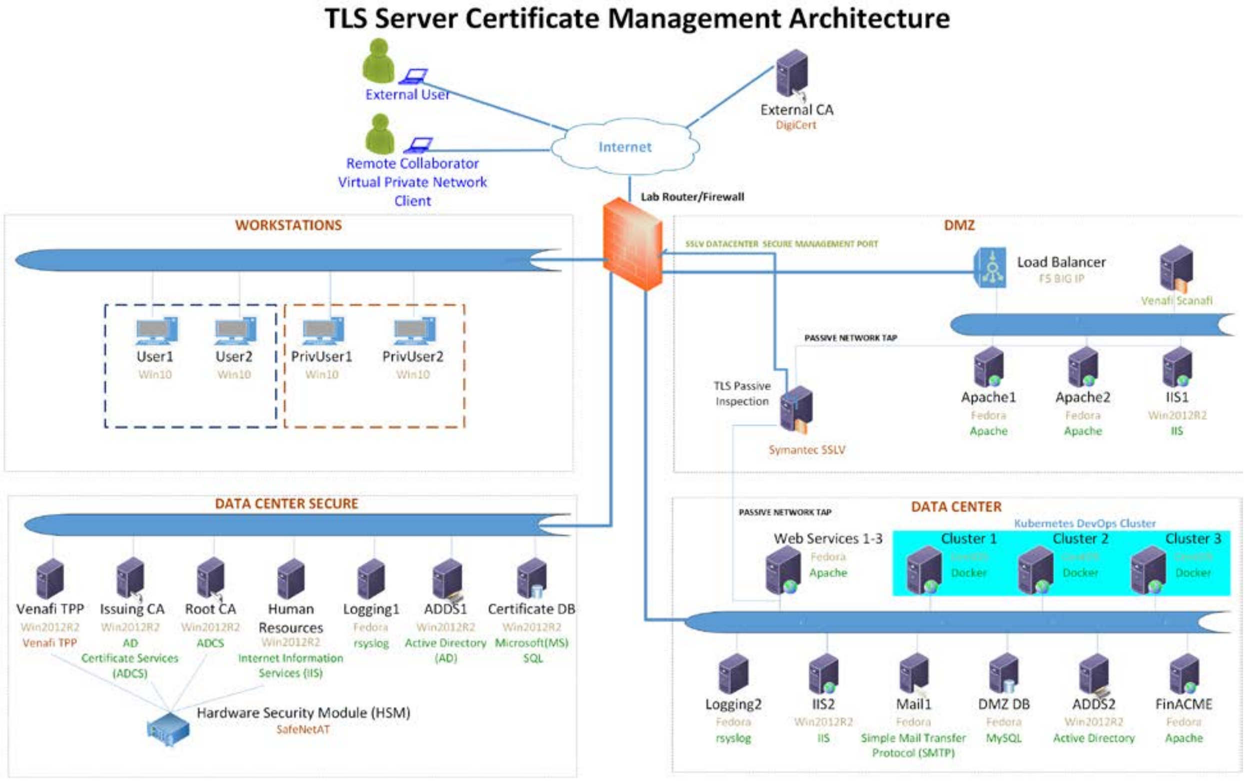
Figure 4-2 TLS Server Certificate Management Example Solution Logical Architecture


Data Center Secure
TLS Server (Microsoft IIS with HSM)
Root CA (Microsoft ADCS)
Hardware Security Module (SafeNet AT)
Issuing CA (Microsoft ADCS)
Certificate Manager, Log Server and Scanning Tool (Venafi TPP)
Certificate DB (Microsoft SQL Server)
Data Center
TLS Server (Microsoft IIS)
TLS Server (Apache with ACME)
DevOps (Kubernetes and NGINX)
Application Servers (Apache)
DMZ
TLS Server (Microsoft IIS)
TLS Servers (Apache)
Load Balancer (F5 BIG-IP LTM)
TLS Inspection Appliance (Symantec SSLV)
Internet
External CA (DigiCert)
External Users

4.2 Physical Architecture

Figure 4-2 depicts the logical architecture deployed in the TLS lab to yield the TLS server certificate management example implementation. Figure 4-3 illustrates the laboratory configuration of that example implementation.

Figure 4-3 Laboratory Configuration of TLS Server Certificate Management Example Implementation



The NCCoE lab provides the following supporting infrastructure for the example implementation:

- firewall-protected connection to the internet, where an external CA resides
- Windows 2012 server with remote desktop manager that acts as a jump box to facilitate installation, deployment, and management of server software for collaborative projects
- segmented laboratory network backbone that models the separation that typically exists between subnetworks belonging to different parts of a medium-to-large-scale enterprise, such as a DMZ, data center hosting widely used applications and services, and a more secure data center hosting critical security infrastructure components
- virtual machine and network infrastructure
- Windows 2012 servers running Active Directory (AD) Certificate Services, including:
    - internal root CA that can issue and self-sign its own TLS certificate
    - internal issuing CA that:
        - issues TLS certificates to the servers that request them (issue CAs are subordinate to and certified by the root CA)
        - manages the life cycle of certificates (including request, issuance, enrollment, publication, maintenance, revocation, and expiration)
- Microsoft structured query language (SQL) Server hosting the database of TLS certificates and keys and corresponding configuration data
- DevOps automation framework, including Kubernetes, Docker, and Jetstack, that demonstrates automated certificate management when performing open-source container orchestration
- Apache, Microsoft IIS, and NGINX servers used to demonstrate various ways of managing TLS server certificates, including remote agentless certificate management, management via the ACME protocol (via the Certbot utility), and management via DevOps
- Apache servers used to demonstrate certificate management on second-tier internal application servers

The following collaborator-supplied components were integrated into the above supporting infrastructure to yield the TLS server certificate management example implementation:

- Venafi Trust Protection Platform (TPP), which performs automated TLS server certificate and private-key management, including monitoring, remediation, and rapid replacement of TLS certificates and keys; TLS certificate and key policy enforcement; automated certificate requests and renewals; automated network scanning for TLS certificates; and logging of certificate and private-key management operations
- Thales Trusted Cyber Technologies (Thales TCT) Luna SA 1700 hardware security module used to securely generate, store, manage, and process the cryptographic key pair and uses it to sign TLS certificates within a hardened, tamper-resistant physical appliance. It is also used to store other

keys, such as the database encryption key and the TLS certificate keys for the key manager component (Venafi TPP) and the CAs

- DigiCert external CA, which issues and renews TLS certificates
- F5 Networks BIG-IP Local Traffic Manager load balancer, which acts as a TLS proxy and distributes received traffic across a number of other TLS servers
- Symantec SSL Visibility, a visibility appliance used to inspect intercepted traffic on encrypted TLS connections

The supporting infrastructure components and the TLS-server-specific collaborator-supplied components are discussed further in the technologies section below. Installation, configuration, and integration of these components are described in detail in Volume D.

4.3 Technologies

Table 4-1 lists the technologies used in this project, and provides a mapping among the generic application term, the specific product used, and the security control(s) the product provides. Refer to Table 3-1 for an explanation of the NIST Cybersecurity Framework Subcategory codes.

Table 4-1 Products and Technologies

Component	Product	Functionality	Cybersecurity Framework Subcategories
Certificate manager	Venafi Trust Protection Platform	Automated monitoring, remediation, and rapid replacement of TLS certificates and keys; TLS certificate and key policy enforcement; automated certificate requests and renewals; workflow for required approvals.	PR.AC-4, ID.AM-2, PR.AC-1, PR.DS-2, PR.DS-3, PR.DS-6, PR.IP-2, PR.IP-3, PR.PT-1, DE.AE-5, RS.MI-2, RS.MI-3: Newly identified vulnerabilities are mitigated or documented as accepted risks.
Internal TLS certificate network scanning tool	Venafi TPP	Automated discovery of TLS certificates via network scanning.	PR.AC-1, PR.AC-4, DE.AE-5, DE.CM-1
Certificate log server	Venafi TPP	Used to log all certificate and private-key management operations.	PR.PT-1

Component	Product	Functionality	Cybersecurity Framework Subcategories
Internal root CA	Windows 2012 server running AD Certificate Services	Issues and self-signs its own TLS certificate.	PR.AC-1, PR.AC-4, PR.DS-2, PR.DS-3, PR.DS-6, PR.PT-1
Internal issuing CA	Windows 2012 server running AD Certificate Services	Issues TLS certificates to the servers that request them; issuing CAs are subordinate to and certified by the root CA. Manages the life cycle of certificates, including request, issuance, enrollment, publication, maintenance, revocation, and expiration.	PR.AC-1, PR.AC-4, PR.DS-2, PR.DS-3, PR.DS-6, PR.PT-1
Certificate database	Microsoft SQL Server	Database of TLS certificates and keys; for confidentiality, this database is encrypted, and the encryption key is stored in the hardware security module.	PR.AC-4, PR.DS-1
TLS inspection appliance	Symantec SSLV Appliance	Intercepts and inspects network traffic encrypted via TLS.	PR.AC-4, DE.CM-1
HSM	Thales TCT Luna SA 1700	Securely generates, stores, manages, and processes the cryptographic key pair and uses it to sign TLS certificates within a hardened, tamper-resistant physical appliance. Also stores other keys, such as the database encryption key and the TLS certificate keys for the key manager component (Venafi) and the CAs. Can issue signed certificates in response to certificate signing requests (CSRs). Administrative access to this component may be supported by using either password-based or secure shell-based public key authentication.	PR.AC-1, PR.AC-3, PR.AC-4, PR.DS-1, PR.DS-2, PR.DS-3, PR.DS-6, PR.PT-1
External certificate authority	DigiCert External CA	Issues, discovers, installs, inspects, remediates, and renews TLS certificates.	PR.AC-1, PR.AC-4, PR.DS-2, PR.DS-3, PR.DS-6

Component	Product	Functionality	Cybersecurity Framework Subcategories
Load balancer	F5 Networks BIG-IP Local Traffic Manager	Acts as a TLS server and distributes received traffic across a number of other TLS servers.	PR.AC-7, PR.DS-2, PR.PT-5
DevOps framework	Kubernetes	Open-source container orchestration system for automating application deployment, scaling, and management.	PR.PT-5
Automated certificate management frameworks	Jetstack Cert-Manager Certbot	Jetstack Cert-Manager provides automated certificate management for Kubernetes. Certbot is an automated client that enrolls and deploys TLS certificates for web servers by using the ACME protocol.	PR.AC-1, PR.AC-4
TLS servers	Apache Microsoft IIS NGINX	The following TLS server configurations were deployed with a TLS server certificate managed as follows: Microsoft IIS: remote agentless certificate management Microsoft IIS attached to the Thales TCT HSM: remote agentless certificate management Apache: remote agentless certificate management Apache: certificate management via the ACME protocol and certbot client NGINX on Kubernetes: Cert-Manager plug-in for automated certificate management of ingresses.	PR.AC-7, PR.DS-2, PR.PT-5
Application servers	Apache	These systems represented a second tier of internal application servers that were also deployed with TLS server certificates.	PR.AC-7, PR.DS-2, PR.PT-5

4.3.1 Certificate Manager and Internal TLS Certificate Network Scanning Tool

The certificate manager is a key element of the architecture, acting as the primary technology component of an organization's central certificate service. It creates and maintains an inventory of certificates and keys; provides a self-service portal for certificate owners; automates monitoring and remediation; rapidly replaces TLS certificates and keys; enforces TLS certificate and key policy; and enables central oversight, reporting, and auditing.

4.3.1.1 Venafi Trust Protection Platform

Venafi TPP serves as the certificate manager and provides the following certificate management functions:

- establishment and enforcement of TLS server certificate policies
- central inventory of TLS server certificates and private keys
- customer creation of custom metadata fields (e.g., Cost Center, Application ID) associated with certificates and other assets for reporting and accounting
- hierarchical organization of assets (e.g., certificates, applications, devices)
- certificate network scanning (discussed below)
- automated import of certificates from CAs
- onboard discovery of certificates and associated configuration parameters (specifically on F5 BIG-IP Local Traffic Manager [LTM] and Microsoft IIS in the lab)
- separation of duties and least-privilege access through granular access controls—assignable to groups or individuals
- self-service portal for onboarding and certificate management by certificate owners
- automated identification of TLS server certificate vulnerabilities, providing visibility through dashboards, reports, and alerts
- automated monitoring of certificate expiration dates, with configurable time frames for alerts sent prior to expiration
- automated monitoring of certificate operation status
- automated integration with internal and public CAs for certificate enrollment
- automated certificate life-cycle management via remote management connections
- agent-based automated certificate life-cycle management
- standard protocol support, including simple certificate enrollment protocol (SCEP) and ACME
- DevOps framework integration
- cloud platform integration, including Amazon Web Services and Azure

- Representational state transfer (REST)-based application programming interfaces (APIs)
- dual-control enforcement through workflow gates that can be applied at specific steps in the certificate life cycle, and can be assigned to groups and individuals with sufficient knowledge of application context to review and approve certificate requests
- integration with HSMs for private-key security
- integration with identity systems (e.g., Microsoft Active Directory, Lightweight Directory Access Protocol [LDAP] directories)
- central logging of all certificate management operations
- configurable event-based alerts, including delivery via simple mail transfer protocol, syslog, security incident and event management systems, ticketing systems, file, or database
- certificate revocation list (CRL) expiration monitoring to prevent outages caused by expired CRLs
- trust anchor management (e.g., root certificates) on TLS clients that act as relying parties for TLS server certificates
- load balanced architecture to support scalability, fault tolerance, and geographic distribution to support enterprise certificate operations
- Common Criteria certified

4.3.2 Internal TLS Certificate Network Scanning Tool

The internal TLS certificate network scanning tool provides automated discovery of TLS server certificates. It integrates with the certificate manager and enables the Certificate Services team and certificate owners to scrutinize newly discovered certificates for policy compliance and inclusion in the certificated inventory, if desired. An effective strategy for certificate network scanning is to use existing vulnerability scanning tools to pass discovered certificate information to the Certificate Services team. In some cases, organizational or technical constraints require that the Certificate Services team performs network scanning. Because a vulnerability scanning tool was not deployed in the lab, the team used Venafi TPP for certificate network scanning.

4.3.2.1 Venafi TPP for Certificate Network Scanning

Venafi TPP provides two different methods for certificate network scanning: scanning from a Venafi TPP server, and scanning from a command line utility called Scanafi. Both methods were used in the lab: the Venafi TPP server for scanning the data center network zones and Scanafi for scanning the DMZ. The Venafi TPP server provides the following functions for discovering TLS server certificates:

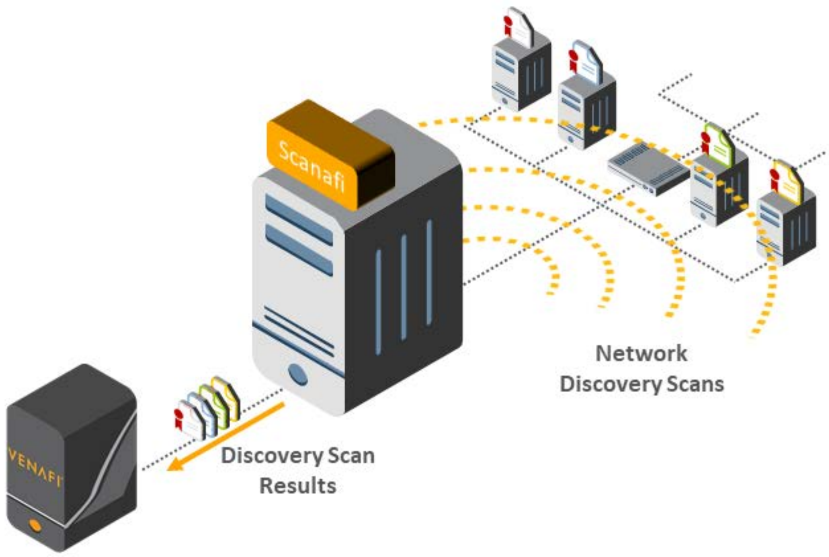
- support for the following as scanning targets:
  - multiple individual internet protocol (IP) addresses or IP ranges
  - multiple host/domain names

    - multiple ports or port ranges
- manual triggering of scans
- scheduled execution of scans, including daily, weekly, monthly, annually
- configuration of blackout periods for scanning
- support for multiple scanning agents
- support for placing scanning agents in distinct network zones (separated by firewalls)
- support for discovering TLS and SSL, including hypertext transfer protocol secure (https), the command `STARTTLS`, secure lightweight directory access protocol (LDAPS), file transfer protocol secure (FTPS), and server name indication (SNI)
- rules-based, automated processing of discovered certificates for placement into the certificate inventory hierarchy to automatically assign to the appropriate certificate owner(s)

Venafi Scanafi provides the following certificate network scanning functionality:

- support for the following as scanning targets:
    - multiple individual IP addresses or IP ranges
    - multiple host/domain names
    - multiple ports or port ranges
- manual triggering of scans (or triggering from a scheduling tool such as cron)
- support for multiple Scanafi agents (e.g., in different network zones)
- REST-based communications to the Venafi TPP server(s) to report scanning results
- support for discovery of TLS and SSL, including https, STARTTLS, LDAPS, FTPS, and SNI
- discovery of enabled TLS/SSL versions and ciphers for vulnerability identification

Figure 4-4 Venafi Scanafi Performing Network Scans and Providing Scan Results to Venafi TPP



4.3.3 Internal Root CA

The architecture includes an internal root CA that issues and self-signs its own TLS certificates for use in the demonstration. The NCCoE built its internal root CA by using a Windows 2012 server running Active Directory Certificate Services (ADCS).

4.3.4 Internal Issuing CA

The architecture also includes an internal issuing CA that issues TLS certificates to the servers that request them. The internal issuing CA is subordinate to and certified by the root CA. It manages the life cycle of certificates, including request, issuance, enrollment, publication, maintenance, revocation, and expiration. Similar to the internal root CA, the TLS team built its internal-issuing CA by using a Windows 2012 server running ADCS.

4.3.5 Certificate Database

The certificate database stores all TLS certificates and keys and associated metadata inventoried by the certificate manager. For confidentiality, private keys and credentials are encrypted in this database, and the encryption key is stored in the HSM.

4.3.5.1 Venafi TPP Database

The Venafi TPP database stores and provides access to the certificate inventory and product configuration data. The functions provided/supported by the Venafi TPP database include:

- storage of TLS server certificates, with the certificate fields' contents (e.g., key length, expiration date, common name) parsed and stored in separate database fields for rapid search
- storage of TLS private keys, encrypted by using an advanced encryption standard symmetric key stored in an HSM (or soft key if preferred)
- storage of TPP configuration data
- support for the following database versions:
  - Microsoft SQL Server 2012 SP2
  - Microsoft SQL Server 2014 SP2
  - Microsoft SQL Server 2016
- support for disaster recovery and high availability across multiple database instances through Microsoft SQL Server AlwaysON Availability Groups

4.3.6 TLS Inspection Appliance

Whether to perform TLS inspection is a policy decision left to each organization. For those organizations that require inspection, a TLS inspection appliance has been demonstrated with traffic that has been encrypted with TLS. The TLS inspection appliance decrypts this traffic, so it can be analyzed and inspected for viruses, malware, or other threats.

4.3.6.1 Symantec SSL Visibility Appliance

The SSLV Appliance inspects encrypted traffic to detect possible attacks. The Symantec device identifies and decrypts all TLS connections and applications across all network ports (even irregular ports). Existing and new security infrastructure can use the decrypted feeds to strengthen detection of and protection against advanced threats. By off-loading process-intensive decryption, the SSL Visibility Appliance also helps improve the overall performance of the organization's network and security infrastructure.

4.3.7 Hardware Security Module

HSMs are specialized devices dedicated to maintaining security of sensitive data throughout its life cycle. They provide tamper-evident and intrusion-resistant protection of critical keys and other secrets and can off-load processing-intensive cryptographic operations. By performing cryptographic operations within the HSM, sensitive data never leaves the secure confines of the hardened device. An HSM can securely generate, store, manage, and process cryptographic key pairs for use with TLS certificates. A CA leverages an HSM to issue signed certificates in response to certificate signing requests, while ensuring the CA signing keys remain safe within the confines of the HSM. In the build architecture, the HSM also stores other keys, such as the certificate database encryption key for the certificate manager component (Venafi).

4.3.7.1 Thales TCT Luna SA 1700 HSM

Thales TCT, formerly SafeNet Assured Technologies (SafeNet AT), is a U.S.-based provider of high-assurance data security solutions with a stated mission to provide innovative solutions to protect the most vital data from the core to the cloud to the field. The company focuses on U.S. government defense, intelligence, and civilian agencies.

The Thales TCT Luna SA for Government is a network-attached HSM with multiple partitions that provide a “many in one” solution to multiple tenants, each with its own security officer management credentials. Depending on security needs, the Luna SA works with or without a secure personal identification number entry device (PED) for controlling management access to the HSM partitions. Utilizing the PED takes the HSM from a FIPS 140-2 Level 2 certified device to Level 3 [12]. The Luna SA also comes in two performance models: the lower performance 1700 and the high-performance 7000 for transaction-intensive use cases.

In addition to the Luna SA, Thales TCT offers Luna G5 for Government, which is a Universal Serial Bus-attached, small form-factor HSM. It is ideal for storing root cryptographic keys in an offline device. The Luna PCI-E for Government is an embedded HSM that can be installed in a server to protect cryptographic keys and accelerate cryptographic operations.

In the TLS Server Certificate Management Project, the Luna SA 1700 for Government was configured with two partitions to protect the keys that secure the Venafi Trust Protection Platform database and the Microsoft IIS root CA private key.

4.3.8 External Certificate Authority

The architecture also includes an external CA.

4.3.8.1 DigiCert External CA

DigiCert is a U.S.-based CA that provides a portfolio of PKI products, including digital certificates (SSL/TLS, Code Signing, Internet of Things [IoT], and more), CA deployment and operation, and tools for CA/PKI management.

DigiCert offers an external CA and management console to operate a deployed CA that is on site or cloud based. This full-service PKI management solution includes configuration of the CA (such as PKI hierarchy, certificate profiles, and revocation checking), certificate life-cycle management, network discovery of certificates, audit logs, and user roles. DigiCert's external CA is operated by the user through the CertCentral console.

CertCentral is a flexible web-based platform for enterprise and small business PKI management. CertCentral supports public and private PKI, and can manage and issue a wide variety of certificate types, including TLS (SSL), Code Signing, Client, Secure/Multipurpose Internet Mail Extensions, and Community standards (including Wi-Fi Alliance and Grid computing). CertCentral also offers a fully functioning API.

Through CertCentral, users can perform all certificate life-cycle operations, including certificate requests, approval/rejection of requests, certificate reissuance, and revocation. Because CertCentral is a centralized tool for certificate issuance and management, organizations can enforce their internal certificate policies and maintain certificates deployed across their networks.

CertCentral includes network scanning tools for identifying certificates installed on a network, regardless of the issuing CA. All discovered certificates are inventoried, and CertCentral will send an alert for expiring certificates and scan for common misconfigurations or security vulnerabilities in the web server and certificate (such as deprecated SSL protocol support or weak encryption ciphers/private keys). By using one tool, network administrators can monitor their PKI operation and receive alerts if problems emerge that can potentially cause network downtime or security risks.

CertCentral supports components of the ACME protocol—an IETF standard for automating issuance, installation, and renewal of SSL/TLS certificates. ACME enables web servers to automatically request and install their certificates, eliminating time-intensive replacement procedures and human error. This facilitates industry best practices such as short-lived certificates (usually 90-day validity or less) and regular key rotation.

An organization's CertCentral account can have as many users as needed, with each one having assigned preset or customizable roles. A user can be limited to what certificates they can request (by certificate type/identity), for which legal organizations/divisions they can make requests, and whether they can approve requests on their own or require an administrator/other approval. This gives users control to issue and manage their own certificates without affecting operations of other divisions within the organization. CertCentral supports two-factor authentication and single sign-on, which are potential requirements for specific roles or users.

Further capabilities and settings of CertCentral are described in the DigiCert Getting Started guide.

4.3.9 Load Balancer

The architecture includes a load balancer that acts as a reverse proxy. It receives client requests at its front end and evenly distributes these requests across a group of back-end TLS servers, which all use the same TLS server certificate and private key.

4.3.9.1 F5 Networks BIG-IP Local Traffic Manager

Businesses depend on applications. Whether the applications help connect businesses to their customers or help employees do their jobs, making these applications available and secure is the main goal. F5 BIG-IP LTM helps enterprises deliver their applications to users in a reliable, secure, and optimized way. It provides the extensibility and flexibility of application services, with the programmability enterprises need to manage their physical, virtual, and cloud infrastructure. With BIG-IP LTM, enterprises can simplify, automate, and customize applications quickly and predictably.

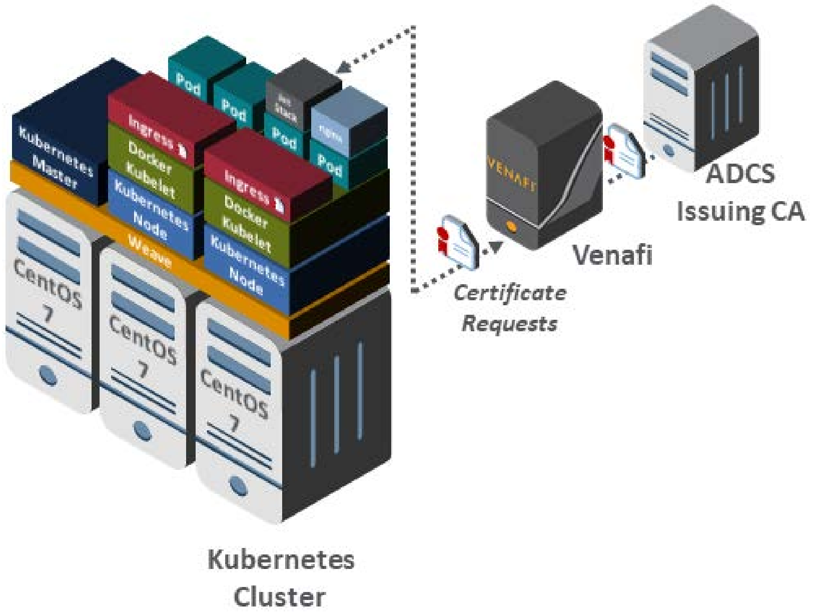
In the example solution architecture, the F5 BIG-IP LTM serves as a load balancer; it acts as a TLS proxy and distributes traffic it receives from external users across a cluster of TLS servers that sit behind it and are serving the same application. To handle traffic securely, each server in the cluster uses the same TLS server certificate and private key. Ideally, copying the keys to each of the servers is not performed manually; rather, automatic copying of private keys can reduce the possibility of a key compromise.

The example solution used in the Venafi TPP certificate manager automatically enrolls and provisions a new certificate to the F5 BIG-IP LTM to automatically replace a certificate on the BIG-IP LTM that was nearing its expiration. It can also configure the LTM’s association with the servers behind it. The Venafi TPP certificate manager was also configured to automatically run a certificate discovery service on the F5 BIG-IP LTM, to identify new certificates and associated configuration parameters.

4.3.10 DevOps Framework

In this phase, the NCCoE undertook a limited DevOps demonstration using a Kubernetes cluster. This limited demonstration included basic DevOps functionality for automated system and application deployment.

Figure 4-5 Example Implementation's DevOps Components Requesting and Receiving Certificates



4.3.10.1 Kubernetes

Kubernetes is an open-source container orchestration system for automating application deployment, scaling, and management. Kubernetes was deployed on three CentOS Linux systems: one acting as the primary, and two nodes.

4.3.11 Automated Certificate Management Frameworks

4.3.11.1 Jetstack Cert-Manager

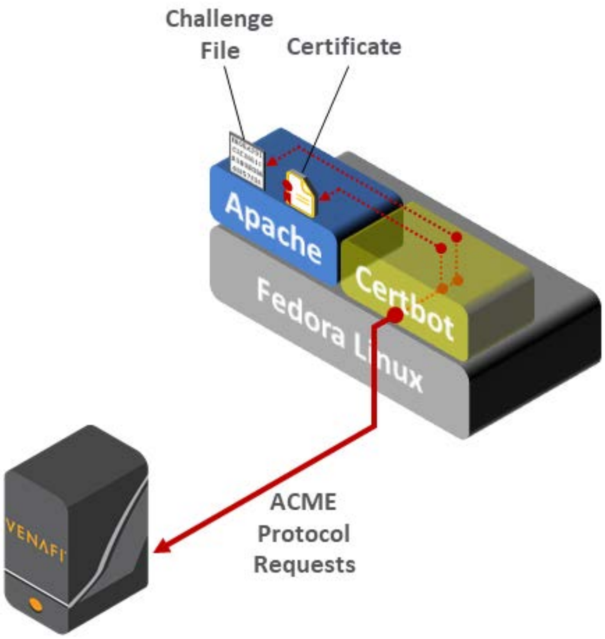
As shown in Figure 4-5, Jetstack Cert-Manager was deployed and configured to automatically manage certificates for ingresses created on the Kubernetes cluster. A Cert-Manager issuer was defined to automatically request certificates from Venafi TPP, so ingress certificates on the Kubernetes cluster were automatically included in the central inventory and tracked (e.g., for expiration).

4.3.11.2 Certbot

Certbot is an open-source automatic client that fetches and deploys TLS certificates for web servers by using the ACME protocol. As shown in Figure 4-6, Certbot was deployed to automate management of certificates on an Apache system in the lab environment.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

Figure 4-6 Certbot Fetching and Deploying TLS Certificates via the ACME Protocol



4.3.12 TLS Servers

The architecture included several TLS servers to demonstrate different methods of certificate management. The certificate management methods used in the example implementation included:

- **Remote Agentless Management:** Many existing "legacy" systems do not support standard protocols for certificate management. Consequently, it is necessary to remotely leverage available interfaces to perform certificate management operations. In this case, the certificate manager must authenticate [10] itself to the system where a certificate is deployed, managed, and used. Once authenticated, it must then execute the necessary operations based on the semantics and syntax required by the system in question. Advantages of this approach include support for automated certificate management when built-in automation is not available, and the ability to centrally and rapidly respond to cryptographic events (e.g., CA compromise), because the certificate manager can proactively connect to each system and manage replacement of affected certificates. Some disadvantages to this approach include that the credentials and access must be granted to the certificate manager system, and integrations must be developed for each distinct type of system.
- **ACME Protocol:** The ACME protocol provides an efficient method for validating that a certificate requester is authorized for the requested domain and to automatically install certificates. This validation is performed by requiring the requester to place a random string (provided by the CA

or certificate manager) on the server for verification via http or in a text record of the server’s Domain Name System (DNS) entry. Client programs such as Certbot can automatically perform all of the operations needed to request a certificate—minimizing the manual work. Let’s Encrypt and several other public CAs support the automated management of public-facing certificates by using the ACME protocol. However, public CAs cannot perform ACME validation for certificates installed on systems inside organizational networks. External entities cannot make http or DNS connections to internal systems. The certificate manager is able to make internal http and DNS connections and can be used for ACME-based certificate management on internal systems. A variety of CAs, certificate managers, and clients across a broad set of TLS servers and operating systems support the ACME protocol, which gives it an advantage. A disadvantage of ACME is that there is no central method for triggering a certificate replacement in response to a certificate event (e.g., CA compromise).

- **DevOps Plug-In:** DevOps frameworks can streamline development and deployment processes through add-on libraries and plug-ins that simplify specific programming tasks. Because certificate management is complex and error prone at times, leveraging certificate management plug-ins in DevOps frameworks increases security while minimizing risk. In this phase of the project, certificate management was implemented by using a plug-in for a single DevOps framework. In future phases, certificate management will be investigated more broadly for DevOps.

4.3.12.1 Microsoft IIS–Remote Agentless Management

Microsoft IIS was deployed on a Windows Server 2012 in the data center network zone. A certificate was manually deployed on IIS to simulate a scenario where existing certificates were deployed. The onboard discovery functionality in Venafi TPP was used to automatically discover the certificate and associated configuration (binding) information. This populated the necessary information for automated certificate management to occur. The certificate was automatically replaced by using Venafi TPP, which used Windows Remote Management to perform the remote certificate management operations.

4.3.12.2 Microsoft IIS with Thales TCT HSM–Remote Agentless Management

Microsoft IIS was deployed on a Windows Server 2012 in the data center secure network zone. The Thales TCT HSM client was installed on the Windows server to make the Thales TCT HSM accessible for cryptographic operations through Windows Cryptographic Application Programming Interface (CAPI) or the next generation Cryptographic API. Configuration information for this IIS system was entered into Venafi TPP, including the address of the Windows system, credentials for authenticating to the Windows system, and information for the certificate needed for the IIS system. Venafi TPP automatically connected to the Windows system, instructed the HSM to generate a new key pair (for which the private key never left the HSM) and CSR, retrieved the CSR, enrolled for a certificate with the issuing CA, and installed the certificate with the necessary binding information for IIS. The https (TLS) connections were confirmed to use the issued certificate, and the corresponding private key was stored in the Thales TCT HSM.

4.3.12.3 Apache–Remote Agentless Management

Apache was deployed on a Fedora Linux system in the DMZ. Configuration information for this Apache system was entered into Venafi TPP, including the address of the Fedora Linux system, credentials for authenticating to the Fedora Linux system, information for the certificate needed for the Apache system, and the location of the privacy enhanced mail files where the certificate and CA chain should be installed. Venafi TPP automatically enrolled for and deployed a certificate to the configured location, so the Apache server could use TLS-secured communications.

4.3.12.4 Apache–ACME Protocol

Apache was deployed on a Fedora Linux system in the DMZ. Certbot was installed on the Fedora Linux system and configured for use with Apache. The ACME server was enabled and configured on Venafi TPP, so Venafi TPP could service ACME protocol requests. Certbot was used to automatically request a certificate from Venafi TPP and install it for use by the Apache web server.

4.3.12.5 NGINX on Kubernetes–DevOps Plug-In

An NGINX deployment and corresponding service were created on the Kubernetes cluster. An ingress was defined to make the NGINX service accessible from outside the Kubernetes cluster. The needed annotation was included in the ingress definition to instruct Cert-Manager to automatically request and install a certificate from Venafi TPP. Once the ingress was enabled, a connection was made to the appropriate address to confirm the certificate from Venafi TPP was successfully installed to secure communications to the NGINX web server.

4.3.13 Application Servers

Most web-based applications include multiple tiers. For example, users of a web-based application may initially connect to a load balancer. The load balancer (tier 1) passes the requests to a web server (tier 2). The web server processes the requests and subsequently makes requests to one or more application servers (tier 3). The application servers process the requests and may read or write to/from a database server (tier 4). Credentials and other confidential information are often passed among adjacent tiers, so each system is typically configured for TLS, including a TLS certificate. The example solution implementation included a load balancer and two web servers in the DMZ. To simulate the existence of application servers, Apache systems were deployed in the data center network zone. NOTE: Apache is not normally used as an application server. However, it was used to minimize complexity of the example implementation. Venafi TPP was used to automatically deploy certificates to the Apache systems acting as application servers.

5 Security Characteristic Analysis

The purpose of the security characteristic analysis is to gauge the extent to which the project meets its objective of demonstrating how the processes for obtaining and maintaining TLS cryptographic certificates can be made less labor-intensive and error prone in medium and large IT enterprises. In addition, it seeks to understand the security benefits and drawbacks of the reference design.

5.1 Assumptions and Limitations

The security characteristic analysis has the following limitations:

- It is neither a comprehensive test of all security components nor a red-team exercise.
- It cannot identify all weaknesses.
- It does not include the lab infrastructure. It is assumed that devices are hardened. Testing these devices would reveal only weaknesses in implementation that would not be relevant to those adopting this reference architecture.

5.2 Functional Capabilities Demonstration

The demonstration shows the extent to which the example solution meets its design goals and stated security requirements.

5.2.1 Definitions

The following definitions apply to terms used in the description of functional capabilities demonstrated.

- discovery–finding new certificates that are not yet known or managed by the certificate management system
- monitoring–maintaining awareness about the status and characteristics of known certificates being managed by the certificate management system, including a determination of whether the certificates conform to policy
- sanctioned certificates–certificates issued by approved CAs
- unsanctioned certificates–certificates issued by CAs that are not approved
- enrolling–creating/issuing a certificate and storing it in the certificate management system inventory
- provisioning–deploying a certificate to a machine; also called *installing*

5.2.2 Functional Capabilities

The following functional TLS server certificate management capabilities were successfully demonstrated in the build phase.

Capability 1: The TLS example implementation demonstrates the ability to **establish a systematized inventory** of certificates (and keys) in use on the network. It enables a user to:

- efficiently **enroll and provision** certificates (and keys) by using:
  - public CA
  - internal CA
  - private key stored in file
  - private key stored in HSM
- store the following **attributes** with certificates in the inventory:
  - subject distinguished name (DN)
  - subject alternative name (SAN)
  - issue date (i.e., notBefore date)
  - expiration date (i.e., notAfter date)
  - issuing CA
  - key length
  - key algorithm (e.g., Rivest, Shamir, and Adleman [RSA], Elliptic Curve Digital Signature Algorithm)
  - signing algorithm
  - validity period (e.g., difference between notBefore and notAfter)
  - key usage flags
  - extended key usage flags
  - installed location(s) of certificate (e.g., IP or DNS address and file path)
  - certificate owner (group responsible for certificate)
  - contacts (the group of individuals that should be notified of issues)
  - approver(s) (parties responsible for reviewing issuance and renewal requests)
  - type of system (e.g., F5 LTM, Microsoft IIS, Apache)

- custom metadata field definition by organizations to associate organizationally relevant information with certificates, such as application identification, cost center, applicable regulations
- use network scanning to **discover certificates** not currently being managed by the inventory, including the ability to:
    - discover TLS server certificates **across different network zones and on a variety of TLS server types** (e.g., load balancer, web server, application server, database, identity services, etc.)
    - **discover and flag unsanctioned certificates** (i.e., certificates not from an approved CA)
        - enroll a new (sanctioned) certificate and provision it to replace the discovered unsanctioned certificate
    - discover and enroll sanctioned certificates
        - end entity (e.g., the TLS server)
        - CA certificate chain certificates (root and intermediate CA certificates)
    - discover the **absence of an expected certificate** from a machine where it should be installed
        - **reprovision** that certificate to that machine from the inventory

Capability 2: The TLS example implementation demonstrates the capability to **maintain the inventory** of TLS certificates (and keys). It enables a user to:

- **enroll (add) new certificates** (and keys) to the inventory and provision them to a network device
- **revoke certificates** that are suspected to be compromised or are no longer needed
- delete certificates and private keys from the machine/HSM where they had been installed
    - private key stored in file
    - private key stored in HSM
- **replace** a given **owner** associated with all certificates when that **person resigns or changes roles**
    - This is ideally handled by associating certificates with groups, so that users can join or leave the group without leaving certificates "orphaned" without an owner. In cases where there is an individual owner for a certificate, the individual's management chain should be included in the group, or Certificate Services or an incident response team should be included to ensure that expiration and other alerts do not go unaddressed.

Capability 3: The TLS example implementation demonstrates the capability to **automatically enroll and provision** a new certificate and **automatically replace a certificate** that is **nearing expiration** on the following systems:

- F5 BIG-IP LTM: The TLS example implementation demonstrates the capability to install and replace a TLS certificate on a load balancer and configure the association with the applicable virtual server.
- Apache with Agentless Management: The implementation demonstrates automated management of certificates on an Apache web server by using a remotely initiated connection.
- Microsoft IIS with Agentless Management: The implementation demonstrates automated management of certificates on a Microsoft IIS web server by using a remotely initiated connection.
- Apache with ACME Protocol: The implementation demonstrates automated certificate management on an Apache web server by using the ACME protocol.
- Kubernetes: The implementation demonstrates automated installation and replacement before expiration of certificates on ingresses defined to allow access to services within Kubernetes.

Capability 4: The TLS example implementation demonstrates the capability to **continuously monitor** the TLS certificates (and keys) managed by the inventory system and to act upon the status of any certificate (e.g., report the status or replace a certificate as needed). The implementation should support these capabilities:

- Enroll and provision a new certificate to **replace** one that is found to **not conform to policy.**
- **Send weekly or monthly expiration reports** to certificate owners showing all of their certificates that are set to expire (e.g., within the next 90 or 120 days).
- Send **notifications** to owners regarding certificates that are **due to expire** within a near term (e.g., 30 days).
- **Send escalation notifications** to managers or incident response if a certificate has not been replaced within a short time of expiration (e.g., 15 days).
- **Enroll and provision new certificates** as existing certificates approach expiration.
  - manual request
  - standardized automated certificate installation

Capability 5: The TLS example implementation demonstrates the disaster recovery capability to **quickly replace a large number of certificates** located across multiple networks and on a variety of server types, because the certificates are no longer trusted. It is able to replace:

- all certificates issued by a given CA
  - This mimics the situation in which a large number of certificates are no longer trusted, because the CA that issued them has been compromised or become untrusted.
- all certificates with associated keys that are dependent on a specific cryptographic algorithm

    - This mimics the situation in which a large number of certificates are no longer trusted, because the algorithm on which they depend is no longer considered secure.
- all certificates with associated keys generated by the faulty cryptographic library after a specific date
    - This mimics the situation where large numbers of certificates are no longer trusted, because the keys associated with them were generated by a faulty cryptographic library after a bug was introduced into that library.
- the ability to track and report on replacement of large numbers of certificates, to monitor the progress of replacement and risk reduction

Capability 6: The TLS example implementation demonstrates the capability to perform **passive, out-of-line decryption** on TLS communications. The demonstration includes the following capabilities:

- verification the decrypted data matches the tapped, TLS-encrypted data
- ability to use the certificate management system to securely transfer private keys from several different TLS servers to the TLS inspection appliance
- ability to use the certificate management system to securely replace expiring keys on servers and immediately copy these to the inspection appliance before expiration
    - manually
    - via standardized automated certificate installation

Capability 7: The TLS example implementation demonstrates the capability to **log all certificate and private-key management operations**, including logging:

- certificate creation
- certificate installation
- certificate revocation
- key pair generation
- certificate requests
- certificate request approvals
- copying certificates and keys
- certificate and key replacement

5.2.3 Mapping to NIST SP 1800-16B Recommendations

The following table provides a mapping between the recommended policy requirements in Volume B of this practice guide (NIST SP 1800-16B) and the example implementation in the TLS Certificate Management lab.

Table 5-1 Mapping Between Volume B Policy Recommendations and the Example Implementation

1800-16B Recommended Requirement	Implementation in TLS Certificate Management Lab
Inventory	Venafi TPP was used to maintain an inventory of all certificates, including metadata fields associated with each certificate for tracking relevant information such as key length, signing algorithm, and installed locations. To create a comprehensive inventory of existing certificates, two Venafi TPP functions were used: 1) CA import, to retrieve all issued certificates from the Microsoft CA, and 2) network discovery, to discover all deployed certificates, including certificates that may have been issued by other CAs. Network discovery added location information for each certificate previously imported from the CA.
Ownership	Venafi TPP was used to track owners for certificates. In Venafi TPP, it is possible to assign individuals or groups as owners of each certificate. It is also possible to assign (individual or group) owners to groups of certificates by associating the owner to a folder, which applies the ownership to all certificates within the folder.
Approved CAs	The Venafi TPP dashboard was used to identify discovered certificates issued from unapproved CAs. These certificates were replaced with certificates from approved CAs by using Venafi TPP.
Validity Periods	The Venafi TPP dashboard was used to identify discovered certificates with a validity period longer than allowed (e.g., a three-year versus one-year validity period). These certificates were replaced with certificates with shorter, allowed validity periods by using Venafi TPP.
Key Length	The Venafi TPP dashboard was used to identify discovered certificates that contained keys smaller than allowed (e.g., 1024 bits versus 2048 bits). These

1800-16B Recommended Requirement	Implementation in TLS Certificate Management Lab
	certificates were replaced with certificates containing longer, allowed key lengths by using Venafi TPP.
Signing Algorithms	The Venafi TPP dashboard was used to identify discovered certificates signed with noncompliant algorithms (e.g., secure hash algorithm 1 [SHA-1]). These certificates were replaced with certificates that had been signed with compliant algorithms by using Venafi TPP.
Subject DN and SAN	Venafi TPP was configured to allow only certain domain names through inclusion on a domain allowlist. Workflow gates were implemented in Venafi TPP to ensure that Subject DNs and SANs in all certificate requests were reviewed and approved prior to issuance by the CA.
Certificate Request Reviews (Registration Authority)	Workflow gates were configured in Venafi TPP, requiring that certificates be reviewed prior to new issuance or renewal. Individuals/groups were assigned as approvers for groups of certificates via Venafi TPP folders.
Private-Key Security	The Thales TCT HSM and Venafi TPP were used to secure private keys. Thales TCT HSM and Venafi TPP: A Microsoft IIS server was connected to the Thales TCT HSM across the network, so the private key used with the TLS server certificate on the IIS server could be stored and used within the HSM for a high level of security. Venafi TPP was used to manage generation of the key pair on the HSM. Venafi TPP: Automated management was used on several systems to remove the need for people to access private keys (which they do when manually managing TLS certificates).
Rotation upon Reassignment/ Termination	Venafi TPP was used create an up-to-date inventory, including tracking owners for all certificates. In case a certificate owner were reassigned or terminated, all certificates to which the person had management responsibility could be quickly identified. In addition to the ability to identify the certificates impacted by a reassignment or termination so they could be rotated, Venafi TPP and the Thales TCT HSM were leveraged to minimize the need to rotate on reassignment. Venafi TPP was used to automate management of certificates and private keys, so that certificate owners did not require direct access to private keys, thereby removing the need to rotate certificates and private keys on reassignment or termination. On one system, additional steps were taken to protect private keys by leveraging the Thales TCT HSM for protection of the private keys. The HSM prevents direct access to private keys, thereby removing the need to replace on reassignment.
Proactive Certificate Renewal	Venafi TPP was leveraged to monitor expiration dates of all certificates and send reports and alerts to certificate owners prior to expiration. Venafi TPP

1800-16B Recommended Requirement	Implementation in TLS Certificate Management Lab
	sent certificate expiration reports weekly showing all certificates expiring within the next 60 days, so certificate owners could proactively plan required replacements. Notification rules were configured in Venafi TPP, so alerts would be sent out if a certificate were within 20 days of expiring.
Crypto-Agility	Venafi TPP was used to establish an inventory of all certificates, so that in case of a large-scale cryptographic event (e.g., CA compromise, vulnerable cryptographic algorithm, or cryptographic library bug), all affected certificates and private keys could be quickly identified and replaced. Automation was configured on multiple systems to enable replacement of certificates and private keys to be completed quickly. In addition, Venafi TPP network validation was configured to automatically confirm the current status of all certificates, so the progress of replacement could be tracked.
Revocation	A workflow gate was configured in Venafi TPP to require review of revocation requests, so a certificate was not accidentally or maliciously revoked, which would cause an outage to the application dependent on the certificate. Permissions to request revocation were limited to certificate owners (for their own certificates) and administrative staff.
Continuous Monitoring	Venafi TPP was leveraged to perform the following to continuously monitor certificates: Network discovery scans were automatically performed on a periodic basis. Alerts were sent when new (previously unknown) certificates were detected. Venafi TPP network validation was configured to automatically check the operational status of all certificates. Onboard discovery was configured to automatically run periodically on the F5 LTM to discover new certificates.
Logging of Certificate Management Operations	Venafi TPP automatically logged all 1) administrative operations performed within the Aperture and WebAdmin consoles (e.g., new certificates, approvals, revocation requests), 2) API operations that made changes to configuration or data, 3) automated certificate management operations performed by Venafi TPP.
TLS Traffic Monitoring	The Symantec SSLV was deployed and configured to monitor all traffic on the data center and internal DMZ network zones. Private keys used for TLS certificates from the several TLS servers in those zones were automatically provisioned by Venafi TPP to the Symantec SSLV. When certificates on those servers were renewed, the new private keys were automatically provisioned to the SSLV.

5.3 Scenarios and Findings

One aspect of our security evaluation involved assessing how well the reference design addresses the security characteristics it was intended to support. The Cybersecurity Framework Subcategories were used to provide structure to the security assessment by consulting the specific sections of each standard cited in reference to a Subcategory. The cited sections provide validation points that the example solution would be expected to exhibit. Using the Cybersecurity Framework Subcategories as a basis for organizing our analysis allowed us to systematically consider how well the reference design supports the intended security characteristics.

5.3.1 Demonstration Scenario

The demonstration scenario starts with an organization that has deployed and currently uses TLS certificates across multiple groups and applications. In the scenario, an organization encounters the challenges described in Section 3. The approach followed to address the issues associated with life-cycle management of the certificates included the following phases:

- **Establish Governance:** The project team defined a set of certificate management policies based NIST guidance documents regarding how to establish consistent governance of TLS certificates.
- **Create and Maintain an Inventory:** A central team provided automated discovery services to certificate owners to establish a complete inventory of all TLS server certificates. The organization leveraged configurable rules to automatically organize discovered certificates and associate owners to enable automated notifications.
- **Register for and Install Certificates:** As new certificates were needed or existing certificates approached expiration, certificates were requested and installed. Because enterprise environments are diverse and have varying technical and organizational constraints, several methods for requesting and installing certificates were demonstrated. These included:
    - *Manual:* Security, operational, or technical requirements/constraints mandate that the server's system administrator manually requests a certificate by using command line tools and a certificate management system portal.
    - *Standardized Automated Certificate Installation:* A TLS server is configured to automatically request and install a certificate by using a protocol, such as IETF's ACME protocol.
    - *Installation Using Proprietary Method:* The certificate management system uses a method that is proprietary to the TLS server, to perform the operations needed to install certificates on one or more systems that do not support a standard automated method for requesting and installing certificates.
    - *DevOps-Based Installation:* A DevOps framework used to install and configure servers/applications is also used to request and install certificates. This was done in a cloud environment—where DevOps frameworks are most commonly used.

- *Management of Private Keys Stored in an HSM:* The majority of private keys used with certificates are stored in files; however, HSMs increase the security of private keys. One or more of the methods listed above was performed on a system that uses an HSM for private-key protection.

- **Continuously Monitor and Manage:** The inventory of certificates was monitored for expiration, proper operation, and security issues. Notifications and alerts were triggered when certificates were nearing expiration or anomalies were detected. Management operations were performed to ensure proper operation and security.
- **Detect, Respond, and Recover from Incidents:** Simulated situations, such as a CA compromise and broken algorithms, were demonstrated (i.e., cryptographic library bug that created weak keys for certificates). A large number of organizational certificates needed to be rapidly replaced. The certificate management system orchestrated replacement of all certificates.

5.3.2 Findings

It is possible to deploy and configure a certificate management service and integrate it with ancillary components and services in such a way that the system

- establishes a TLS server certificate inventory by supporting functions such as certificate (and key) discovery, enrollment, provisioning, and revocation
- supports automatic enrollment and provisioning of new certificates
- supports automatic replacement of certificates nearing expiration
- discovers and monitors certificates and sends alerts as required to help avoid having certificates expire while they are still in use
- continuously monitors certificates to ensure their validity
- can quickly identify and replace a large number of certificates that share a common characteristic (e.g., they were all generated by a faulty cryptographic library) that may cause them to become untrusted
- can enroll and provision new certificates as well as automatically replace certificates that are nearing expiration on various types of systems, including Microsoft IIS and Apache web servers, application servers, load balancers, TLS proxies, and DevOps frameworks
- can perform certificate management via various types of mechanisms, including remote agentless management, the ACME protocol, and a DevOps certificate management plug-in
- can use an HSM to generate, store, manage, and process cryptographic key pairs for use with TLS server certificates and use these keys within the HSM to issue signed certificates in response to certificate signing requests
- can use an HSM to store and protect additional keys, such as the symmetric keys that secure sensitive data in the certificate database

- can efficiently and automatically copy private keys from servers to inspection appliances to enable inspection of traffic within encrypted TLS connections if desired
- can log all certificate and private-key management operations

Passive inspection of VMware vSphere workloads by using a remote physical monitoring appliance is challenging. Within the TLS lab deployment, passive decryption monitoring was deployed. This required that network packets captured within VMware vSphere workloads be forwarded to a physical remote monitoring appliance. The packet had to traverse the switch fabric between the VMware ESXi cluster and the physical remote monitoring appliance. VMware standard switches will monitor only east–west traffic locally in a standard switched port analyzer (SPAN) port configuration. VMware needs additional configuration to its virtual distributed switch configurations to support SPAN or mirroring ports. This method is discussed in more detail in Appendix A of Volume D.

There is an additional challenge with passive decryption of TLS traffic. TLS 1.3 prohibits use of the RSA algorithm, requiring use of ephemeral Diffie-Hellman instead. TLS passive inspection is not possible when ephemeral Diffie-Hellman is used. As a result, organizations must continue to use TLS 1.2 or earlier versions to perform TLS passive inspection of traffic on their internal networks. TLS passive inspection is possible with TLS 1.2 and earlier versions because the RSA algorithm is supported for key exchange.

6 Future Build Considerations

The expanding use of cloud environments and DevOps methodologies/tools, and reliance on TLS to secure communications necessitates implementation of sound TLS server certificate management methodologies. Future builds will focus on strategies for effectively managing TLS server certificates for cloud and DevOps, including strategies for adapting management methodologies as cloud environment and DevOps methodologies/tools continue to rapidly evolve and change. Future builds will look at strategies for managing TLS server certificates in individual cloud implementations, as well as implementations where multiple cloud environments are used or those requiring the ability to move implementation between clouds. For DevOps, we will investigate commonalities and differences for TLS server certificate management between the various types of DevOps methodologies and tools.

We have also received suggestions that we should investigate TLS server certificate management recommended best practices in the context of company acquisitions and divestitures, as well as investigate providing more detail regarding what certificate management aspects to audit against.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

Appendix A List of Acronyms

ACME	Automated Certificate Management Environment
AD	Active Directory
ADCS	Active Directory Certificate Services
API	Application Programming Interface
CA	Certificate Authority
CAPI	Cryptographic Application Programming Interface (also known variously as CryptoAPI, Microsoft Cryptography API, MS-CAPI, or simply CAPI)
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DevOps	Development Operations
DMZ	Demilitarized Zone
DN	Distinguished Name
DNS	Domain Name System
FIPS	Federal Information Processing Standards
FTPS	File Transfer Protocol Secure
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IIS	Internet Information Server (Microsoft Windows)
IoT	Internet of Things
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
LTM	Local Traffic Manager (F5)
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
PED	Personal Information Number Entry Device
PKI	Public Key Infrastructure
POP	Post Office Protocol
REST	Representational State Transfer (API)
RMF	Risk Management Framework

RSA	Rivest, Shamir, and Adleman (public key encryption algorithm)
Thales TCT	Thales Trusted Cyber Technologies
SAN	Subject Alternative Name
SCEP	Simple Certificate Enrollment Protocol
SHA-1	Secure Hash Algorithm 1
SNI	Server Name Indication
SP	Special Publication
SPAN	Switched Port Analyzer
SQL	Structured Query Language
SSL	Secure Socket Layer (protocol)
TLS	Transport Layer Security (protocol)
TPP	Trust Protection Platform (Venafi)
URL	Uniform Resource Locator

This publication is available free of charge from: http://doi.org/10.6028/NIST.SP.1800-16.

Appendix B Glossary

Active Directory

A Microsoft directory service for management of identities in Windows domain networks.

Application

1. The system, functional area, or problem to which information technology is applied. The application includes related manual procedures as well as automated procedures. Payroll, accounting, and management information systems are examples of applications. (National Institute of Standards and Technology [NIST] Special Publication [SP] 800-16).

2. A software program hosted by an information system (NIST SP 800-137).

Application Programming Interface (API)

A system access point or library function that has a well-defined syntax and is accessible from application programs or user code to provide well-defined functionality. (NIST Interagency/Internal Report [IR] 5153)

Authentication

Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to a system's resources. (NIST SP 800-63-3)

Automated Certificate Management Environment

A protocol defined in Internet Engineering Task Force (IETF) Request for Comments (RFC) 8555 that provides automated enrollment of certificates.

Certificate

A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. Additional information in the certificate could specify how the key is used and its validity period. (NIST SP 800-57 Part 1 Revision 4 [1] under Public-Key Certificate) (Certificates in this practice guide are based on IETF RFC 5280).

Certificate Authority (CA)

A trusted entity that issues and revokes public key certificates. (NISTIR 8149)

Certificate Authority Authorization

A record associated with a Domain Name Server (DNS) entry that specifies the CAs authorized to issue certificates for that domain.

Certificate Chain

An ordered list of certificates that starts with an end-entity certificate, includes one or more CA certificates, and ends with the end-entity certificate's root CA certificate, where each certificate in the chain is the certificate of the CA that issued the previous certificate. By ascertaining whether each certificate in the chain was issued by a trusted CA, the receiver of an end-user certificate can determine if it should trust the end-entity certificate, by verifying the signatures in the chain of certificates.

Certificate Management

Process whereby certificates (as defined above) are generated, stored, protected, transferred, loaded, used, and destroyed (Committee on National Security Systems Instruction [CNSSI] 4009-2015) (In the context of this practice guide, it also includes inventory, monitoring, enrolling, installing, and revoking).

Certificate Revocation List

A list of digital certificates revoked by an issuing CA before their scheduled expiration date and should no longer be trusted.

Certificate Signing Request (CSR)

A request sent from a certificate requester to a CA to apply for a digital identity certificate. The certificate signing request contains the public key as well as other information to be included in the certificate and is signed by the private key corresponding to the public key.

Certificate Transparency

A framework for publicly logging the existence of Transport Layer Security (TLS) certificates as they are issued or observed, in a manner that allows anyone to audit CA activity and notice the issuance of suspect certificates, as well as to audit the certificate logs themselves (experimental RFC 6962).

Chief Information Officer

An organization's official who is responsible for (i) providing advice and other assistance to the head of the organization and to other senior management personnel to ensure that information technology (IT) is acquired and that information resources are managed in a manner consistent with laws, directives, policies, regulations, and priorities established by the head of the organization, (ii) developing, maintaining, and facilitating implementation of a sound and integrated IT architecture for the organization, and (iii) promoting the effective and efficient design and operation of all major information resources management processes for the organization, including improvements to work processes of the organization (NIST SP 800-53 Revision 4 adapted).

Note: A subordinate organization may assign a chief information officer to denote an individual filling a position with security responsibilities with respect to the subordinate organization that are similar to those the chief information officer fills for the organization to which they are subordinate.

Client

1. A machine or software application that accesses a cloud over a network connection, perhaps on behalf of a consumer. (NIST SP 800-146)

2. A function that uses the public key infrastructure (PKI) to obtain certificates and validate certificates and signatures. Client functions are present in CAs and end entities. Client functions may also be present in entities that are not certificate holders. That is, a system or user that verifies signatures and validation paths is a client, even if it does not hold a certificate itself. (NIST SP 800-15)

Cloud Computing

A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (NIST SP 800-145)

Common Name

An attribute type commonly found within a subject distinguished name in an X.500 directory information tree. When identifying machines, it is composed of a fully qualified domain name or internet protocol (IP) address.

Configuration Management

A collection of activities focused on establishing and maintaining the integrity of IT products and information systems through control of processes for initializing, changing, and monitoring the configurations of those products and systems throughout the system development life cycle. (NIST SP 800-53 Revision 4)

Container

A method for packaging and securely running an application within an application virtualization environment. Also known as an application container or a server application container. (NIST SP 800-190)

Cryptographic Application Programming Interface (CAPI)

An API included with Microsoft Windows operating systems that provides services to enable developers to secure Windows-based applications by using cryptography. While providing a consistent API for applications, CAPI allows specialized cryptographic modules (cryptographic service providers) to be provided by third parties, such as hardware security module (HSM) manufacturers. This enables applications to leverage the additional security of HSMs while using the same APIs they use to access built-in Windows cryptographic service providers (also known variously as CryptoAPI, Microsoft Cryptography API, MS-CAPI, or simply CAPI).

Cryptography API: Next Generation

The long-term replacement for CAPI.

Demilitarized Zone

A perimeter network or screened subnet separating a more-trusted internal network from a less-trusted external network.

Development Operations (DevOps)

A set of practices for automating the processes between software development and IT operations teams so that they can build, test, and release software faster and more reliably. The goal is to shorten the systems development life cycle and improve reliability while delivering features, fixes, and updates frequently in close alignment with business objectives.

Digital Certificate

Certificate (as defined above).

Digital Signature

The result of a cryptographic transformation of data that, when properly implemented, provides origin authentication, assurance of data integrity, and signatory nonrepudiation. (NIST SP 800-133)

Digital Signature Algorithm

One of the Federal Information Processing Standards (FIPS) for digital signatures based on the mathematical concept of modular exponentiations and the discrete logarithm problem. (FIPS 186-4)

Directory Service

A distributed database service capable of storing information, such as certificates and certificate revocation lists, in various nodes or servers distributed across a network (NIST SP 800-15) (In the context of this practice guide, a directory services stores identity information and enables authentication and identification of people and machines.)

Distinguished Name

An identifier that uniquely represents an object in the X.500 directory information tree. (RFC 4949 Version 2)

Domain

A distinct group of computers under a central administration or authority.

Domain Name

A name owned by a person or organization and consisting of an alphabetical or alphanumeric sequence, followed by a suffix indicating a top-level domain; used as an internet address to identify the location of web pages.

Domain Name Server

The internet’s equivalent of a phone book. It maintains a directory of domain names, as defined by the DNS, and translates them to IP addresses.

Domain Name System (DNS)

The system by which internet domain names and addresses are tracked and regulated as defined by IETF RFC 1034 and other related RFCs.

Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm specified in ANSI X9.62 and approved in FIPS 186.

Enrollment
The process a CA uses to create a certificate for a web server or email user (NISTIR 7682) (In the context of this practice guide, enrollment applies to the process of a certificate requester requesting a certificate, the CA issuing the certificate, and the requester retrieving the issued certificate).

Extended Validation Certificate
A certificate used for https websites and software that includes identity information subjected to an identity verification process standardized by the CA Browser Forum in its Baseline Requirements that verifies the identified owner of the website for which the certificate has been issued has exclusive rights to use the domain; exists legally, operationally, and physically; and has authorized issuance of the certificate.

Federal Information Processing Standards
A standard for adoption and use by federal departments and agencies that has been developed within the Information Technology Laboratory and published by the National Institute of Standards and Technology, a part of the U.S. Department of Commerce. A FIPS covers some topic in IT to achieve a common level of quality or some level of interoperability. (NIST SP 800-161)

Hardware Security Module
A physical computing device that provides tamper-evident and intrusion-resistant safeguarding and management of digital keys and other secrets, as well as crypto-processing. FIPS 140-2 specifies requirements for HSMs.

Host Name
Host names are most commonly defined and used in the context of DNS. The host name of a system typically refers to the fully qualified DNS domain name of that system.

Hypertext Transfer Protocol (HTTP)
A standard method for communication between clients and web servers. (NISTIR 7387)

Internet Engineering Task Force
The internet standards organization made up of network designers, operators, vendors, and researchers that defines protocol standards (e.g., IP, transmission control protocol, DNS) through processes of collaboration and consensus.

Internet Message Access Protocol
A method of communication used to read electronic mail stored in a remote server. (NISTIR 7387)

Internet of Things (IoT)
As used in this publication, user or industrial devices connected to the internet. IoT devices include sensors, controllers, and household appliances.

Internet Protocol

The internet protocol, as defined in IETF RFC 6864, is the principal communications protocol in the IETF internet protocol suite for specifying system address information when relaying datagrams across network boundaries.

Lightweight Directory Access Protocol (LDAP)

In this document, LDAP refers to the protocol defined by RFC 1777, which is also known as LDAP V2. LDAP V2 describes unauthenticated retrieval mechanisms. (NIST SP 800-15)

Microservice

A set of containers that work together to compose an application. (NIST SP 800-190)

Organization

An entity of any size, complexity, or positioning within an organizational structure (e.g., a federal agency or, as appropriate, any of its operational elements). (NIST SP 800-39) This publication is intended to provide recommendations for organizations that manage their own networks (e.g., that have a chief information officer).

Outage

A period when a service or an application is not available or when equipment is not operational.

Payment Card Industry Data Security Standard

An information security standard, administered by the Payment Card Industry Security Standards Council, for organizations that handle branded credit cards from the major card schemes.

Personal Information Number Entry Device

An electronic device used in a debit-, credit-, or smart card-based transaction to accept and encrypt the cardholder's personal identification number.

Pivoting

A process where an attacker uses one compromised system to move to another system within an organization.

Post Office Protocol (POP)

A mailbox access protocol defined by IETF RFC 1939. POP is one of the most commonly used mailbox access protocols. (NIST SP 800-45 Version 2)

Private Key

The secret part of an asymmetric key pair that is used to digitally sign or decrypt data. (NIST SP 800-63-3)

Public CA

A trusted third party that issues certificates as defined in IETF RFC 5280. A CA is considered public if its root certificate is included in browsers and other applications by the developers of those browsers and applications. The CA/Browser Forum defines the requirements that public CAs must follow in their operations.

Public Key

The public part of an asymmetric key pair that is used to verify signatures or encrypt data. (NIST SP 800-63-3)

Public Key Cryptography Cryptography that uses separate keys for encryption and decryption; also known as asymmetric cryptography. (NIST SP 800-77)

Public Key Infrastructure (PKI) The framework and services that provide generation, production, distribution, control, accounting, and destruction of public key certificates. Components include the personnel, policies, processes, server platforms, software, and workstations used for administering certificates and public-private key pairs, including the ability to issue, maintain, recover, and revoke public key certificates. (NIST SP 800-53 Revision 4)

Registration Authority (RA) An entity authorized by the CA system to collect, verify, and submit information provided by potential subscribers that is to be entered into public key certificates. The term RA refers to hardware, software, and individuals that collectively perform this function. (CNSSI 4009-2015)

Rekey To change the value of a cryptographic key being used in a cryptographic system application; this normally entails issuing a new certificate on the new public key. (NIST SP 800-32 under Rekey) (a certificate)

Renew The act or process of extending the validity of the data binding asserted by a public key certificate by issuing a new certificate (NIST SP 800-32). (The new certificate is typically used to replace the existing certificate, and both certificates typically contain the same subject domain name and subject alternative name information. It is a best practice to generate a new key pair and CSR, i.e., rekey, when renewing a certificate, but re-keying is not required by all CAs. Renewal is typically driven by expiration of the existing certificate but could also be triggered by a suspected private-key compromise or other event requiring the existing certificate to be revoked.)

Replace The process of installing a new certificate and removing an existing one, so that the new certificate is used in place of the existing certificate on all systems where the existing certificate is being used.

Representational State Transfer A software architectural style that defines a common method for defining APIs for web services.

Risk Management Framework The Risk Management Framework, presented in NIST SP 800-37, provides a disciplined and structured process that integrates information security and risk management activities into the system development life cycle. (NIST SP 800-82 Revision 2)

Rivest, Shamir, and Adleman

An algorithm approved in FIPS 186 for digital signatures and in NIST SP 800-56B for key establishment. (NIST SP 800-57 Part 1 Revision 4)

Root Certificate

A self-signed certificate, as defined by IETF RFC 5280, issued by a root CA. A root certificate is typically securely installed on systems, so they can verify end-entity certificates they receive.

Root Certificate Authority

In a hierarchical PKI, the CA whose public key serves as the most trusted datum (i.e., the beginning of trust paths) for a security domain. (NIST SP 800-32)

Rotate

The process of renewing a certificate in conjunction with a rekey, followed by the process of replacing the existing certificate with the new certificate.

Secure Hash Algorithm 1

A hash function specified in FIPS 180-2, the Secure Hash Standard. (NIST SP 800-89)

Secure Hash Algorithm 256

A hash algorithm that can be used to generate digests of messages. The digests are used to detect whether messages have been changed since the digests were generated. (FIPS 180-4)

Secure Transport

Transfer of information by using a transport layer protocol that provides security between applications communicating over an IP network.

Server

A computer or device on a network that manages network resources. Examples include file servers (to store files), print servers (to manage one or more printers), network servers (to manage network traffic), and database servers (to process database queries). (NIST SP 800-47)

Service Provider

A provider of basic services or value-added services for operation of a network; generally refers to public carriers and other commercial enterprises. (NISTIR 4734)

Simple Certificate Enrollment Protocol (SCEP)

A protocol defined in an IETF internet draft specification that is used by numerous manufacturers of network equipment and software that are developing simplified means of handling certificates for large-scale implementation to everyday users, as well as referenced in other industry standards.

Simple Mail Transfer Protocol

The primary protocol used to transfer electronic mail messages on the internet. (NISTIR 7387)

Special Publication

A type of publication issued by NIST. Specifically, the Special Publication 800 series reports on the Information Technology Laboratory's research, guidelines, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations. The 1800 series reports the results of National Cybersecurity Center of Excellence demonstration projects.

Subject Alternative Name A field in an X.509 certificate that identifies one or more fully qualified domain names, IP addresses, email addresses, uniform resource identifiers, or user principal names to be associated with the public key contained in a certificate.

System Administrator Individual responsible for installation and maintenance of an information system, providing effective information system utilization, adequate security parameters, and sound implementation of established information assurance policy and procedures. (CNSSI 4009-2015)

Team A number of persons associated together in work or activity (Merriam-Webster). As used in this publication, a team is a group of individuals that has been assigned by an organization's management the responsibility to carry out a defined function or set of defined functions. Designations for teams as used in this publication are simply descriptive. Different organizations may have different designations for teams that carry out the functions described herein.

Transport Layer Security (TLS) An authentication and security protocol widely implemented in browsers and web servers. TLS is defined by RFC 5246 and RFC 8446.

Trust Protection Platform The Venafi Machine Identity Protection platform used in the example implementation described in this practice guide.

User Principal Name In Windows Active Directory, this is the name of a system user in email address format, i.e., a concatenation of user name, the "@" symbol, and domain name.

Validation The process of determining that an object or process is acceptable according to a predefined set of tests and the results of those tests. (NIST SP 800-152)

Web Browser A software program that allows a user to locate, access, and display web pages.

Appendix C References

[1] E. Barker, *Recommendation for Key Management: Part 1: General*, NIST SP 800-57 Part 1, Revision 4, Gaithersburg, Md., Jan. 2016. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.

[2] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.*3, Internet Engineering Task Force, Apr. 2006. Available: https://www.ietf.org/rfc/rfc4346.txt.

[3] Executive Office of the President, Office of Management and Budget (OMB), *Managing Federal Information as a Strategic Resource*, OMB Circular A-130, July 28, 2016. Available: https://www.federalregister.gov/documents/2016/07/28/2016-17872/revision-of-omb-circular-no-a-130-managing-information-as-a-strategic-resource.

[4] Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1, NIST, Gaithersburg, Md., Apr. 16, 2018. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf.

[5] Joint Task Force Transformation Initiative, *Guide for Conducting Risk Assessments*, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30 Revision 1, Gaithersburg, Md., Sept. 2012. Available: https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final.

[6] Joint Task Force Transformation Initiative, *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*, NIST SP 800-37 Revision 2, Gaithersburg, Md., Dec. 2018. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf.

[7] Joint Task Force Transformation Initiative, *Security and Privacy Controls for Information Systems and Organizations*, Draft NIST SP 800-53 Revision 5, Gaithersburg, Md., Aug. 2017. Available: https://csrc.nist.gov/CSRC/media//Publications/sp/800-53/rev-5/draft/documents/sp800-53r5-draft.pdf.

[8] M. Georgiev et al., "The most dangerous code in the world: validating SSL certificates in non-browser software," *Proceedings of the 2012 ACM conference on Computer and Communications Security*, 2012, pp. 38–49. Available: http://doi.acm.org/10.1145/2382196.2382204.

[9] NIST Computer Security Resource Center Risk Management Framework guidance [Website]. Available: https://csrc.nist.gov/projects/risk-management/risk-management-framework-quick-start-guides.

[10] P. Grassi et al., *Digital Identity Guidelines*, NIST SP 800-63-3, Gaithersburg, Md., June 2017. Available: https://csrc.nist.gov/publications/detail/sp/800-63/3/final.

[11] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, Request for Comments 5246, Internet Engineering Task Force, Aug. 2008. Available: https://www.ietf.org/rfc/rfc5246.txt.

[12] U.S. Department of Commerce, *Security Requirements for Cryptographic Modules*, FIPS Publication 140-2, (including change notices as of Dec. 3, 2002), May 2001. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf.

[13] U.S. Department of Commerce*, Standards for Security Categorization of Federal Information and Information Systems*, FIPS Publication 199, Feb. 2004. Available: https://csrc.nist.gov/publications/detail/fips/199/final.

[14] W. 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: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf.