NIST SPECIAL PUBLICATION 1800-37B

Addressing Visibility Challenges with TLS 1.3 within the Enterprise

Volume B:
Approach, Architecture, and Security Characteristics

Murugiah Souppaya
Computer Security Division
Information Technology Laboratory

William Barker
Dakota Consulting
Silver Spring, MD

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

John Kent
The MITRE Corporation
McLean, VA

David Wells
Johann Tonsing
Mira Security
Cranberry Township, PA

Sean Turner
sn3rd
Washington, DC

Erik Freeland
Nubeva
San Jose, CA

Russ Housley
Vigil Security LLC
Herndon, VA

Murali Palamisamy
AppViewX
New York, NY

Dung Lam
F5
Seattle, WA

Paul Barrett
Ray Jones
NETSCOUT
Westford, MA

Patrick Kelsey
Not for Radio
Manheim, PA

January 2024

PRELIMINARY DRAFT

This publication is available free of charge from

https://www.nccoe.nist.gov/addressing-visibility-challenges-tls-13

NIST NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified by name or company logo or other insignia 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 1800-37B, Natl. Inst. Stand. Technol. Spec. Publ. 1800-37B, 64 pages, January 2024, CODEN: NSPUE2

FEEDBACK

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

Comments on this publication may be submitted to: applied-crypto-visibility@nist.gov.

Public comment period: January 30, 2024 – April 1, 2024

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 recommended practices to develop modular, adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to 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 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

The Transport Layer Security (TLS) protocol is widely deployed to secure network traffic. The latest version, TLS 1.3, has been strengthened so that even if a TLS-enabled server is compromised, the contents of its previous TLS communications are still protected—better known as forward secrecy. The approach used to achieve forward secrecy interferes with passive decryption techniques that are widely used by enterprises to achieve visibility into their own TLS 1.2 traffic. Many enterprises depend on that visibility to permit their authorized network security staff to implement controls needed to conform to cybersecurity, operational, and regulatory requirements. This forces enterprises to choose between using the old TLS 1.2 protocol or adopting TLS 1.3 with some alternative method for internal traffic visibility. The NCCoE has, in collaboration with technology providers and enterprise customers, initiated a project demonstrating options for maintaining visibility within the TLS 1.3 protocol within an enterprise to overcome these impediments. The project demonstrates several standards-compliant architectural options for use within enterprises to provide both real-time and post-facto systems

monitoring and analytics capabilities. This publication describes the approach, architecture, and security characteristics for the demonstrated proofs of concept.

KEYWORDS

bounded lifetime; break and inspect; ephemeral; key management; middlebox; passive inspection; protocol; Transport Layer Security (TLS); visibility.

ACKNOWLEDGMENTS

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

Name	Organization
Ravishankar Chamarajnagar	AppViewX
Michael Ackermann	Blue Cross Blue Shield
Dean Coclin	DigiCert
Avesta Hojjati	DigiCert
Jonathan Chen	F5
Ryan Johnson	F5
Brad Otlin	F5
Kevin Stewart	F5
Tim Cahill	JPMorgan Chase & Company
Joshua Klosterman	The MITRE Corporation
Michael Dimond	The MITRE Corporation
Julian Sexton	The MITRE Corporation
Nanjaiah Vijayalakshmi	NETSCOUT Corporation
David Cooper	NIST
William Polk	NIST (Former employee)
Gina Scinta	Thales Trusted Cyber Technologies
Steven Fenter	U.S. Bank Corporation
Jake Wills	U.S. Bank Corporation

The organizations who have collaborated 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:

Project Collaborators	
AppViewX	NETSCOUT Corporation
DigiCert	Not for Radio LLC

Project Collaborators	
F5	Nubeva Inc.
JPMorgan Chase & Company	Thales Trusted Cyber Technologies
Mira Security, Inc.	U.S. Bank Corporation

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.

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:

1. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
2. 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: applied-crypto-visibility@nist.gov.

Contents

List of Figures

1 Summary

Enterprises have typically depended upon visibility into data in transit within their networks, both traditional office networks and enterprise data centers, to implement critical cybersecurity, operational, and regulatory controls (e.g., intrusion detection and response, malware detection, troubleshooting, fraud monitoring). Deploying some network security protocols within enterprise data centers to protect integrity and confidentiality has posed challenges to network visibility required by these controls. To maintain visibility, enterprise architectures facilitate comprehensive inspection, collection, and analysis of internal network traffic through a small number of passive or active monitoring devices. To facilitate decryption of network traffic, passive decryption devices are provided copies of the servers' long-term cryptographic keys. In these cases, these long-term cryptographic keys allow decryption of past, present, and future network traffic for the lifetime of a key.

To improve the security of communications on the public Internet, modern protocol designers have made changes to protocols to implement stronger security properties that protect the secrecy of historical traffic even if the servers' long-term secret keys are compromised, a property referred to as *forward secrecy*. This property, however, has correspondingly created significant challenges for the network visibility strategies used by enterprises.

The National Cybersecurity Center of Excellence (NCCoE) has, in collaboration with technology providers and enterprise customers, initiated a project demonstrating options for maintaining visibility within an enterprise, given these challenges. The example solutions demonstrated are designed to be suitable for voluntary adoption across a wide range of user environments and meet the following criteria:

- Scalable;
- Relatively easy to implement/deploy;
- Application protocol-agnostic;
- Usable in real-time and post-packet capture;
- Effective for both security and troubleshooting purposes; and
- Widely available and supported in mainstream commercial products and services.

Enterprises using the Transport Layer Security (TLS) 1.2 [1] protocol without forward secrecy, which was how TLS 1.2 was originally specified, are currently using tools and architectural solutions that provide visibility into enterprise traffic within their network. However, TLS 1.2 visibility solutions provide more privilege than is needed to just view the traffic. Enterprise policies regarding visibility into received network traffic still need to remain capable of being enforced if the organizations' security monitoring, analysis, and management policies are to be enforced. This is because many monitoring and analysis tools that are used to conform to their security policies are dependent on visibility solutions that enable an enterprise-authorized party to decrypt the network traffic past, present, and future. The solutions demonstrated by this project facilitate implementation of TLS improvements by enterprises that have to date been discouraged by visibility limitations from doing so.

1.1 Challenge

Enterprise cybersecurity is dependent on identification, protection, detection, response, and recovery policies, mechanisms, and processes. Cryptography is an important mechanism for protecting enterprise information and processes. Network and system monitoring and analysis of both encrypted traffic and the underlying plaintext is often necessary for detecting cyber-attacks and anomalous behavior, understanding their nature, responding effectively, and recovering from the incidents. TLS is a cryptographic protocol that is widely deployed to secure internal enterprise traffic within traditional office networks and enterprise data centers, as well as connections across the public Internet.

The latest version, *The Transport Layer Security (TLS) Protocol Version 1.3* (Internet Engineering Task Force [IETF] Request for Comments [RFC] 8446 [2]), has been strengthened to provide *forward secrecy*. In the legacy TLS 1.2 implementations, forward secrecy is optional, but in TLS 1.3 it is provided by default. As stated above, many enterprises have troubleshooting, audit, and other policies that require visibility into unencrypted traffic and stored data. The TLS 1.3 approach to achieving forward secrecy conflicts with the passive decryption techniques that are widely used by enterprises to achieve this visibility into their own internal enterprise TLS-protected traffic. This results in enterprises choosing between using the TLS 1.2 protocol without forward secrecy or adopting TLS 1.3 together with some alternative method for achieving visibility into internal traffic. If an enterprise chooses the old TLS 1.2 protocol, they miss out on the performance enhancements in TLS 1.3 and face additional risks in relying on protocol implementations that will become increasingly out-of-date over time.

Ways that loss of visibility into received network traffic may affect organizations include loss or degradation of functions such as network performance monitoring, application performance monitoring, and security logging and diagnostics, as well as negative impacts on network and security operations/engineering roles. The inability to decrypt network data for purposes such as deep packet inspection (DPI), security, monitoring, and diagnostics leaves the security of enterprise networks dependent on endpoint devices for performance and security information and management. This loss of visibility introduces security and operational risks for network and data center operations. Consequences of this loss of visibility include the following:

- The incoming network data stream often provides information or perspectives that individual endpoint devices like workstations, servers, and other devices that can support a security client are not capable of providing.
- Network data frequently possesses a holistic view of sessions that no single platform in the chain can provide. This holistic view permits a comprehensive understanding of the consequences of anomalous traffic or traffic patterns. For example, network data is needed to determine where issues are occurring relative to middleboxes involved in sessions, such as firewalls, routers, proxies, and load balancers. Correlating and comparing these subsidiary sessions is essential when performing fault domain isolation and general diagnostic triage.
- Network data is even more critical when the endpoints are having problems or are in any way compromised. A degraded or compromised endpoint device may fail to report incidents that would be recognized from the incoming network data stream.
- Network data is essential for issues that involve multiple platforms and even more so for issues that involve multiple organizations within the enterprise.

- Network data is required when sessions span devices that do not log information well or at all. Even with devices that do log well, it is frequently necessary to augment that content with related network data.
- For those situations where endpoint data is adequate, it still needs to be collected, consolidated, centralized, and correlated. Network data is often needed to assure this is occurring properly. Even if/when such an endpoint infrastructure is effectively built, network data is critical to its ongoing operation.
- For situations where logging needs to be turned off, or even reduced (which is often the case), network data may be the only alternative. Network data is preferred for situations where the endpoint (or middlebox) platform is incapable of adequate logging without causing utilization or performance issues on the platform.
- For sessions which span domains of control, network data is the only common point at which multiple operators can establish common ground for monitoring, security, and diagnostics. This becomes even more critical in larger, more complex situations, such as outsourcing, partnerships, and cloud computing.
- From a security standpoint, there are many threats that are more easily or only identifiable from network data. Where and how the network data is collected will provide key information. For example, network data could be collected at the edge of the network, outside a demilitarized zone (DMZ), or within the internal network. It is also important to know whether the information is collected inline or passively. All of these provide differing forms of valuable information.
- It is often necessary to utilize network data to assure that endpoint security agents are operating properly.
- If an endpoint is in any way compromised or its security agent is not running properly, network data is both the only line of defense and the most critical tool for performing related triage and forensics, which are useful or necessary for quickly resolving related issues.
- If network data cannot be decrypted, a security breach, malware, or other compromise can spread throughout the entire organization once any single platform has been accessed. Malicious sessions and traffic can go undetected and hence be unconstrained since the sessions between platforms within the organization cannot be decrypted. Often called *pivoting*, this process represents a serious threat to network or data center operators.
- Nearly all attacks occur over the network, and attackers leave traces or tracks on the network. Questionable traffic can often be better understood based on where it is coming from (e.g., Marketing Client subnet improperly accessing a Human Resources database subnet). To obfuscate this, by removing the ability to decrypt, eliminates the ability to be aware of it and control it. Malicious actions would then be untraceable.
- DPI is one of many inspection tools. However, many tools often depend upon DPI to rectify issues with their own operation and/or to determine that such issues even exist. Many tools can even become part of the problem, and without network data DPI, such network-based attacks can go unnoticed indefinitely.
- Forensics performed after a breach or other security exposure event depends heavily on DPI and network data to figure out what happened, why, and what can be done about it. Root cause

analysis has become critical to most large organizations. The lack of DPI will make this much more difficult to accomplish in many situations.

Alternatives to DPI require much time and effort and are, for many organizations, prohibitively disruptive and expensive. Alternatives to monitoring and analysis of decrypted incoming network data streams that have been mentioned in the past include:

- **Re-architecting the enterprise network.** This is difficult, expensive, and time-consuming, and even where feasible is not a short-term solution.
- **Depending on endpoints for management and logging.** Even if available endpoint solutions selected are stable, capable, and effective and are consistent and reliable recorders of all events related to incidents (enhanced logging), this would require building a separate infrastructure for producing, collecting, storing, and parsing terabytes (or more) of data. None of this is a simple proposition, and such an infrastructure would require both DPI for certain data and significantly enhanced infrastructure management. Furthermore, if the true root cause is occurring at a middlebox device, endpoints will not see essential information at all.
- **Use of intermediate proxies between application tiers.** This approach would add cost, latency, and potential points of failure. It becomes less viable, and more expensive, the more tiers that a given application has. The cost and complexity increases could be enormous in many cases. There may also be situations where intermediate proxies are not possible (such as secure subnets and virtual environments).

A significant constraint in meeting the visibility challenges attendant on TLS 1.3 is the need to provide workable approaches that do not change the current TLS 1.3 standard or require development or adoption of additional or alternative standards. The project objective is to provide TLS 1.3 implementation approaches that permit visibility. Several technical and management challenges are being addressed in this TLS 1.3 visibility project. Some of the following challenges are shared by visibility in TLS 1.2 environments, while others are unique to TLS 1.3.

- **Secure management of servers' cryptographic keys**. Private and secret keys must be protected throughout the cryptographic lifecycle: creation, distribution, use, retention, and destruction. Unauthorized disclosure places all past, present, and future traffic encrypted under those keys at risk.
- **Management of recorded traffic**. This demonstration project assumes that recorded traffic is stored in encrypted form, not plaintext. To be useful, the enterprise must be able to identify the corresponding key material. However, recorded traffic remains at risk of compromise until the corresponding key material or the recorded traffic itself is destroyed. Any solution must allow the enterprise to recover plaintext traffic when required but ensure that traffic is not at risk of compromise indefinitely.
- **Managing expectations of privacy**. IT users often have preconceived notions about the privacy of TLS connections, and the security enhancements associated with TLS 1.3 may increase those expectations. Enterprises that rely on visibility for critical controls should ensure that TLS 1.3 connections within that scope are accepted only by informed users.

In addition to the TLS-specific challenges, the NCCoE is also considering the practical challenges of scalability, ease of deployment, and usability of the visibility solutions themselves.

1.2 Solution

The NCCoE assembled a highly qualified team that included public- and private-sector cryptographers, secure network technology providers, and private-sector user organizations that are facing TLS 1.3 visibility challenges. To meet the challenges in a manner that does not change or replace the IETF RFC 8446 standard, provides secure management of servers' cryptographic keys, securely manages recorded traffic, and manages expectations of privacy, the project team identified a broad set of options. These options included:

- key-management mechanisms that defer forward secrecy until all copies of keying material needed to maintain current levels of network visibility are deleted (such as copies retained to support passive inspection)
- network architectures that inherently provide visibility, such as using overlays or incorporating middleboxes (see RFC 3234: Middleboxes: Taxonomy and Issues [3])

The TLS 1.3 Visibility project described in this volume focuses primarily on the following passive inspection and middlebox solutions for avoiding consequences of loss of TLS 1.3 visibility characteristics while migrating to TLS 1.3 and avoiding vulnerabilities to which TLS 1.2 is susceptible:

- To achieve visibility through key management (via passive inspection), we demonstrate two technical mechanisms that can be implemented for each server whose traffic is of interest to the enterprise.
  - In the first case, the enterprise provisions bounded-lifetime Diffie-Hellman key pairs for TLS 1.3 servers for use in ephemeral key exchanges. This approach includes a purely static deployment while also including deployments that use key pairs for a short period of time.
  - In the second case, the enterprise collects and retains the symmetric session keys used to encrypt the connections instead of provisioning Diffie-Hellman key pairs.
- Some aspect of analytics functions needing enterprise visibility into its encrypted traffic may require combining network architecture and key-management techniques to achieve operational visibility. Therefore, the scope of the project includes demonstration of an architecture that achieves visibility inside the data center through tools that break and inspect traffic. These middleboxes are commonly used at the enterprise edge to achieve real-time visibility; in this demonstration project, we expand the scope to examine deployment within the enterprise and address access to historical data by leveraging key management-based solutions.

In the passive inspection solutions, the managed Diffie-Hellman keys and symmetric traffic keys are retained by a key distribution function until all corresponding encrypted traffic has been decrypted or is destroyed or otherwise no longer available. Systems that are authorized to examine traffic obtain the appropriate keys from the key distribution function. The solution also incorporates components to retain traffic for retrospective applications, like troubleshooting and cybersecurity forensics. The stored traffic is retained in encrypted form until policy conditions (e.g., retention time limits) are met. The data is then deleted by the storage function. The resulting solutions protect keys and data against misuse or compromise and do not leave recorded traffic at risk of compromise indefinitely. The solutions include mechanisms and procedures used to transmit, store, provide access to, and use cryptographic keys and that perform comprehensive deletion of decryption keys when established temporal or data protection limits are met.

Since TLS 1.3 is designed to achieve allow client/server communication in a way that prevents eavesdropping, the solutions also assume out-of-band notification of the visibility policy.

1.3 Benefits

Enterprises can accrue the following benefits from implementing one of the visibility solutions described in this volume:

- Retaining visibility into network traffic without re-architecting the enterprise network, depending on endpoints for management and logging, or using intermediate proxies among all tiers.
- Achieving tunable, time-bounded forward secrecy.
- Avoiding the potential for security vulnerabilities to which TLS 1.2 is susceptible.
- Moving from implementations using the RSA key transport algorithm that has been deprecated due to cryptographic security concerns.
- Conforming to TLS 1.3 implementation mandates, where applicable.

2 How to Use This Guide

This practice guide is being developed in five parts. Depending on your role in your organization, you might use this guide in different ways:

Business decision makers, such as chief information security, product security, and technology officers, can use NIST Special Publication (SP) 1800-37A: *Executive Summary*, to understand the project's challenges and outcomes, as well as our solution approach.

Technology, security, and privacy program managers who are concerned with how to identify, understand, assess, and mitigate risk can use this part of the guide, NIST SP 1800-37B: *Approach, Architecture, and Security Characteristics*. It describes the architecture and different implementations. Also, the future NIST SP 1800-37E: *Risk and Compliance Management*, will map components of the TLS 1.3 visibility architecture to security characteristics in broadly applicable, well-known cybersecurity guidelines and practices.

IT professionals who want to implement an approach like this can make use of the NIST SP 1800-37C: *How To Guide* currently under development. It will provide product installation, configuration, and integration instructions for building example implementations, allowing them to be replicated in whole or in part. They will also be able to use a future NIST SP 1800-37D: *Functional Demonstrations*, which will provide the use cases that have been defined to showcase TLS 1.3 visibility capabilities and the results of demonstrating these capabilities with each of the example implementations.

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 the TLS 1.3 visibility solutions described herein. 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, 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 preliminary draft guide. NIST is adopting an agile process to publish this content. Each volume is being made available as soon as possible rather than delaying release until all volumes are completed. Work continues on designing and implementing the example solution and developing other parts of the content. As a preliminary draft, this volume will have at least one additional draft released for public comment before it is finalized. 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 applied-crypto-visibility@nist.gov.

3 Approach

The approach taken to determine whether and how to address TLS 1.3 visibility issues was developed in an open and collaborative manner. The NCCoE hosted an industry roundtable in 2018 to assess the scope of the visibility challenges faced by enterprises. NCCoE staff subsequently participated in the Center for Cybersecurity Policy's 2019 workshop [4] on enterprise visibility that identified a set of baseline criteria for acceptability of solutions for visibility challenges and has adopted them without change as the baseline criteria for a generally effective solution. The NCCoE hosted a virtual workshop focused on TLS 1.3 in September 2020 [5]. The interactions identified a broad set of options for maintaining visibility, including:

- Endpoint mechanisms that establish visibility, such as enhanced logging;
- Network architectures that inherently provide visibility, such as using overlays or incorporating middleboxes;
- Key management mechanisms that forgo forward secrecy to maintain current levels of network visibility;
- Innovative tools that analyze network traffic without decryption; and
- Deployment of alternative standards-based network security protocols where forward secrecy is optional or not supported.

Following the September 2020 workshop [5], in May 2021 the NCCoE published a project description for a TLS 1.3 visibility project that included use case scenarios for implementation of potential solutions discussed in the workshops. Collaborators participating in this project submitted their capabilities in response to an open call in the Federal Register for all sources of relevant security capabilities from academia and industry (vendors and integrators). Those respondents with relevant capabilities or product components signed a Cooperative Research and Development Agreement (CRADA) to collaborate with NIST in a consortium to build this example solution. The resulting project team then began to develop the architectures and demonstration scenarios that are described in this volume.

3.1 Audience

The audience for this volume is the community of technology, security, and privacy program managers who are concerned with how to identify, understand, assess, and mitigate risk. It describes what we built and why, including the risk analysis performed and the security/privacy control mappings.

3.2 Scope

The scope of the project is to demonstrate various approaches and practices that meet common compliance, operations, and security requirements while gaining the security and performance benefits of TLS 1.3 deployment. The project focuses on enterprise data center environments which include on-premises data center and hybrid cloud deployments hosted by a third-party data center or a public cloud provider. The project demonstrates real-world visibility approaches utilizing current or emerging components. Solutions may utilize proprietary vendor products as well as commercially viable open-source solutions. The project focuses on the security implications of TLS 1.3 protocol implementations that provide system and application administrators and users the necessary visibility into the content of

information being exchanged. This includes approaches that restore visibility into encrypted data in transit, such as alternative key establishment and management. The project leverages current and ongoing NIST and industry standards, as well as NCCoE application projects.

Information transmitted over the public Internet (e.g., connections between an enterprise and its customers) is out of scope and must not be impacted by proposed solutions. Also out of scope are emerging deployment models leveraging encrypted transport to protect protocols that were previously in the clear, such as DoT (Domain Name System [DNS] over TLS) [6], DoH (DNS over Hypertext Transfer Protocol Secure [HTTPS]) [7], and DoQ (DNS over QUIC) [8] . DoT, DoH, and DoQ may be the subject of future NCCoE work.

3.3 Assumptions

This project is guided by the following assumptions:

- Recent enhancements to cryptographic security protocols, such as TLS 1.3, disrupt current approaches to achieving visibility into internal network communications within enterprise data centers. While these protocol enhancements increase performance and address security concerns within the enterprise and on the public Internet, they also reduce enterprise visibility into internal traffic flows. These enhanced security protocols and new deployment models were not designed to accommodate decryption of internal network traffic by passive monitoring devices; this has created potential compliance, security, and operational impacts in enterprises that currently rely on such devices.
- Enterprises have raised questions about how to meet enterprise security, operational, and regulatory requirements for critical services while using the enhanced security protocols and leveraging new deployment models. Such enterprises may need to consider applying new architectures and novel techniques to augment or replace conventional monitoring devices while satisfying their business, regulatory, security, and network operations requirements.
- Many enterprises choose to rely on the same standard transport security protocols to exchange information over the public Internet and within internal enterprise network environments. For these enterprises, the ability to naturally migrate to the most current versions offers continuity and simplifies network evolution. As a result, this project assumes that enterprises cannot rely on older protocol versions as a long-term solution.
- The majority of the components of the project's demonstration environment that are part of the on-premises data center are located in a laboratory at the NCCoE facility in Rockville, Maryland. This is to ease the integration of the components and provide an open and transparent environment for the participants to collaborate on building and testing the proposed approaches.

3.4 Risk Assessment

NIST SP 800-30 Revision 1, *Guide for Conducting Risk Assessments*, [9] 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*—material that is available to the public. The Risk Management Framework (RMF) guidance, as a whole, proved to be invaluable in giving 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

General threats to data exchanged over networks include eavesdropping, tampering, and forgery. TLS uses cryptographic mechanisms to protect data from being stolen, modified, or spoofed. This section describes generic threats to the security of information protected by TLS mechanisms.

- **Stolen keys.** Threat actors who gain unauthorized access to symmetric keys used with cryptographic algorithms that provide data confidentiality or to private keys used with public key cryptographic algorithms can defeat authentication mechanisms or access additional keys. If symmetric or private keys are stored in unencrypted form, they are particularly vulnerable to being used to compromise the protection afforded by TLS.
  - Inadequately protected keys are subject to unauthorized access by direct theft, hacking, or other means. The more information that is protected by a given key, the more serious the consequence of unauthorized access usually is.
  - TLS 1.3 symmetric keys are used with Authenticated Encryption with Associated Data (AEAD) algorithms, so they provide confidentiality and integrity for the keys.
- **Certificate compromise.** Public keys used during authentication are often exchanged in certificates issued by a certificate authority (CA). There is a real threat that the issuing CA can be compromised or that the registration system, persons, or process can be compromised to obtain an unauthorized certificate in the name of a legitimate entity and thus compromise the clients.
- **Handshake data replay.** Parties to cryptographically protected communications exchange keys using protocols often called handshakes. There are often multiple steps in a handshake, but TLS 1.3 allows the client to send data (known as 0-RTT data) in the first flight of a handshake with a server to which the client has previously connected. Replayable 0-RTT data presents several security threats to TLS-using applications, unless those applications are specifically engineered to be safe under replay (minimally, this means idempotent, but in many cases may also require other stronger conditions, such as constant-time response). Many applications do not allow 0-RTT to avoid the replay concern (e.g., draft-ietf-netconf-over-tls13). Potential attacks include:
  - Duplicating actions which cause side effects (e.g., purchasing an item or transferring money) to be duplicated, thus harming the site or the user.
  - Storing and replaying 0-RTT messages to reorder them with respect to other messages (e.g., moving a delete to after a create).
  - Exploiting cache timing behavior to discover the content of 0-RTT messages by replaying a message to a different cache node and then using a separate connection to measure request latency, to see if the two requests address the same resource.

569 • If data can be replayed a large number of times, additional attacks become possible, such
570 as making repeated measurements of the speed of cryptographic operations. In addition,
571 they may be able to overload rate-limiting systems.

572 ▪ **Misuse of client credentials.** Another threat that must be protected against is misuse or
573 exposure of the credentials that reside on the client machine.

574 ▪ **Absence of support for endpoint solutions.** Not all server-type systems in an enterprise will be
575 supported by endpoint vendors. Enterprises commonly have old systems running custom
576 operating system (OS) software for which there is limited or no support in endpoint solutions.

577 ▪ **Compromise of systems running endpoint software.** When deployed, endpoint solutions may
578 provide good security but only up to the point that the system running the endpoint or the
579 endpoint software itself is compromised. Endpoint compromise may not be detectable by the
580 endpoint solution, as it relies on trusting what the endpoint software tells it, leading to false
581 trust in the endpoint. Alternate solutions that analyze network traffic can detect compromised
582 or rogue endpoints and provide resilience if used alongside endpoint solutions.

583 3.4.2 Vulnerabilities

584 Several vulnerabilities have been found in both the TLS 1.2 protocol and in the implementation of
585 features permitted by TLS 1.2. While TLS 1.2 can be made secure using extensions and careful
586 configuration, TLS 1.3 has been designed to avoid these vulnerabilities. Note also that vendors are now
587 focusing on TLS 1.3, so getting new algorithms or extensions for TLS 1.2 to be implemented by vendors
588 will be increasingly difficult.

589 ▪ Unlike TLS 1.3, TLS 1.2 offers some cipher suites, such as those that use RSA key exchange, that
590 do not provide forward secrecy. Where forward secrecy is not provided, if a TLS-enabled server
591 is compromised, the contents of its previous TLS communications are vulnerable to exposure.
592 The ephemeral key exchange mechanisms that provide forward secrecy also protect future TLS
593 communication against passive attackers.

594 ▪ The passive decryption techniques that are widely used by enterprises to achieve visibility into
595 their own internal TLS 1.2 enterprise traffic only work with the cipher suites that use RSA key
596 exchange. In addition to not providing forward secrecy, the RSA key exchange used in TLS 1.2
597 has been vulnerable to a number of implementation flaws. As a result, its use has been
598 deprecated. (See SP 800-131A Rev. 2 [10], SP 800-52 Rev. 2 [11], IETF draft *Deprecating Obsolete*
599 *Key Exchange Methods in TLS 1.2* [12], and *TLS 1.2 Is Frozen* [13].)

600 ▪ Reuse of keys outside the protected data center would create vulnerabilities regarding
601 comparison of key shares in different handshakes. This would permit an attacker to track an
602 endpoint or reveal the identity of the TLS server that a user connected to. On the public
603 Internet, this would represent a violation of user privacy. The current TLS 1.3 specification
604 contains a new normative requirement stating that to prevent tracking and identification,
605 “Clients SHOULD NOT reuse a ticket for multiple connections.” Further, as of this writing, the
606 draft of the revised TLS 1.3 specification contains an additional normative requirement for the
607 same purpose, “Clients and Servers SHOULD NOT reuse a key share for multiple connections.
608 Reuse of a key share allows passive observers to correlate different connections.” This
609 discourages client and server reuse of a key share for multiple Internet connections. Reusing key
610 shares outside protected facilities can also expand the impact of security breaches.

- Except in cases of exclusively symmetric key management environments, the sharing of symmetric keys also needs to be restricted to data center environments. A node with access to the symmetric traffic keys can not only view all traffic, but also impersonate the endpoints by modifying and injecting traffic.
- The TLS 1.2 visibility mechanisms that are based on RSA private key sharing allow middleboxes to masquerade as the server. The TLS 1.3 mechanisms do not allow this because the signature private key does not need to be shared to gain visibility.

3.4.3 Risk

While TLS 1.3 significantly reduces risks associated with TLS 1.2, its approach to achieving forward secrecy necessarily introduces institutional risk to systems and enterprises having operational security requirements for visibility into traffic exchanges by their IT security teams. From a purely cryptographic point of view, enterprises implementing TLS 1.3 might effectively mandate reliance on endpoint solutions to achieve their operational security requirements. However, this can be infeasible and ill-advised for many enterprises. If an enterprise were to rely entirely upon endpoint security, without any visibility by middleboxes, then the items listed below would be needed, which might have significant time or cost impact on the migration to TLS 1.3.

A frequent first step for a sophisticated attacker on a target is to modify, disable, or evade endpoint security tools. This is an enduring reality faced by all blue team security practitioners (e.g., security operations or incident response teams) for decades. Researchers continue to publish results of successful evasion and neutering techniques for all endpoint controls, including the most sophisticated tools available on the market. Malware that has detonated on a computer only has the OS between it and the endpoint security tool. To combat this risk, organizations should budget for and perform the following steps:

- Follow OS hardening requirements to limit the ability of software-accessible accounts to control endpoint security tools.
- Implement strategies that can help identify missing, fraudulent, or anomalous status or log data from the endpoint tool(s).
- Upgrade, modernize, or replace all legacy applications that leverage insecure libraries, protocols, applications, and subsystems throughout the endpoint state.
- Ensure that remote access to all endpoint agent management tools adheres to the most stringent authentication/authorization strategies.
- Implement strategies to persistently refresh endpoint agent policy or accurately detect policy drift.
- Implement a credible application allowlist solution to prevent execution/reading of applications and libraries.
- Endpoint control techniques such as enhanced logging can be effective detective controls where the following conditions are met:
    - Adequate funding exists for a robust security information and event management (SIEM) infrastructure, and

650 - Adequate personnel are available to manage a SIEM or similar tool to create and
651 administer correlation rules to produce timely, accurate alerts related to anomalous
652 activity.

653 ▪ Implement appropriate configuration of all cogent log generating agents on the endpoint (e.g.,
654 instant log transfer to SIEM vs. periodic transfer of batched logs which can be deleted by an
655 attacker to hide their activity).

656 See NIST SP 800-92 Revision 1, *Cybersecurity Log Management Planning Guide*, for more information on
657 logging and log protection.

658 Different types of malware execute with different intentions. Destructive malware often doesn't require
659 the instantiation of command-and-control (C2) communication with attacker infrastructure, but the
660 goals of the most sophisticated adversaries (e.g., financial gain, data exfiltration) are best achieved by
661 maintaining persistence via a C2 channel over a network to the attacker. Therefore, security controls
662 surrounding outbound network communications from organizational endpoints should be maximally
663 restrictive.

664 The above concepts are beneficial to any organization, of course, but are often considered by IT and
665 security practitioners outside of highly targeted enterprises to be a luxury. But when network visibility is
666 lost, network controls are also lost, and these luxuries quickly become imperatives.

667 As stated previously, re-architecting networks is difficult, expensive, and time-consuming. Even if viable,
668 it is not a short-term solution. Depending on endpoints for management and logging is also not currently
669 practical for large data centers. Endpoint solutions must become stable, capable, and effective. Endpoint
670 solutions must then become consistent and reliable recorders of all related events (enhanced logging).
671 Additionally, a separate infrastructure for producing, collecting, storing, and parsing terabytes (or more)
672 of data must be built. None of this is a simple proposition, and this infrastructure will require DPI for
673 certain data, as well as to manage the infrastructure. Furthermore, if the true root cause of a
674 compromise or other network problem is occurring at a middlebox device, endpoints will not see this
675 information at all. Use of intermediate proxies between all tiers adds cost, latency, and potential points
676 of failure. It also becomes less viable and more expensive as the number of tiers a given application has
677 increases; the cost and complexity increases are enormous in many cases. Finally, there will be
678 situations where intermediate proxies are not possible at all, such as secure subnets and virtual
679 environments.

680 In adopting any visibility solution, protecting stored session keys from access by entities external to the
681 data center is essential, and that requires implementing access controls that enforce least privilege
682 within the data center. Consequently, management's risk assessment involves trading off the relative
683 consequences of delaying TLS 1.3 implementation or replacing enterprise data centers against the cost
684 of protecting stored session keys from access by processes other than those that require and are
685 specifically approved for specific continuous monitoring or forensics functions. In implementing visibility
686 solutions, enterprises must focus on supporting least privilege principles and compliance with zero trust
687 and supply chain security requirements. See NIST SP 800-207 [14]. Ideally, implementations will provide
688 network owners with control over what information is actually shared to monitoring systems.
689 Cryptographic protection of all keys, whether at rest or in transit, is essential except where the keys are
690 being employed in a cryptographic process or approved compensating controls are employed.

Standardized open interfaces for endpoint interception are needed but are not currently available in applications that scale to the requirements of large data centers. As a result, organizations may determine that acceptance of some cryptographic security risks associated with the visibility solutions described herein is acceptable in the face of the consequences of delaying TLS 1.3 implementation and of loss of visibility into information exchanges by the IT security staff responsible for security monitoring and forensics. One of the visibility approaches demonstrated in this project involves middlebox solutions. The risks associated with introducing in-house middleboxes having man-in-the-middle capabilities and/or caching or otherwise storing session keys may be weighed against the consequences of losing security monitoring and forensics capabilities.

3.4.4 Security Control Map

A future SP 1800-37E, *Risk and Compliance Management*, will describe the mappings between cybersecurity functions performed by the reference design's logical components (Sections 4.2 and 4.3) and the security characteristics enumerated in relevant cybersecurity documents. These mappings are intended for any organization that is interested in implementing TLS 1.3 visibility solutions or that has begun or completed an implementation.

The mappings provide information on how cybersecurity functions from the reference design are related to NIST-recommended security outcomes and controls: the security outcome subcategories from the NIST Cybersecurity Framework (*Framework for Improving Critical Infrastructure Cybersecurity* [CSF] 1.1) [15] and security controls identified in NIST SP 800-53r5 (*Security and Privacy Controls for Information Systems and Organizations*) [16]. All of the elements in these mappings—the TLS 1.3 visibility cybersecurity functions, CSF Subcategories, and SP 800-53 controls—are concepts involving ways to reduce cybersecurity risk.

There are two primary use cases for this mapping. They are not intended to be comprehensive.

1. **Why should organizations implement TLS 1.3 visibility solutions?** This use case identifies how implementing TLS 1.3 visibility solutions can support organizations in achieving CSF Subcategories and SP 800-53 controls. This helps communicate to an organization's chief information security officer, security team, and senior management that expending resources to implement TLS 1.3 visibility solutions can also aid in fulfilling other security requirements.

2. **How can organizations implement TLS 1.3 visibility solutions?** This use case identifies how an organization's existing implementations of CSF Subcategories and SP 800-53 controls can help support trusted implementation of TLS 1.3 visibility solutions. An organization wanting to implement TLS 1.3 visibility solutions might first assess its current security capabilities so that it can plan how to add missing capabilities and enhance existing capabilities. Organizations can leverage their existing security investments and prioritize future security technology deployment to address the gaps.

4 Technologies

The technology collaborators for this project have offered products and insights that improve organizations' visibility into traffic protected by the improved TLS 1.3 protocol. This section identifies the project collaborators, components of the functional architecture employed, and products provided by the collaborators to implement the functional architecture.

4.1 Project Collaborators

The following organizations have collaborated with the NCCoE in demonstrating mechanisms for implementing TLS 1.3 without loss of essential real-time and post-facto visibility by the organizations into traffic being exchanged within enterprise networks. Real-time visibility permits monitoring for detection of threats or incidents concurrent with the data exchange, and post-facto visibility permits after-the-fact analytics (e.g., forensics analysis to permit understanding of anomalies and responses to and/or recovery from security incidents).

4.1.1 AppViewX

AppViewX is an automated certificate lifecycle management (CLM) solution that simplifies public key infrastructure (PKI) and certificate management. It combines the best of automation, security, and insights to meet all enterprise PKI and key management needs. AppViewX CERT+ features are purpose-built to address both the operational and security challenges of certificate and key management to, in turn, help organizations prevent application outages and security breaches. AppViewX capabilities include discovering all certificates across complex enterprise environments, building and maintaining central inventories, provisioning both private and public trust certificates from any CA, alerting to expiring certificates, and fully automating renewals and revocation. AppViewX can be deployed as a virtual appliance or in a public cloud as a virtual appliance or container service, or consumed as Software-as-a-Service (SaaS). For more details, visit https://www.appviewx.com.

4.1.2 DigiCert

DigiCert is a provider of scalable TLS and PKI solutions for identity and encryption. The company is known for its expertise in identity and encryption for web servers and Internet of Things devices. DigiCert supports TLS/SSL and other digital certificates for PKI deployments at any scale through its certificate lifecycle management platform, CertCentral®. The company provides enterprise-grade certificate management platforms, responsive customer support, and advanced security solutions. Learn more about DigiCert at https://www.digicert.com.

4.1.3 F5

F5, Inc. is a publicly-held American technology company specializing in application security, multi-cloud management, online fraud prevention, application delivery networking, application availability & performance, network security, and access & authorization. F5 is headquartered in Seattle, Washington with an additional 75 offices in 43 countries focusing on account management, global services support, product development, manufacturing, and software engineering. F5 originally offered application delivery controller technology, but expanded into application layer, automation, multi-cloud, and security services. The company offers modules on their proprietary operating system, TMOS (Traffic

Management Operating System). These modules include, but are not limited to, Local Traffic Manager, Advanced Web Application Firewall, Domain Name Service, and Access Policy Manager. These offer organizations the ability to deploy load balancing, Layer 7 application firewalls, single sign-on (for Active Directory [AD], Azure AD, and Lightweight Directory Access Protocol [LDAP]), as well as enterprise-level virtual private networks. While F5's BIG-IP offering was traditionally a hardware product, F5 now offers it as a virtual machine, which they have branded as the BIG-IP Virtual Edition. The BIG-IP Virtual Edition is cloud-agnostic and can be deployed on-premises in a public and/or hybrid cloud environment.

4.1.4 JPMorgan Chase & Co.

JPMorgan Chase & Co. is an American multinational financial services firm headquartered in New York City and incorporated in Delaware. It is the largest bank in the United States and the world's largest bank by market capitalization.

4.1.5 Mira Security

Mira Security delivers standalone TLS visibility solutions allowing existing, unmodified enterprise security tools to detect and block threats hidden inside encrypted traffic flows. With over 15 years' experience in building scalable, safe, and secure visibility solutions, initially as part of parent company Netronome Systems and now as Mira Security, their technology is embedded in solutions from many companies as well as being available directly from Mira. Their Encrypted Traffic Orchestrator (ETO) software supports all the latest TLS standards, providing visibility into encrypted traffic without weakening the security profile of the connection, and can be deployed as a physical or virtual appliance or in public cloud environments. Learn more at https://mirasecurity.com.

4.1.6 NETSCOUT

NETSCOUT Systems, Inc. (NETSCOUT) assures digital business services against disruptions in availability, performance, and security. NETSCOUT combines its patented smart data technology with smart analytics and provides the real-time, pervasive visibility, and insights that its customers need to accelerate and secure their digital transformation. NETSCOUT's approach aims to transform the way organizations plan, deliver, integrate, test, and deploy services and applications. Its nGenius service assurance solutions provide real-time, contextual analysis of service, network, and application performance. Founded in 1984 as Frontier Software, NETSCOUT has evolved from a software consulting business to an enterprise service assurance and cybersecurity solutions provider serving large federal and local government, service provider, and enterprise customers.

The mission of NETSCOUT is to protect the global leaders of industry from the risks of disruption, allowing them to solve their network performance and security problems to ensure that the connected world runs safely and smoothly. In support of its mission, NETSCOUT provides software solutions that support the service assurance, advanced cyber threat and distributed denial of service (DDoS) protection, and business analytics/big data areas of its customers' business.

4.1.7 Not for Radio

Since 2013, **Not for Radio** (NFR) has been providing solutions to complex challenges in communication networks for both corporate and government customers, with deployments in internet and

telecommunication infrastructure as well as high-performance computing fabrics. NFR's Encryption Visibility Architecture (EVA) product delivers a flexible software solution to the challenge of maintaining data visibility in enterprise networks following the deployment of TLS 1.3 (while likewise supporting additional protocols such as legacy TLS and IPsec). EVA is designed to be minimally intrusive with respect to the diversity of existing security postures, compliance regimes, performance requirements, and orchestration technologies typically found in service operator environments.

4.1.8 Nubeva

Nubeva develops next-generation enterprise decryption solutions for TLS and ransomware. The company's TLS solution consists of a micro-endpoint agent that automatically discovers and extracts symmetric keys from handshake processes in memory in real-time, and then securely forwards them to systems for fast and easy decryption. The solution works without any modification to applications, libraries, network and systems architectures, or PKI. Nubeva's Session Key Intercept (SKI) works on nearly all versions of Linux, containers and Kubernetes, and Windows server/client systems. The solution enables decryption of TLS 1.3 and of 1.2 with forward security, as well as pinned certificate sessions for both passive and inline use-cases. The company delivers the solution as a software toolkit to enable solution and service providers, as well as mature SecOps/DevOps teams, to enhance existing or new inspection solutions.

4.1.9 Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies is a U.S. provider of cybersecurity solutions dedicated to the U.S. Government. It protects the government's most vital data from the core to the cloud to the edge with a unified approach to data protection. Thales' solutions reduce the risks associated with the most critical attack vectors and address the most stringent encryption, key management, and access control requirements. In addition to the core solutions developed and manufactured in the U.S. specifically for the Federal Government, Thales sells and supports industry-leading, third-party, commercial-off-the-shelf solutions. To mitigate the risks associated with procuring data security solutions developed outside of the U.S, Thales operates under a Proxy Agreement with the Defense Counterintelligence and Security Agency (DCSA) for Foreign Ownership, Control, or Influence (FOCI) and Committee on Foreign Investments in the United States (CFIUS) National Security Agreement.

4.1.10 U.S. Bank Corporation

U.S. Bancorp, with approximately 77,000 employees, is the parent company of U.S. Bank National Association. The Minneapolis-based company serves millions of customers locally, nationally, and globally through a diversified mix of businesses: Consumer and Business Banking; Payment Services; Corporate and Commercial Banking; and Wealth Management and Investment Services. Union Bank, consisting primarily of retail banking branches on the West Coast, joined U.S. Bancorp in 2022. The company has been recognized for its approach to digital innovation, social responsibility, and customer service, including being named one of the 2022 World's Most Ethical Companies and Fortune's most admired superregional bank. Learn more at https://usbank.com/about.

4.2 System Architecture Functions

The following subsections identify the component functions that comprise the TLS 1.3 visibility architecture. Each subsection describes the function that it serves in the reference design architecture.

4.2.1 Server Components

Server components are system entities that provide services such as HTTPS, email, and other applications in response to requests from other system entities called *clients*.

In this project, servers are devices that manage the demonstration network resources. The TLS server is the peer for encrypted traffic that generates session keys, negotiates encryption protocols, and connects to key management infrastructure.

4.2.2 Client Components

Client components are system entities that request and use a service provided by another system entity called a *server*. Examples of client components include enterprise workstations that receive network traffic, management workstations, analytics workstations, and NETSCOUT Omnis Cyber Intelligence and InfiniStreamNG appliances. Usually, it is understood that the client and server are automated components of the system, and the client makes the request on behalf of a human user. In some cases, the server may itself be a client of some other server.

In this project, client components are devices that initiate encrypted traffic. They are interfaces for human users, devices, applications, and processes to access network functions, including requesting certificates and keys. The TLS client devices are likely to be located outside of the data center.

4.2.3 Network Tap Function

The network tap is a component that provides a copy of traffic from a network segment in support of logging requirements and the network security applications to monitor traffic and identify malicious activity or security threats.

4.2.4 Break and Inspect Middlebox Function

The middlebox function is executed by a computer networking device that transforms, inspects, filters, and manipulates traffic for purposes other than packet forwarding. A break and inspect middlebox is an inline security mechanism that allows enterprises to decrypt traffic, inspect the decrypted content for threats, and then re-encrypt the traffic before it enters or leaves the network. In this project, the break and inspect middlebox is the component that taps, decrypts, terminates, and re-encrypts/reinitiates traffic.

4.2.5 Real-Time Decryption Function

A real-time decryption function conducts decryption operations that must guarantee response times within a specified time or window of time, usually relatively short. In this project, the real-time decryption component decrypts and forwards the copied traffic in real time.

4.2.6 Real-Time Analytics Function

Real-time analytics is the function that applies logic and mathematics to data to provide insights for immediate threat detection and response. For some use cases, real time simply means the analytics is completed within a few seconds or minutes after the arrival of new data. In this project, the function is executed by a set of tools for examining unencrypted payloads to identify a set of characteristics such as:

- Causes of network or application performance degradation or failures
- Key management-based communications failures
- Anomalous received data and their sources
- Detection of traffic from unauthorized sources

Note that transfers of information even within the enterprise and any information stored on or by the analytics platform require cryptographic protection or compensating physical controls.

4.2.7 Post-Facto Decryption and Analytics Function

Post-facto decryption and analytics is decryption and storage of encrypted data for detailed analysis at a later time (e.g., for forensics purposes). The information must be protected (e.g., encrypted with a key accessible by analytics processes or physically isolated and protected) while in storage and be destroyed immediately when no longer needed or to meet an organization's defined policy.

4.2.8 Key Management Agent Function

The key management agent function is the gateway via which the key governance platform provisions TLS server applications with bounded-lifetime DH key pairs. In addition to providing a secure provisioning point for new key material, it implements the key activation and expiration policies as communicated by the key governance platform.

4.2.9 Enterprise Public Key Infrastructure (PKI)

The PKI is an authorized entity that stores, signs, and issues digital public key certificates. The CA validates identities and binds them to cryptographic key pairs with digital certificates.

4.2.10 Key Governance Function

Certificate and key governance functions include securely issuing, monitoring, facilitating, and using digital public key certificates and managing the cryptographic keys exchanged using the certificates. In this project, it is the security module that performs storage and distribution of session keys (e.g., discovery, creation, renewal, provisioning, revocation, and destruction of certificates and keys).

4.2.11 Key Source

The key source is a FIPS 140-validated [17] entity that securely generates cryptographic keys and key pairs that are used for demonstration of cryptography employed in the TLS 1.3 visibility project.

4.2.12 TLS Traffic Sources and Sinks

The laboratory configuration includes TLS traffic sources and sinks for the data exchanged in and examined by the TLS 1.3 visibility project. The traffic needs to be on a scale that provides some confidence that the demonstrated solutions will be effective in large data centers.

4.3 Products Comprising the Demonstration Architecture

The following subsections identify the component products that comprise the TLS 1.3 visibility architectural solutions. Each subsection describes the component product being contributed, identifies the function that it serves in the reference design architecture (how it functions in the reference design), and optionally indicates other functionality not demonstrated in the TLS 1.3 visibility project.

4.3.1 Mira Encrypted Traffic Orchestrator (ETO)

Mira Encrypted Traffic Orchestrator (ETO) software provides a transparent TLS visibility solution that feeds decrypted traffic to existing security tools, allowing the detection and removal of threats contained in the encrypted end-to-end traffic flow. ETO is transparent at the network layer, making deployment straightforward with no requirements to change the existing network architecture or network addressing. Fine-grained policy controls are provided, allowing an enterprise to control which encrypted traffic flows are made visible, ensuring compliance with corporate and regulatory requirements covering privacy and security.

ETO can be deployed as a physical or virtual appliance and in public cloud environments. Whatever the deployment method, the features and functionality of ETO are the same. In addition to ETO, Mira also provides a Category Database service that enhances the built-in policy control provided by ETO and a Central Management system. These elements are detailed below.

4.3.1.1 ETO Physical Appliance

Mira ETO physical appliances are available with a range of interface speeds, from 1 Gbps to 40 Gbps, and they can decrypt up to 100 Gbps of TLS traffic total across all interfaces. Larger models support multiple independent segments, allowing visibility into different parts of the network from a single appliance.

4.3.1.2 ETO Virtual Appliance

Mira virtual ETO (vETO) is available to run on KVM and ESXi. It supports decryption of up to 5 Gbps of TLS traffic from a single virtual appliance.

4.3.1.3 ETO in Public Cloud

Mira ETO can be deployed in AWS to decrypt up to 5 Gbps of TLS traffic.

4.3.1.4 Mira Category Database Service

Mira's Category Database service is an optional subscription service that can be used by any ETO device to enhance the policy controls available on the device. When using the Category Database service,

policies can be set based on the category of the destination of a TLS flow. So, for example, a policy to not decrypt traffic to “health care” destinations can be specified.

4.3.1.5 Central Manager

The Mira ETO can be managed directly via an easy-to-use WebUI; it also provides a Representational State Transfer (REST) application programming interface (API), allowing programmatic control of the device and enabling integration into an existing management framework. In situations where many ETO devices are deployed, the Mira Central Management System (CMS) is available to simplify control and management of the devices. CMS allows devices to be grouped with shared policies and configurations, and it centralizes management of licensing for ETO devices.

4.3.2 AppViewX Key Governance Platform

AppViewX has partnered with NETSCOUT to develop a prototype key governance platform for the TLS 1.3 visibility challenge that it plans to formalize as an open industry standard. This project dovetails with the Secure Key Orchestration initiative that aims to secure and automate the management of all the encryption keys across distributed and hybrid enterprise environments. The AppViewX Cloud-native Identity and Security Platform is used by organizations across financial services, banking, healthcare, oil and gas, manufacturing, and high tech to reduce cybersecurity risk and meet security compliance requirements.

The AppViewX Platform facilitates enterprise-wide central certificate and key governance and lifecycle management to prevent outages, reduce security incidents, and protect the reputation and bottom lines of organizations through streamlined automation workflows. Delivered as a service, the modular AppViewX Platform and its CERT+ and PKI+ products address critical digital and machine identity challenges. AppViewX provides instant value by discovering all certificates across complex enterprise environments, building and maintaining inventories, provisioning both private and public trust certificates from any CA, alerting to expiring certificates, and fully automating renewals and revocation. By eliminating manual processes with AppViewX, enterprise organizations reduce errors, free up staff resources, and become more crypto agile to strengthen their overall security postures and meet essential compliance requirements.

AppViewX CERT+ is a next-generation automated certificate lifecycle management (CLM) solution that simplifies PKI and certificate management. It combines the best of automation, security, and insights to meet all enterprise PKI needs. CERT+ features are purpose-built to address both the operational and security challenges of certificate management to, in turn, help organizations prevent application outages and security breaches.

Setting up a secure, scalable, and compliant cloud-based private PKI is easier and faster than ever with AppViewX PKI+. Whether needing to comply with data protection mandates, enable ecosystem trust, or secure assets with strong authentication and encryption, PKI+ is a turnkey PKI-as-a-Service for all private trust use cases. PKI+ eliminates the need for expensive PKI hardware and allows you to simplify your private PKI architecture and set up tailored custom CAs in minutes while meeting the highest standards of security and compliance.

4.3.3 DigiCert CertCentral Enterprise Certificate Authority

DigiCert's CertCentral web-based platform allows provisioning and managing publicly trusted X.509 certificates for TLS and code signing as well as a variety of other purposes. After establishing an account, clients can log in, request, renew, and revoke certificates by using only a browser. Multiple roles can be assigned within an account, and a discovery tool can inventory all certificates within the enterprise. In addition to certificate-specific features, the platform offers baseline enterprise SaaS capabilities, including role-based access control (RBAC), Security Assertion Markup Language (SAML), single sign-on (SSO), and security policy management and enforcement. All account features come with full parity between the web portal and a publicly available API.

4.3.4 F5 SSL Orchestrator

The BIG-IP SSL Orchestrator enhances TLS infrastructure, makes encrypted traffic visible to security solutions, and optimizes existing security investments. It delivers dynamic service chaining and policy-based traffic steering—applying context-based intelligence to encrypted traffic handling to intelligently manage the flow of encrypted traffic across the security stack—and ensures optimal availability and security. The F5 BIG-IP SSL Orchestrator is designed and purpose-built to enhance TLS infrastructure, provide security solutions with visibility into TLS encrypted traffic, and optimize and maximize existing security investments. BIG-IP SSL Orchestrator delivers dynamic service chaining and policy-based traffic steering, applying context-based intelligence to encrypted traffic handling to allow you to intelligently manage the flow of encrypted traffic across your entire security stack, ensuring optimal availability. BIG-IP SSL Orchestrator centralizes TLS decryption across multiple security tools, inspects next-generation encryption protocols, simplifies change management through security stack orchestration, improves scalability and availability of existing enterprise security tools, configures dynamic service chaining based on context, deploys with flexible options to ease integration, and integrates F5 security solutions into enterprise service chains. The BIG-IP SSL Orchestrator is designed to easily integrate with existing architectures and to centrally manage the TLS decrypt/re-encrypt function. It delivers current TLS encryption technologies across the enterprise's entire security infrastructure. This enables discovery of hidden threats and prevention of attacks at multiple stages, leveraging existing security solutions. The BIG-IP SSL Orchestrator ensures that encrypted traffic can be decrypted, inspected by security controls, then re-encrypted. This delivers enhanced visibility to mitigate threats traversing the network.

4.3.5 NETSCOUT Visibility Without Borders Platform

NETSCOUT's Visibility Without Borders Platform includes its nGeniusONE Service Assurance platform vSTREAM™ virtual appliance, Omnis Cyber Intelligence console, and CyberStream network security sensors.

4.3.5.1 NETSCOUT nGeniusONE Service Assurance Platform

The nGeniusONE Service Assurance platform provides an overarching view into the performance characteristics of all infrastructure and application components associated with delivering IP-based services. With emphasis on service triage and network troubleshooting, the nGeniusONE platform combines real-time monitoring, historical analysis, and multi-layered analytics capabilities for a holistic performance management solution.

4.3.5.2 NETSCOUT vSTREAM

NETSCOUT's vSTREAM™ virtual appliance complements existing Adaptive Session Intelligence™ (ASI)-based instrumentation to provide the same smart data visibility within virtualized and cloud infrastructures that is already possible in physical environments. The vSTREAM is used for monitoring service-critical traffic running within virtualized or cloud infrastructures, monitoring services locally on a host, or as an aggregation point for multiple hosts. It seamlessly operates with NETSCOUT's nGeniusONE Service Assurance solution, nGeniusPULSE, and, with the addition of the NETSCOUT Cloud Adaptor, is an integral part of the NETSCOUT Smart Edge Monitoring solution.

4.3.5.3 NETSCOUT Omnis Cyber Intelligence

Serving as a centralized console for the Omnis Security platform, Cyber Intelligence analyzes Smart Data collected by Omnis CyberStreams, ISNGs or vSTREAMs running Cyber Modules, network baselines, and ATLAS or third-party threat intelligence feeds to detect all types of cyber threats and enable workflows for further visualization and investigation. Cyber Intelligence alerts can be sent to third-party SIEMs, and its data can be exported to third-party data lakes for further analysis by third-party applications.

Leveraging advanced threat detection techniques and cutting-edge machine learning algorithms, Omnis CyberStream ensures the detection of both known and zero-day threats. The Omnis Cyber Intelligence Network Detection and Response (NDR) platform provides a unified interface for efficient security event management. Seamlessly integrating with SIEM tools and offering automation through SIEM/SOAR (Security Orchestration, Automation, and Response) and Extended Detection and Response (XDR) systems, this solution empowers organizations to swiftly investigate and respond to security threats.

4.3.5.4 NETSCOUT vSTREAM Omnis CyberStream Module

Strategically deployed in any network environment (including public cloud), NETSCOUT CyberStream network security sensors use patented, highly scalable, DPI technology to convert raw packets into a rich source of layer 2-7 metadata that provides cybersecurity teams comprehensive network visibility and a rich source of data for better network threat detection and response.

4.3.6 Not for Radio Encryption Visibility Agent (EVA)

The demonstration systems constructed for this project employ NFR's Encryption Visibility Agent™ (EVA™) in its Bounded Lifetime Key Control mode, with an external key management system configured as the source of the bounded-lifetime key material. With this configuration, the Agent runs within the applications of interest and enforces the use of the controlled, bounded-lifetime Diffie-Hellman key material in TLS 1.3 sessions. Importantly, the Agent's operation does not introduce new pathways for lateral movement of malware by requiring relaxation of any platform security mechanisms.

Other modes of operation of the EVA, such as high performance and fully deterministic reporting of per-session key material, as well as distributed bounded-lifetime key generation, are not used in this demonstration. Additional components of the Encryption Visibility Architecture™ family designed to address scalability and integration challenges within larger deployments are likewise not used.

4.3.7 Nubeva TLS Visibility Solution

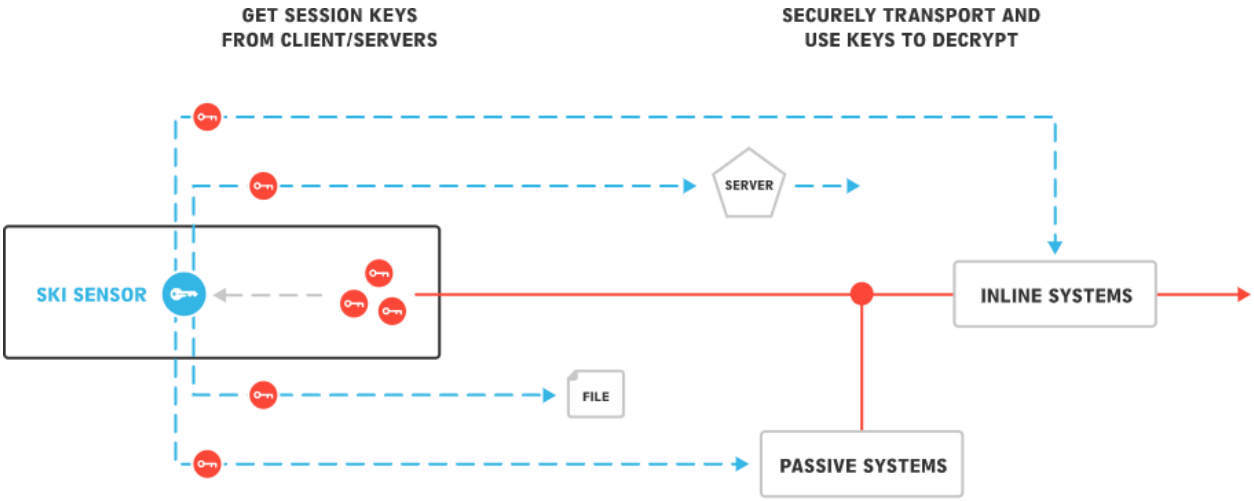
The TLS Visibility solution uses session key intercept (SKI) to support next-generation TLS decryption. By detecting and protecting against malicious behaviors concealed within encrypted network communications, this solution addresses the growing limitations of traditional decryption solutions and helps eliminate major blind spots in cybersecurity.

SKI for TLS is a software system that provides efficient decryption of all TLS traffic, including TLS 1.3, pinned certificates, and mutual authentication sessions. It is designed to serve both passive and inline applications, addressing the deficiencies of traditional decryption methods, and enabling thorough packet inspection in modern environments.

4.3.7.1 Session Key Intercept

SKI is employed both as an augmentation to legacy forward and reverse-proxy engines and as a fix to passive intercept systems due to pending obsolescence from forward secrecy. SKI provides the ability to GET SESSION KEYS from TLS clients and servers in real-time and to USE SESSION KEYS to decrypt TLS on authorized systems to enable DPI. With session keys available, one can decrypt any session with simple and efficient bulk decryption. As such, SKI is universal to all traffic flows and use cases and applications: inbound, outbound, east-west, clients, servers, data center, cloud. Since TLS session keys are symmetric (shared by both endpoints), keys only need to be obtained from one side of a connection and therefore can apply to client connections to foreign servers and services. Figure 4-1 depicts the SKI configuration.

Figure 4-1: Session Key Intercept



Unlike legacy man-in-the-middle/forward proxy, session pre-termination/reverse proxy, and RSA Passive Intercept, SKI does not involve certificates or server keys, nor does it manipulate or change traffic, connections, authentication, or PKI. Instead, it simply works with the individual TLS session encryption keys (ephemeral, symmetric, or bulk encryption keys) that are developed during the handshake, shared by the TLS client and servers, used for the bulk encryption/decryption of the communication, and then discarded. SKI can passively decrypt forward secrecy-based traffic as well as traffic to external servers and services and thus re-establishes the out-of-band decryption option for the industry. SKI delivers high-speed decryption of all TLS-encrypted traffic from any standard passive traffic source, including

taps, in any environment (physical, virtual, on-prem, or cloud). SKI is delivered as a modular suite of software components.

4.3.7.2 FastKey Protocol

Network probes need to inspect traffic at very high speeds with minimal delays. These probes would like to receive keys not more than 1 millisecond after receiving an encrypted packet, and preferably less. Low-latency key extraction reduces the time decryptors wait for keys to less than 200 microseconds. The best-case scenario is that keys are received by a decryptor before the corresponding packet.

FastKey combines a binary protocol over Datagram Transport Layer Security (DTLS) and a REST API to send keys to key targets. The REST API uses a JSON object. The object is a key:value pair, where the key is a "client random" and the value is a set of key fields and a metadata structure. Metadata is not required for decryption.

4.3.8 Thales Component HSM

A hardware security module (HSM) is a dedicated crypto processor that is specifically designed for the protection of the crypto key lifecycle. HSMs act as trust anchors that protect the cryptographic infrastructure of some of the most security-conscious organizations in the world by securely managing, processing, and storing cryptographic keys inside a hardened, tamper-resistant device. Thales HSMs always store cryptographic keys in hardware. They provide a secure crypto foundation, as the keys never leave the intrusion-resistant, tamper-evident, FIPS 140-validated appliance. See the link for more information on FIPS validation. Since all cryptographic operations occur within the HSM, strong access controls prevent unauthorized users from accessing sensitive cryptographic material. Thales also implements operations that make the deployment of secure HSMs as easy as possible. They are integrated with Thales Crypto Command Center for quick and easy crypto resource partitioning, reporting, and monitoring.

4.3.9 JPMorgan Chase Contribution

JPMorgan Chase manages large-scale network operations with many customers and partners. The network traffic is TLS-protected. Security and reliability considerations require continuous monitoring and analytics to support threat and incident detection, auditing, and forensics. The analytics processes require both real-time and post-facto visibility into traffic metadata and contents. As such, JPMorgan Chase is providing content, protocol, and performance requirements and constraints information that supports project functional objectives.

4.3.10 U.S. Bank Corporation Contribution

U.S. Bank Corporation manages large-scale network operations with many customers and partners. The network traffic is TLS-protected. Security and reliability considerations require continuous monitoring and analytics to support threat and incident detection, auditing, and forensics. The analytics processes require both real-time and post facto visibility into traffic metadata and contents. As such, U.S. Bank is providing content, protocol, and performance requirements and constraints information that supports project functional objectives.

5 Architecture

Some aspect of analytics functions requiring enterprise visibility into its encrypted TLS 1.3 traffic may require combining network architecture and key-management techniques to achieve operationally necessary visibility. Necessary analytics functions may include identification of causes of network performance degradation or failures, key management-based communications failures, detection and identification of anomalous received data, identification of sources of anomalous data, and detection of encrypted traffic from unauthorized sources and exfiltration of enterprise data to anomalous destinations.

Therefore, the scope of the project includes demonstration of an architecture that achieves visibility inside the enterprise data center through tools that intercept and decrypt traffic without altering the traffic flow between the TLS clients and servers. No change to the TLS 1.3 protocol is proposed. TLS 1.3 continues to be used for exchanges between data centers. In this demonstration project, we examine TLS 1.3 deployment within the enterprise data center and address mechanisms that may be used to support access to historical data by leveraging key management-based and middlebox solutions.

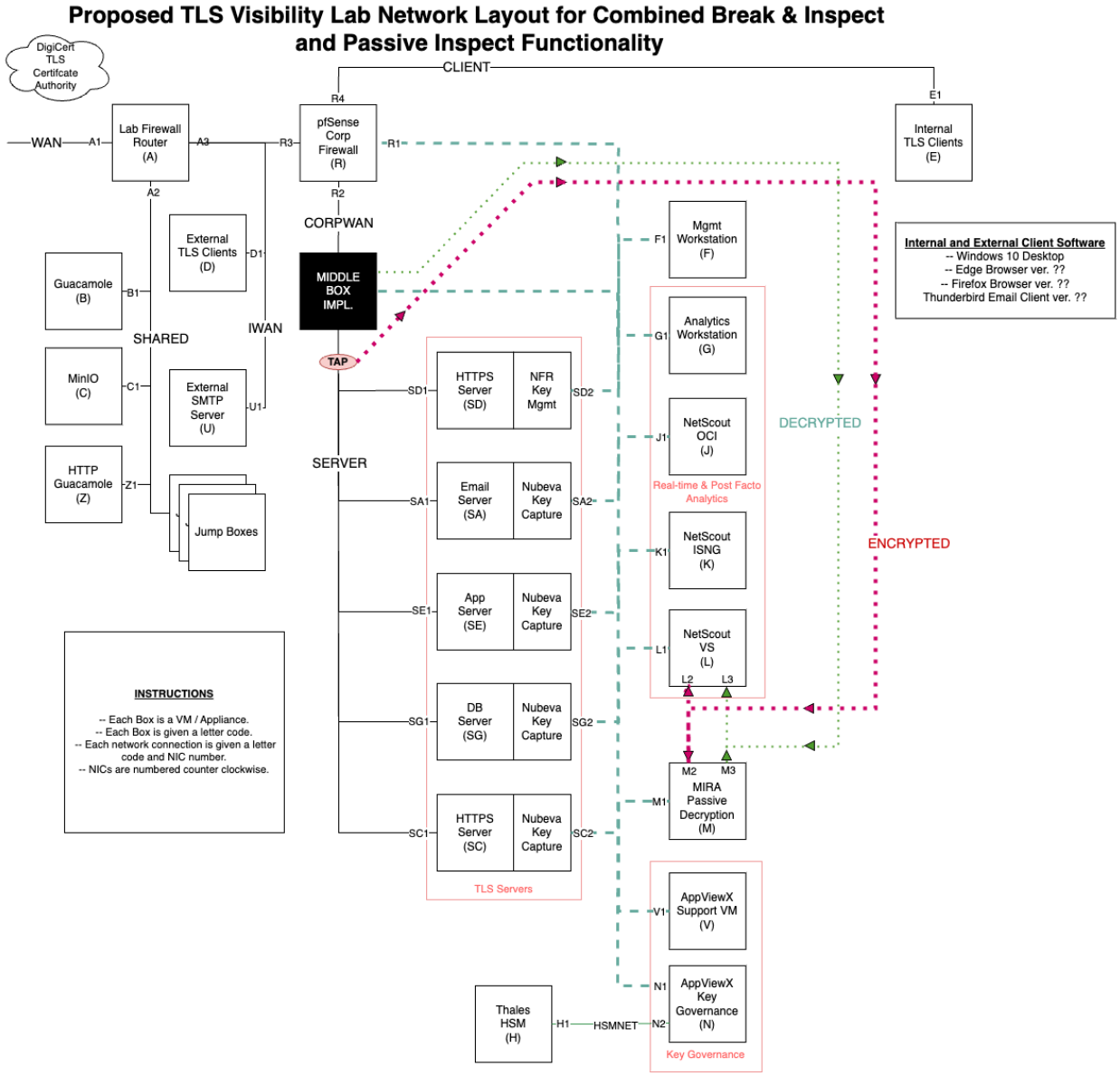
5.1 Data Center Architecture Description

The project's data center architecture employs virtual machine technology to interconnect and manage the components provided by the collaborators. Figure 5-1 depicts the laboratory demonstration environment for collaborator-contributed components. Note that many components are connected to more than one internal network, and that permissions for each connection are tailored to enforce the principle of least privilege.

5.1.1 NCCoE Laboratory Network

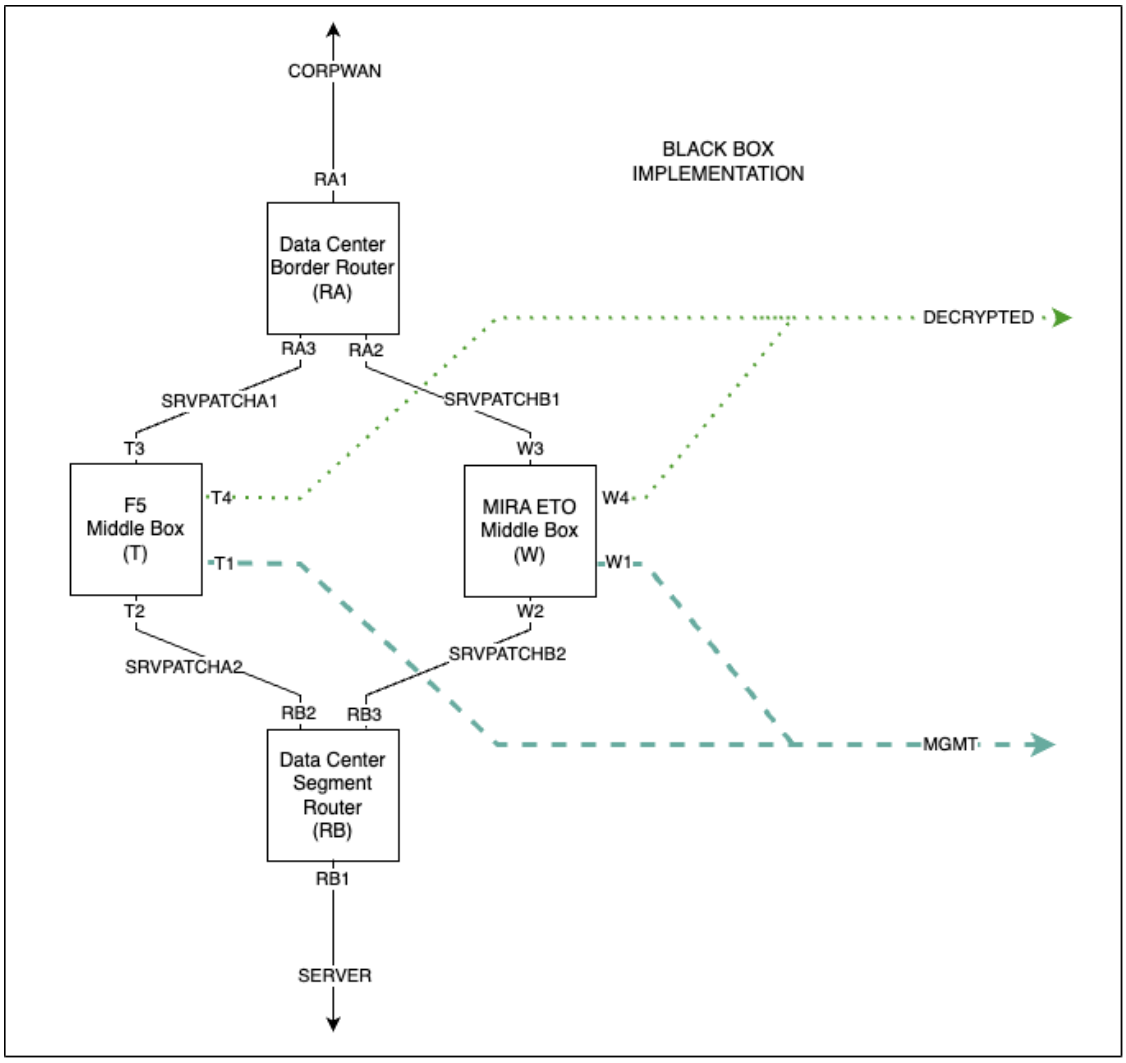
The NCCoE laboratory network supports NCCoE projects and protects in-house connections such that intellectual property resident in one NCCoE project's laboratory is not exposed to the other laboratories. Figure 5-1 depicts the TLS 1.3 Visibility Laboratory Network layout.

1141 **Figure 5-1: TLS 1.3 Visibility Laboratory Network**



1142 Two different middlebox options are included in the TLS 1.3 visibility demonstration. One employs an F5
1143 middlebox, and the other employs a Mira middlebox. Figure 5-2 depicts the detailed laboratory
1144 middlebox connectivity.

1145 **Figure 5-2: Middlebox Connection Detail**



1146 5.1.2 Internet

1147 The TLS visibility laboratory permits component connections to the Internet via a firewall router. The
1148 laboratory accesses external components such as DigiCert's TLS certification authority via its Internet
1149 connection.

1150 5.1.3 TLS Subnetwork

1151 The TLS subnetwork includes connections necessary for interfacing and managing all project
1152 components other than the external CA. Processors representing external TLS clients, internal TLS
1153 clients, and an internal Simple Mail Transfer Protocol (SMTP) server connect to other components
1154 through a pfSense corporate firewall.

5.1.4 Management Network

The management network provides the connections necessary to manage collaborator-provided components. It is also a protected network for key exchanges and analytics.

5.1.5 Server Network

The server network is the production network and hosts all TLS server endpoints.

5.1.6 Client Networks

The client network hosts all internal TLS client endpoints.

5.1.7 HSM Network

The HSM network is the protected network used for master key exchange between the Thales HSM and the AppViewX key governance component.

5.1.8 Encrypted Traffic Capture Network

The encrypted traffic capture network carries all tapped encrypted traffic for demonstration captures.

5.1.9 Decrypted Traffic Network

The decrypted traffic network aggregates and carries traffic that has been decrypted regardless of source.

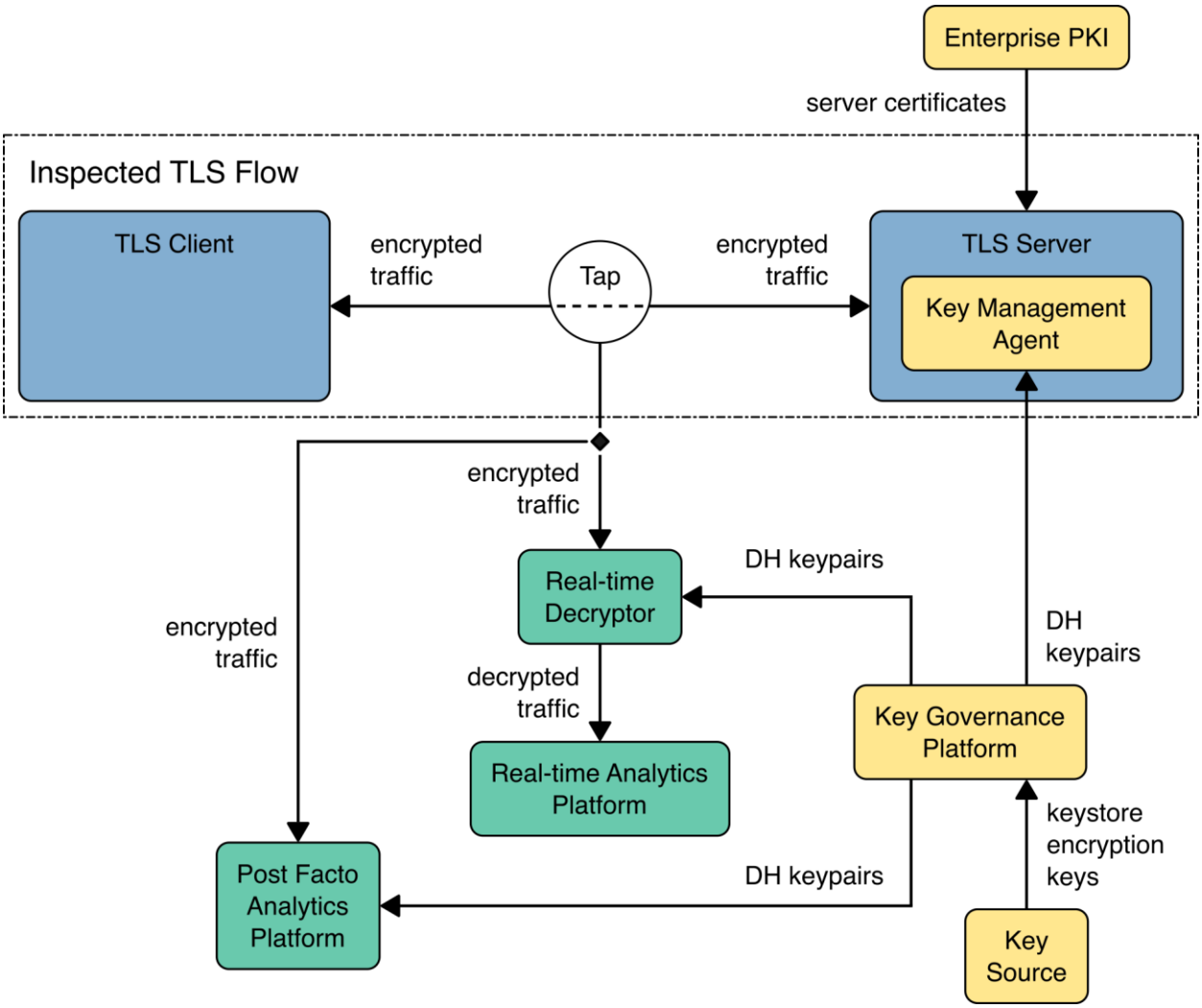
5.1.10 Server Patch Networks

The server patch networks connect the data center network to the production server network through the middlebox implementations. These networks are necessary to isolate the middlebox implementation details from the overall data center network implementation.

5.2 High-Level Passive Inspection Architecture Overview

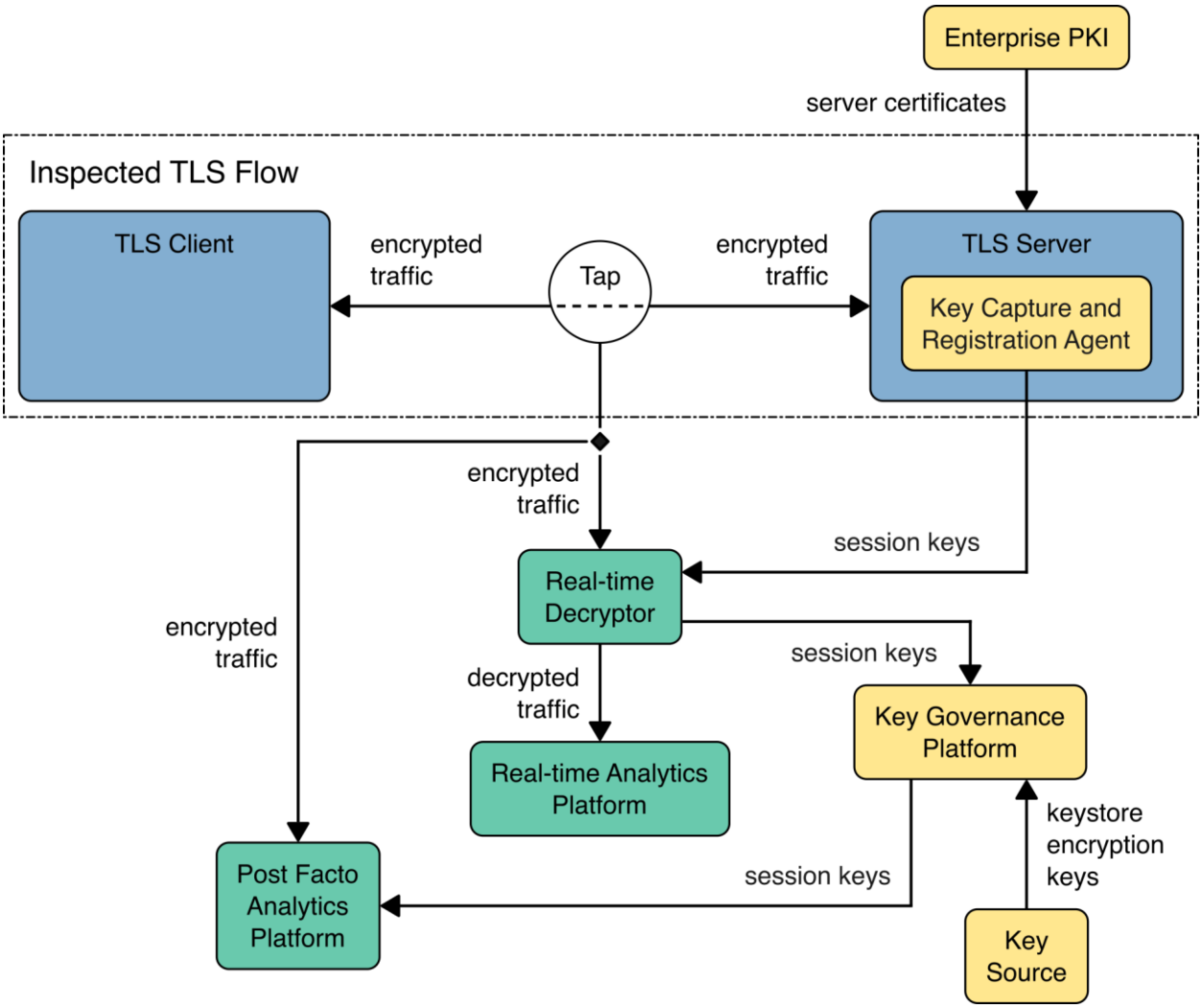
The figures below depict the functional components of a passive decrypt and inspect demonstration architecture. Figure 5-3 depicts passive inspection using rotated bounded-lifetime DH keys on the destination TLS server. This approach can be used to capture decrypted traffic for real-time analysis, incoming traffic for post-facto or historical analysis, or both. Note that the clients internal to the enterprise that are recipients of the TLS 1.3-protected traffic from the TLS server are not depicted.

1180 **Figure 5-3: Passive Inspection Functional Architecture – Bounded-Lifetime DH**



1181 Figure 5-4 depicts passive decryption and inspection using exported session keys. The architecture
1182 permits both real-time analysis of decrypted TLS traffic and post-facto analysis of stored encrypted
1183 traffic. It is worth noting that exported session keys can be used to decrypt TLS traffic irrespective of the
1184 TLS version and cipher suite being used by the session.

1185 **Figure 5-4: Passive Inspection - Exported Session Key Functional Architecture**



1186 5.2.1 Passive Inspection Components

1187 The function of each passive inspection component is described as follows:

1188 - **TLS Client Devices:** Devices that initiate encrypted traffic.
1189 - **Network Tap:** Component that provides a copy of traffic from a network segment.
1190 - **Real-Time Decryption:** Passive decrypt component that decrypts and forwards the copied
1191 traffic.
1192 - **Real-Time Analytics Platform:** Set of tools for examining decrypted payloads to identify
1193 undesired characteristics:
1194 - Identification of causes of network or application performance degradation or failures
1195 - Key management-based communications failures
1196 - Detection and identification of received anomalous data

- Identification of sources of anomalous data, and detection of traffic from unauthorized sources
- Identification of legitimate enterprise data being exfiltrated to anomalous destinations

Note that transfers of information even within the enterprise and any information stored on or by the analytics platform require cryptographic protection or compensating physical controls.

- **Traffic Capture Platform:** Encrypted storage of captured traffic to allow subsequent analytics of captured traffic. This can be encrypted storage of the captured decrypted traffic or storage of the captured original encrypted traffic.
- **Key Governance Platform:** Security module performing storage and distribution of keys (e.g., discover, create, renew, provision, revoke, and destroy certificates and keys). Bounded-lifetime DH keys are pushed to the TLS server and passive decrypt device providing real-time decryption; they are also stored for future use by decrypt solutions that work with captured encrypted sessions. Exported session keys and flow identification data are received from the Session Key Capture agent or the decrypt platform, providing real-time decryption using these keys and stored for future use by decrypt solutions that work with captured encrypted sessions.
- **TLS Server:** Peer for encrypted traffic that generates session keys, negotiates encryption protocols, and connects to key management infrastructure.
- **Bounded-Lifetime DH Key Management Agent:** Receives the bounded-lifetime keys from the key governance platform and enables their use by the TLS server according to key governance platform policy.
- **Session Key Capture and Registration Agent:** Captures and exports the ephemeral session keys and registers the captured keys and flow identification data with the key governance platform.
- **Enterprise Public Key Infrastructure:** CA that provides enterprise public key certificates.

5.2.2 Passive Inspection Functionality

These passive inspection architectural options support the capture of traffic from monitored network segments, providing mechanisms to decrypt the traffic for immediate analysis, or forwarding encrypted traffic for storage structured with session information and the appropriate key information. This architecture does not terminate or otherwise modify traffic between the TLS clients and servers.

5.2.2.1 Bounded-Lifetime Key Pair (Bounded-Lifetime Diffie-Hellman)

For TLS using bounded-lifetime key pairs, the key governance platform provisions the TLS server with bounded-lifetime DH key pairs via a key management agent running on the server. The TLS server then uses the provisioned keys pairs instead of performing ephemeral generation during the normal TLS handshake. The key governance platform provisions the server with new bounded-lifetime DH key pairs on a frequent basis via the agent. A decrypt platform that has the bounded-lifetime DH key pairs used by the TLS server to establish TLS sessions will be able to passively decrypt all TLS sessions to the server for the period during which the server uses those DH key pairs.

5.2.2.2 Exported Session Key (Ephemeral Diffie-Hellman)

For the exported session key option, passive decryption is accomplished by retrieving TLS session keys from the key governance platform or receiving them from the Session Key Capture agent in the case of real-time decryption. This architecture uses agents operating on the TLS servers to capture the ephemeral session keys at the time they are negotiated for a TLS flow. In the case of capture of encrypted flows for post-facto or historical analysis, these agents register the captured keys with the key governance platform. These keys can be retrieved from the key governance platform using the TLS session's client-random-id as the flow identification mechanism. This passive decrypt mechanism works regardless of what TLS version and cipher suite is negotiated between the client and server.

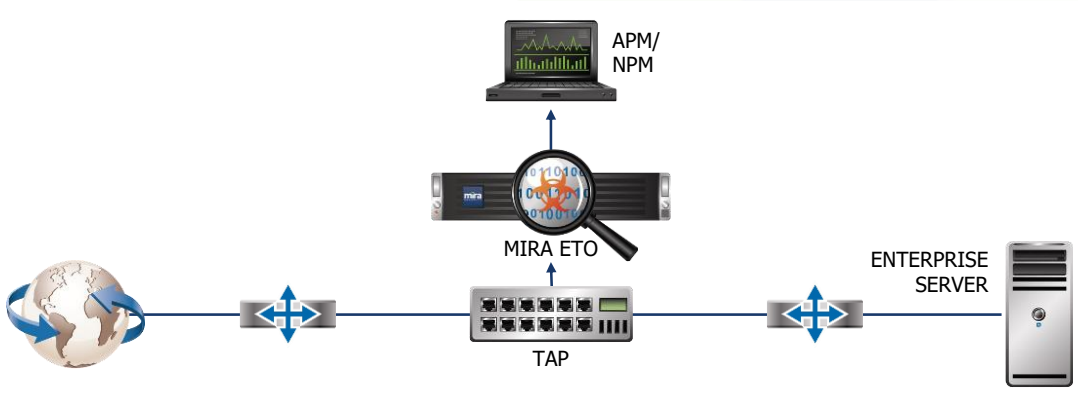
5.2.2.3 Mira ETO Use in Passive Decryption Mode

The use of the Mira ETO in a classic passive decryption mode is shown in Figure 5-5 (Passive-Passive). The Mira ETO is installed out-of-band with copies of network packets being received from a network tap. Decrypted versions of the TLS flows are sent to one or more passive tools by the Mira ETO.

The Mira ETO supports real-time passive decryption of TLS 1.3 traffic when it receives copies of the ephemeral session keys from the Nubeva TLS Visibility Solution agent running on the server. This allows real-time decryption of the traffic by the Mira ETO. The Mira ETO also stores the ephemeral session keys so that it can forward them to a key management system, allowing tools that do post-facto decryption to do so. The ephemeral keys are sent to AppViewX using the protocol employed by the Nubeva TLS Visibility Solution agent. This allows AppViewX to either receive keys forwarded by the Mira ETO or directly from Nubeva.

The Mira ETO supports passive real-time decryption when bounded-lifetime DH keys are in use on the server. The Mira ETO needs to receive these keys in advance of their use by the server to allow for real-time decryption. The implementation permits these keys to be pushed to ETO via an API.

Figure 5-5: Mira ETO Use in Passive Decryption



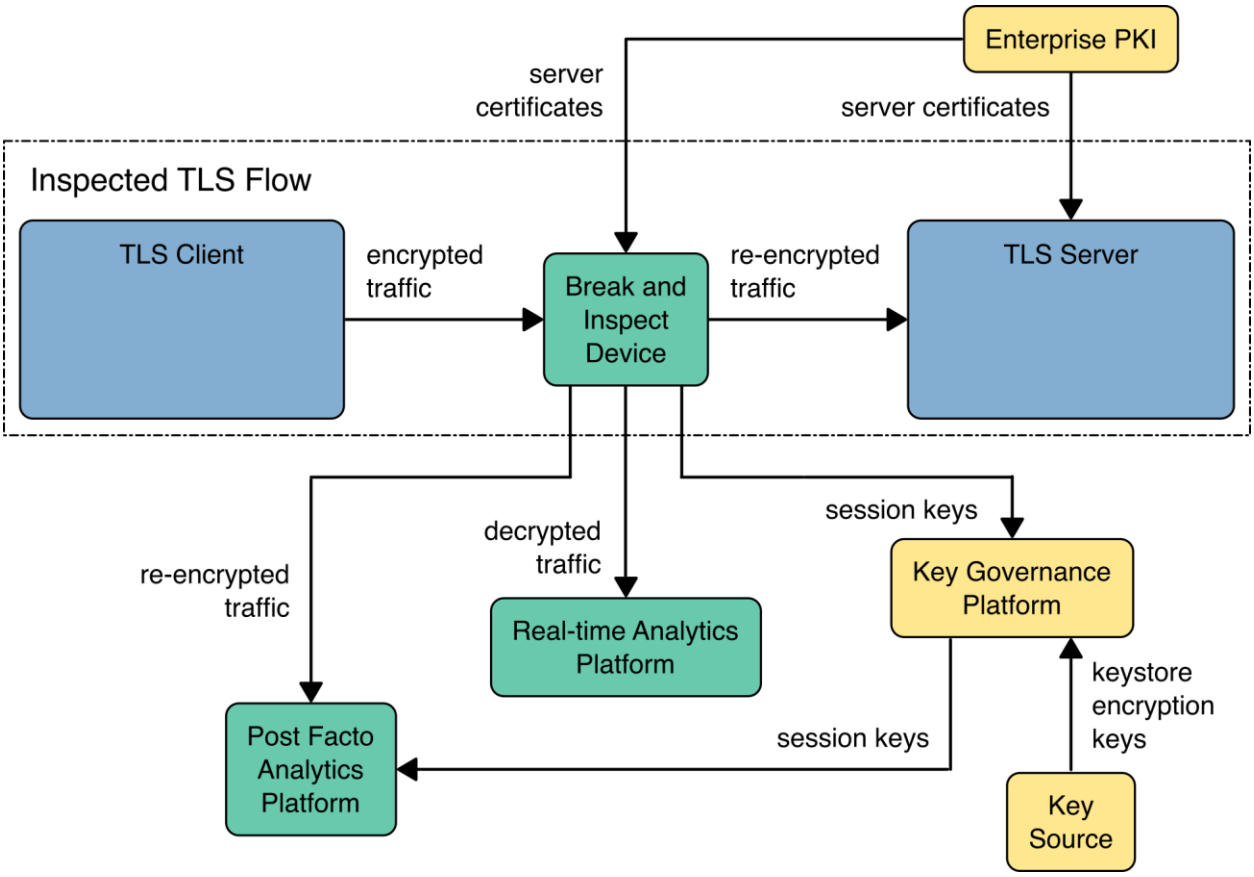
5.3 High-Level Middlebox Architecture Overview

Some aspect of analytics functions requiring enterprise visibility into encrypted TLS 1.3 traffic may require a middlebox approach that combines network architecture and key-management techniques to achieve operationally necessary visibility. Necessary analytics functions may include identification of

causes of network performance degradation or failures, key management-based communications failures, detection and identification of anomalous received data, identification of sources of anomalous data, and detection of encrypted traffic from unauthorized sources. Therefore, the scope of the project includes demonstration of an architecture that achieves visibility inside the data center through tools that break and inspect traffic. These "middleboxes" are commonly used at the enterprise edge to achieve real-time visibility. In this demonstration project, we examine deployment within the enterprise and address access to historical data by leveraging key management-based solutions.

Figure 5-6 depicts the functional components of the break and inspect (B&I) demonstration architecture.

Figure 5-6: Middlebox (Break and Inspect) Functional Architecture



5.3.1 Break and Inspect Middlebox Component Descriptions

The function of each B&I middlebox component is described as follows:

- **TLS Client Devices:** Devices that initiate encrypted traffic. These TLS client devices may be located outside of the data center. However, note that B&I is not using bounded-lifetime DH or ephemeral key reporting means to gain visibility, so the TLS 1.3 session from an external client to the B&I device and from the B&I device to the server both have forward secrecy.
- **Break and Inspect Component:** Component that terminates, decrypts, and re-encrypts/reinitiates TLS traffic.

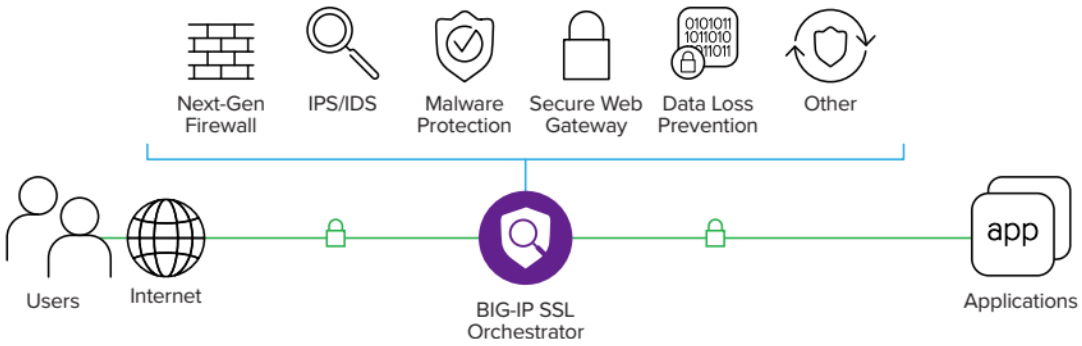
- **Real-Time Analytics Platform:** Set of tools for examining unencrypted payloads to identify undesired characteristics:
  - Identification of causes of network or application performance degradation or failures
  - Key management-based communications failures
  - Detection and identification of anomalous received data
  - Identification of sources of anomalous data, and detection of traffic from unauthorized sources

  Note that transfers of information even within the enterprise and any information stored on or by the analytics platform require cryptographic protection or compensating physical controls. Also note that in the example above, the B&I device feeds analytics tools with a copy of the decrypted TLS traffic (i.e., the analytic tool is passive and simply consumes the feed). B&I devices are also capable of feeding the decrypted TLS traffic to inline security tools which may modify the decrypted traffic before returning it to the B&I device for re-encryption and forwarding to the final destination. In the use case above, with passive analytic tools the end-to-end payload between client and server is unmodified, whereas the use of inline tools may result in modification.
- **Traffic Capture Platform:** Encrypted storage of captured decrypted traffic or storage of the captured original encrypted traffic to allow subsequent analytics of captured traffic.
- **Key Governance Platform:** Security module performing storage and distribution of ephemeral session keys and associated flow identification data provided by the B&I device for later use by a passive decrypt device working on captured encrypted traffic.
- **TLS Server:** Counterparty for encrypted traffic that generates session keys, negotiates encryption protocols, and connects to the enterprise PKI infrastructure.
- **Enterprise PKI:** CA that provides enterprise key certificates.

5.3.1.1 F5 Middlebox Build Component

F5 BIG-IP SSL Orchestrator (SSLO) provides an all-in-one appliance solution designed specifically to optimize the SSL infrastructure, provide security devices with visibility of TLS-encrypted traffic, and maximize the efficient use of existing security resources. The SSL Orchestrator makes encrypted traffic visible to security solutions and optimizes existing security elements. It delivers dynamic service chaining and policy-based traffic steering by applying context-based intelligence to encrypted traffic handling to intelligently manage the flow of encrypted traffic across the security stack. The placement of the BIG-IP SSLO as a middlebox is depicted in Figure 5-7.

1311 **Figure 5-7: F5 BIG-IP SSL Orchestrator Use in TLS 1.3 Visibility Functional Architecture**



1312 The SSL Orchestrator is used in one demonstration option as a middlebox component that terminates,
1313 decrypts, and re-encrypts/reinitiates traffic. As a middlebox it delivers high-performance decryption of
1314 inbound and outbound TLS traffic, decrypting incoming encrypted traffic to permit inspection for
1315 ransomware, malware, or other threats that can lead to attacks, infections, and other data breaches. Its
1316 use employs full-proxy and diverse cipher support without requiring architectural changes.

1317 The SSL Orchestrator also provides central control to unify decryption across multiple inspection devices
1318 to detect unsupported cipher use, fake TLS connections, and unwanted infrastructure complexity. It uses
1319 a flexible context engine to group, monitor, and steer traffic regardless of network topology, protocol, or
1320 cipher in use. It employs dynamic service chaining with existing security solutions based on the type of
1321 incoming traffic.

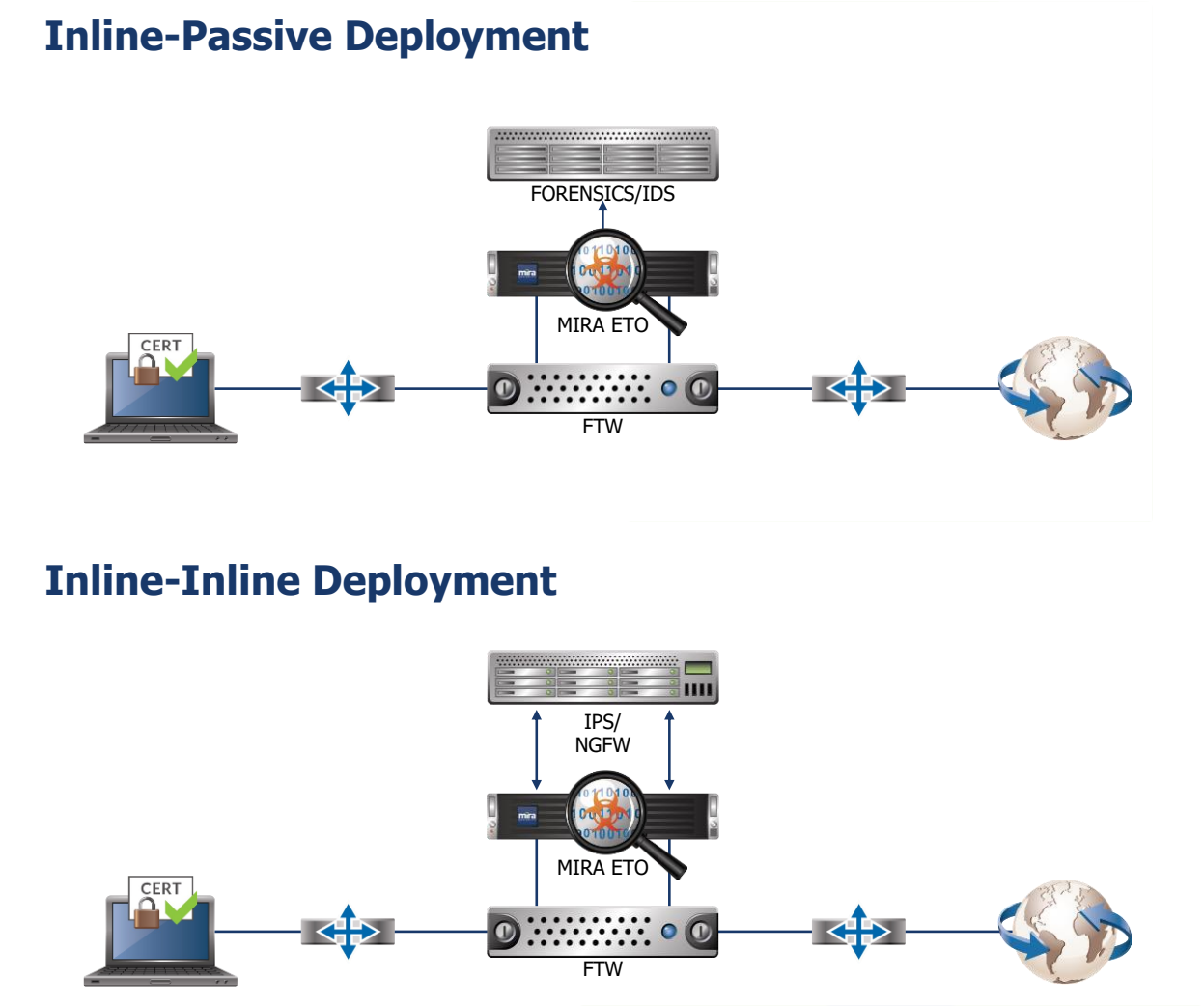
1322 F5 SSL Orchestrator is available on BIG-IP appliance and VELOS chassis hardware platforms. It can either
1323 be a dedicated standalone SSL Orchestrator or an add-on to existing F5 LTM devices performing load
1324 balance duty. In software-only format, SSL Orchestrator can run as a virtual appliance in private or public
1325 clouds.

1326 For this project, SSL Orchestrator is deployed as a virtual appliance running on VMware in a two-arm
1327 configuration.

1328 *5.3.1.2 Mira Middlebox Build Components*

1329 Mira ETO software supports B&I mode on all types of appliances, physical, virtual (KVM and ESXi) and
1330 when deployed in public cloud (AWS). In this project architecture, the Mira ETO is installed inline and
1331 can provide real-time decryption and re-encryption of TLS traffic to maintain an end-to-end TLS
1332 connection between the client and the server. Inline interfaces (real or virtual) create a bump in the
1333 wire. Decrypted versions of the traffic can be sent to both passive and inline tools. Passive tools
1334 consume the decrypted stream and generate alerts. Inline tools process the decrypted stream, then
1335 return it (potentially modified) to the ETO for re-encryption before it is sent onwards to the client or
1336 server. There is a single interface to passive tools and two interfaces to inline tools. Figure 5-8 shows the
1337 two B&I modes (Inline-Passive and Inline-Inline). Note that when Inline-Inline is in use, it is possible at
1338 the same time to feed passive security tools via a copy port.

1339 **Figure 5-8: Mira ETO Use in Middlebox (Break and Inspect) Functional Architecture**



1340 For traffic destined for an enterprise server with a server certificate issued by either a public CA or an
1341 Enterprise CA, the Mira ETO needs a copy of the server certificate and the associated private key. This is
1342 the primary NIST use case for B&I. Options exist to have the server certificates and private keys installed
1343 in the ETO be protected by keys stored in an HSM. For traffic from an enterprise client destined for an
1344 external TLS server, the Mira ETO needs an intermediate Enterprise CA installed so that it can generate
1345 modified server certificates for the client to do B&I decryption. The enterprise client needs to be
1346 configured to trust the Enterprise root CA and any intermediate CAs, such as the one installed in the
1347 ETO. Options exist to have the intermediate CA and private keys be stored in an HSM rather than on the
1348 ETO. The Mira ETO captures the negotiated session keys for the TLS sessions on either side of the
1349 middlebox B&I point to share these with systems that will use them to perform post-facto decryption of
1350 captured encrypted flows. The ETO shares these keys with the key management system using the
1351 Nubeva TLS Visibility Solution agent protocol.

5.3.2 Break and Inspect Functionality

The B&I middlebox architecture supports capturing incoming traffic, providing tapped decrypted traffic to an analytics platform, storing traffic re-encrypted using new session keys negotiated between the B&I device and the client and/or server, and passing re-encrypted traffic from the B&I component to an enterprise network server for routing to the enterprise's in-house consumers. The architecture also includes connection by the server and B&I component to the enterprise PKI infrastructure. The capture of traffic may be accomplished between the client and the B&I device, the B&I device and the server, or both. The session keys between the client and the B&I device, or the B&I device and the server, or both need to be provided to the key governance platform to enable subsequent analysis of captured encrypted data.

5.3.2.1 F5 Middlebox Build Functionality

The SSL Orchestrator can be used as a traditional B&I middlebox. It can be configured to feed inline tools, post-facto analytic tools, or both. The SSL Orchestrator decrypts incoming traffic that may be routed to a real-time analytics platform or platforms. The SSL Orchestrator re-encrypts the decrypted traffic and feeds the re-encrypted traffic to the enterprise TLS endpoint server. It can also feed the decrypted traffic to an analytics platform for later decryption and forensics analysis. When operating in B&I mode, for traffic destined for an enterprise server with a server certificate issued by either a public CA or an Enterprise CA, the SSL Orchestrator needs a copy of the server certificate and the associated private key. As stated in Section 5.3.1.1, this is the primary NIST use case for B&I. The BIG-IP SSL Orchestrator is inline between the TLS client and TLS server.

In the situation where visibility is required for inbound TLS traffic terminating on TLS servers within the enterprise, the SSL Orchestrator Is normally configured with copies of the TLS server's certificate and keys, allowing it to appear to the TLS client(s) as if it is the server. As in the case of the Mira ETO (see Section 5.3.2.2), the SSL Orchestrator uses the server certificate and keys as part of a TLS handshake with the TLS client and carries out a second TLS handshake with the TLS server. Again, as described for the ETO, because TLS 1.3 requires forward secrecy, the TLS handshakes on either side of the SSL Orchestrator will result in different ephemeral session keys being used for the two TLS connections. If the SSL Orchestrator is providing copies of ephemeral keys to the key management system for later use by analytic tools to decrypt captured TLS flows, it is important to ensure that the ephemeral keys matching the captured flow are exported by the SSL Orchestrator.

Full SSL Orchestrator functionalities can be found in the BIG-IP SSL Orchestrator data sheet.

5.3.2.2 Mira Build Functionality

The Mira ETO can be used as a traditional B&I middlebox and can be configured to feed passive tools, inline tools, or both with the decrypted traffic. When operating in B&I mode, the ETO is configured as either an "Inline-Passive" or "Inline-Inline" segment. The first part of the segment type refers to the network interfaces; "Inline" means the ETO is inline between the TLS client and TLS server, i.e., it is a bump in the wire. The second part of the segment type refers to how the decrypted traffic is fed to attached analytic tools; "Passive" means that the tool consumes the traffic and does not return it to the ETO, while "Inline" means the tool processes the decrypted traffic and then returns it to the ETO. When connected to an Inline tool, the ETO can be configured to allow modifications of the decrypted payload

by the tool to be re-encrypted and propagated to the TLS endpoint, or to ignore any modifications made by the tool so that the payload to the TLS endpoint is not changed.

In the situation where visibility is required for inbound TLS traffic terminating on TLS servers within the enterprise, the ETO Is normally configured with copies of the TLS server's certificate and keys, allowing it to appear to the TLS client as if it is the server. The ETO uses the server certificate and keys as part of a TLS handshake with the TLS client and carries out a second TLS handshake with the TLS server. Because TLS 1.3 requires forward secrecy, the TLS handshakes on either side of the ETO will result in different ephemeral session keys being used for the two TLS connections. If the ETO is providing copies of ephemeral keys to the key management system for later use by analytic tools to decrypt captured TLS flows, it is important to ensure that the ephemeral keys matching the captured flow are exported by the ETO.

The Mira ETO B&I mode will work no matter what certificate/keys are being used by the enterprise TLS server. The ETO will use the appropriate decrypt mechanism depending on the certificate/keys used by the server:

- If the server has a TLS server certificate that was issued by a public or private CA, the ETO needs a copy of the certificate and the private key.
- If the server has a self-signed server certificate, the ETO will simply change the public/private key pair associated with the certificate, as no authentication is being done using the certificate.

TLS server certificates and private keys stored on the ETO are encrypted and only exist in memory in a decrypted state when being used. Additional security can be enabled in the ETO, allowing a key stored in an external HSM to be part of the encryption flow for stored TLS private keys, ensuring that these can only be decrypted if the ETO has access to the enterprise HSM.

Full details of the Mira ETO capabilities are provided in the physical appliance getting started guide and the physical appliance administration guide. The virtual appliance (KVM, ESXi, AWS) has a different virtual appliance getting started guide and virtual appliance administration guide.

6 Security Characteristic Analysis

The purpose of the security characteristic analysis is to understand the extent to which the project meets its objective of demonstrating visibility using the TLS 1.3 protocol. In addition, it seeks to understand the security benefits and drawbacks of the example solution.

6.1 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.

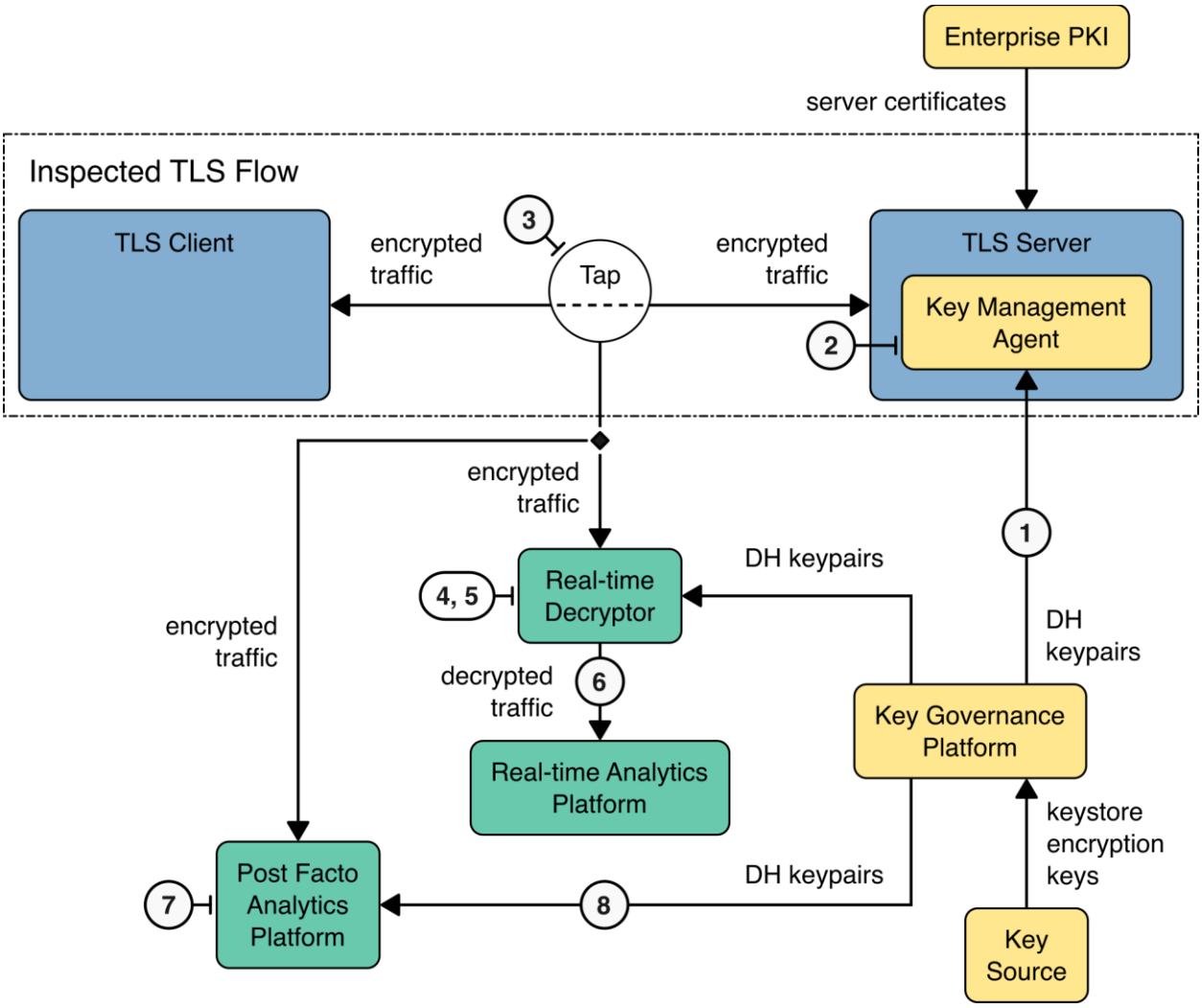
6.2 Build Demonstration

As documented in this preliminary draft volume, the project team has demonstrated capabilities that include visibility into 1) post-facto decryption of traffic protected under bounded-lifetime DH server keys, 2) real-time decryption of traffic protected under both bounded-lifetime and exported session keys, and 3) real-time decryption of traffic using middleboxes (break and inspect) in a TLS 1.3 environment. Work is underway to demonstrate post-facto decryption of traffic protected under exported session keys and using middleboxes. These capabilities will be demonstrated in a subsequent draft of this volume (see Section 7).

6.2.1 Bounded-Lifetime DH Flow

Figure 6-1 depicts the elements involved in the bounded-lifetime DH demonstration. Both post-facto and real-time decryption capabilities are illustrated.

Figure 6-1: Bounded-Lifetime DH Passive Inspection Elements



6.2.1.1 Real-Time (RT) Decryption

The demonstration of real-time decryption using bounded-lifetime DH keys executed the following sequence:

1. Before an epoch* begins, the Key Governance Platform generates a new bounded-lifetime DH key pair and pushes it to the TLS Server and the RT Decryptor.
2. When the epoch begins, the Key Management Agent configures the TLS Server to use the new bounded-lifetime DH key pair to negotiate a new TLS session with the client.
3. The Network Tap captures encrypted packets between client and server and forwards them to the RT Decryptor.
4. The RT Decryptor calculates the session symmetric keys from the captured traffic and the known server key pair.
5. The traffic is decrypted using the calculated session symmetric keys.
6. The decrypted traffic is forwarded to the RT Analytics Platform.

*An epoch is the rotation period for keys determined by the enterprise key governance platform.

6.2.1.2 Post-Facto Decryption Flow

The demonstration of storage of traffic for post-facto decryption and analysis using bounded-lifetime DH keys executed the following sequence:

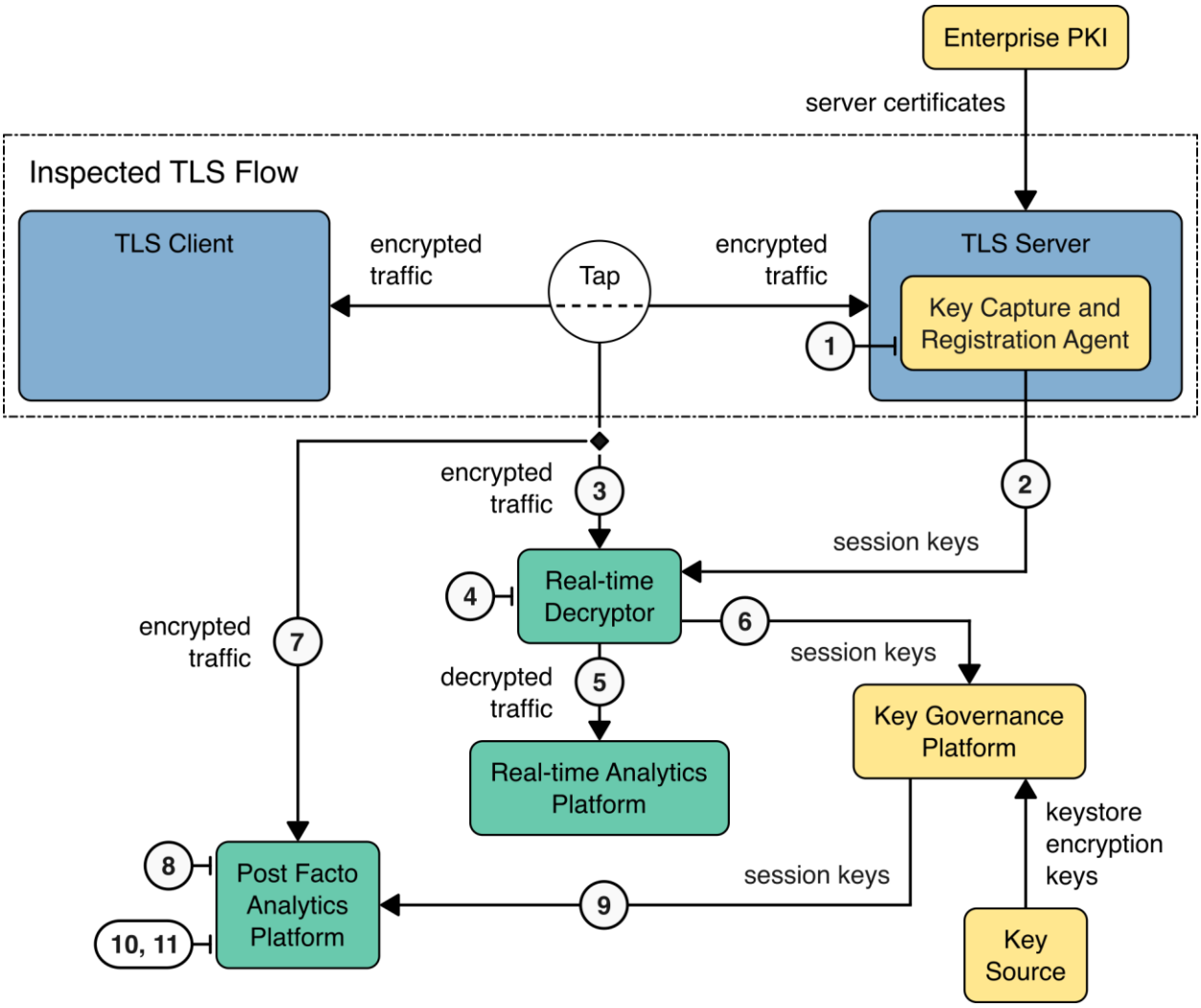
1. Before an epoch begins, the Key Governance Platform generates a new bounded-lifetime DH key pair and pushes it to the Key Management Agent on the TLS Server (and RT Decryptor).
2. When the epoch begins, the Key Management Agent configures the TLS Server to use the new bounded-lifetime DH key pair to negotiate a new TLS session with the client.
3. The Network Tap captures encrypted packets between client and server and forwards them to the Post-Facto Analytics Platform.
4. The Post-Facto Analytics Platform selects the traffic stream to be decrypted.
5. On a per-traffic stream basis, the Post-Facto Analytics Platform requests the server key pair from the Key Governance Platform using the TLS Server and the epoch of the traffic stream.
6. The Post-Facto Analytics Platform calculates session symmetric keys from the captured traffic and the server key pair supplied by the Key Governance Platform.
7. The traffic is decrypted using the calculated session symmetric keys.
8. Decrypted traffic is now available for analysis.

(Note: Key stores are generated by the Key Governance Platform.)

6.2.2 Exported Session Key Flow

Figure 6-2 depicts the elements involved in demonstrating passive inspection using exported session keys. Both post-facto and real-time decryption capabilities are illustrated.

Figure 6-2: Passive Inspection Using Exported Session Keys



6.2.2.1 Real-Time (RT) Decryption

The demonstration of real-time decryption using exported session keys executed the following sequence:

1. When the TLS Server negotiates a new TLS session with the client, the Key Capture and Registration Agent sniffs the negotiated session key.
2. The Key Capture and Registration Agent sends the session key and client-random-id* to the Realtime Decryptor.
3. The Network Tap captures encrypted packets between client and server and forwards them to the RT Decryptor.

4. The RT Decryptor uses the session key to decrypt the traffic.
5. The decrypted traffic is forwarded to the RT Analytics Platform.
6. The RT Decryptor sends the session key and client-random-id to the Key Governance Platform to support post-facto decryption.

*The client-random-id is a plaintext field in the TLS specification that uniquely identifies the TLS session.

Note that each connection between a server and client generates a new session key/client-random-id pair. The volume of key material that requires storage can be very large as it increases one-to-one with the volume of traffic. Forward secrecy is maintained.

6.2.2.2 Post-Facto Decryption (follows RT Decryption steps)

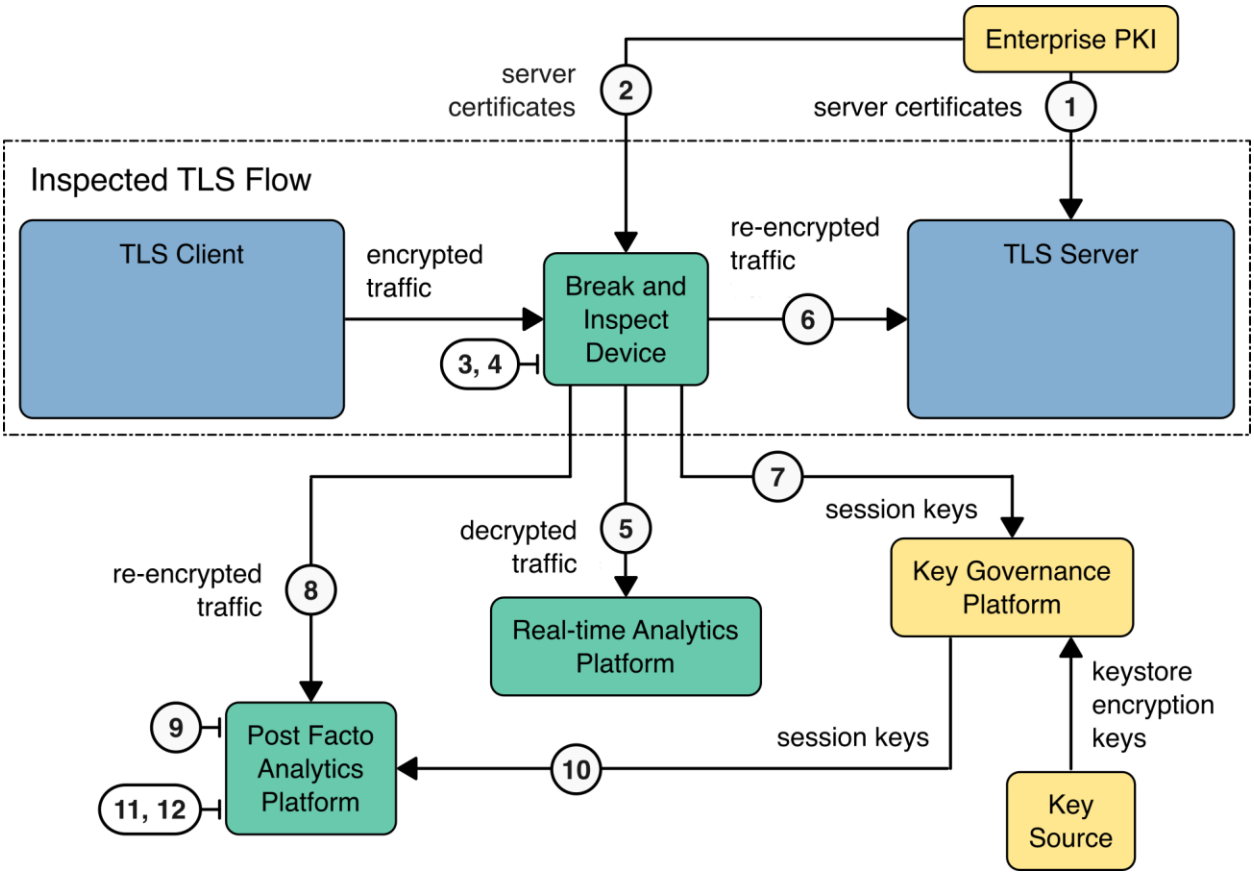
The demonstration of decryption for post-facto analysis using exported session keys will execute the following sequence that follows the real-time decryption sequence shown above:

7. The Network Tap captures encrypted packets between client and server and forwards them to the Post-Facto Analytics Platform.
8. The Post-Facto Analytics Platform selects the traffic stream to be decrypted.
9. On a per-traffic stream basis, the Post-Facto Analytics Platform requests the session key from the Key Governance Platform.
10. The traffic is decrypted using the session key.
11. Decrypted traffic is available for analysis.

6.2.3 Middlebox Active Decryption (Break and Inspect) Flow

Figure 6-3 depicts the architectural elements involved in demonstrating visibility using a middlebox. Note that, although both real-time and post-facto decryption are shown in the architecture drawing, only real-time decryption has been demonstrated as of this writing. Note also that traffic will be re-encrypted for transmission to the post-facto traffic capture platform within the data center.

1507 **Figure 6-3: Middlebox Break and Inspect Demonstration Elements**



1508 ### *6.2.3.1 Real-Time (RT) Decryption*

1509 The real-time break and inspect process executes the following steps:

1510 1. TLS Server certificates are provisioned on the appropriate TLS Server.

1511 2. All TLS Server certificates and private keys are loaded into the middlebox as well.

1512 3. The TLS client negotiates a TLS session with the middlebox. Simultaneously, the middlebox ne-
1513 gotiates a new TLS session with the intended destination TLS Server.

1514 4. The traffic from the incoming TLS session is decrypted by the middlebox using the session key
1515 for the TLS session to the client.

1516 5. The decrypted traffic is forwarded to the RT Analytics Platform.

1517 6. The decrypted traffic is copied to the TLS session with the intended destination TLS Server after
1518 being encrypted with the session key for this session.

1519 7. The middlebox exports the session key and client-random-id pair of the TLS session between the
1520 middlebox and the intended destination TLS Server to support post-facto decryption. Note that
1521 the session keys will be different for the client-to-B&I session and the B&I-to-server session. The
1522 session keys for the two sessions can be exported.

6.2.3.2 Post-Facto Decryption (follows RT Decryption steps)

The demonstration of decryption for post-facto analysis using middlebox B&I processes will execute the following sequence that follows the real-time decryption sequence shown above:

8. The Network Tap captures encrypted packets between the middlebox and server, and forwards them to the Post-Facto Analytics Platform.
9. The Analytics Platform selects the traffic stream to be decrypted.
10. Per traffic stream, the Analytics Platform requests the session key from the Key Governance Platform.
11. The traffic is decrypted using the session key.
12. Decrypted traffic is available for analysis.

6.3 Scenarios and Findings

The TLS 1.3 visibility project encompasses several application scenarios that impact enterprise compliance, security, and operational challenges. All scenarios address enterprise data center environments which include on-premises data center and hybrid cloud deployments hosted by a third-party data center or a public cloud provider. There are a variety of potential communications scenarios where visibility into communications for compliance, security, and operations are required. These include outbound traffic, connections across the Internet to the enterprise network boundary, and communications within the enterprise network between internal systems. This project specifically focuses on communications within the enterprise network and does not include outbound connections or communications across the public Internet. Project demonstrations address each of the following scenarios using both passive inspection and B&I middlebox approaches.

- Enterprises providing services to customers, partners, and employees must have the ability to rapidly troubleshoot and fix issues when availability and operational issues occur. An operations troubleshooting scenario demonstrates the enterprise need to trace transactions through all tiers of an application, including collection of detailed information such as transaction identifiers, data payload, and the results of operations performed by each application tier. Because operational issues can be intermittent and difficult to replicate, the scenario includes the ability to proactively collect and view detailed historical data that may or may not be available in logs. Examples of troubleshooting situations include application unavailability and intermittent system failures. Visibility may be required into communications for network-attached storage (NAS), identity management systems, databases, routers and switches, application servers, web servers, load balancers, and firewalls to build a complete picture of the end-to-end session across the enterprise.
- Application performance and response times are critical to customer service and time-sensitive mission-critical applications. Enterprises must be able to proactively detect and isolate performance issues for multi-tier applications. The performance monitoring scenario involves rapidly and accurately detecting user performance issues, predicting and resolving customer performance issues based on upstream degradation, maintaining the ability to rapidly identify sources of performance issues, monitoring across all mission-critical applications and platforms, and minimizing performance loads on applications and platforms.

- With the widespread threat of cyber-attacks, enterprises must be able to rapidly triage indicators of compromise (IOCs), quickly distinguishing false positives from real attacks. The threat triage scenario includes triage, identification, and response to IOCs. IOCs may arise in network-attached storage, identity management systems, databases, routers and switches, application servers, web servers, load balancers, and firewalls. They may be found in processes, open ports, and logs. Performing threat triage may require visibility into current and historical inbound and outbound communications. Effective performance of threat triage requires rapidly obtaining a clear picture of system state, reducing triage time with an accurate and detailed picture of current and historical communications, minimizing reliance on data sources that can be manipulated by attackers, and using independent data sources for verification.
- Following a major compromise, enterprises must be able to establish a clear picture of how the attack occurred, including each system that was compromised, vulnerabilities that were exploited, attack methods that were used, and data that was exfiltrated. To be effective, accurate information must be obtained from independent data sources about all operations performed by attackers (in case logs were manipulated). This security forensics scenario includes the ability to trace paths of attacks as they pivot laterally across the internal network of compromised systems. Affected systems may involve network-attached storage, identity management systems, databases, routers and switches, application servers, web servers, load balancers, and firewalls.

6.3.1 Demonstration of Passive Inspection

Passive inspection options support the capture of copies of traffic from monitored network segments, providing mechanisms to decrypt the copied traffic for immediate analysis, or forwarding encrypted traffic for storage structured with session information and the appropriate key information. This architecture does not terminate or otherwise modify traffic between the TLS clients and servers. All processing works on the copied traffic, not the original traffic.

6.3.1.1 Using Bounded-Lifetime Diffie-Hellman (DH) Keys

For TLS using bounded-lifetime (sometimes called static, when the lifetime is very long) DH session keys, the key governance platform provisions an agent operating on the TLS server with DH key pairs for the server to use in place of ephemeral generation. The key governance platform also makes these keys available to the systems that will decrypt the traffic.

6.3.1.2 Using Exported (Ephemeral) Session Keys

For TLS sessions using ephemeral keys, this architecture uses agents operating on the TLS servers to capture the session keys at the time they are negotiated. For ephemeral keys, real-time decryption is accomplished by rapidly sending the TLS session keys to the decryption platform, ideally before it sees the TLS handshake on the network. In the case of capture of ephemeral key-protected information for post-facto or historical analysis, these agents register the captured keys with the key governance platform. These keys can be retrieved from the key governance platform using the TLS session's client-random-id.

6.3.2 Demonstration of Inspection Using Middleboxes

Some aspect of analytics functions requiring enterprise visibility into its encrypted TLS 1.3 traffic may require combining network architecture and key-management techniques to achieve operationally necessary visibility. Necessary analytics functions may include identification of causes of network performance degradation or failures, key management-based communications failures, detection and identification of anomalous received data, identification of sources of anomalous data, and detection of encrypted traffic from unauthorized sources. Therefore, the scope of the project includes demonstration of an architecture that achieves visibility inside the data center through tools that break and inspect traffic. These middleboxes are commonly used at the enterprise edge to achieve real-time visibility. In this demonstration project, we examine deployment within the enterprise and address access to historical data by leveraging key-management based solutions.

The middlebox architecture supports capturing incoming traffic, providing tapped decrypted traffic to an analytics platform, storing traffic re-encrypted using new session keys negotiated between the middlebox device and the client and/or server, and passing re-encrypted traffic from the B&I component to an enterprise network server for routing to the enterprise's in-house consumers. The architecture also includes connection by the server and B&I component to a CA. The capture of traffic may be accomplished between the client and the B&I device, the B&I device and the server, or both. The session keys between the client and the B&I device, the B&I device and the server, or both need to be provided to the key governance platform to enable subsequent analysis of captured encrypted data.

7 Future Build Considerations

This preliminary draft practice guide reports TLS 1.3 visibility capabilities that have been demonstrated as of the date of its publication. As stated in Section 6.2, we plan to demonstrate and describe in a subsequent draft post-facto decryption of traffic protected under exported session keys and using middleboxes. We also plan to demonstrate and describe some examples of analytics on the captured traffic that can be conducted using collaborators' analytics tools.

An additional capability that is not within the scope of the current demonstration project but may be included in future project extensions is client-based monitoring. The options examined by this project are server-focused rather than client-focused. An approach to TLS 1.3 visibility that has been suggested involves reliance on enterprise support directly by the client endpoint or using clients via trusted proxy methods (e.g., SOCKS proxies). This approach is reported to require no potential deviation from RFC 8446 and to ensure that only those TLS clients mandated by local policy (e.g., enterprise management, parental control, anti-malware protection services) have these monitoring features available, and those only via opt-in (directly or via their guardian).

Appendix A List of Acronyms

AD	Active Directory
AEAD	Authenticated Encryption with Associated Data
API	Application Programming Interface
ASI	(NETSCOUT) Adaptive Session Intelligence
AWS	Amazon Web Services
B&I	Break and Inspect
C2	Command and Control
CA	Certificate Authority
CFIUS	Committee on Foreign Investments in the United States
CLM	Certificate Lifecycle Management
CMS	(Mira) Central Management System
CRADA	Cooperative Research and Development Agreement
CSF	Cybersecurity Framework
DCSA	Defense Counterintelligence and Security Agency
DDoS	Distributed Denial of Service
DH	Diffie-Hellman
DMZ	Demilitarized Zone
DNS	Domain Name System
DoH	DNS over HTTPS
DoQ	DNS over QUIC
DoT	DNS over TLS
DPI	Deep Packet Inspection
DTLS	Datagram Transport Layer Security
ESXi	VMware Purpose-Built Bare Metal Hypervisor
ETO	(Mira) Encrypted Traffic Orchestrator
EVA	(NFR) Encryption Visibility Agent, (NFR) Encryption Visibility Architecture
FIPS	Federal Information Processing Standard
FOCI	Foreign Ownership, Control, or Influence
Gbps	Gigabits per second
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IoC	Indicator of Compromise
IT	Information Technology

ITL	Information Technology Laboratory
JSON	JavaScript Object Notation
KVM	Kernel-based Virtual Machine
LDAP	Lightweight Directory Access Protocol
NAS	Network-Attached Storage
NCCoE	National Cybersecurity Center of Excellence
NDR	Network Detection and Response
NFR	Not for Radio
NIST	National Institute of Standards and Technology
NISTIR	NIST Internal or Interagency Report
OS	Operating System
PKI	Public Key Infrastructure
RBAC	Role-Based Access Control
REST	Representational State Transfer
RFC	Request for Comments
RMF	Risk Management Framework
RSA	Rivest, Shamir, Adleman
RT	Real Time
RTT	Real-Time Text
SaaS	Software-as-a-Service
SAML	Security Assertion Markup Language
SIEM	Security Information and Event Management
SKI	(Nubeva) Session Key Intercept, Symmetric Key Infrastructure
SMTP	Simple Mail Transfer Protocol
SOAR	Security Orchestration, Automation, and Response
SP	Special Publication
SSLO	(F5) SSL Orchestrator
SSO	Single Sign-On
TLS	Transport Layer Security
TMOS	(F5) Traffic Management Operation System
UDP	User Datagram Protocol
vETO	(Mira) Virtual Encrypted Traffic Orchestrator
VM	Virtual Machine
WebUI	Web User Interface
XDR	Extended Detection and Response

1635

Appendix B Glossary

We use the terms from NISTIR 7298, *Glossary of Information Security Terms* [18] or IETF RFC 4949, *Internet Security Glossary*, Version 2 [19] where those references define the terms.

Analytics	The discipline that applies logic and mathematics to data to provide insights for event recognition and for making response decisions. In this project, the function is executed by a set of tools for examining unencrypted payloads to identify undesired characteristics.
Bounded-Lifetime Key	A key variable that is used within the enterprise for decryption in real time or is stored for a period established by an explicit enterprise policy to enable decryption for post-facto security analytics/forensics purposes and is then destroyed in accordance with the policy.
Break and Inspect	A function that taps, decrypts, terminates, and re-encrypts/reinitiates network traffic.
Certificate	A set of data that uniquely identifies a public key (which has a corresponding private key) and an owner that is authorized to use the key pair. The certificate contains the owner's public key and possibly other information and is digitally signed by a certificate authority (i.e., a trusted party), thereby binding the public key to the owner.
Certificate Authority	An authorized entity that stores, signs, and issues digital cryptographic key certificates. It acts to validate identities and bind them to cryptographic key pairs with digital certificates.
Certificate and Key Governance	Functions for securely issuing, monitoring, facilitating, and executing digital X.509 certificates and managing the cryptographic keys exchanged using the certificates.
Client	System entities that request and use a service provided by another system entity called a server. Usually, it is understood that the client and server are automated components of the system, and the client makes the request on behalf of a human user. Clients may initiate encrypted traffic. They are interfaces for human users, devices, applications, and processes to access network functions, including the requesting of certificates and keys.
Cryptography	The discipline that embodies the principles, means, and methods for the transformation of data to hide their semantic content, prevent their unauthorized use, or prevent their undetected modification. It embodies the principles, means, and methods for providing information security, including confidentiality, data integrity, non-repudiation, and authenticity.
Decryption	The process of a confidentiality mode that transforms encrypted data into the original usable data.
Deep Packet Inspection	A form of packet filtering that locates, identifies, classifies, and reroutes or blocks packets with specific data or code payloads that conventional packet filtering, which examines only packet headers, cannot detect.

DevOps	A combination of the terms development and operations; meant to represent a collaborative or shared approach to the tasks performed by a company's application development and IT operations teams.
Diffie-Hellman	A method used to securely exchange or establish secret keys across an insecure network. Ephemeral Diffie-Hellman is used to create temporary or single-use secret keys.
Encryption	Cryptographic transformation of data (called "plaintext") into a form (called "ciphertext") that conceals the data's original meaning to prevent it from being known or used. If the transformation is reversible, the corresponding reversal process is called "decryption," which is a transformation that restores encrypted data to its original state.
Endpoint Agent	A lightweight background application installed on a device's operating system to constantly assess it for vulnerabilities.
Ephemeral Key	A cryptographic key that is generated for each execution of a key-establishment process and that meets other requirements of the key type (e.g., unique to each message or session).
Key	A numerical value used to control cryptographic operations, such as decryption, encryption, signature generation, or signature verification. Usually, a sequence of random or pseudorandom bits used initially to set up and periodically change the operations performed in cryptographic operations for the purpose of encrypting or decrypting electronic signals, or for producing another key.
Key Capture	Captures of session keys at the time they are negotiated.
Key Management	The handling of cryptographic keys and other related security parameters (e.g., passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key Registration	A function in the lifecycle of a cryptographic key; the process of a registration authority officially recording the keying material.
Key Source	A FIPS 140-validated entity that securely generates cryptographic keys and key pairs that are used in cryptography.
Kubernetes	A portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
Middlebox	A networking device that transforms, inspects, filters, and manipulates traffic for purposes other than packet forwarding. In this project, the device is used to break and inspect enterprise network traffic.
Network Tap	A component that provides a copy of traffic from a network segment. It is typically used in network security applications to monitor traffic and identify malicious activity or security threats.
Post-Facto	From or by an after act, or thing done afterward; in consequence of a subsequent act; retrospective.

Private Key	The secret part of an asymmetric key pair that is typically used to digitally sign or decrypt data.
Public Key	The public part of an asymmetric key pair that is typically used to verify signatures or encrypt data.
Public Key Certificate	A digital document issued and digitally signed by the private key of a certificate authority that binds an identifier to a cardholder through a public key. The certificate indicates that the cardholder identified in the certificate has sole control and access to the private key.
Public Key Infrastructure	A framework that is established to issue, maintain, and revoke public key certificates.
QUIC	A UDP-based multiplexed and secure transport protocol.
Real-Time	A function which conducts operations that must guarantee response times within a specified time or window of time, usually relatively short.
SecOps	A combination of the terms security and operations; a methodology that IT managers implement to enhance the connection, collaboration, and communication between IT security and IT operations teams.
Secret Key	A cryptographic key that is used with a (symmetric) cryptographic algorithm that is uniquely associated with one or more entities and is not made public. The use of the term “secret” in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.
Server	A system entity that provides services in response to requests from other system entities called clients.
Symmetric Cryptography	A cryptographic algorithm that uses the same secret key for its operation and, if applicable, for reversing the effects of the operation (e.g., an AES key for encryption and decryption).
Transport Layer Security	A security protocol providing privacy and data integrity between two communicating applications.
TLS Server	The counterparty for encrypted traffic that generates session keys, negotiates encryption protocols, and connects to key management infrastructure.

Appendix C References

[1] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 5246, August 2008 (Updated October 2015). Available at https://datatracker.ietf.org/doc/rfc5246/.

[2] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 8446, August 2018. Available at https://datatracker.ietf.org/doc/rfc8446/.

[3] B. Carpenter and S. Brim, *Middleboxes: Taxonomy and Issues*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 3234, February 2002. Available at https://datatracker.ietf.org/doc/rfc3234.

[4] Center for Cybersecurity Policy and Law, Enterprise Data Center Transparency and Security Workshop Report, November 2019. Available at https://www.centerforcybersecuritypolicy.org/insights-and-research/enterprise-data-center-transparency-and-security-workshop-summary-report.

[5] National Institute of Standards and Technology, *Virtual Workshop on Challenges with Compliance, Operations, and Security with TLS 1.3*, September 2020. Available at https://www.nccoe.nist.gov/get-involved/attend-events/virtual-workshop-challenges-compliance-operations-and-security-tls-13.

[6] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, *Specification for DNS over Transport Layer Security (TLS)*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 7858, May 2016. Available at https://datatracker.ietf.org/doc/rfc7858/.

[7] P. Hoffman and P. McManus, *DNS Queries over HTTPS (DoH)*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 8484, October 2018. Available at https://datatracker.ietf.org/doc/rfc8484/.

[8] C. Huitema, S. Dickinson, and A. Mankin, *DNS over Dedicated QUIC Connections*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 9250, May 2022. Available at https://datatracker.ietf.org/doc/rfc9250/.

[9] Joint Task Force Transformation Initiative, *Guide for Conducting Risk Assessments*, NIST Special Publication (SP) 800-30 Revision 1, September 2012. https://doi.org/10.6028/NIST.SP.800-30r1.

[10] E. Barker and A. Roginsky, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*, NIST Special Publication (SP) 800-131A Revision 2, March 2019. https://doi.org/10.6028/NIST.SP.800-131Ar2.

[11] K. McKay and D. Cooper, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, NIST Special Publication (SP) 800-52 Revision 2, August 2019. https://doi.org/10.6028/NIST.SP.800-52r2.

[12] C. Bartle and N. Aviram, *Deprecating Obsolete Key Exchange Methods in TLS 1.2*, Internet Engineering Task Force (IETF), March 2023. Available at https://www.ietf.org/archive/id/draft-ietf-tls-deprecate-obsolete-kex-02.html.

[13] R. Salz and N. Aviram, *TLS 1.2 is Frozen,* Internet Engineering Task Force (IETF), June 2023. Available at https://www.ietf.org/archive/id/draft-rsalz-tls-tls12-frozen-01.html.

[14] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, NIST Special Publication (SP) 800-207, August 2020. https://doi.org/10.6028/NIST.SP.800-207.

[15] *Framework for Improving Critical Infrastructure Cybersecurity*, Version 1.1, National Institute of Standards and Technology (NIST), April 2018. https://doi.org/10.6028/NIST.CSWP.6.

[16] Joint Task Force, *Security and Privacy Controls for Information Systems and Organizations*, NIST Special Publication (SP) 800-53 Revision 5, September 2020. https://doi.org/10.6028/NIST.SP.800-53r5.

[17] U.S. Department of Commerce, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standard (FIPS) 140-3, March 2019. https://doi.org/10.6028/NIST.FIPS.140-3.

[18] C. Paulson and R. Byers, *Glossary of Key Information Security Terms*, National Institute of Standards and Technology Interagency Report (NISTIR) 7298 Rev. 3, July 2019. https://doi.org/10.6028/NIST.IR.7298r3.

[19] R. Shirey, *Internet Security Glossary*, *Version 2,* Internet Engineering Task Force (IETF) Request for Comments (RFC) 4949, August 2007. Available at https://datatracker.ietf.org/doc/rfc4949/.