JW secure Privileged account management done right from the start.

End-to-End Enforcement of Hardware-Based Data Protection

Table of Contents

Abstract

After decades of hacking, cyberattacks, malware, ransomware, and other data security threats, we still do not have data protection managed by policy and applicable to all devices—from smartphones to servers.

The Trusted Platform Module (TPM) is an example of the kind of hardware-backed data security building block we would like to see. However, TPM has not experienced the uptake necessary to realize this vision. Nevertheless, the typical enterprise server and PC client refresh cycle is finally likely to encounter the option of a TPM-enabled backplane. Plus, while no mass market smartphones expose TPM-like capabilities to third-party app developers, improvements in data loss prevention can be realized by implementing storage and lifecycle policies in software on these platforms, and complementing those protections with other defense-in-depth measures.

In this paper, we describe an enforceable system for better data security, including new technology that binds hardware-protected cryptographic keys to environmental measurements such as time and the behavior of the user.

1 Introduction

In the domain of information security, the defenders are always at a disadvantage. Both diligence and a motivation to move beyond simple *checkbox compliance* are essential to mount an effective defense.

Typical IT compliance regimes include stated requirements for encryption and auditing. But turning generic industry terms into real security requires judgment and skill. Compliance is not the same thing as security. It has been well noted that both Target and The Home Depot were compliant with industry standards before their systems were breached.[1]

1.1 Field Devices Connect to the Cloud

In fact, referring to recent well-publicized security incidents in the retail sector, the ability to guard field-deployed systems such as point of sale (PoS) is a good litmus test for any enterprise data protection model. Such systems tend to be a heterogeneous mix of new and old hardware and software, ranging from credit card swipe readers and cash registers (in the case of PoS) to PCs and iPads (in the case of broader scenarios such as DevOps). Further, those systems are now connected to the internet, regardless of whether they were designed for that level of threat exposure.

We see risk in the variety of device types used for DevOps and remote system administration of cloud resources, as well as in the variety of networks used—from airport lounges to overseas hotel rooms. That an entire multi-server infrastructure can be created by the click of a smartphone at any time and in any place is a major technological accomplishment that the IT industry should be proud of. But it’s not a recipe for secure operations.

The Verizon 2016 Data Breach Investigations Report points to the importance of network segregation and strong authentication for forward-deployed devices such as PoS terminals.[2] That is another way of saying we must use *defense in depth*.

[1] Details can be found in the “Lessons From Home Depot: Expect Hackers To Crack More Retailers This Holiday Season” Forbes article by Paula Rosenblum.

[2] The full “Verizon 2016 Data Breach Investigations Report” is available online: http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/

1.2 Data Classification

Defense in depth for data security begins with data classification—which is easier said than done. As Forrester notes, data professionals "can't expect to adequately protect data if they don't have knowledge about what data exists, where it resides, its value to the organization, and who can use it. Data classification also helps to create data identity (data-ID), the missing link for creating actionable data security and control policies. Yet S&R [Security and Risk] pros who attempt to lead efforts to classify data are thwarted by their own efforts with overly complex classification schemes and haphazard approaches. As a result, many see data discovery and classification as a Sisyphean task."[3]

Indeed, for many organizations, data classification is like servicing your car: it's inconvenient and can be expensive, but avoiding it greatly increases risk. Preventive maintenance is the least costly approach. The data classification equivalent to changing the engine oil twice a year can be as simple as having a written policy for handling business-sensitive data. The policy should encompass user behavior: don't print sensitive documents, don't use removable media, don't provide sensitive information over the phone, and so on. This approach complements the hardware- and software-based protections discussed next.

1.3 Securing Big Data

Absent other requirements, protecting large quantities of data at rest is theoretically easy. However, practice is messier: content must be accessible in real time, protected from a continuously evolving list of threats, and administered by IT staff and DevOps with limited time and budget. In light of these complex realities, we propose to address proofing data access requests as they arrive at data servers. This ensures timely responses while preventing data loss due to sophisticated attacks and security policy drift. This is a more scalable approach than retrofitting application-tier encryption onto large data stores. And it's more effective, since experience has shown that application-tier is still the low-hanging fruit for remote compromise. Encryption is moot once the attacker controls the app.

Administrators of big data systems must manage access to data from devices with widely varying operational environments. We assume sensitive content must not only be protected with strong encryption, but also that such data be exposed only to strong identities and high-integrity systems, including—and especially—within the datacenter. Enforcing protection for such a model requires data only be accessible to known systems in a provably secure state.

1.4 Defense in Depth

After data classification, we must distinguish between sensitive (for example, a line-of-business database) and everyday (for example, the internet) network resources.

Next we create policies for authentication, authorization, confidentiality, and auditing. We do not allow network connections to sensitive assets unless meaningful security policies are in place.

Finally, we add the linchpin: enforcement. Focusing on business-sensitive data and network assets, meaningful security policies to enforce include:

- Hardware-based disk encryption.
- Whitelisted firmware, operating system, and application software components.
- Strong authentication of the user plus the host device.

We will discuss enforcement in the remainder of this paper.

[3] "Predictions 2016: The Trust Imperative For Security & Risk Pros." Forrester Research.

2 The Current State of the Technology

Current commercial technologies are inadequate to mitigate attacks that come from within the enterprise-trusted ecosystem. There are several reasons for this:

1. Server identities expressed using Public Key Infrastructure (PKI) are an improvement over server identities authenticated using static passwords. However, server-side PKI typically uses software-based keys that are easily exported. Forrester estimates that “70 percent to 80 percent of data breaches involve the use of privileged and administrative passwords and credentials (Unix root and Windows Administrator passwords).”[4] Only hardware-based device identities should be allowed when valuable data are being stored or transmitted.

2. While server authentication is common in client connectivity protocols such as TLS and 802.1x, strong authentication is less commonly enforced in machine-to-machine scenarios such as accessing backend storage (e.g., SMB, SQL) and datacenter management (e.g., virtual machine migration). Authorization must instead be enforced at every hop.

3. Advanced cyberattacks such as Pass the Hash and rootkits are increasingly prevalent in the datacenter and are difficult both to detect and to stop from spreading. According to the Microsoft Security Intelligence Report, “Even though current PtH methods concentrate on password hashes, similar attacks could (and can) work against other authentication mechanisms, including tokens, delegation, and two-factor authentication. If the attacker is able to capture the ultimate secret—be it a password hash, a token, or some other entity—the attack will succeed.”[5]

   Server-side data protection must account for the integrity of the end-to-end operating environment when authorizing a connection request.

4. Maintaining the invariant of dual-layer encryption is difficult enough to configure and deploy, let alone to actually enforce. However, without enforcement, security policy drift is unfortunately the norm in enterprise computing. Software versions and computer configurations vary in subtle ways, even for a single provider or host. Over time, it is common for software, and even hardware, settings to change, due to operator error, incompatibilities, or side effects. Scanners can detect such issues, but it’s not enough to know about them—gaps must be remediated immediately in order to block opportunistic adversaries. Hence the need for enforcement.

3 Policy and Enforcement Building Blocks

Windows and Linux allow many policies to be checked using hardware root of trust and platform attestation. For example:

- Disk encryption, based on TPM (plus boot PIN for laptops), to help mitigate the threat of offline attacks.
- Trusted code-signer whitelisting. This includes identifying providers of early-boot software components and device drivers (usually large Original Equipment Manufacturers [OEMs]) and providers of application software (usually a combination of external vendors and internal lines of business teams).
- Early launch anti-malware to initiate antivirus protection before other operating system components initialize.
- Disabled kernel debugging, so users aren’t given more privilege than they need in order to be productive.
- TPM storage and endorsement hierarchy attestation to bind credentials to a specific device.
- TPM trusted Endorsement Keys in order to whitelist the device itself.

[4] From “Quick Take: 12 Lessons For Security & Risk Pros From The US OPM Breach” by Forrester Research.

[5] From the Microsoft Security Intelligence Report.

- Credential Guard, to help mitigate Pass the Hash attacks.
- DeviceGuard, so only whitelisted applications and extensions can run.
- Secure Boot Database cryptographic hash to help mitigate the threat of offline attacks.

However, it is not enough to scan for a report on these policies. After all, many attacks start by disabling host security features. FedEx was recently hacked after malware disabled local security tools.[6] And for usability reasons, most systems prompt for user credentials only occasionally, relying on cached session state in the interim. It is critical to enforce security policies in real time whenever an attempt is made to access sensitive data.

4 Implementing True Enforcement

In order to enforce security policy, you must always make authorization conditional on compliance. That is, if either the computer or user account does not comply with security policy, any attempt to access sensitive network resources must be prevented. While this sounds simple, the complexity (and old age) of network security protocols can make this challenging. Further, we cannot sacrifice usability or we risk end-user mutiny.

4.1 Mechanisms of enforcement

Despite the challenge, several mechanisms are available today for implementing security policy enforcement with reasonable usability:

- Enforce the hardware root of trust mechanisms described previously, using remote platform attestation.
- Bind user and computer credentials to a specific device in a specific state. TPM sealed keys are one way to do this.
- Integrate new authorization capabilities in a standards-based way so existing line-of-business applications can use them. This can be done with PKI or token/federation-based integration, for example.
- Limit the lifetime of derived credentials and protocol session state, including automatic forced cleanup when a policy violation occurs.

4.2 Patching

Patching is the single most effective technique for data loss prevention. For example, Microsoft says, “CVE-2010-2568 is the most commonly targeted individual vulnerability in 1H15.”[7] However, that CVE, a Windows Shell vulnerability, has been patched ever since 2010.

Enabling automatic patching, the baseline critical defense so many organizations and users continue to resist, does present some risk. Application compatibility problems can result from such patches, particularly given the long tail of possible OS and app version combinations. However, exploit statistics speak to an inconvenient truth: unless network endpoints are forced to be in a fully patched state each time a connection is established, then those endpoints are probably not fully patched, and are therefore vectors for malware infection, credential theft, and data loss.

4.3 Application Whitelisting

A corollary to the complexity of compatibility and configuration testing discussed in the previous section is the need to whitelist applications. Like any good management practice, whitelisting must begin with inventory—knowing what you

[6] Details can be found in the “FedEx Delivery Notices Dropping Zeus and Fareit Trojans” Infosecurity Magazine article by Tara Seals.

[7] From the Microsoft Security Intelligence Report, Volume 20.

already have. Experience has shown that, for many large organizations, a comprehensive app inventory does not exist, and if it did, it would be frightening, due to business dependencies upon a variety of legacy applications for which neither support nor source code are available.

Nevertheless, ignorance is not bliss. All software has bugs, and older software tends to have a greater number of published vulnerabilities (including indirectly, due to the use of third-party libraries), allowing it to be exploited in seconds by attackers with only rudimentary skills (see Verizon's Data Breach Investigations Report[2] for more statistics on time to exploit).

With a whitelist in hand, enforcement can be implemented using a combination of mechanisms including reduced user privilege, managed desktops (Microsoft SIR[5] shows that unmanaged computers encounter malware at roughly double the rate of managed computers), and digital signature-based code execution policies.

While all of these mechanisms help, there is still the challenge of attacks from within.

5 Securing Devices Behind the Enterprise Firewall

We have seen that, with effort, we can help ensure a computer operates correctly on behalf of a user. Examples include everything from a smartphone-initiated personal banking transaction to a web server accessing a database server within the confines of a datacenter. Before any enterprise allows critical data onto a device, the device must be both identified and secured from attack.

Modern business requirements are such that employees and partners throughout the world expect ready access to data from a wide range of devices. IT professionals must ensure timely responses while preventing data loss due to sophisticated attacks and enterprise security policy drift. However, even devices behind enterprise firewalls are regularly hacked. Web application servers are a notorious example. Each hacked web server becomes a Trojan horse within the walls of the IT fortress.

How do we prevent the bad guys from getting inside the fortress that is our datacenter? Furthermore, given the standard security guidance of assumed breach, how do we prevent the bad guys from pouring out and taking over the whole place once they are inside? Proofing data access requests arriving at each individual server is critical.

Threats against data include spoofing the user or the device requesting access, modification or denial of timely response to the user, and unauthorized disclosure of the plaintext data. Sensitive content must not only be protected with strong encryption, it must also be exposed only to strong identities and high-integrity systems, whether they are remote or within the datacenter. Enforcing protection for such a model requires that data be accessible only to systems known to have been good when issued and that remain secure during use.

We recommend data protection based on a foundation of hardware-protected device identity. Then, building on that foundation, the following capabilities are required:

- Data protection enforcement is local to each server. For example, the virtual TPM capability supported by hypervisors such as Xen and Hyper-V allows attested policy to be applied per server workload.
- Data are encrypted, at rest and in flight, until rendered for display.
- Data are rendered into plaintext only in a protected environment.
- User authentication does not hamper the security analyst (but neither is the security analyst account a single point of failure).
- Techniques for handling device or user termination or abort are robust.

6 Hardware Virtualization

Most commercial operating system and hardware platform combinations support virtualization and secure execution. However, integration of these capabilities into a system that enforces *continuous authorization* based on hardware root of trust is both possible and a necessary next step.

The National Institute of Standard and Technology says, “Initial authorization to operate is based on evidence available at one point in time, but systems and environments of operation change. Ongoing assessment of security control effectiveness supports a system’s security authorization over time in highly dynamic environments of operation with changing threats, vulnerabilities, technologies, and mission/business processes.”[8]

We recommend the following technologies be employed in concert:

- Data encryption keys protected by the TPM.
- Virtualized container for rendered documents.
- Measurement-based access to keying material.
- Deletion of plaintext content when compromise is detected.

Users demand ready access to business data content over on-again/off-again connectivity. However, that need must be balanced with business continuity, intellectual property protection, and reputation needs. In other words, service high-availability must not come at the expense of data protection. Data should be exposed only to authorized users on known-good hardware in a secure state.

6.1 Time-Limited Access

For managed devices, the ability to ensure hardware remains in a secure state is diminished when the device is disconnected from the management network. Downloaded plaintext data is vulnerable. Therefore, we need to limit by policy the period of time during which downloaded plaintext data remains accessible.

6.2 Clear Cache

Whenever any secure app is suspended, all protected content contained in its cache will be cleared and must be decrypted again after the app is activated. An example scenario would be the laptop of a business executive left in a hotel room.

6.3 Continuous Authorization

Device software should decrypt content only when authorization conditions are met. The service should determine the integrity of the software on the device continuously.

To this we add information about user identity, device identity, and whatever additional context is relevant to the authorization decision.

The combination of device integrity and authentication data for the user and project parameters, such as time and location, are used to create wrappers for the content decryption keys. Note that content decryption keys can be included in several key wrappers with different users and projects, where projects should be interpreted broadly to include groupings like physical locations or even device types.

[8] From “Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations” by the National Institute of Standards and Technology (NIST)

6.4 Render Engine

Consider a roaming user who needs to access content over an on-again/off-again network. The time that a content license is valid is defined as the window of vulnerability for the content supplied under that license. That is the period the device may remain out of contact with the servers and still access the content.

In addition to existing commercial components, we propose a Device Authentication Provider (DAP) and device software to decrypt content when authorization conditions are met. The DAP is a standard remote authentication service for determining the integrity of the software on the device at the time it is awakened.

We propose a render engine for DOCX, XLSX, and PDF documents, for example. This render engine must include the standard document encryption technology available with those formats. It must protect the decryption key in use by putting the render engine in a space that cannot be accessed by any other application. Several technologies are available to place the render engine in an isolated environment. We propose to show hardware virtualization performing this function. The display image is eliminated whenever the device power cycles, requiring renewed authorization and decryption.

To decrypt new content, the user and the device must be authenticated. We bind hardware-protected cryptographic keys to environmental measurements such as time and behavior (see section *Measurement Bound Keys*, following).

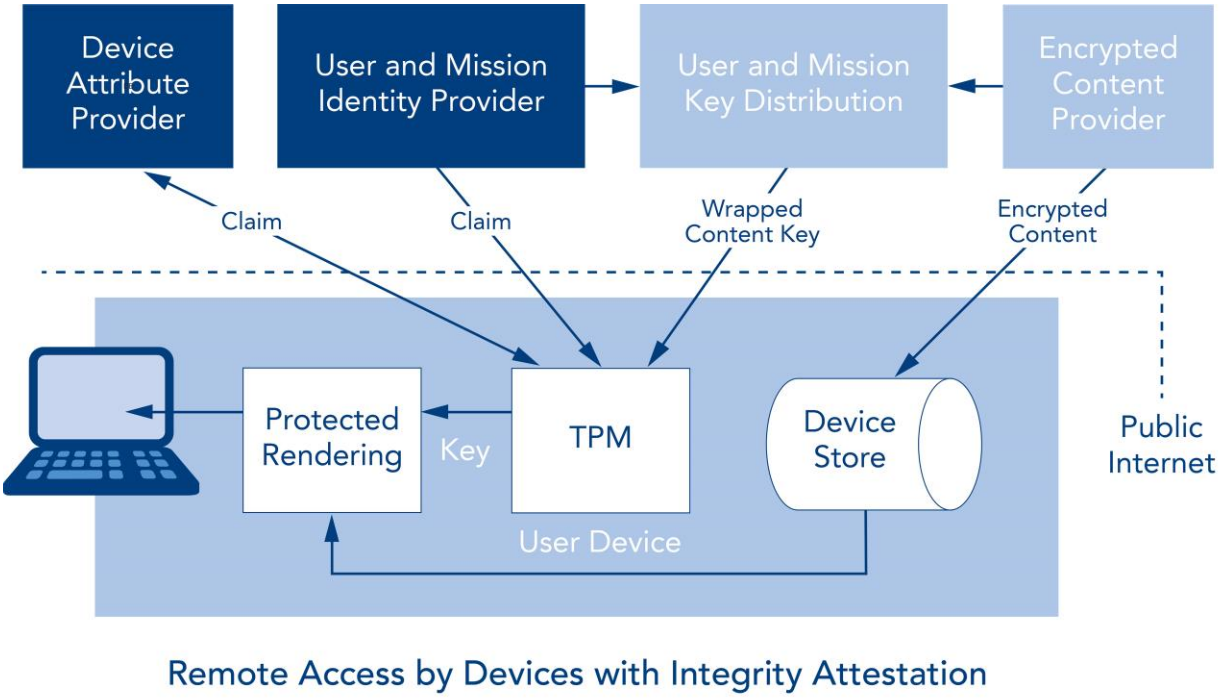
Device Attribute Provider
User and Mission Identity Provider
User and Mission Key Distribution
Encrypted Content Provider
Claim
Claim
Wrapped Content Key
Encrypted Content
Protected Rendering
Key
TPM
Device Store
User Device
Public Internet
Remote Access by Devices with Integrity Attestation

Figure 1: Hardware-Based Data Protection

This section describes the proposed system for acquiring a license (i.e., policy-protected, wrapped decryption key) to enable access to sensitive data. The decryption key is provided based on configurable license policy including such requirements as:

- The mission has not expired.
- Device hardware identity is whitelisted.
- Device remains compliant with boot and runtime security policies.

The Device Attribute Provider (DAP) and Decryptor (in the Protected Rendering component in Figure 1) interact with the TPM of the User Device in the following way to ensure that only an authorized user on a known device in a secure state can decrypt content:

1. When the TPM is enrolled in the DAP, a private key is created in the TPM that is bound to a public key contained in the DAP.
2. On every reboot the TPM is cryptographically challenged in order to verify the integrity of the device and generate a claim bound (that is, sealed—see 7.2.1 following) to the attested measurements.
3. The DAP can now sign the policy with a key (A1) to be sent to the TPM that is evaluated before decryption is possible. The policy consists of the rules the TPM and Protected Rendering components must use to determine if decryption is authorized.
4. The Content Provider acquires a license from the DAP for each TPM it chooses to share content with. The license can be used for all or a part of the content the Content Provider shares with the device containing the TPM. The DAP generates a symmetric license key (S1) and a license that contains the public key (T1) of the TPM, the license key (S1) encrypted with the public key (T1), and a policy (P1) signed with a key (A1). The policy can be a pre-existing standard policy or one unique to the license.
5. The content is encrypted with a symmetric key (S1) and sent to the user device.
6. The license (policy plus the wrapped content key) is sent to the Decryptor.
7. When the User Device is asked to display the plaintext of the content, that content is sent to the Decryptor with a license identity that allows the Decryptor to discover the license containing the private key and from it the policy that must be followed to allow the decryption of the license key (S1). For example, the following checks may be included in the policy:
    a. The policy is signed by the key (A1) that is bound to key (T1).
    b. Time: for example, the key is good for only four hours.
    c. Boot counter: for example, the key is good only until the reader device reboots.
    d. Firmware and boot path: for example, the key is usable only while known-good Decryptor device firmware and operating system boot binaries are used.
    e. Valid plaintext display application: for example, the application code that displays the content is known to not leak it and will enforce other policy terms.
8. Once the content is decrypted into plaintext for display, the decryption key used by the Decryptor is destroyed.
9. If the User Device goes into sleep mode (for example, as the result of a device policy that forces sleep after a certain inactivity period) the plaintext display will be cleared by the reader.
10. When the User Device is awakened, the plaintext may try to redisplay. That will cause a request to the Decryptor to recover the decryption key from the Decryption Material which will force the policy to be revalidated. The plaintext will only be displayed if the policy check is still valid.
11. If the policy check is no longer valid, the Decryptor may optionally try to recover the plaintext by seeking a new license from the Device Attribute Provider, which will run the same checks described previously. If a new license is not available then the content must be reacquired.

As stated before, the render engine must be in a space that cannot be accessed by any other application in order to protect the decryption key. When hardware virtualization performs this function, the display image is eliminated whenever the device power cycles, requiring renewed authorization and decryption.

7 Measurement Bound Keys

Decrypting new content must not be possible until the user and the device are successfully authenticated. We can do this by binding hardware-protected cryptographic keys to environmental measurements such as time and behavior. We call the technique Measurement Bound Keys (MBK). An example implementation is TPM sealing.

From an implementation perspective, in order to drive adoption, an MBK should be usable in any scenario that calls for asymmetric cryptography and/or PKI without modification of the host application. Then, any change in measurement forces the key to be reauthorized, but in a way that is seamless to the user. As long as the key remains usable, the data owner knows the host is provably secure.

7.1 Components

The solution includes the following components (see Figure 2):

- Attestation Service: The purpose of the Attestation Service (AS) is to perform a cryptographic challenge-response protocol to ensure client computers have trustworthy basic input/output system (BIOS), TPM, and boot software. The AS also enforces security policy and acts as a repository for client device history.
- Certification Authority: We propose an enforcement model that integrates with PKI, since that approach offers a wide range of enterprise scenario and network protocol compatibility. An extension is applied to the CA (such as EJBCA or the Microsoft Enterprise CA) that ensures that certain X.509 certificate templates are issued only to computers compliant with security policy, as measured by the Attestation Service.
- Client Crypto: Client Crypto interacts with the Attestation Service to create attested cryptographic keys bound to the security policy settings of a client computer. Keys generated in this way are usable only while the client stays compliant with security policy.
- High-Value Asset: The HVA is any business-critical network asset or database.

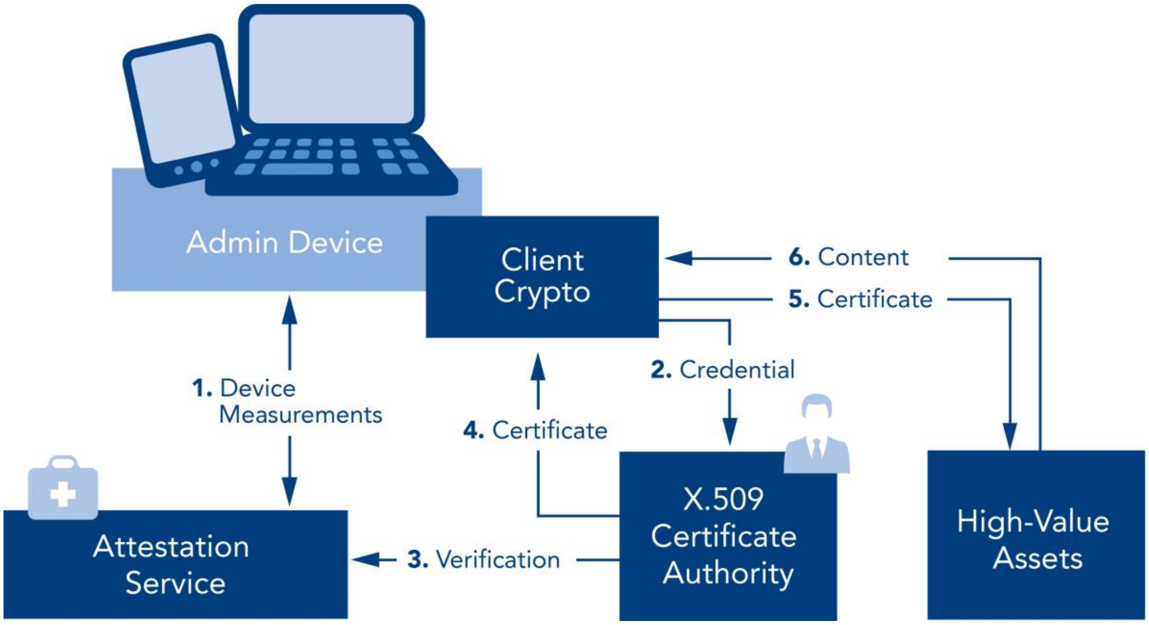

Figure 2: Component Diagram

7.2 Data Flow

The following describes the flow of data in Figure 2.

7.2.1 Device Measurements

The Admin Device attempts to create a measurement-bound keyset, that is, a cryptographic key sealed to a specific TPM in a specific state.

7.2.2 Credential

The device uses the key to sign a certificate enrollment request.

7.2.3 Verification

The Certificate Authority verifies the request signer is trusted by the Remote Attestation Service.

7.2.4 Certificate [granted]

If the signer is trusted, the Certificate Authority issues the requested certificate to the device.

7.2.5 Certificate [submitted]

The device uses the certificate for authenticated access to High-Value Assets on the network.

7.2.6 Content [released]

Only policy-compliant devices can reach sensitive resources. If the device deviates from security policy, authenticated access is immediately terminated. The system supports optional enhanced scenario integration with Kerberos and internet protocol security (IPSEC).

7.3 Creating a Measurement Bound Key

The creation of an MBK entails the following challenge-response message exchange:

1. The client requests a Nonce.
2. The client requests a TPM Attestation Identity Key (AIK) challenge. The Attestation Service (AS) challenges the binding of the Nonce, the AIK, the manufacturer Endorsement Key, and the TPM Storage Root Key.
3. The client sends its signed boot logs to the AS. The AS enforces log integrity, continuity, and boot policy.
4. If successful, the client is issued an encrypted Measurement Bound Key. Only the specific TPM challenged in step 2 can decrypt and use the key, and only until the security policy measurements change again.
5. When the boot measurements change, the client repeats steps 1–4.

8 Conclusion

We learn best from personal experience, but corporate boards of directors are no longer accepting the legal risk of waiting for a major security incident before instituting proper defenses. Mandiant, Microsoft, and Verizon have all published sobering reports that point to the importance of staying vigilant: keeping patching up to date, fixing security bugs in internet-facing web apps, reducing the impact of phishing, slowing lateral movement of intruders on your network, and responding quickly to attacks once they occur. Using hardware root of trust and continuous monitoring solves those problems by ensuring a broad range of policies are enforced at the time of authentication, thereby protecting all high-value network assets.

Psychological and social research shows that risk assessment is something humans tend to do poorly. Careful, considered analysis of risks and assets and staying informed about online security are the only ways we can effectively prioritize and mitigate those risks. Just because cyber offense has an advantage over defense does not mean attacks cannot be stopped, or at least greatly slowed down. Security policies must be enforced.

9 References

[1] Forrester Research. 2015. *Predictions 2016: The Trust Imperative For Security & Risk Pros.* (November 9, 2015).

[2] Forrester Research. 2015. *Quick Take: 12 Lessons For Security & Risk Pros From The US OPM Breach.* (June 8, 2015). https://www.forrester.com/report/Quick+Take+12+Lessons+For+Security+Risk+Pros+From+The+US+OPM+Breach/-/E-RES123441

[3] *Microsoft Security Intelligence Report.* https://www.microsoft.com/security/sir/strategy/default.aspx#!pass_the_hash_attacks

[4] National Institute of Standards and Technology (NIST). 2011. *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations.* (September 2011). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-137.pdf

[5] National Institute of Standards and Technology (NIST). 2014. *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.0.* (February 12, 2014). http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf

[6] Raab, Carolyn. 2016. *True Network Hardware Virtualization.* Linux.com News. (June 15, 2016). https://www.linux.com/news/true-network-hardware-virtualization

[7] Rosenblum, Paula. 2014. *Lessons From Home Depot: Expect Hackers To Crack More Retailers This Holiday Season.* Forbes. (November 6, 2014). http://www.forbes.com/sites/paularosenblum/2014/11/06/lessons-from-home-depot-expect-hackers-to-crack-more-retailers-this-holiday-season/

[8] Seals, Tara. 2016. *FedEx Delivery Notices Dropping Zeus and Fareit Trojans.* Infosecurity Magazine. (June 21, 2016). http://www.infosecurity-magazine.com/news/fedex-delivery-notices-dropping/

[9] TCG. 2015. *TCG Infrastructure WG: TPM Keys for Platform Identity for TPM 1.2. Specification Version 1.0, Revision 3.* (August 21, 2015). http://www.trustedcomputinggroup.org/wp-content/uploads/TPM_Keys_for_Platform_Identity_v1_0_r3_Final.pdf

[10] Verizon. 2016. *Verizon 2016 Data Breach Investigations Report.* Accessible at http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016