Security Fabric – Tailored Trustworthy Space
Part 3: A Close-up on Security Management

Charles Speicher

McAfee
3965 Freedom Circle
Santa Clara, CA 95054-1203

Abstract

The Security Fabric framework is a commercial implementation of the "tailored trustworthy space" strategy developed by the White House Networking and Information Technology Research and Development (NITRD) Program and promoted by the Department of Energy for maintaining security of end-to-end intelligent grid environments. For end-to-end security, no one size fits all implementation is possible because of the variety of specific installation needs. The approach must have the flexibility to dynamically adjust to the policies that are appropriate to each individual situation. It must be suitable for the very smallest of situations, but it must also scale uniformly to support the largest of situations which involve millions of managed objects. In that there will be no single victor in the commercial marketplace for a single proprietary design for matters such as key management or other major functional concepts, the Security Fabric provides an interoperable framework that comfortably supports many solutions for individual components using varying standards that can be tested and certified for interoperability.

This ***Part 3: Close-up on Security Management*** – Focuses specifically on the framework for security management aspect of the TTS management system. It includes coverage of the security elements plus a discussion of the defense in depth and moving target strategies.

1. A CLOSE-UP ON END-TO-END SECURITY MANAGEMENT

Security management covers many well-known areas, but identity management and authentication are the first among equals. The following list itemizes the specific areas addressed by the Security Fabric:

- Identity management
- Authentication
- Access management
- Authorization
- Auditing
- Confidentiality
- Integrity
- Availability
- Non-repudiability
- Provenance.

2. IDENTITY AND AUTHENTICATION MANAGEMENT

Devices join the TTS environment by powering on and then performing registration and mutual authentication with the TTS manager directly. The Security Fabric Identity Metasystem residing at the utility's manager system initially uses Kerberos as the trusted broker to welcome the device to the TTS – assuming the device was successfully registered to the Security Fabric directory by the utility at commissioning time. Other options planned for authentication include mutual TLS and also SAML 2.0, but the initial release uses Kerberos because of the opportunity for distributing specialized credentials to devices during the initial authentication interaction. The directory system that is part of the management system provides a useful way for the identity management mechanisms to rapidly retrieve credentials during authentication.

The initial credentials exchange can provide the basis for securely providing additional path information to the managed device that provides messaging linkage to the network manager itself. It can also provide linkage for registration to the configuration synchronizer, and also to the authorized firmware repository.

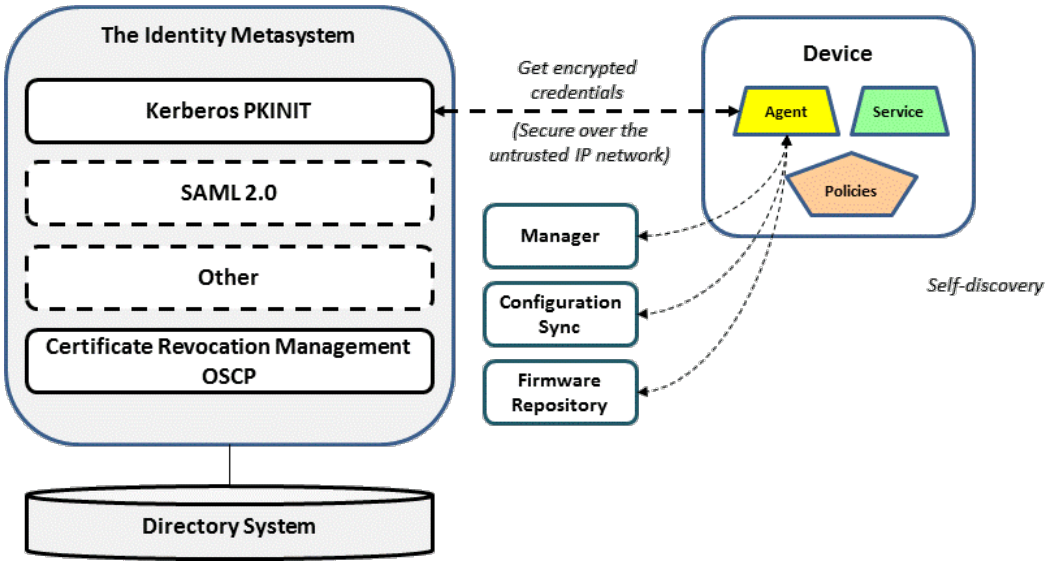

Figure 17. Identity management is crucial for security.

Because the machine-to-machine paradigm favors the use of digital certificates for identity management, certificate management is a critical element of the solution. The hardest part of a large scale certificate management function is the timely management of certificate revocation because of the complexity it would otherwise bring to each device in a TTS. The Security Fabric framework addresses this need by centralizing certificate revocation management and maintaining a whitelist of all valid certificates. In most TTSs, the list is not long. Further, the whitelist becomes one of those special credentials that is delivered during authentication (or periodically between registrations). The managed device therefore takes advantage of the power of the central management system for managing revocations, but then automatically receives the most up to date whitelist credential every time the registers, or when receiving a multicast directive to refresh the whitelist thereafter.

Note that the Security Fabric framework does not presuppose that there will ever be a single master technique for all time for identity management. It does presuppose that current standards are a good starting point, but that other techniques will rise in popularity as security attacks become more sophisticated. The Identity Metasystem allows for evolutionary expansion of identity management standards over time.

Within the Security Fabric framework, the end device must first receive an asymmetric identity (secret key + public key) during the manufacturing process by a separate trusted process. The public key can be used to identify the device, but the secret key can be used to authenticate that the device is genuinely who it claims to be. For mutual authentication of devices and services to each other inside a TTS, the Security Fabric will use Kerberos as the trusted third party within the Security Fabric framework.

The authentication will actually occur on two levels:

1. Network access for the device as a whole to be allowed to use the network at all.
2. Application sessions individually interconnecting their client side to a central server.

The reason the authentication is needed on two levels is because some of the applications may be administered by different authorities, as when some of the applications are controlled by the utility and some are controlled by third parties selected by the customer.

The sequence of Kerberos interactions is shown in the diagram below.

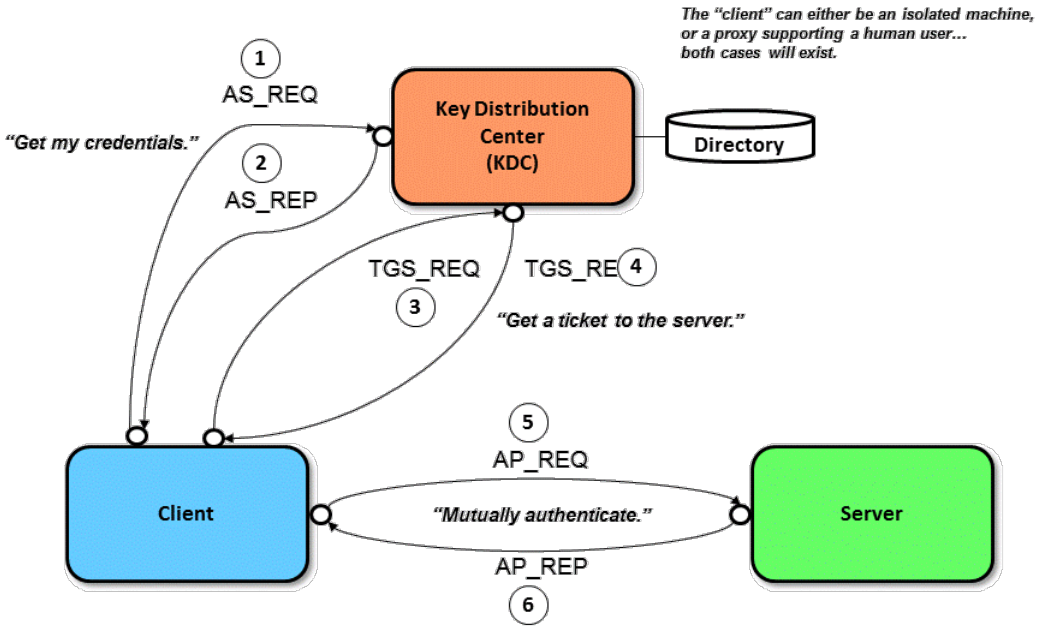

Figure 18. Kerberos process for mutual authentication.

Kerberos allows two peer-to-peer services to negotiate trusted interconnection. When a device powers on or resets, it begins by contacting the central Key Distribution Center, a trusted third party, to obtain secure credentials that will help it identify itself to others. Kerberos assumes that the communications network is inherently unsecure and makes no assumptions during the initial sequence. Once having been initialized, the client portion of a device's management agent can then request "tickets" so that it can be introduced to server services in other devices. Kerberos provides authentication information in the ticket encrypted once for the client side using the public key in its configuration records for the client device, and then also for the server side using the public key in its configuration records for the server device. If each device is genuinely who it claims to be, the decryption of the ticket using the respective secret keys will allow each side to securely authenticate to the other to whom it is interacting.

The Kerberos technique also allows (directly or indirectly) for additional credentials to be delivered to the device at registration time such as the following:

- Secure path to the intended intermediate manager for the device
- Secure path to the configuration synchronizer service for device discovery
- Secure path to the repository for policies and firmware evolution.

These centralized and distributed management services are essential for supporting the management functions required by the end device as was described earlier. However, it is the security technique of mutual authentication that allows the management connections used by the Security Fabric framework to establish themselves dynamically and securely.

3. ACCESS MANAGEMENT AND AUTHORIZATION

Access management and authorization policies work together to decide whether an authenticated device or user has permission to use a service or its resources.

Policy decisions can sometimes be made without exact knowledge of the identity of the other party. Sometimes only the secure knowledge of certain attributes of the requesting party is needed without having full knowledge of the other party's full identity or private information. This is useful in situations where the two parties do not trust each other in all matters, but for the situation at hand limited trust will work just fine. An example of this situational trust would be during connection of a home gateway device to a utility demand response system that would require the identification of the user as having subscribed to the intended service, but not necessarily the user's social security number, date of birth, or plans for being away from home. The service ID and password are sufficient for access control purposes.

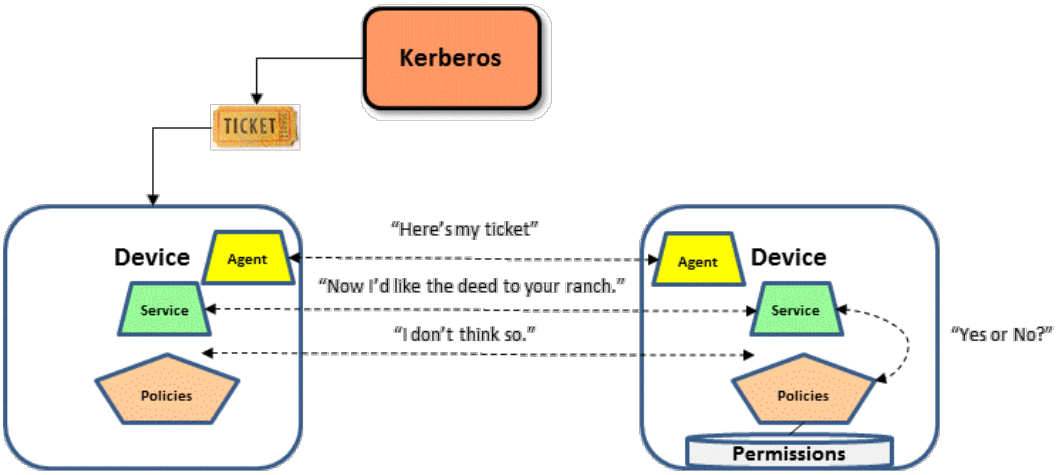

Figure 19. Authorization policies support access control.

Also, a sequence of individual or group permissions allows the policy system to differentiate between a user who is a customer versus a user who is a systems administrator. These policy rules are authorization rules that the end device uses to protect its data, or that the central service uses to defend its resources.

3.1. Auditing

Auditing is the act of recording essential events for after the fact analysis of what happened. The newspaper reporter's who, what, when, where, why, and how are all essential data to record about each event worth remembering.

Auditing uses the event and log management service on a device to forward the history of important events at the device to the central management system for either instantaneous action, or for analysis of trends.

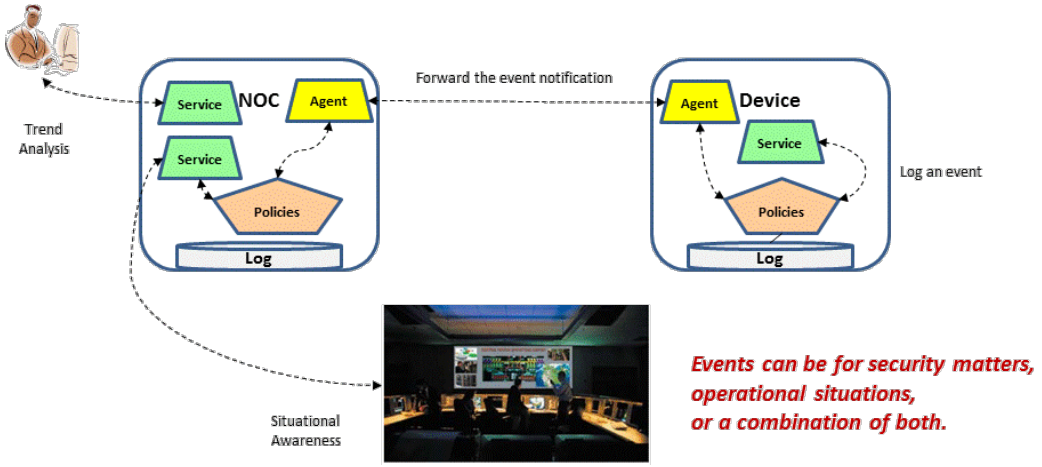

Figure 20. Event management supports operational and analytical operations.

Auditing and the associated event and log management service are essential mechanisms for supporting the fault management functions described in the previous section.

3.2. Confidentiality

Confidentiality has to do with encryption. The Security Fabric framework supports a variety of encryption capabilities. Some of these capabilities are implemented in

software or firmware. Sometimes performance or capacity requirements dictate that these operations use hardware acceleration to meet service level agreements.

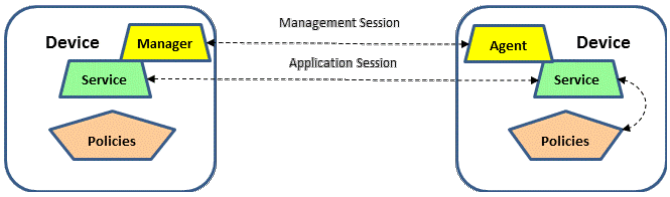

Figure 21. Separate encryption tunnels for applications and management.

Initially, the Security Fabric uses a separate IPsec tunnel for management versus application communications sessions. The session keys are established at registration time, but the session keys are evolved based on time, volume, or both based on policy. AES 256 is the minimum required today, but the implementation for key management vs. data vs. communications can vary.

Encryption is also used to provide privacy of data stored on persistent flash memory. The keys for private storage are derived from information stored securely on a hardware security module which is described further later in this paper. But the important fact remains that confidentiality and privacy can be maintained for data during communications interaction and also while at rest on the persistent storage on the end device.

3.3. Integrity

Integrity management essentially uses message digests and digital signatures to ensure that messages between services in devices have not been altered, even if they are not encrypted.

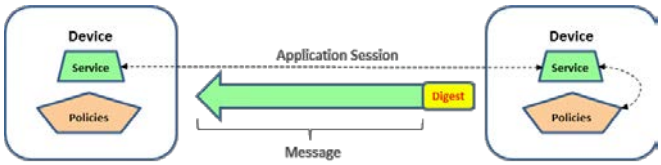

Figure 22. Message digests are used to provide integrity for interactions.

A message digest is computed based on a secure hash total of the binary contents of the message using an algorithm like SHA-256, and then is digitally signed using either RSA or elliptic curve signatures. A message received by a service can be deemed to have integrity if a recomputation of the digest produces the same result as is found attached to the message.

The digital signature can later be used to show non-repudiability of the approval given to proceed with processing a transaction. This aspect is sometimes very useful in diagnosing problems where there is a difference of opinion as to whether approval was given or not.

3.4. Availability

Availability has to do with detecting denial of service (DoS) attacks, and then defeating the harmful aspects, and then continuing to provide service – even while the attack is still going on.

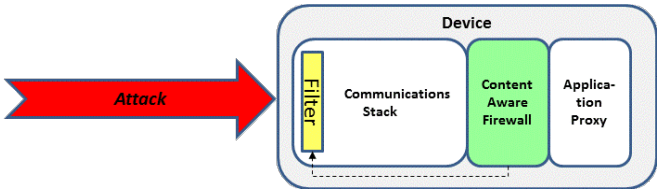

Figure 23. Device internal firewall dealing with a DoS attack.

The Security Fabric framework uses an embedded implementation of a content-aware firewall to detect evidence of such an attack underway. It then activates a filter early in the communications stack to immediately discard further reception and processing of packets from the offending source system.

3.5. Provenance

Provenance is all about constructing a "Supply Chain of Trust". The recommended path of the supply chain is shown in the following diagram.

Provenance depends on knowing and understanding the sources of supply, both in terms of hardware as well as software and configuration data administration. You must be able to trust those who design the silicon circuits, those that fabricate the components, and those that assemble the parts into finished goods. You must also know those who design the firmware, who develop the firmware, and who has authorized loading the firmware into the devices. Similarly you must know who commissions the devices, who installs and activates the devices, and who makes over-the-air changes to the devices once they are in the field. At critical junctures in the process, you need to have red team inspections to see if individual devices or the system as a whole can be penetrated despite all the planning that has gone into the deployment. You must have qualification steps to see that the parts and the system work. And for interoperability purposes, you need to make sure that the devices and subsystems are certified to interact with multiple vendors' components.

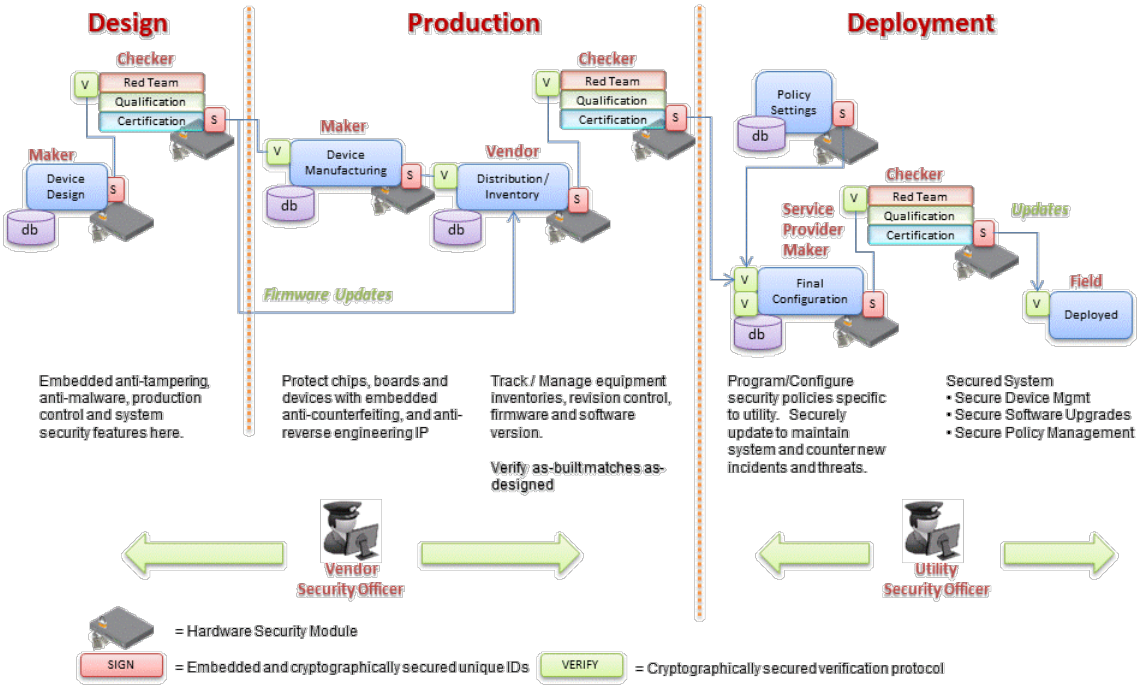

Figure 24. The approach to provenance: the "supply chain of trust".

Finally, and most importantly, the device itself must have a process for attesting that all firmware, policy rules, and control data loaded or operating on it can be attested as trustworthy during changes, and also at random times between changes.

The Security Fabric framework provides a basis for the life cycle supply chain of trust.

4. SECURE SILICON: THE BASIS FOR THE SECURITY FABRIC

With the current state of the art of security attacks on the power grid here in the United States, it is now believed by many that it is impossible to secure a remote unattended field device with software alone. The attacks are too sophisticated. Therefore the Security Fabric framework provides for secure silicon approaches in addition to standard firmware-based management services.

The end devices themselves have either internal or external hardware-level assistance in maintaining security and management control. The diagram below shows the high-level integrated circuit components used in a typical secure device.

A typical embedded system would use either an ARM or Intel Atom-based hard processor integrated with field programmable gate arrays (FPGAs) in a single complex logic arrangement. The FPGA uses a soft implementation of logic and connections that allows for customization, optimization, and reconfiguration of devices in the field. Flash memory is used for persistent storage of configuration data, application data, and audit logs. The separate hardware security module (HSM) provides for secure key management used by external authentication functions.

It also provides for storage of electronic value in a tamper-proof enclosure. The NIST has different specifications for the HSM depending on what tolerance for risk the device can bear. Communications-related circuit elements are usually modularized off onto a separate daughter card so that upgrades can be made over time to memory, processor speed, and the like without needing to replace the entire set of circuit elements.

The HSM and elements of the FPGA logic are especially useful to the security of the device. In all secure systems there must be something that has to be kept secret. In this case, it is the HSM that provides the opportunity to keep master keys and the key management system secret from prying eyes. In addition, the FPGA logic allows for the opportunity to include state machine logic and instrumentation that can watch the execution of the system

on the device without the software knowing that it is being watched. There are several critical points that can be monitored for events that provide evidence of tampering:

- Signaling reset without warning while the device is in production operation.
- Use of the Joint Test Action Group (JTAG) hardware testing interface to monitor and reset hardware registers unexpectedly while the device is in production operation.
- A change in the pattern of process context changes or thread change sequences from the normal pattern of operation as seen by monitoring the bus.

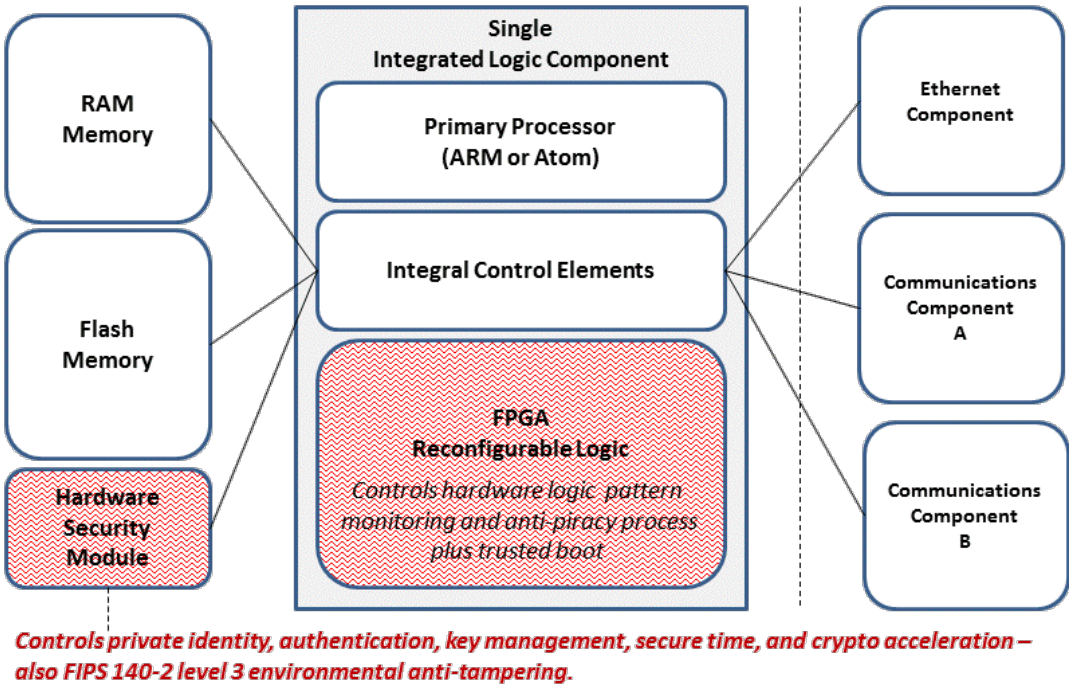

Figure 25. Secure silicon as the base for the Security Fabric.

In each of these situations it would be appropriate for at least a notification over the management channel that an anomaly is in process. Depending on the situation, it may be useful to actually block some of these operations unless they are specifically authorized by an authoritative source or policy.

The use of the FPGA logic also helps create anti-piracy capabilities for the circuitry in that the component can be manufactured off shore, but finished on shore in a trusted facility with repurposed FPGA logic.

It is the combination of multiple elements that provides the strength of the security as opposed to overdependence on a single point strategy.

5. FUTURE DIRECTION

As the need for security advances, Intel is investigating the creation of new components that will support security in the embedded world. Currently Intel offers individual components suitable for deployment for control of security and control of workstations and servers. Examples of these components include the following:

- TPM – Trusted processing management
- AMT – Out of band device management processor
- TXT – Virtualization component
- Cryptographic Acceleration.

The Security Fabric as it is initially deployed contains a number of firmware features that are candidates for hardware deployment:

- Layer 2 IPsec
- Layer 3-5^{+} Content aware firewall
- Layer 6 Application Proxy
- The Hardware Security Module.

All of these elements join to provide the fundamentals for security, but many have asked whether they might all be rolled into one component suitable for supporting the embedded environment. Such an approach might be envisioned as a single complex component as depicted in the diagram below.

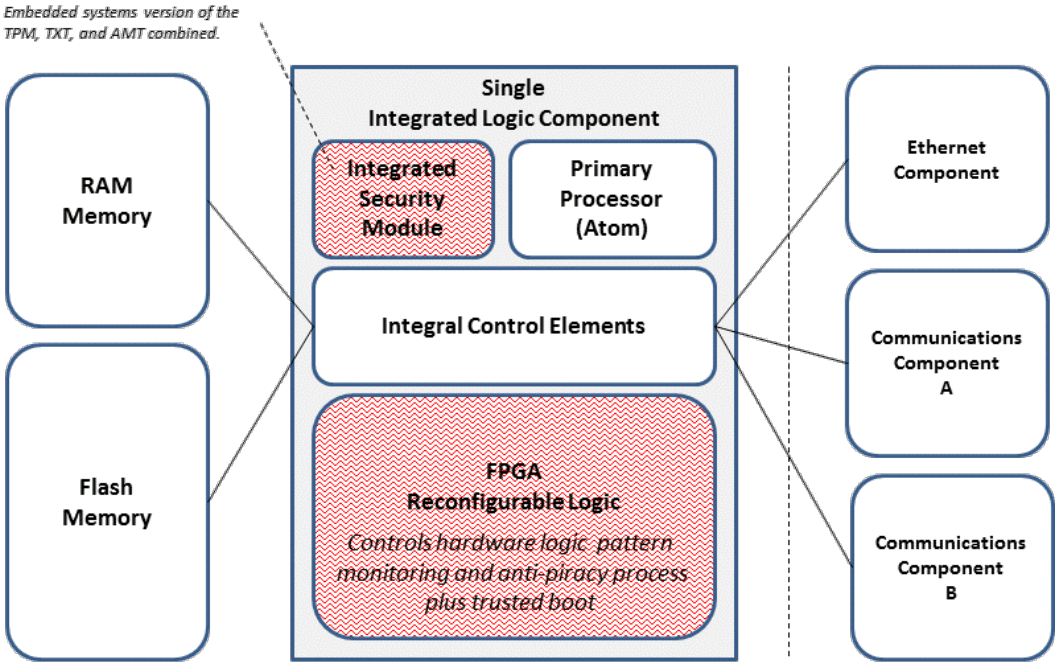

Figure 26. Fully Integrated Embedded System Security Component.

The FIPS 140-2 level 3 status currently certified for the HSM requires certain analog mechanisms to maintain it tamper resistant capability. These are for thermal, shock, and electrical attacks.

Intel is investigating the viability of such an offering with its embedded systems OEMs right now and will pursue such roadmap items as is warranted by the embedded business.

6. DEFENSE IN DEPTH

The idea behind the *Defense in Depth* approach is to defend a system against any particular attack using several, varying methods. It is a layering tactic, conceived by the National Security Agency (NSA) as a comprehensive approach to information and electronic security [1].

Defense in depth is originally a military strategy that seeks to delay, rather than prevent, the advance of an attacker by yielding space in order to buy time. The placement of protection mechanisms, procedures and policies is intended to increase the dependability of an IT system where multiple layers of defense prevent espionage and direct attacks against critical systems. In terms of computer network defense, defense in depth measures should not only prevent security breaches, but also buy an organization time to detect and respond to an attack, thereby reducing and mitigating the consequences of a breach.

The Defense in Depth strategy as introduced in a previous section separates the structure of a system or component into layers that can provide buffers between the outside world and the critical elements. Each layer is managed and monitored as if it is a separate device, but each can be used as a point of defense when irregularities are discovered. If a layer is determined to be compromised, it can be replaced wholesale and transitioned back into production – many times without stopping the rest of the device. The key vehicles for establishing defense in depth are the separate management channel and a separation kernel hypervisor that supervises the end device. (A hypervisor is a hardware virtualization technique that allows multiple operating systems or standalone partitions, termed *guests*, to run concurrently on a host processor.)

The separation kernel hypervisor provides a defense in depth partitioning of key firmware components within an end device.

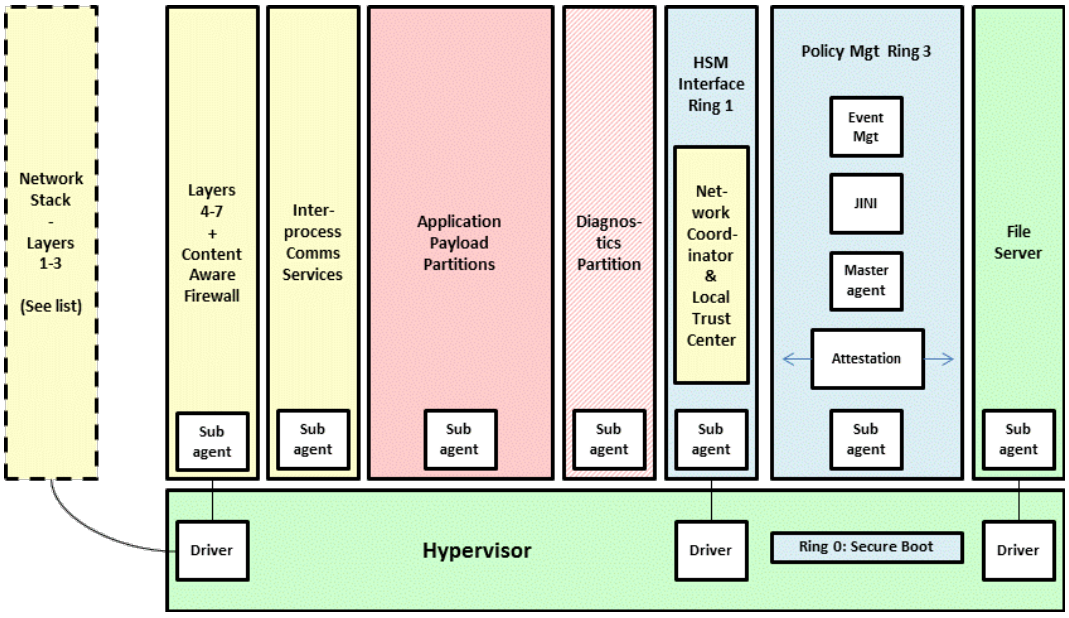

Figure 26. Defense in depth using a hypervisor as the separation kernel.

The hypervisor allows the administrator to arrange the processing into different partitions that are convenient for both application processing as well as the management services. Each partition is its own memory managed address space so that inter-partition reference is impossible except for message passing that is strictly controlled by the hypervisor itself. Compromises can be isolated and controlled by partition.

Some of the management partitions depicted in blue in the previous diagram are configured to provide rings of recovery similar to the original Multics system as shown in the following diagram.

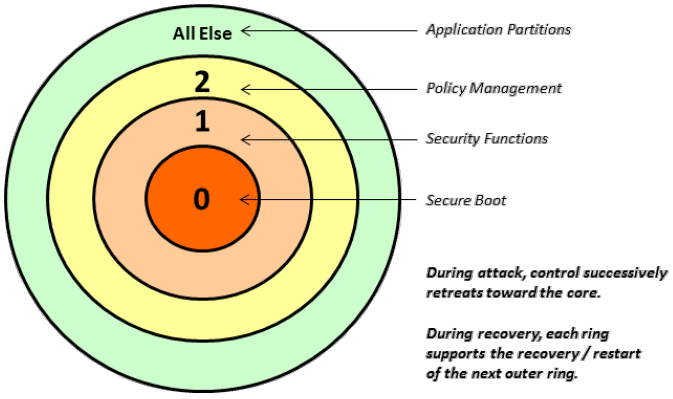

Figure 27. Rings of Recovery.

The secure boot is at the core. The secure management services in and around the HSM are at the next level. The policy management service is at the next level. And then the rest of the services are at the last level. As compromise occurs, the system retreats successively to inner rings. At worst or at power on, the system starts from the core and successively relaunches each next higher ring. In this way, a compromise of communications, applications, diagnostics, or other aspects of the system do not instantly stop the system in a state that the supervisory elements cannot recover or at least operate in a degraded mode.

Both coded logic and control data are signed during development prior to promotion to production status so that whitelist attestation can protect both. This is illustrated in the following diagram.

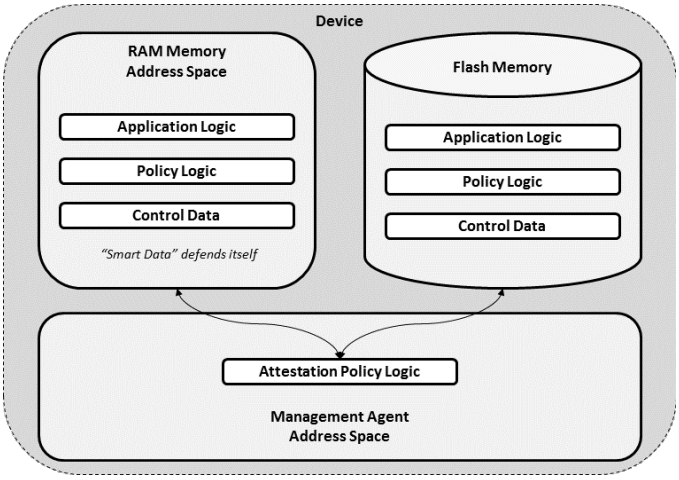

Figure 27. Whitelisting can protect both memory and persistent storage.

All soft components like firmware load modules or control data provisioning sets are digitally signed at the completion of QA testing. When new modules are pulled into a remote device, they are not immediately used. Instead, they are submitted for examination by the whitelisting attestation policy logic before used. Even if they pass the attestation check, they may not immediately be used. The transition management policy logic looks to see if the remote device is in a state that will allow it to switch to a newer version of a component without disrupting a critical power process already in operation. If it is safe to shift, transition management policies will load and initialize the new component.

The defense in depth approach separates processing on devices into a subsystem of "moats and drawbridges." This is an important first step – but so is the strategy of providing a moving target within the configuration to confuse the attacker.

Figure 28. Out-of-band management dealing with an attack on a device.

The fundamental concept of the moving target strategy is to confuse would be attackers by evolving the configuration and the policies within an element at random times during normal operation – but also more rapidly during an attack over long lived situations. The out-of-band management channel is especially useful. During an attack, the moving target bobs and weaves as it evaluates, mutates, and then assimilates countermeasures. This is an appropriate model for the moving target strategy using the Security Fabric framework. While an attack may occur on an application service in a device, the private out-of-band management channel can be used to communicate what is going on, receive countermeasure instructions, receive new firmware, or even new policy logic for closely watching an attack underway, or for aggressively launching a counter attack if the situation warrants.

The theory of the moving target strategy is that it is harder for an attacker to penetrate a remote device if it appears to be constantly changing like a kaleidoscope. If between identifying a target and launching an attack, the target suddenly becomes something else with a different array of defenses, it is difficult for a simple assault to be successful. If a compromise is detected, the affected section can be determined and isolated, and the cooperating parties can renegotiate a degraded arrangement and continue safe operation while remediation addresses the problem. Part of that new arrangement during an attack can be an increased level of monitoring of events, and also the dynamic loading of additional counterattack logic. Dynamic changes to a system under stress cannot be made in a capricious way. Stability during attack depends on determining in advance what the likely attack scenarios are, developing profiles of what needs to be in place during the attack, and then system testing the arrangement to ensure that all alternate plans work well even during the rearrangement. The profiles provide configuration information that is needed for normal operation and different configuration plans to be used during attack. Many times those scenarios will be different if the device has access to communications with the management over the secure management channel or not. If totally offline, a different posture is typically appropriate.

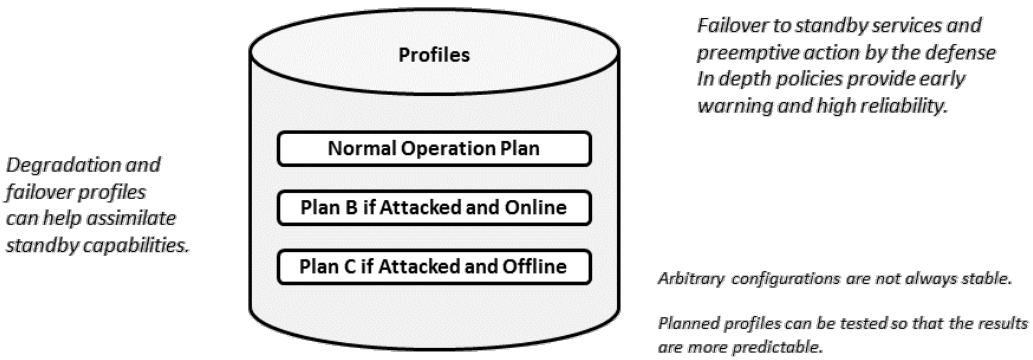

Figure 29. Profiles support continued operation while under attack.

The policy may isolate affected areas to smaller trust islands – providing a measure of resilience during an attack. The profiles are stored on encrypted persistent storage for access by the policy ruleset.

The following sequence illustrates the resiliency of the moving target capability.

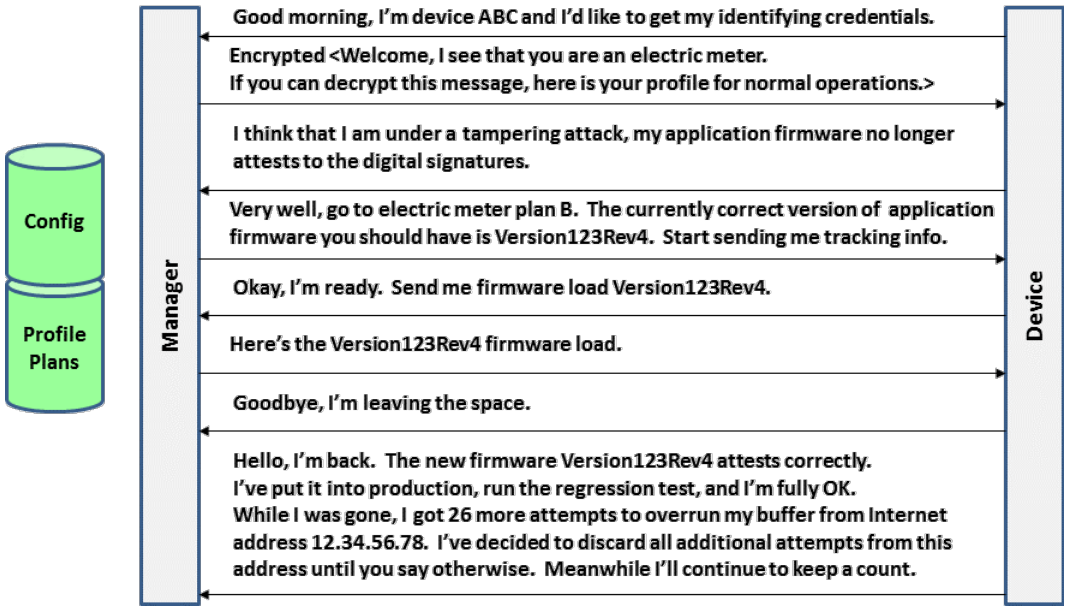

Figure 30. Example of "moving target" strategy.

The scenarios are usually governed by finite state machine logic that allows transitions from one posture to another as the dynamics of an attack in progress ensues.

Since there are so many other opportunities for attacks and so little defense currently in place, when professionally developed defense is actually in place, the defense usually succeeds since it is far easier for the anonymous attacker to just move on to another simpler opportunity somewhere else as opposed to engaging the moving target. Still, no defense will ever be perfectly secure if the rewards to the attacker are great enough. So the correct approach to planning for use of the Security Fabric framework is to assume that each of the defenses will eventually be compromised – and then work out what the steps to remediation need to be to recover. No other approach is practical other than eternal vigilance. The Security Fabric framework offers professional tools for defense.

7. IN SUMMARY OF THE SERIES

The Security Fabric framework will be a commercial product that captures the essence of the concept of the "tailored trustworthy space" that the Department of Energy is pursuing. It is composed of elements based on security principles that have been known for many years. Yet with the complexity of the intelligent grid, it has not been appropriate for manufacturers to engineer an end-to-end solution all by themselves. The standards process takes too long and is riddled with compromise. Plus, the fine points are never realized in a standards committee – the practical experience only comes from attempting to build a commercial product in a free enterprise market.

The Security Fabric framework uses policy managed distributed processing principles at its core. It also recognizes the fact that there will never be one and only one technique for handling key management or any other essential component of the architecture. Instead, the emphasis of the framework is to offer integration and interoperability in the face of diversity of opinion and multiple situations. It allows for new security approaches to be integrated as they are developed.

Although the Security Fabric framework is designed to allow choice when originally planning at TTS in terms of key management, changing plans at a later date is harder to do. For devices that are intended to be in the field for many years and upgraded "over the air," the upgrade must be carefully planned. For devices whose secret identity has been created just after manufacturing time, the recommendation by NIST is to plan a controlled evolution of secret keys once a year. But the upgrade of key management firmware itself in the HSM must be controlled by secure upload and attested before being made operational. Similarly, other management firmware and control data can all be upgraded using the change management process identified in Part 2 of this series. But change of multiple elements at the same time must be tested as a whole system prior to distribution. Transition control must be coordinated using signaling from the configuration management system knowing that not all components will ever be at precisely the same revision level. So each component must negotiate its capabilities each time it registers. And in all cases, policy must be in place that allows a rollback to a known stable release if operational anomalies appear.

The Security Fabric framework is based on solid system and network management principles. It offers a starting kit of interoperable security management services, a way to evolve those services, and a way for testing for interoperability in the face of diversity and situational evolution.

The specific plans for security management cover both software and hardware approaches to security. But security is not just in the execution environment. Provenance stretches all the way back through the supply chain to the origins of all the elements that are used in a TTS.

The Security Fabric is not static. It offers defense strategies like Defense in Depth through the use of the secure separation kernel. The configuration management and fault management mechanisms, along with the policy management elements, allow administrators to establish a moving target to protect against attack – not just in normal operations, but also at an augmented pace during an attack.

The Security Fabric will be useful for all utilities, large or small, regardless of their circumstance or situation. It is intended to serve as the basis for securing the power grid using the efficiencies of mass customization. It also offers the opportunity for the required continual optimization over the entire life cycle.

References

[1] National Security Agency - Information Assurance Solutions Group. Available at http://www.nsa.gov/ia/_files/support/defenseindepth.pdf