Building a Secure Enterprise[1]

Kevin E. Foltz and William R. Simpson
Institute for Defense Analyses, 4850 Mark Center Dr.
Alexandria, Virginia 22311

[1] The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

ABSTRACT

Building a system that is functional and resilient to change can be challenging. For many established disciplines, knowledge and techniques have been developed to build something to achieve a design goal. Architects do this for houses and other structures. Engineers do this for a variety of systems, both physical and electronic. The challenge comes when trying to design a system that not only achieves a goal, but is also adaptable to accomplishing new goals in the future or the same goal in a different environment or context. This requires thinking beyond the standard design process for the current goal, environment, and available technology. This paper describes an approach to build a system for both today and tomorrow. The methods in this paper are applicable when 1) there is a defined goal, 2) accomplishing the goal requires a substantial up-front investment in planning and resources, and 3) the goal or its context or environment are likely to change over time. This approach is applied to the specific case of the Enterprise Level Security (ELS) system, which is an architecture for a secure enterprise to share information. The benefits for ELS included improved security, cost savings, and improved vendor interactions.

Keywords: Enterprise, Security Concepts, IT Security, System Design, Security Tenets, Security Requirements.

1. INTRODUCTION

The Enterprise Level Security model is a new approach to enterprise security. Its goal is to make information sharing secure and convenient within a large enterprise. It does this by focusing on endpoint security rather than perimeter security. After many years of development the first ELS systems are beginning to take shape.

The complexity of ELS and the length of time in development present challenges to its development. Many of the people involved come and go. The scope of the project increases in size and functionality. The security environment and threats continue to evolve as the work progresses. These pose challenges for focusing efforts in the right direction while adapting to continuous changes throughout development.

The work in this paper describes the development approach for ELS. It has generic components that are likely to be relevant to many projects, as well as security-specific and ELS-specific components that serve to illustrate the level of detail involved in the process.

The following sections discuss ELS, its development model, and the methods and benefits of using it.

2. ELS BASIC SECURITY MODEL

The goal of ELS is to provide access to information in an enterprise through secure, trusted sharing mechanisms that protect the integrity of the information from creation through utilization. ELS is both an architecture and a philosophy that allows intelligent sharing of information among the entities in the enterprise and partners while maintaining a strong security posture that is both uniformly applied and standards-driven throughout the enterprise. ELS is specifically for a high-assurance environment, in which security is of primary importance and attacks on the system are likely to be frequent and sophisticated.

ELS is focused on active entities and their communications. An active entity for ELS is a credentialed requester or provider of information through a web-based interface. This includes human users, non-human requesters, applications, and web services. Active entities have a persistent credential for identity and a temporal credential for access to applications and services.

Active entities within the enterprise are registered within the enterprise and have unique identities with associated credentials. Active entities are known identities, and "anonymous" is not one of those identities. Communication between active entities uses identity credentials to perform bi-lateral end-to-end authentication prior to exchanging information. Authorization in the operational environment is implemented by a verifiable short-lived credential with embedded access claims.

Claims represent satisfaction of access control rules and are included as part of an authorization credential issued and signed by a trusted credential issuer. The access control rules for a data set are provided by the data set's owner. The data owner may also request, as part of the access control requirement definition, additional information about the requesting entity to determine the level of privilege.

The description of ELS in this section is not comprehensive, but it gives some of the important ideas of ELS. More technical information about ELS can be found in [7- 22]. The focus of this work is on how to build a complex system that accurately reflects the original design goals when built and as it evolves to adapt to changing needs. As this process involves more details the ELS example will be used for reference.

3. SYSTEM DESIGN AND MAINTENANCE

For system design and maintenance, a set of core tenets is the starting point. These describe the desired highest-level properties and design philosophies of the system. They do not indicate what to build but provide guidance that influences all decisions about what to build, how to build it, and the choices of finer grained details. From these tenets, key concepts describing the system to be built are derived. The concepts describe what to build at the highest level. They are not sufficient to build the system, but they provide a vision of some of the critical parts and how they interact. From the concepts a set of requirements are developed, which are closely tied to the concepts and provide sufficient detail to start building the system. The idea is that an enterprise can use these requirements as the foundation for building a system and supplement them with additional details as the design is refined.

This method bridges the gap between the builder of a system, who is focused on implementation details, and the designers of the architecture, who focus on the high-level properties of the system. It also enables a systematic assessment by tying requirements to the overall design goals of the system. This facilitates modification to the system by showing which tenets, concepts, and requirements are affected when one or more of them change due to changes in technology or adjustments to goals.

The idea of the basic security model can be visualized in Figure 1. The tenets are like solid, heavy rocks that are positioned in the beginning and form the structure for everything else to build upon. These rarely change, and when they do it reflects a major change in direction or external circumstances.

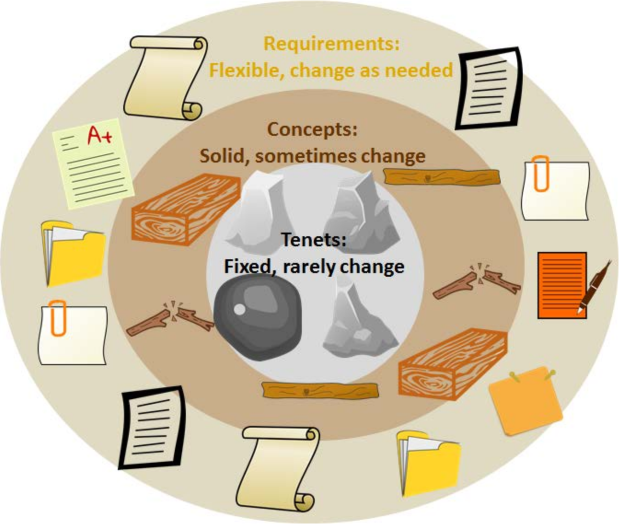

Figure 1 Basic Security Model Visualization

The concepts are represented as wood, which is solid like rock, and can last a long time, but the structures they build require maintenance and repair. For wooden structures, components can break or rot, but with maintenance and repair they can last a long time. Concepts are meant to last and be structural elements, but they are not as solid or resilient as the tenets. The concepts are tailored to the system under construction, and they are easier to change than the tenets. Just as the particular system is more likely to change than the overall goals, concepts are more likely to change than tenets.

The requirements are generally derived from the concepts, and paper, a wood derivative, is used to represent them in the next layer. The requirements are more flexible and represent the particular choices for the system functions, which may change more rapidly than the functions themselves. Paper, being easily folded, shredded, moved, and otherwise changed, represents the idea that requirements may change more often than concepts.

The linkages are not shown, but they are an important part of the model, as they define the structural connections from the tenets to concepts and requirements, much like an architectural diagram can show relationships between a rock foundation, a wooden external structure, and paper elements within.

This bullseye representation is the center of the system model. Additional rings can be layered outside of this core, including the following:

- Additional detailed functional requirements
- Implementation, including products, their versions, and configurations
- Operational guidance on how to use the products

By continuing the linkages outward a mapping can be made from tenets to operational guidance through the intermediate layers. Changes to the internal layers, especially tenets and concepts, have a large effect, as such changes generally propagate outward along their connections. The goal of this design method is to design the architecture to address changes at the outermost level possible for the type of change it is. Major changes will necessarily be addressed near the center and have significant effects on system design, but small changes in a properly designed system should only affect the fringes and result in quick fixes using standard methods.

The sections below describe the tenets, concepts, and requirements in more detail. Examples are provided for the development of the ELS system.

4. CORE TENETS

The tenets are the core drivers of all architectural decisions. Some of the ELS tenets are as follows:

0. **The enemy is present**. Malicious entities are present and our systems need to function with these embedded threats rather than rely on filtering them out.
1. **Simplicity**. Added features come at the cost of greater complexity, less understandability, greater difficulty

in administration, higher cost, and/or lower adoption rates that may be unacceptable to the organization.

2. **Extensibility**. Any construct should be extensible to the domain and the enterprise, and ultimately across the enterprise and coalition.
3. **Information hiding**. This involves revealing to the requester and the outside world only the minimum set of information needed for making effective, authorized use of a capability.
4. **Accountability**. This means being able to unambiguously identify and track which active entity in the enterprise performed each operation.

The tenets generally fall into two categories: must-haves and design principles. Tenets 0 and 4 are examples of must-haves. ELS must be able to function with malicious entities that are attacking from outside and inside the system, and it must provide accountability. Simplicity, extensibility, and information hiding are examples of design principles. These are not must-haves, as they are always in some tension with each other. It is not the absolute value of these tenets that matters but the relative values and their balance. For example, a complex solution may be acceptable if the goal itself is inherently complex. Simplicity means that the complexity in the system reflects the complexity of the goal.

The tenets for most projects will be similar. There may be differences, such as valuing anonymity over accountability in a privacy-based system, but things like simplicity and extensibility are common design themes that are likely to be repeated broadly beyond just enterprise security.

5. KEY CONCEPTS

The key concepts are based on the tenets and address specific architectural decisions that relate to the requirements. These are likely to be similar to the ELS concepts for many security-based systems, but different for projects with other goals. The concepts form a bridge between the high-level tenets and the technical requirements by describing the high level system in a way that maps to the tenets. A subset of the ELS concepts follows:

0. ELS-specific concepts. These are important choices based on current technology. Due to their overall importance to ELS they are considered as a single concept.
   a. PKI credentials are used for active entity credentials.
   b. Security Assertion Markup Language (SAML) with claims is used for authorization credentials.
   c. TLS v1.2 is used for end-to-end confidentiality, integrity, and authentication.
   d. A Security Token Server (STS) is the trusted entity for generating authorization credentials.
   e. Exceptions in implementation must have a documented plan and schedule for becoming compliant.
1. A standard naming process is applied to all active entities.
2. Authentication is implemented by a verifiable identity claims-based process.
3. Identity claims are tied to a strong vetting process to establish identity.
4. Active entities verify each other's identity.
5. The verification of identity is by proof of ownership of the private key associated with an identity claim.
6. Active entities act on their own behalf.

Concepts are linked to the tenets. Linkages are shown in Figure 2. The connections between tenets and concepts are important for future changes as they allow traceability and a way to determine the effects of changing any of the tenets or concepts on associated concepts or tenets.

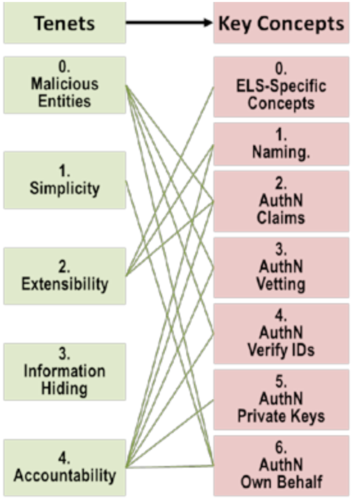

Figure 2 Mapping ELS tenets to concepts

The ELS-specific concepts are a collection of important protocols and standards that are to be used across the enterprise. Although each of these could be taken individually as a requirement, they together form such an important part of the ELS model that they are elevated to the level of a concept. The other concepts listed all relate to authentication, which is an important part of the ELS model.

6. TECHNICAL REQUIREMENTS

The technical requirements are based on the key concepts and are traceable through the concepts to the core tenets. A subset of the requirements for ELS follows:

1 Active entities shall be named in accordance with DoD Naming standard.
2 Active entities within the enterprise shall have unique identities.
3 Active entities shall use credentials from approved certificate-issuing authorities.
4 Active entity communication shall use two-way, end-to-end PKI authentication.
5 No active entity shall be anonymous.
6 Authentication tokens shall not be allowed.
7 Traditional single sign-on shall not be allowed.

8 Private keys shall be stored in tamperproof, threat-mitigating storage to which only the associated entity has access.
9 Impersonation of active entities through sharing of private keys or issuing of duplicate credentials shall not be allowed.
10 Proxies or portals shall not be allowed, because they cause ambiguity in identity.
11 Active entity authentication shall use only primary or derived credentials.

The concepts and requirements are generally more closely related than the tenets and concepts. The authentication related requirements generally reference the authentication related concepts, whereas the tenets connect more uniformly across the concepts.

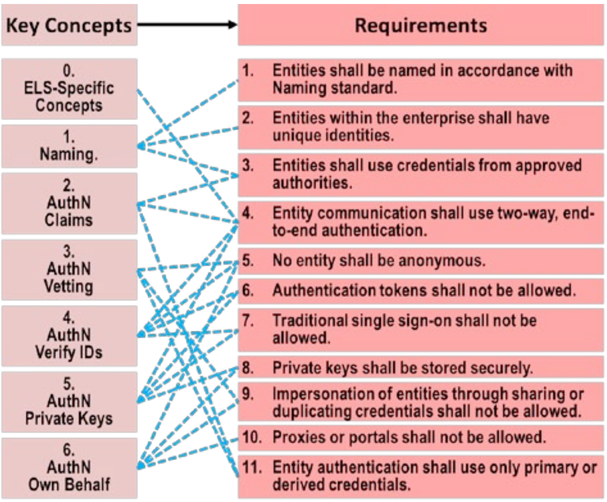
Key Concepts
Requirements
0. ELS-Specific Concepts
1. Naming.
2. AuthN Claims
3. AuthN Vetting
4. AuthN Verify IDs
5. AuthN Private Keys
6. AuthN Own Behalf
1. Entities shall be named in accordance with Naming standard.
2. Entities within the enterprise shall have unique identities.
3. Entities shall use credentials from approved authorities.
4. Entity communication shall use two-way, end-to-end authentication.
5. No entity shall be anonymous.
6. Authentication tokens shall not be allowed.
7. Traditional single sign-on shall not be allowed.
8. Private keys shall be stored securely.
9. Impersonation of entities through sharing or duplicating credentials shall not be allowed.
10. Proxies or portals shall not be allowed.
11. Entity authentication shall use only primary or derived credentials.

Figure 3 Mapping concepts to requirements

The connections between concepts and requirements are shown in Figure 3. Because these concepts and requirements are all related to authentication, there are many links between them. By combining the tenet to concept and concept to requirement connections the paths between tenets and requirements can be shown. The full mapping of all ELS tenets, concepts, and requirements is shown in Figure 4.

7. MAPPINGS

The full mapping can be used to trace requirements back to concepts and tenets, which can help in making and justifying implementation decisions. For example, the enterprise may consider inserting a proxy in front of a server and sharing the server's certificate and private key with the proxy to enable in-depth security scans on incoming encrypted traffic. This is a common practice, but it results in the following ELS violations:

- Req #2 – the proxy shares the same name as the server by using its certificate and private key.
- Req #4 – the proxy breaks the end-to-end authentication by acting as the server.
- Req #8 – the proxy is not the appropriate entity to access the server's private key.
- Req #9 – the proxy impersonates the server.
- Req #10 – the proxy causes ambiguity in the server's identity.
- Req #12 – the proxy has no claims but is accessing the server.
- Req #14 – the proxy has no attributes.
- Req #22 – the proxy breaks the end-to-end TLS connection.

Tracing these requirements back to related concepts, we see that the most often referenced is Concept 6, "Active entities act on their own behalf." The proxy is a direct violation, since it acts on behalf of the server when communicating with requesters. Others with multiple references are Concept 5, "The verification of identity is by proof of ownership of the private key associated with an identity claim," which again is violated directly by sharing the private key of the server with the proxy, and Concept 8, "Service providers use identity and authorization credential claims to determine access and privilege," which is violated because the proxy gains access to the service without valid identity or authorization credentials.

Extending this process, we can link back from these concepts to the related tenets. The most referenced are Tenet 0, "Malicious entities are present," Tenet 4, "Accountability," Tenet 2, "Extensibility," and Tenet 11, "Trust but verify." When using proxies we provide more points of exposure to enemies, we reduce accountability by spreading identities across multiple nodes, and we reduce the ability to verify and validate identity. Extensibility is affected less directly, but many of the choices made for extensibility are negated by using proxies.

The example of proxies was chosen to illustrate a serious violation. Other changes might have minimal impact. For example, choosing not to scan outputs for consistency would violate Requirement 27, which maps only to Concept 21, and Tenets 0 and 15.

8. BENEFITS

The benefits of using tenets, concepts, and requirements to guide the development process depend on the goal of the system to be built. A general benefit is the continued adherence to initial design goals throughout all the decisions in the development process.

For ELS the benefits can be grouped into three major categories, as illustrated in Figure 5. The first is security. Security is the main design goal for ELS, and adhering to the original tenets through all the changes and decisions in the design process helped to maintain a strict adherence to this goal despite constant outside influences that attempted to impose their own goals at the expense of security.

A second benefit is cost savings. By designing the system to address changes at the fringes of Figure 1, less time is spent redesigning the system, since changes are

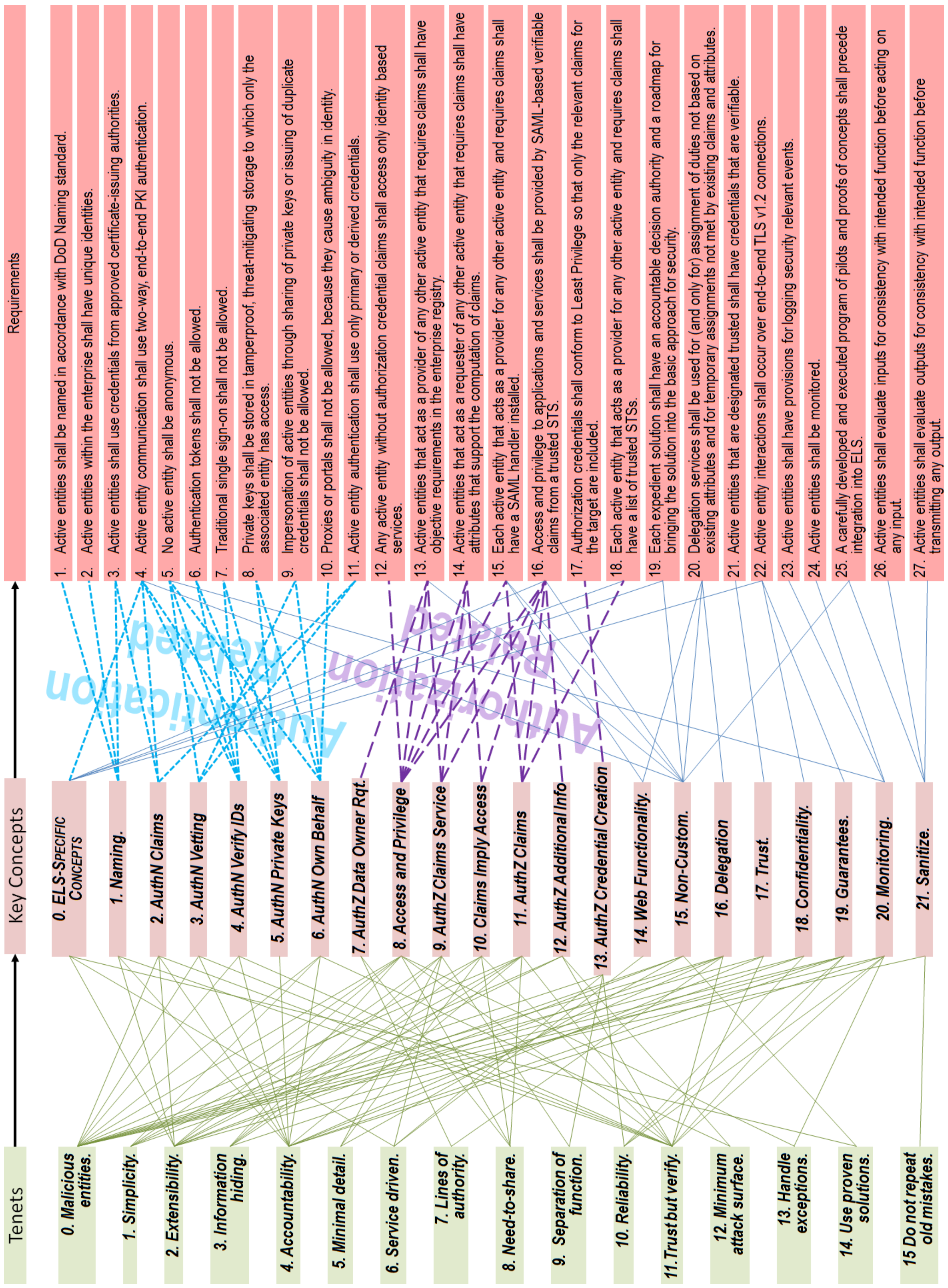
Tenets
0. Malicious entities.
1. Simplicity.
2. Extensibility.
3. Information hiding.
4. Accountability.
5. Minimal detail.
6. Service driven.
7. Lines of authority.
8. Need-to-share.
9. Separation of function.
10. Reliability.
11. Trust but verify.
12. Minimum attack surface.
13. Handle exceptions.
14. Use proven solutions.
15 Do not repeat old mistakes.
Key Concepts
0. ELS-Specific Concepts
1. Naming.
2. AuthN Claims
3. AuthN Vetting
4. AuthN Verify IDs
5. AuthN Private Keys
6. AuthN Own Behalf
7. AuthZ Data Owner Rqt.
8. Access and Privilege
9. AuthZ Claims Service
10. Claims Imply Access
11. AuthZ Claims
12. AuthZ Additional Info
13. AuthZ Credential Creation
14. Web Functionality.
15. Non-Custom.
16. Delegation
17. Trust.
18. Confidentiality.
19. Guarantees.
20. Monitoring.
21. Sanitize.
Authentication Related
Authorization Related
Requirements
1. Active entities shall be named in accordance with DoD Naming standard.
2. Active entities within the enterprise shall have unique identities.
3. Active entities shall use credentials from approved certificate-issuing authorities.
4. Active entity communication shall use two-way, end-to-end PKI authentication.
5. No active entity shall be anonymous.
6. Authentication tokens shall not be allowed.
7. Traditional single sign-on shall not be allowed.
8. Private keys shall be stored in tamperproof, threat-mitigating storage to which only the associated entity has access.
9. Impersonation of active entities through sharing of private keys or issuing of duplicate credentials shall not be allowed.
10. Proxies or portals shall not be allowed, because they cause ambiguity in identity.
11. Active entity authentication shall use only primary or derived credentials.
12. Any active entity without authorization credential claims shall access only identity based services.
13. Active entities that act as a provider of any other active entity that requires claims shall have objective requirements in the enterprise registry.
14. Active entities that act as a requester of any other active entity that requires claims shall have attributes that support the computation of claims.
15. Each active entity that acts as a provider for any other active entity and requires claims shall have a SAML handler installed.
16. Access and privilege to applications and services shall be provided by SAML-based verifiable claims from a trusted STS.
17. Authorization credentials shall conform to Least Privilege so that only the relevant claims for the target are included.
18. Each active entity that acts as a provider for any other active entity and requires claims shall have a list of trusted STSs.
19. Each expedient solution shall have an accountable decision authority and a roadmap for bringing the solution into the basic approach for security.
20. Delegation services shall be used for (and only for) assignment of duties not based on existing attributes and for temporary assignments not met by existing claims and attributes.
21. Active entities that are designated trusted shall have credentials that are verifiable.
22. Active entity interactions shall occur over end-to-end TLS v1.2 connections.
23. Active entities shall have provisions for logging security relevant events.
24. Active entities shall be monitored.
25. A carefully developed and executed program of pilots and proofs of concepts shall precede integration into ELS.
26. Active entities shall evaluate inputs for consistency with intended function before acting on any input.
27. Active entities shall evaluate outputs for consistency with intended function before transmitting any output.

Figure 4. Mappings among Tenets, Concepts, and Requirements

smaller and can be easily addressed by established procedures. In contrast, redesigning the architecture every time a product or component is swapped out requires a large level of effort. This is often the case when there is no forethought in designing a system.

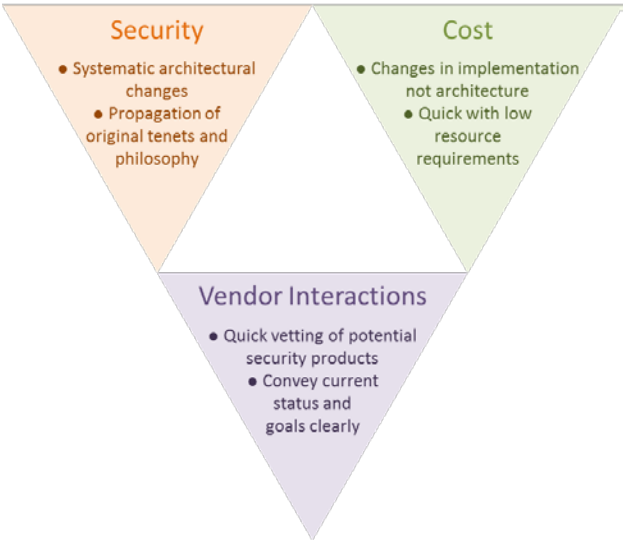
Security
• Systematic architectural changes
• Propagation of original tenets and philosophy
Cost
• Changes in implementation not architecture
• Quick with low resource requirements
Vendor Interactions
• Quick vetting of potential security products
• Convey current status and goals clearly

Figure 5 Benefits of using the model

A third benefit of using this model is dealing with vendors. This basic model provides an architecture for the system into which vendor products can fit. The alternative is to adjust the architecture to fit available vendor product suites. Vendors will often sell a product that is a collection of smaller pieces, and then slowly add more pieces in an effort to integrate all functions under their product suite. This is convenient and efficient in many cases, but it locks the system architecture to a particular vendor and product, which can cause problems when enterprise needs and vendor product functionality diverge. The explicit mapping of the basic security model and choice of widely used protocols and standards maintains a focus on functions instead of products.

9. SUMMARY

The methods described in this paper provide a starting point for building a system that is capable of functioning and adapting to change. This was illustrated with particular details of the ELS model, but the same approach easily be used for other projects that have a high-level goal, an up-front investment, and a slowly changing environment or goal.

The mappings between basic tenets, key concepts, and requirements allow:

1. Improved visibility into the impact of design choices on the high level goals of the system
2. Improved efficiency of the system as it evolves over time
3. Improved interactions with vendors when building the system.

This research is part of a body of work for high assurance enterprise computing using web services. Elements of this work include bi-lateral, end-to-end authentication using PKI credentials for all person and non-person entities, a separate SAML credential for claims-based authorization, full encryption at the transport layer, and a defined federation process. Many of the elements of this work are described in [7-22].

REFERENCES

[1]. X.509 Standards
a. DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011
b. JTF-GNO CTO 06-02, Tasks for Phase I of PKI Implementation, 17 January 2006
c. X.509 Certificate Policy for the United States Department of Defense, Version 9.0, 9 February 2005
d. FPKI-Prof Federal PKI X.509 Certificate and CRL Extensions Profile, Version 6, 12 October 2005
e. RFC Internet X.509 Public Key Infrastructure: Certification Path Building, 2005
f. Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, October 27, 2012
g. PKCS#12 format PKCS #12 v1.0: Personal Information Exchange Syntax Standard, RSA Laboratories, June 1999 http://www.rsa.com/rsalabs/node.asp?id=2138 PKCS 12 Technical Corrigendum 1, RSA laboratories, February 2000

[2]. OCSP Internet Engineering Task Force (IETF) Standards
a. RFC 2560, PKIX OCSP, June 1999
b. RFC 4806, OCSP Extensions to IKEv2, February 2007
c. RFC 6066, TLS Extension Definitions, January 2011
d. RFC 6961, Multiple Certificate Status Extension, June 2013

[3]. CRL Internet Engineering Task Force (IETF) Standards
a. RFC 3280, Internet X.509 Public Key Infrastructure, April 2002
b. RFC 5280, PKIX Certificate and CRL Profile, May 2008

[4]. PKI Standrds
a. DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011
b. JTF-GNO CTO 06-02, Tasks for Phase I of PKI Implementation, 17 January 2006
c. X.509 Certificate Policy for the United States Department of Defense, Version 9.0, 9 February 2005
d. FIPS 140-2 FIPS PUB 140, Security Requirements for Cryptographic Modules, Change Notice, 3 December 2002 (current version).
e. FPKI-Prof Federal PKI X.509 Certificate and CRL Extensions Profile, Version 6, 12 October 2005.
f. RFC Internet X.509 Public Key Infrastructure: Certification Path Building, 2005
g. Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, October 27, 2012
h. PKCS#12 format PKCS #12 v1.0: Personal Information Exchange Syntax Standard, RSA Laboratories, June 1999 http://www.rsa.com/rsalabs/node.asp?id=2138 PKCS 12 Technical Corrigendum 1, RSA laboratories, February 2000

[5]. Organization for the Advancement of Structured Information Standards (OASIS) open set of Standards
a. “WS-Security Specification 1.1” OASIS, November 2006
b. “WS-Trust Specification 1.4.” OASIS, February 2009

c. "WS-ReliableMessaging Specification 1.1," OASIS, November 2004

d. "WS-SecureConversation Specification 1.4," OASIS, February 2009

e. "WS-BaseNotification," 1.3 OASIS, October 2006

f. "WS-BrokeredNotification," 1.3 OASIS, October 2006

g. N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, March 2008

h. P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.

i. S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 2005

[6]. TLS family Internet Engineering Task Force (IETF) Standards

In draft for reference only:

a. TLS Renegotiation Support Extension to HTTP/2, 2015-03-24

b. Terminology related to TLS and DTLS, 2015-03-26

c. X.509v3 TLS Feature Extension, 2015-04-06

d. TLS over HTTP, 2015-03-09

e. A TLS ClientHello padding extension, 2015-02-17

f. Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier, 2015-03-09

g. Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension, 2015-04-16

h. The Transport Layer Security (TLS) Protocol Version 1.3, 2015-03-09

Standards:

i. RFC 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, 2000-05

j. RFC 3749 Transport Layer Security Protocol Compression Methods, 2004-05

k. RFC 4279 Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), 2005-12

l. RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08

m. RFC 5289 TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008-08

n. RFC 5929 Channel Bindings for TLS, 2010-07

o. RFC 6358 Additional Master Secret Inputs for TLS, 2012-01

p. RFC 7251 AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, 2014-06

q. RFC 7301 Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, 2014-07

r. RFC 7457 Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015-02

[7]. William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, Electronic Digest of the 2008 System and Software Technology Conference, "A Persona-Based Framework for Flexible Delegation and Least Privilege," Las Vegas, Nevada, May 2008.

[8]. William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, The 1st International Multi-Conference on Engineering and Technological Innovation: IMET2008, "Cross-Domain Solutions in an Era of Information Sharing," Volume I, pp. 313–318, Orlando, Florida, June 2008.

[9]. Coimbatore Chandersekaran and William R. Simpson, World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, "The Case for Bi-lateral End-to-End Strong Authentication," 4 pp., London, England, December 2008.

[10]. William R. Simpson and Coimbatore Chandersekaran, The 2nd International Multi-Conf.on Engineering and Technological Innovation: IMETI2009, Volume I, pp. 300–305, "Information Sharing and Federation," Orlando, Florida, July 2009.

[11]. Coimbatore Chandersekaran and William R. Simpson, The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, "A SAML Framework for Delegation, Attribution and Least Privilege," pp. 303–308, Orlando, Florida, July 2010.

[12]. William R. Simpson and Coimbatore Chandersekaran, The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, "Use Case Based Access Control," pp. 297–302, Orlando, Florida, July 2010.

[13]. Coimbatore Chandersekaran and William R. Simpson, The First International Conference on Computer Science and Information Technology (CCSIT-2011), "A Model for Delegation Based on Authentication and Authorization," Springer Verlag Berlin-Heildleberg, Lecture Notes in Computer Science, 20 pp.

[14]. William R. Simpson and Coimbatore Chandersekaran, The 16th International Command and Control Research and Technology Symposium: CCT2011, Volume II, pp. 84–89, "An Agent Based Monitoring System for Web Services," Orlando, Florida, April 2011.

[15]. William R. Simpson and Coimbatore Chandersekaran, International Journal of Computer Technology and Application (IJCTA), "An Agent-Based Web-Services Monitoring System," Vol. 2, No. 9, September 2011, pp. 675–685.

[16]. William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2011, Volume I, "High Assurance Challenges for Cloud Computing," pp. 61–66, San Francisco, October 2011.

[17]. Coimbatore Chandersekaran and William R. Simpson, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Claims-Based Enterprise-Wide Access Control," pp. 524–529, London, July 2012.

[18]. William R. Simpson and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Assured Content Delivery in the Enterprise," pp. 555–560, London, July 2012.

[19]. William R. Simpson and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2012, Volume 1, "Enterprise High Assurance Scale-up," pp. 54–59, San Francisco, October 2012.

[20]. Coimbatore Chandersekaran and William R. Simpson, International Journal of Scientific Computing, Vol. 6, No. 2, "A Uniform Claims-Based Access Control for the Enterprise," December 2012, ISSN: 0973-578X, pp. 1–23.

[21]. William R. Simpson, Kevin Foltz and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2014, Volume 1, “Distributed versus Centralized Protection Schema for the Enterprise,” pp. 173-184, Berkeley, CA. October 2014,

[22]. William R. Simpson and Kevin Foltz, Proceedings of the World Congress on Engineering 2015 Vol I, WCE 2015, July 1-3, 2015, London, U.K., “Wide Area Network Acceleration in a High Assurance Enterprise,” pp. 502–507.