Enterprise Level Security – Basic Security Model[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 secure information sharing system is challenging. Maintaining, updating, and modifying such a system based on changing enterprise needs and advancing technology is even more challenging. Decisions and informal rules that were made and enacted in the initial build are often lost, forgotten, or ignored when changes are needed. When the original system designers have moved on, the system is entrusted to an administrator who understands how the system works but not why it was designed to work that way. Without this higher-level understanding, the secure system devolves into a collection of loosely integrated partial solutions with security vulnerabilities at the seams and edges. This work presents a method of documenting the design logic of a secure enterprise information system, from basic principles to implementable requirements. Important design decisions are captured, along with the logic supporting them. Before changes to the system are made, an assessment is made against the core design decisions to ensure the original security goals are maintained. This provides clarity to the system owner and administrators to help guide future changes, and it provides a way to convey security goals to product vendors in a structured and logical way, which can help to reduce the back-and-forth arguing over whether a product meets security requirements. The Enterprise Level Security (ELS) architecture is used as an example of the application of this method to a real-world security system.

Keywords: Enterprise, Security Concepts, IT Security, Integrity, Security Tenets, Security Requirements.

1. INTRODUCTION

Designing a secure information-sharing system is a challenge. Vendors attempt to sell product suites that do everything you need, but they usually miss some things and include many other, completely unnecessary, functions. Behind their polished interfaces are pieced-together solutions with bloated code and pieced-together security. The alternative is to put such a system together yourself from different components. This allows better visibility into how the system works and more control over protocols and formats, but it can be difficult to find products that work well together and support a particular security model. As a result, compromises are made based on what is currently most important and what technology is currently available.

Building a secure system is just the start. Maintaining security in the face of technology changes and new enterprise goals is the more difficult and important challenge. Changes to system functionality can often be implemented by adding components or logic that perform such functionality and tie into the existing system. However, security is often reduced by such changes unless they are done very carefully. Proper system design can prevent systemic security problems and reduce the problems to the implementation and configuration realms. Unlike architecture problems, which are difficult and time-consuming to fix, these implementation and configuration problems can be addressed more quickly. Thus, a properly designed system is more resilient to attack when available exploits are limited to those that can be quickly and efficiently fixed through established patching and configuration methods.

This paper looks at a way to properly design and maintain security using the example of the ELS system, which is currently under development as the Air Force security solution. The next section provides a brief description of ELS. The following sections describe the method, its components, and its application to the design and maintenance of the ELS system.

2. BASIC SECURITY MODEL

The goal of ELS is to provide access to information 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 allow intelligent sharing of information among the entities in the enterprise and among 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 a web application or web service. 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. Since both credentials are required for access to services, this separation makes compromise of a single credential insufficient for immediate access.

In contrast to active entities, passive entities are critical to operations but do not themselves initiate or respond to web service or web application requests. These include routers, switches, wireless access points, and network layer scanners. Passive entities do not initiate activities and cannot assume the role of requestor or provider.

Active entities within the enterprise are registered within the enterprise and have unique identities. The identities are defined by the issuance of an acceptable identity credential, and their private keys are stored in tamper proof, threat mitigating storage to which only the associated entity has access. Thus possession of the private key is an assurance of identity. 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. Authorization in the operational environment is implemented by a verifiable access and privilege claims-based process.

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 are provided by the data owner. A trusted third party examines the attributes of an entity and determines whether the requirement is satisfied. Another trusted third party provides that claim in a credential that can be validated and verified. 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. It would take many documents to describe the full ELS implementation, but the focus of this work is on tracing requirements to concepts and tenets. The following sections describe the core tenets, key concepts, and requirements for ELS. These provide the starting point and foundation for all the detailed requirements and implementation decisions within ELS and provide guidance for future decisions.

3. SECURITY DESIGN AND MAINTENANCE

For security design and maintenance, a set of core tenets is the starting point. These describe the desired highest-level security properties of the system. They do not indicate how to achieve these goals but rather provide guidance that can be used to choose the methods to achieve security. From these tenets, key concepts describing the system to be built are derived. The concepts describe some of the high-level rules of the system and how it works. From the concepts a set of requirements are developed. The idea is that an enterprise can use these requirements as the foundation for building a secure system.

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 of security by tying requirements to the overall design goals of the system. This facilitates redesigning 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 security goals.

The sections below describe a systematic method of designing and analyzing the security of a system. To make this more tangible, details are provided for the development of the ELS system. Using this method, every design decision is connected to a basic assumption or tenet of the security architecture. This provides visibility into what parts of the system contribute to which security goals and provides a way to change the system as technology changes or security goals change while preserving the desired security goals.

4. CORE TENETS

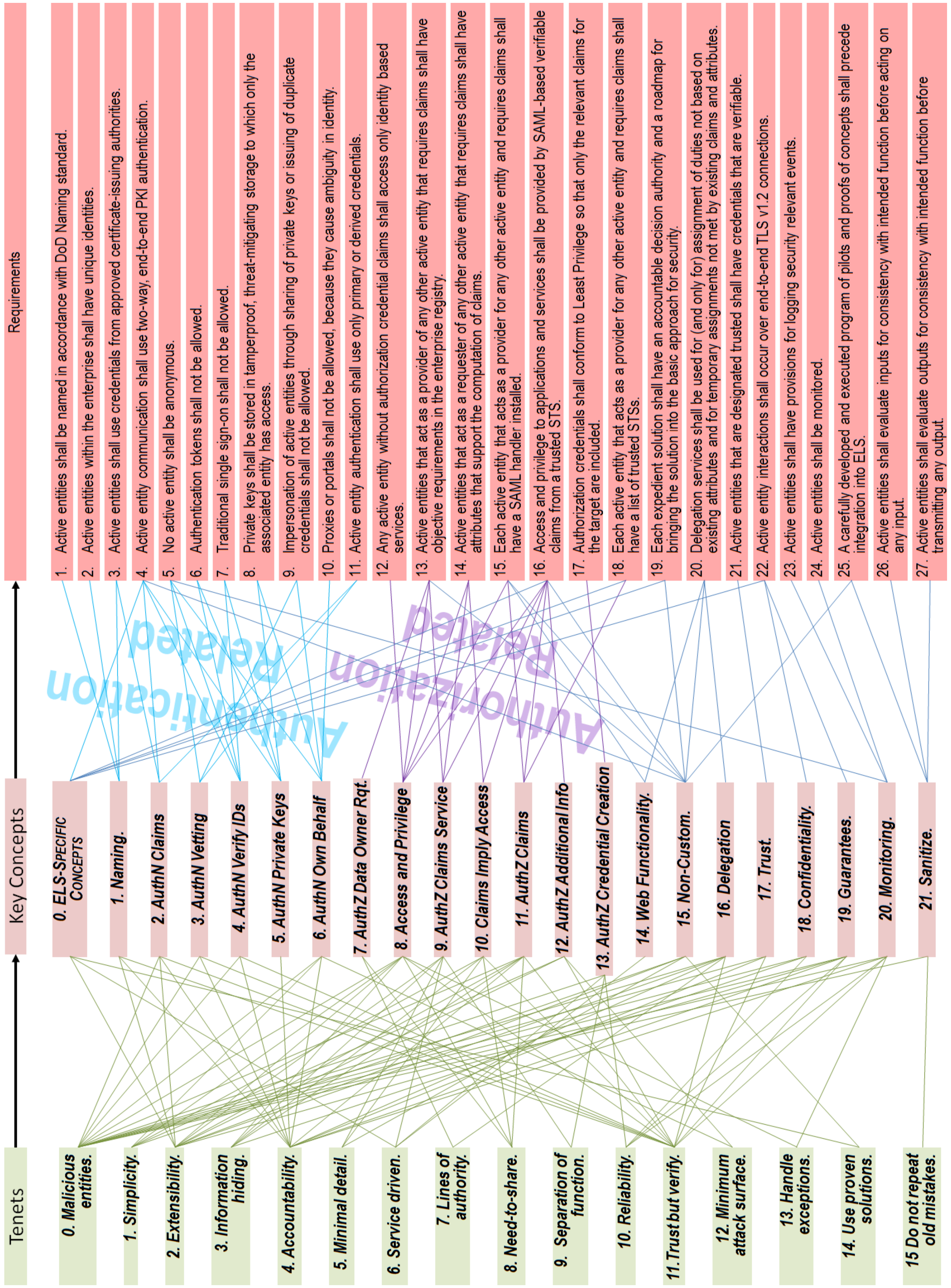
Each component of every enterprise solution should be tested against a set of fundamental evaluation criteria or tenets. These tenets are the core philosophical drivers of all architectural decisions. ELS tenets are as follows:

0. 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 what active entity in the enterprise performed each operation.
5. Minimal detail. This means adding detail to the solution to only the required level. This preserves flexibility of implementation at lower levels.
6. Service-driven rather than a product-driven solution.
7. Lines of authority should be preserved and information assurance decisions should be made by policy and/or agreement at the appropriate level.
8. Need-to-share as overriding need-to-know.
9. Separation of function. Sometimes referred to as atomicity, this allows for fewer interfaces, easier updates, maintenance of least privilege, reduced and easier identified vulnerabilities, and improved forensics.
10. Reliability. Security needs to work even if adversaries know how the process works.
11. Trust but verify (and validate). Trust should be given out sparingly and even then, trusted outputs need checking.
12. Minimum attack surface. The fewer the interfaces and the less the functionality in the interfaces, the smaller the exposure to threats.
13. Handle exceptions and errors. Exception handling involves logging, alerting the Enterprise Support Desk (ESD), and notifying the user.
14. Use proven solutions. Select products, technologies, techniques, and algorithms that have sufficient evidence of maturity in their intended use.
15. Do not repeat old mistakes. This means using a flaw remediation system, patching and repairing, and not fielding a software solution with known vulnerabilities and exploits.

5. KEY CONCEPTS

The key concepts for ELS are based on the tenets but address specific architectural decisions which relate to the requirements. The concepts form a bridge between the high-level tenets and the technical requirements. The numbers of the tenets that relate to each concept are shown in braces.

0. ELS-specific concepts {2, 6, 14}. These are choices based on current technology and are subject to change and expansion as technology changes and the ELS model is developed further. For simplicity they are considered as a single concept.
   a. PKI credentials are used for active entity credentials. [1, 2, 3, 4]
   b. Security Assertion Markup Language (SAML) with claims is used for authorization credentials. [5]
   c. TLS v1.2 is used for end-to-end confidentiality, integrity, and authentication. [6]
   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, 4, 11}
2. Authentication is implemented by a verifiable identity claims-based process. {0, 2, 4, 11}
3. Identity claims are tied to a strong vetting process to establish identity. {0, 4, 11}
4. Active entities verify each other's identity. {0, 4, 11}
5. The verification of identity is by proof of ownership of the private key associated with an identity claim. {4}
6. Active entities act on their own behalf. {0, 1, 12}
7. The claims objective requirement is provided by the data owner. {7, 8}
8. Service providers use identity and authorization credential claims to determine access and privilege. {0, 1, 2, 3, 8, 11, 13}
9. A trusted entity examines the attributes of an entity and determines whether the claims objective requirement is satisfied. {2, 3, 5, 6, 9}
10. A claim in an authorization credential is a statement that an access requirement has been satisfied. {1, 3, 5, 8, 11}
11. Authorization is implemented by a verifiable identity, access, and privilege claims-based process. {0, 2, 3, 4, 8, 11}
12. The data owner may request as part of his requirement definition, additional information about the requesting entity. {1, 2, 11, 12}
13. Authorization credentials are created by a trusted entity for a specific requester, a specific target resource, and a specific level of access. {0, 6, 9, 10}
14. Functionality is to be provided through web services. {6}
15. It is undesirable to work a point solution or custom approach. {1, 2, 5, 14}
16. A formalized delegation policy both within and outside of the enterprise is a requirement. {0, 2, 4, 7, 11}
17. Being able to be verified and validated is a requirement for trusted entities. {0, 4, 11}
18. All active entity interactions require confidentiality of data/content exchanged. {0, 3, 10}
19. Guarantee integrity, authenticity, timeliness, and pedigree. {0, 2, 4, 10, 11}
20. Monitoring is a precursor to cyber security. {0, 4, 10, 11, 13}
21. Eliminate or mitigate malware. {0, 15}



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 1. Mappings among Tenets, Concepts, and Requirements

6. TECHNICAL REQUIREMENTS

The basic security model technical requirements for ELS are based on the key concepts, as listed in parentheses (), and are directly traceable to the core tenets.

1 Active entities shall be named in accordance with DoD Naming standard. (1)
2 Active entities within the enterprise shall have unique identities. (1)
3 Active entities shall use credentials from approved certificate-issuing authorities. (1, 2)
4 Active entity communication shall use two-way, end-to-end PKI authentication. (0, 2, 4, 5, 15)
5 No active entity shall be anonymous. (3, 4, 6, 20)
6 Authentication tokens shall not be allowed. (4, 5)
7 Traditional single sign-on shall not be allowed. (4, 5)
8 Private keys shall be stored in tamperproof, threat-mitigating storage to which only the associated entity has access. (5, 6)
9 Impersonation of active entities through sharing of private keys or issuing of duplicate credentials shall not be allowed. (3, 6)
10 Proxies or portals shall not be allowed, because they cause ambiguity in identity. (6)
11 Active entity authentication shall use only primary or derived credentials. (2, 3)
12 Any active entity without authorization credential claims shall access only identity-based services. (8)
13 Active entities that act as a provider of any other active entity that requires claims shall have objective requirements in the enterprise registry. (7, 8, 9, 15)
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. (8, 9, 10)
15 Each active entity that acts as a provider for any other active entity and requires claims shall have a SAML handler installed. (8, 11, 14, 15)
16 Access and privilege to applications and services shall be provided by SAML-based verifiable claims from a trusted STS. (0, 8, 9, 10, 11, 12, 15)
17 Authorization credentials shall conform to Least Privilege so that only the relevant claims for the target are included. (13)
18 Each active entity that acts as a provider for any other active entity and requires claims shall have a list of trusted STSs. (0, 11)
19 Each expedient solution shall have an accountable decision authority and a roadmap for bringing the solution into the basic approach for security. (0, 15)
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. (14, 15, 16)
21 Active entities that are designated trusted shall have credentials that are verifiable. (17)
22 All active entity interactions shall occur over end-to-end TLS v1.2 connections. (0, 18, 19)
23 Active entities shall have provisions for logging security relevant events. (20)
24 Active entities shall be monitored. (20)
25 A carefully developed and executed program of pilots and proofs of concepts shall precede integration into ELS. (15, 21)
26 Active entities shall evaluate inputs for consistency with intended function before acting on any input. (21)
27 Active entities shall evaluate outputs for consistency with intended function before transmitting any output. (21)

7. MAPPINGS BETWEEN TENETS, CONCEPTS AND REQUIREMENTS

Figure 1 shows the mappings between tenets and concepts and between concepts and requirements. The overall mapping is complex, but certain features stand out. First is the clustering of concepts and requirements related to authentication, as indicated on the links. Second is the similar clustering of authorization-related concepts and requirements, as indicated on the links. Many of the remaining concepts and requirements are in nearly one-to-one or one-to-many correspondence, with a few cross-cutting links from concepts 0, 14, 15, and 20. Between tenets and concepts, the links appear denser and less organized, with some tenets applying across many concepts while others are limited to a small number.

Figure 1 can be used to trace requirements back to concepts and tenets, which can help in making and justifying implementation decisions. For example, suppose the enterprise is considering 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 TLS-encrypted traffic. This is a common practice, but it violates the following requirements because:

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

Tracing these requirements back to related concepts, we see that the following concepts are identified (with duplicates as indicated): 0, 1, 2, 3, 4, 5 (2x), 6 (3x), 8 (2x), 9, 10, 15, 18, and 19. The most often referenced is Concept 6, "Active entities act on their own behalf," of which 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. The remaining concepts round out the set that are relevant.

Extending this process, we can link back from these concepts to the related tenets to identify the following (again with duplicates indicated): 0 (10x), 1 (7x), 2 (8x), 3 (5x), 4 (10x), 5 (3x), 6 (2x), 8 (3x), 9, 10 (2x), 11 (8x), 12 (5x), 13 (2x), 14 (2x). The most referenced are Tenet 0, "Malicious entities are present," Tenet 4, "Accountability," Tenet 2, "Extensibility," and Tenet 11, "Trust but verify." By allowing proxies we provide more points of exposure to internal 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 to the concepts and tenets of ELS. 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. Although a simple count of requirements, concepts, and tenets is not a reliable way to compare, in this case it is true that adding proxies is a more serious security violation than not scanning outputs for consistency. In general an assessment is needed to determine how bad a violation is.

Another example of using the mapping is making changes based on changes in technology or enterprise goals. For example, suppose delegation is strictly forbidden, and everyone must be explicitly assigned duties instead of allowing people to assign them dynamically through delegation. This might take place if delegation were abused. In this case, Requirement 20 is revoked, which affects Concepts 14, 15, and 16 and Tenets 0, 1, 2 (2x), 4, 5, 6, 7, 11, 14. The most affected tenet is Tenet 2, "Extensibility," which is listed twice. Delegation offers a convenient way to extend ELS to both undefined internal role assignments and external credentialed entities. Without delegation, this extensibility suffers and ELS is restricted to well-defined positions and privileges within the enterprise. Someone familiar with architectural goals and an understanding of delegation would not necessarily learn much from this exercise, but in many cases, those who understand delegation do not understand the high-level goals, so this provides a way to put their issues into the larger context of the enterprise security picture.

Working in the other direction involves changing a tenet and finding which requirements are affected. For example, suppose Tenet 8 is removed, since need-to-share causes problems versus need-to-know. In this case, Concepts 7, 8, 10, and 11 are affected, and they map to Requirements 12, 13 (2x), 14 (2x), 15 (2x), 16 (3x), and 18. The most affected is Requirement 16, describing the use of SAML from a trusted STS for access and privilege. The STS-based SAML is fundamental to the need-to-share approach, since it provides access to any active entity that should be allowed to access services and data. So, changing this tenet would require rethinking the SAML authorization approach. Other requirements strongly affected include Requirement 13, using requirements in the registry to define claims, Requirement 14, requiring appropriate attributes to produce claims, and Requirement 15, use of the SAML handler to process incoming authorization credentials. In this case, all concepts and requirements affected lie within the AUTHZ section, suggesting that the entire authorization approach would need to be changed if Tenet 8 is removed.

The exercises above provide some examples of using the mapping. Another is to educate people about how the system is supposed to work. Product vendors often try to sell products and force-fit them to the existing enterprise security solution. The tenets convey what is important to maintaining system security, and the mapping shows how requirements are derived from these important tenets. This can eliminate some of the initial false starts in which the vendor solution meets most of the technical requirements but its implementation misses some important tenets or concepts.

8. EXTENSIONS

The methods described in this paper provide a starting point for understanding enterprise security. A simple extension would be to show the strength of the connections in the mapping, not just whether a connection is present. This would improve the ability to map requirements to the most affected tenets and vice versa. This could also be extended to include negative weight links, which could make security trade-offs visible, showing how a requirement might support one concept but hinder another.

Other mappings could be used as well, such as a direct tenet to requirement mapping, a multi-layer approach, or a cross-layer approach, in which links may reach across or within layers. Such mappings would likely be more difficult to analyze and maintain, but the added complexity may capture important properties that the simple three-layer approach presented cannot.

Of course, the size of the model is also an important extension. The model presented is comprehensible and suitable for analysis by a human, but for automated analysis it might be desirable to include many more tenets, concepts, and requirements. Effective methods for capturing these and linking them correctly would become more important as the data set size grows.

9.SUMMARY

We have reviewed the basic tenets and key concepts as they relate to the establishment of requirements for a basic security model for enterprise level security. The choices are clearly defined in the initial ELS Key concepts. Mappings allow:

1. Evaluating the impact of choices.
2. Evaluating the impact of relaxation of requirements such as single-sign-on or use of proxies.
3. Traceability of extended requirements for functional areas such as load balancing, database operations, and others.

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.