Minimal Instantiation of Enterprise Level Security

William R. Simpson *Member, IAENG* and Kevin E. Foltz

***Abstract* — Enterprise Level Security (ELS) is a web-based security architecture designed using standard commercially available technology to build a cohesive set of policies and rules for an enterprise information system. This paper discusses the key components of the Enterprise Attribute Ecosystem (EAE), which is the access and privilege management infrastructure (or back-office) for ELS. The EAE collects attributes about all entities from trusted sources, maintains access and privilege rules for all enterprise resources, and provides access and privilege claims to enterprise users for enterprise resources. The EAE provides validated claims for appropriate access and privileges on a per-request basis. The techniques employed have been shown to be resilient, secure, extensible, and scalable. This paper describes the minimal set of EAE capabilities needed to stand up an initial ELS capability.**

***Index Terms*: Access privilege, authentication, authorization, digital signatures, identity claims, public key infrastructure.**

I. Introduction

Enterprise Level Security (ELS) is a security architecture for web-based information systems. Its development is guided by basic tenets that stress high security. Two important features of ELS are end-to-end transport layer security (TLS) connections with mutual authentication and a claims-based system for access and privilege [1]. The ELS design addresses five security principles that are derived from the basic tenets:

- Know the Players – enforce bi-lateral end-to-end authentication using Public Key Infrastructure (PKI) certificates issued by an enterprise approved Certificate Authority (CA). [2];
- Maintain Confidentiality – use unbroken end-to-end encryption (no in-transit decryption/payload inspection) using TLS [3];
- Separate Access and Privilege from Identity – use a Security Assertion Markup Language (SAML) authorization credential issued by the Security Token Server (STS) in addition to the PKI authentication credential [4];
- Maintain Integrity – validate the integrity of all received content through end-to-end TLS integrity measures [5];
- Require Explicit Accountability – log, aggregate, and centrally monitor activity of all endpoints [6, 7].

Manuscript received 1 April 2019; revised 15 June 2019. This work was supported in part by the U.S. Secretary of the Air Force and the Institute for Defense Analyses (IDA). The publication of this paper does not indicate endorsement by any organization in the Department of Defense or IDA.
Kevin E. Foltz is with the Institute for Defense Analyses. (e-mail: kfoltz@ida.org).
William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA, and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org).

In this paper, we primarily focus on the infrastructure to support the generation of authorization and privilege claims. It is assumed that all communication is conducted through end-to-end TLS with mutual authentication using PKI credentials. This ensures that both sides of the communication know who they are communicating with, the communication is confidential, and the content maintains end-to-end integrity. The authorization claims process builds on this secure connection to provide access and privilege information for requesters to services [1].

This paper covers the core EAE functions required to generate authorization claims in an ELS system. Such an instantiation will provide the following:

a. a core capability that meets the ELS security model,
b. a claims-based access and privilege system that is mostly automated and is dynamic, resilient, secure, and extensible, and
c. an ecosystem that can be enhanced for many of the additional capabilities that are part of the overall ELS architecture.

More technical details of ELS, which extend beyond the core instantiation, are covered in [8].

II. Needed Capabilities for a Minimal Instantiation

In order to provide a minimal instantiation of the EAE, the following functionalities are required:

1. an attribute store with sufficient user information for data owners to define access and privilege rules,
2. a registration service for enterprise resources and their access and privilege rules,
3. a service to generate claims and store them when a match between the information available for an individual in the attribute store matches rules for access and privilege, and
4. a set of user convenience services that allow for corrections and adjustments and make the authorization requirements user-friendly.

At the initial establishment of the EAE, all servers and users are provisioned with PKI certificates. The private keys are stored in Hardware Storage Modules (HSMs). All servers are configured to require TLS mutual authentication and strict rules about cipher suites and protocol versions. If the handshake does not match, no communication takes place [9, 10]. Within the EAE, all entities and communication paths are known, so the interfaces, protocols, and authorizations can be strictly controlled.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

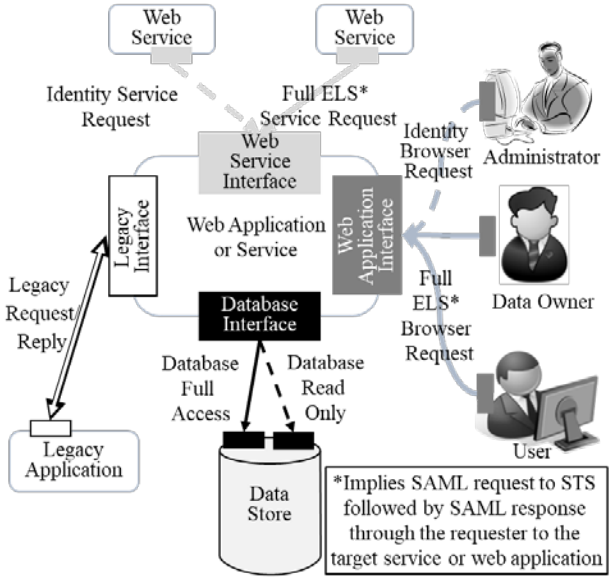

Figure 1 Enterprise Legend and Nomenclature

Figure 1 provides a general legend for the objects in the figures that follow. There are three classes of human entities. Users send browser requests to web applications to request data or services. Administrators conduct similar requests but also perform configuration and receive privileged access. Data owners host web applications and services and set the rules for access for both users and administrators.

There are two types of non-human entities. Web applications and web services provide services and data to requesters according to the rules set by the data owner. Data stores maintain data pertaining to attributes and access rules.

There are four types of interfaces, each with one or more communication types. Legacy interfaces use legacy requests and replies and are secured to the extent possible. Database interfaces are used to access data stores, and they may be full access or read only depending on the sensitivity of the data and the requesting entity. Browser requests typically use SAML authorization, but in cases where security is strict and requesters are known and limited, the identity may be used instead. Web service interfaces are similar to web application interfaces, but they use web service clients instead of browsers.

III. Creating an Attribute Store

The Enterprise Attribute Store (EAS) consists of a collection of current information about registered enterprise personnel and entities as shown in Figure 2. It is a logical construct and may be a single store or a collection of stores. It is independent of the other stores in the EAE and has its own set of access controls.

Many Authoritative Content Stores (ACSs) may be used to populate the EAS. These ACSs may have different access methods and data formats, and each has its own associated exposure service that communicates with the ACS and extracts data into a standard format. These data are gathered and placed in an interim store awaiting a periodic update from the EAS Data Import Aggregation & Mediation service.

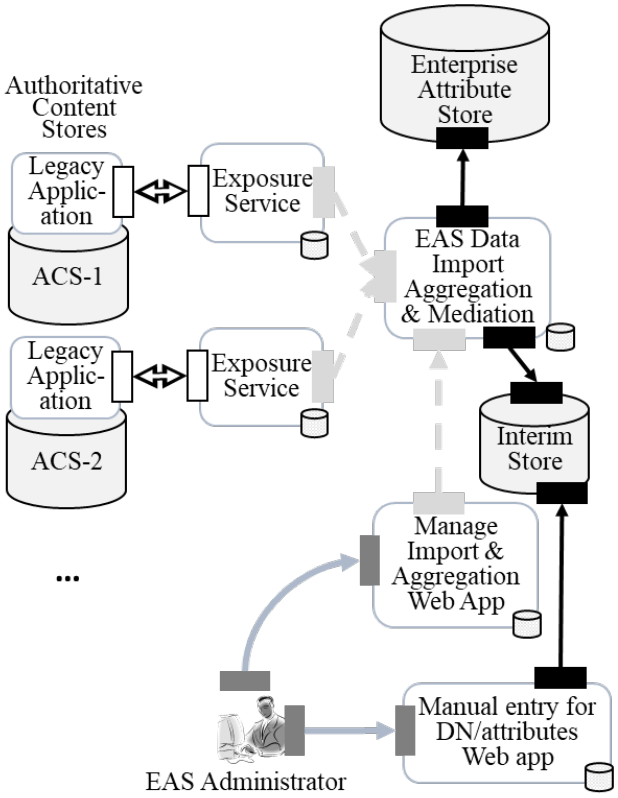

Figure 2 Creating an Attribute Store

This service sanitizes input data, tags Distinguished Names (DNs) for changes, and uploads the data to the EAS. This upload interface is the only write interface to the EAS. The tagging is for use by the claims engine, so it can update the claims for any DN that has changed values. The ACSs may be legacy systems, so the exposure services and sanitization serve to keep the aggregated data in the EAS consistent, clean, and properly correlated. Each service and requester has a small store indicated for monitoring files as required by the security model.

IV. Registering a Service

The data owner is responsible for registering their enterprise applications and services through an auto registration application (as shown in Figure 3). This application provides the EAS attribute list, and the data owner defines access control rules (ACRs) as logical combinations of these attributes and other dynamic information, such as time of day. The service and/or application details may be provided as documentation to an administrator for entry into the system.

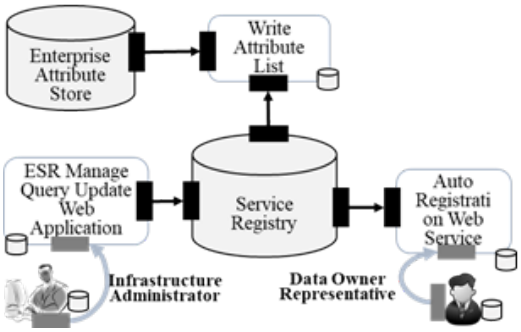

Figure 3 Registering a Service

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The write attribute list service is only needed when frequent changes to the schema of the enterprise attribute store occur. They are a convenience for the data owner to register and when schema changes are not frequent they may be entered manually into the service registry.

V. Computing Claims

With attributes and access rules based on these attributes, there is now enough information in the system to compute access claims [11–13]. The process is shown in Figure 4. The claims engine is triggered periodically or on demand by the Data Import, Aggregation, and Mediation Service. For each DN that has a change in attributes, the claims are recomputed by reading the ACR for each service and gathering the data to fulfill the ACR from among the stores in the EAE. The new or modified claims are written into the claims repository.

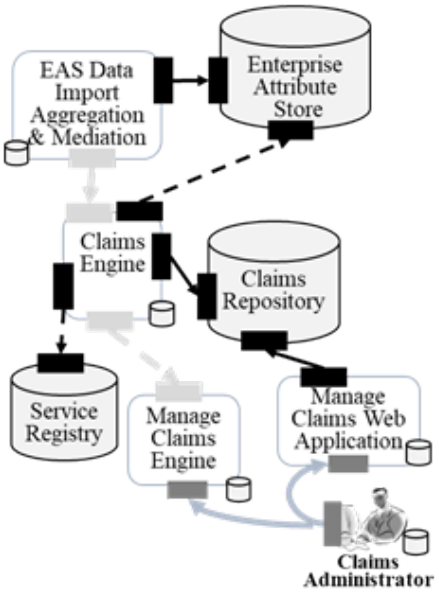

Figure 4 Computing Claims

The claims repository is a precomputed set of access rights for all combinations of requesters and providers. After claims have been computed, the operational system relies on the claims repository instead of the attribute store. This provides some benefits:

- the EAS has fewer access points and, hence, fewer points of vulnerability,
- a copy of the claims repository provides all the needed information to determine access, and it can be used for remote locations with limited connectivity back to the EAE, and
- claims need not be repeatedly computed from scratch, because they are computed as a background process when attributes or access rules are changed.

VI. User Convenience Services

A user may need to know what claims the EAE has in its data bases. For privacy reasons, an individual user is only allowed to see his own information. Because all users should be able to access such a service regardless of their attributes, access and privilege is identity based, and the service returns a summary of claims for the individual, as shown in Figure 5. Such applications and services with identity-based access control and simple request/response data flows do not establish application layer sessions. This reduces the attack surface by eliminating session cookies and their associated vulnerabilities.

The claims query service may be used to advise superiors or data owners when sufficient claims are not granted to complete work assignments. The data owner may consider revisions. Additionally, the claims query service provides a link to each service that the user has claims to access.

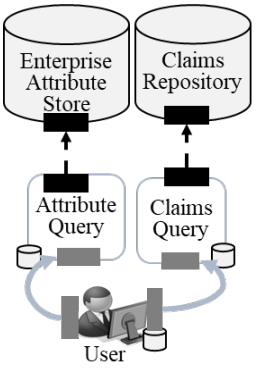

Figure 5 User Convenience Services

Similarly, the attribute query returns a summary of attributes with contact information for correcting discrepancies. The user may initiate a request for such a correction, but the owners of the ACSs must actually make the corrections. Such corrections propagate first to the EAS and then to the claims repository by the normal update process. Claims can be viewed to troubleshoot application access difficulties, and problems are best discussed with a supervisor for modification of the ACRs. Such access problems will decrease as the system gets refined over time.

VII. The Enterprise Attribute Ecosystem

Figure 6 pulls it all together and shows the back office infrastructure for ELS. The figure includes those applications and services described above as well as administrative and other function.

There are multiple ways that the user may invoke a web application session. In all cases, the STS will go to the provide claims web service for claims that the user can assert for the application target. The STS then packages these claims in a SAML token. The invocation methods are as follows:

1. the user sends a request to the STS, indicating the target application, and the STS provides a token and a redirect to the application,
2. the user clicks a link obtained in the claims query service, which initiates a request to the STS as above,
3. the user sends a request to the web application, which redirects the user to the STS as above.

SAML handlers need to be integrated into each of the applications. These handlers exist in both .NET and JAVA applications and may be made available upon request to the authors. The handlers solve many of the XML vulnerability issues and are the subject of separate documentation [18].

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

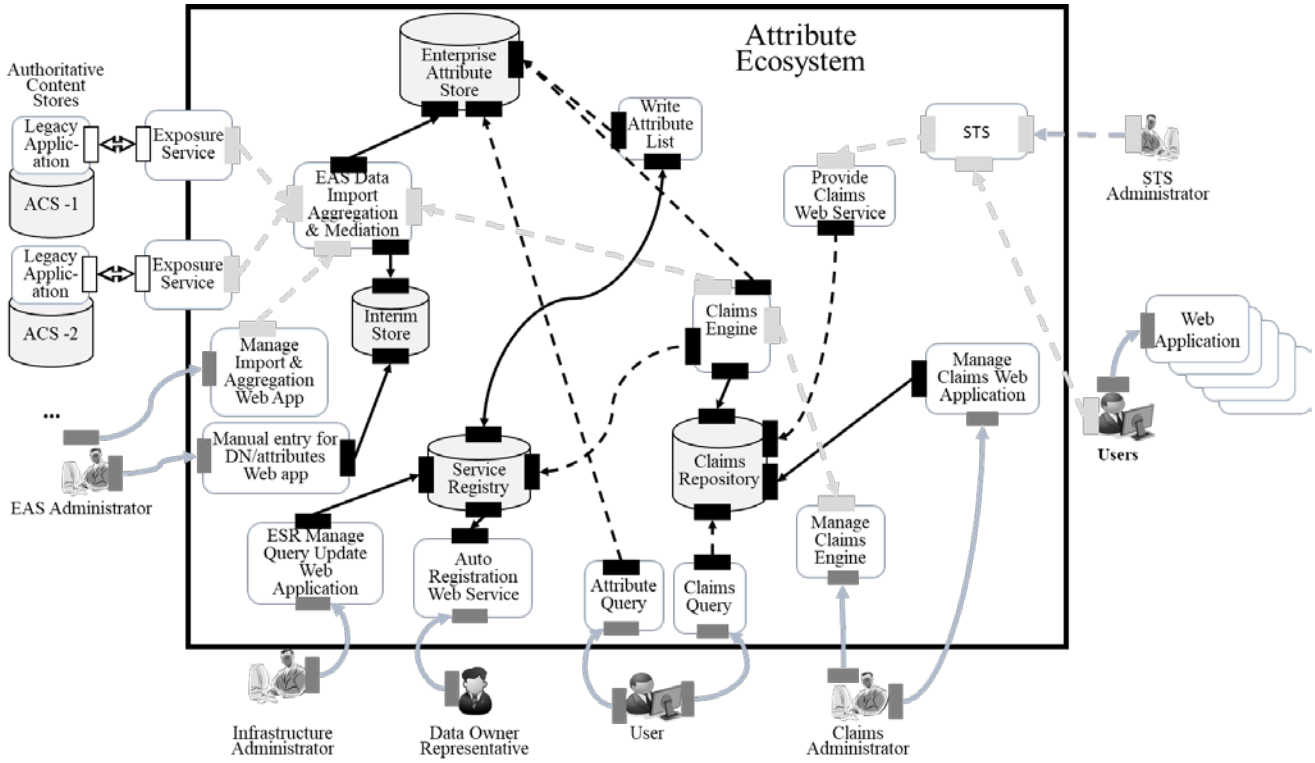

Figure 6 Minimal Instantiation Attribute Ecosystem

The third method is the most complicated because the application must detect that no SAML is provided after authentication and send back the redirect to the STS.

The minimal instantiation of the EAE consists of the following fourteen information services:

1. *Authoritative Content Exposure Service(s).* One is required for each ACS, and they depend on the legacy interfaces available. At such time an authoritative content store becomes ELS compatible, and satisfies the ELS integrity requirements, the exposure service may be dropped and the data may be imported directly by the *EAS Data Import, Aggregation, and Mediation Service.*

2. *EAS Data Import, Aggregation, and Mediation Service.* There are three tasks accomplished by this service:

   a. Mediation of different import formats (not needed for ELS compatible authoritative content stores.

   b. Common sense checks on data (e.g., a receptionist does not normally get promoted to CIO). This check is not needed when the authoritative content store is ELS compatible and meets the ELS integrity requirements.

   c. Periodic updates to the EAS.

3. *Manage Import, Aggregation, and Mediation Service.* This service configures the service above (may be integrated into the EAS Data Import, Aggregation, and Mediation Service).

4. *Manual Entry Web Application for Attributes.* This service corrects shortcomings in the automated services (should be used less and less over time as configurations are improved). **This web application may be integrated with the Auto Registration Service.**

5. *Enterprise Service Registry Management Web Application.* This service allows configuration and management of the service registry system. **This web application may be integrated with the Auto Registration Service.**

6. *Auto Registration Service.* This service permits the data owner to input the ACRs associated with a service. This service will ideally present a user friendly interface for building logical requirements for individual entity attributes.

7. *Claims Engine.* This service gathers the data for each individual to make a comparison to the ACR. When a match is found, claims are generated. The new or modified claims are written into the claims repository.

8. *Manage Claims Engine Service.* This service manages the rule sets and configuration of the claims engine. **This service may be built into the Claims Engine.**

9. *Manage Claims Web Application.* This service corrects shortcomings in the automated services (should be used less and less over time as configurations are improved).

10. *Provide Claims Web Service.* This service extracts the claims appropriate to the requester and provide for use by the STS. This service has read only interface with the claims repository.

11. *Attribute List.* This service provides a menu of alternatives to the auto registration service to assist the data owners in formulating ACRs. This service has a read only interface to the EAS. This service is

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

needed as a check against access and privilege requirements, to assure that values are in the attribute store. **This service may be included in the Enterprise Service Registry Management Web Application and/or the Auto Registration Service.**

12. *Attribute Query.* This service returns a summary of attributes with contact information for correcting discrepancies. This service has a read only interface with the EAS. **This service may be implemented jointly with the Claims Query.**
13. *Claims Query.* This service returns a summary of claims for the individual making the request. This service has a read only interface with the claims repository. **This service may be implemented jointly with the Attribute Query.**
14. *STS.* This provides signed SAML tokens and is a trusted element of the EAE. There are commercial products available to perform this service.

This list may be reduced to as little as eight services if the services are combined as listed above. One of the services (STS) will likely be purchased as commercial off-the-shelf reducing the service development to seven services.

The minimal instantiation of the EAE consists of five data modules as follows:

1. *Interim Store.* This holds changed data from authoritative content stores and manual inputs for updating the Enterprise Attribute Store. It may also include new attributes and identities. This store should be held separately from the *Enterprise Attribute Store* for security and integrity reasons.
2. *Service Registry.* This holds information provided about each web service at registration, such as name, web address, security information, and owner contact information. This store must have a read-only interface for the *Claims Engine.*
3. *Enterprise Attribute Store.* This holds attributes from authoritative content stores and manual inputs for each identity in the enterprise. This store must have read-only interfaces for the *Attribute Query Service* and the *Claims Engine.*
4. *Claims Repository.* This holds computed claims based on web service ACRs and delegated claims for each identity in the enterprise. This store must have a read-only interface for the *Provide Claims Web Service.*
5. *Monitor Records.* This holds records in accordance with ELS requirement for attribution. Each service, application, and requester has such a store with appropriate access and integrity provisions.

VIII. Conclusions

We have presented the core EAE requirements for an ELS system. This initial build is useful for first adoption of the ELS model and allows for full instantiation of the ELS security model and claims-based access control. Additional capabilities of an intermediate EAE build include an agent-based architecture, access claim delegation, multi-factor authentication, and end-point device management. A larger enterprise may require an advanced build, with additional capabilities including a certificate authority for temporary certificates and active entity veracity measures. This work is part of a body of work for high-assurance enterprise computing using web services. Elements of this work are described in [14–33].

References

[1] William R. Simpson and Kevin E. Foltz, Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI, “Enterprise Level Security - Basic Security Model”, Volume I, WMSCI 2016, Orlando, Florida, 8-11 March 2016.

[2] [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, Oct 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, Feb 2000

[3] TLS family Internet Engineering Task Force (IETF) Standards
- a. RFC 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, 2000-05
- b. RFC 3749 Transport Layer Security Protocol Compression Methods, 2004-05
- c. RFC 4279 Pre-Shared Key Cypher suites for Transport Layer Security (TLS), 2005-12
- d. RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08
- e. RFC 5289 TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008-08
- f. RFC 5929 Channel Bindings for TLS, 2010-07
- g. RFC6358 Additional Master Secret Inputs TLS, 2012-01
- h. RFC 7251 AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, 2014-06
- i. RFC 7301 Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, 2014-07
- j. RFC 7457 Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015-02

[4] Organization for the Advancement of Structured Information Standards (OASIS) open set of Standards
- a. N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, March 2008
- b. P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.
- c. S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 2005

[5] William List and Rob Melville, IFIP Working Group 11.5, Integrity In Information, Computers and Security, Volume 13, Issue 4, pp. 295–301, Elsevier, doi:10.1016/0167-4048(94)90018-3, 1994.

[6] William R. Simpson and Coimbatore Chandersekaran, CCCT2010, pp. 84–89, “An Agent Based Monitoring System for Web Services,” Orlando, FL, Apr 2011.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[7] William R. Simpson and Coimbatore Chandersekaran, 1st International Conference on Design, User Experience, and Usability, part of the 14th International Conference on Human-Computer Interaction (HCII 2011), "A Multi-Tiered Approach to Enterprise Support Services," 10 pp. Orlando, FL, July 2011.

[8] Simpson, William R., CRC Press, "Enterprise Level Security – Securing Information Systems in an Uncertain World," by Auerbach Publications, ISBN 9781498764452, May 2016, 397 pp. 56-61.

[9] William R Simpson and Coimbatore Chandersekaran, 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, World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, "Federated Trust Policy Enforcement by Delegated SAML Assertion Pruning", 4 pp., London, England, December 2008.

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

[12] 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, "Claims-Based Enterprise-Wide Access Control", pp. 524-529, Imperial College, London, July 2012.

[13] William R Simpson and Coimbatore Chandersekaran, 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, Co-authored by Coimbatore Chandersekaran.

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

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

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

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

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

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

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

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

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

[23] William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, "High Assurance Challenges for Cloud Computing," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2011, WCECS 2011, 19–21 October 2011, San Francisco, USA, pp. 61–66.

[24] Coimbatore Chandersekaran and William R. Simpson, "Claims-Based Enterprise-Wide Access Control," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4-6 July 2012, London, U. K., pp. 524–529.

[25] William R. Simpson and Coimbatore Chandersekaran, "Assured Content Delivery in the Enterprise," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4–6 July 2012, London, U. K., pp. 555–560.

[26] William R. Simpson and Coimbatore Chandersekaran, "Enterprise High Assurance Scale-up," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2012, WCECS 2012, 24-26 October 2012, San Francisco, USA, pp. 54–59.

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

[28] Simpson, William R., and Kevin E. Foltz, "Enterprise Level Security: Insider Threat Counter-Claims," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2017, 25–27 October, 2017, San Francisco, USA, pp. 112–117.

[29] Simpson, William R. and Kevin E. Foltz, Proceedings of the 22nd International Command and Control Research and Technology Symposium (ICCRTS), "Escalation of Access and Privilege with Enterprise Level Security," Los Angeles, CA. September 2017, pp. TBD.

[30] Simpson, William R. and Kevin E. Foltz, Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017), Volume 1, pp. 177–184, Porto, Portugal, 25–30 April, 2017, "Enterprise Level Security with Homomorphic Encryption," SCITEPRESS – Science and Technology Publications.

[31] Foltz, Kevin E. and William R Simpson, "Enterprise Considerations for Ports and Protocols," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2016, 19–21 October, 2016, San Francisco, USA, pp.124–129.

[32] "Simplified Key Management for Digital Access Control of Information Objects," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2016, 29 June–1 July, 2016, London, U.K., pp. 413–418.

[33] Simpson, Wessex Institute, Proceedings of the International Conference on Big Data, BIG DATA 2016, "Access and Privilege in Secure Big Data Analysis," 3–5 May 2016, Alicante, Spain, pp. 193–205.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)