Enterprise High Assurance Scale-up

Coimbatore Chandersekaran and William R Simpson

***Abstract*— Many Organizations are moving to web-based approaches to computing. As the threat evolves to higher levels of sophistication, many governmental and commercial organizations are also moving toward high assurance. This paper describes an approach that uses strong bi-lateral end-to-end authentication with end-point encryption and with SAML-based authorization using OASIS Security Standards. This service-based approach offers many of the advantages of the cloud-based approaches. Cloud-based approaches allow for more agile scale-up, while maintain a low marginal cost of accommodating increased users. However, many of the applications require high assurance, attribution, formal access control processes, and a wide range of threat mitigation procedures for many of the industries (banking, credit, content distribution, etc.) that are considering conversion to cloud computing environments. Current implementations of cloud services do not meet these high assurance requirements. This high assurance requirement presents many challenges to normal computing and some rather precise requirements that have developed from high assurance issues for web service applications. The most difficult part of scaling up to higher user levels is the maintenance of the security paradigms that provide mitigation of these generic and specific threats.**

***Index Terms*— Authentication, Authorization, Attribution, Public Key Infrastructure, Security Assertion Markup Language (SAML)**

I. INTRODUCTION

In certain enterprises, the network is continually under attack. Examples might be:

- Banking industry enterprise such as a clearing house for electronic transactions,
- Defense industry applications,
- Credit card consolidation processes that handle sensitive data, both fiscal and personal,
- Medical with concerns for privacy and statutory requirements,
- Content distributors worried about rights in data, or theft of content.

The attacks have been pervasive and continue to the point that nefarious code may be present, even when regular monitoring and system sweeps clean up readily apparent malware. This omnipresent threat leads to a healthy paranoia of resistance to observation, intercept, and masquerading. Despite this attack environment, the web interface is the best way to provide access to many of the enterprise users. One way to continue operating in this environment is to not only know and vet your users, but also your software and devices.

Manuscript received 4 June 2102; revised and finalized 30 July 2012. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses. 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

Coimbatore Chandersekaran is with the Institute for Defense Analyses.(email: cchander@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)

Even that has limitations when dealing with the voluminous threat environment. Today we regularly construct seamless encrypted communications between machines through SSL or other TLS. These do not cover the "last mile" between the machine and the user (or service) on one end, and the machine and the service on the other end. This last mile is particularly important when we assume that malware may exist on either machine, opening the transactions to exploits for eavesdropping, ex-filtration, session high-jacking, data corruption, man-in-the-middle, masquerade, blocking or termination of service, and other nefarious behaviors. Before we examine the challenges of cloud computing systems, let us first examine what high assurance architecture might look like.

II. TENETS FOR HIGH ASSURANCE COMPUTING

This section provides nine tenets that guide decisions in an architectural formulation for high assurance and implementation approaches [1]. These tenets are separate from the "functional requirements" of a specific component (e.g., a name needs to be unique); they relate more to the goals of the solution that guide its implementation.

- The *zeroth* tenet is that the *malicious entities* can look at all network traffic and send virus software to network assets. In other words, rogue agents (including insider threats) may be present and to the extent possible, we should be able to operate and, in their presence, although this does not exclude their ability to view some activity. Assets are constantly monitored and cleaned; however, new attacks may be successful at any time and nefarious code may be present at any given time.
- The *first* tenet is *simplicity*. This seems obvious, but it is notable how often this principle is ignored in the quest to design solutions with more and more features. That being said, there is a level of complexity that must be handled for security purposes and implementations should not overly simplify the problem for simplicity's sake.
- The *second* tenet, and closely related to the first is *extensibility*. Any construct we put in place for an enclave should be extensible to the domain and the enterprise, and ultimately to cross-enterprise and coalition. It is undesirable to work a point solution or custom approach for any of these levels.
- The *third* tenet is *information hiding*. Essentially, information hiding involves only revealing the minimum set of information to the outside world needed for making effective, authorized use of a capability. It also involves implementation and process hiding so that this information cannot be farmed for information or used for mischief.
- The *fourth* tenet is *accountability*. In this context, accountability means being able to unambiguously identify and track what active entity in the enterprise performed any particular operation (e.g., accessed a file or IP address, invoked a service). Active entities include people, machines, and software process, all of which are named

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

registered and credentialed. By accountability we mean attribution with supporting evidence. Without a delegation model, and detailed logging, it is impossible to establish a chain of custody or do effective forensic analysis to investigate security incidents.

- The *fifth* tenet is *minimal detail* (to only add detail to the solution to the required level). This combines the principles of simplicity and information hiding, and preserves flexibility of implementation at lower levels. For example, adding too much detail to the access solution while all of the other IA components are still being elaborated may result in wasted work when the solution has to be adapted or retrofitted later.
- The *sixth* tenet is the emphasis on a *service-driven* rather than a product-driven solution whenever possible. Using services makes possible the flexibility, modularity, and composition of more powerful capabilities. Product-driven solutions tend to be more closely tied to specific vendors and proprietary products. That said, commercial off-the-shelf (COTS) products that are as open as possible will be emphasized and should produce cost efficiencies. Procurement specifications should require functionality and compatibility in lieu of requiring operations in a Microsoft forest [2] environment.
- The *seventh* tenet is that *lines of authority* should be preserved and IA decisions should be made by policy and/or agreement at the appropriate level.
- The *eighth* tenet is *need-to-share* as overriding the need-to-know. Health, defense, and finance applications often rely upon and are ineffective without shared information.

A. Architectural Features

Building an architecture that conforms to these tenets requires specific elements to ensure the tenets are built into systems. In the architecture we espouse, the basic formulation follows a web 2.0 approach and uses Organization for the Advancement of Structured Information Standards (OASIS) standards of security [4]. These elements are listed below:

Naming and Identity

Identity will be established by the requesting agency. All recognized certificate authorities naming scheme must be honored. To avoid collision amongst the schemes, the identity used by all federated exchanges shall be the distinguished name as it appears on the primary credential provided by the certificate authority. The distinguished name must be unique over time and space, which means that retired names are not reused and ambiguities are eliminated. Naming must be applied to all active entities (persons, machines, and software).

Credentials

Credentials are an integral part of the federation schema. Each identity (all active entities) requiring access shall be credentialed by a trusted credentialing authority. Further, a Security Token Server (STS) must be used for storing attributes associated with access control. The STS that will be used for generating Security Assertion Markup language (SAML) tokens must also be credentialed (primarily through the same credentialing authority, although others may be entertained).

PKI required – X.509 Certificates

The primary exchange medium for setting up authentication of identities and setting up cryptographic flows is the Public Key Infrastructure (PKI) embodied in an X.509 certificate.

Certificate Services

The certificate authority must use known and registered (or in specific cases defined) certificate revocation and currency-checking software.

Bi-Lateral End-to-End Authentication

The requestor will not only authenticate to the service (not the server), but the service will authenticate to the requestor. This two-way authentication avoids a number of threat vulnerabilities. The requestor will initially authenticate to the server and set up a Secure Socket Layer (SSL) connection to begin communication with the service. The primary method of authentication will be through the use of public keys in the X.509 certificate, which can then be used to set up encrypted communications (either by X.509 keys or a generated session key). The preferred method of communication is secure messaging, contained in Simple Object Access Profile (SOAP) envelopes. All messages are encrypted for delivering to the recipient of the message.

Authorization Using SAML Packages

All authorizations will be through the use of SAML packages in accordance with the SAML 2.0 specification provided by OASIS [3-7].

Registration of the STS

All STS that create and sign SAML packages must be registered. The certificate of the STS will be used to sign SAML tokens, and complete bi-lateral authentication between requestors and the STS.

Recognizing STS Signatures

STS signatures will be recognized only for registered STSs and may be repackaged by the local STS when such registration has been accomplished. Unrecognized signatures will not be honored and the refusal will be logged as a security relevant event.

Certificate Caches

Local STSs within the enterprise forests will maintain a certificate cache of all registered STSs to facilitate the re-issuance of SAML packages when appropriate.

III. HIGH ASSURANCE ARCHITECTURE ELEMENTS

Despite the obvious advantages of cloud computing, the large amount of virtualization and redirection poses a number of problems for high assurance. In order to understand this, let's examine a security flow in a high assurance system.

The basic elements include a user, who initially authenticates to his/her domain using a hardware token and establishes a Virtual Private Network (VPN) session; a Security Token Server (STS), in this case the Identity Provider (IdP); and attribute stores for generating the Security Assertion Markup Language (SAML) token.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

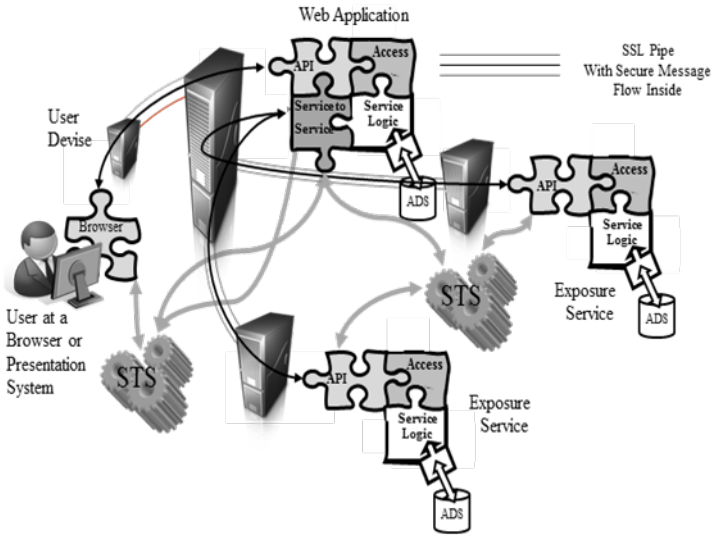

Fig. 1. High Assurance Security Flows

The application system consists of a web application (for communication with the user), one or more aggregation services that invoke one or more exposure services and combine their information for return to the web application and the user. As a prerequisite to end-to-end communication an SSL or other suitable TLS is set up between each communication of the machines. The exposure services retrieve information from one or more Authoritative Data Sources (ADSs). Each communication link in Fig. 1 will be authenticated end- to-end with the use of public keys in the X.509 certificates provided for each of the active entities.

This two-way authentication avoids a number of threat vulnerabilities. The requestor initially authenticates to the service provider. Once the authentication is completed, a TLS/SSL connection is established between the requestor and the service provider, within which a WS-Security package will be sent to the service. The WS-Security package contains a SAML token generated by the Security Token Server (STS) in the requestor domain. The primary method of authentication will be through the use of public keys in the X.509 certificate, which can then be used to set up encrypted communications (either by X.509 keys or a generated session key). Session keys and certificate keys need to be robust and sufficiently protected to prevent malware exploitation. The preferred method of communication is secure messaging using WS Security, contained in SOAP envelopes. The encryption key used is the public key of the target (or a mutually derived session key), ensuring only the target can interpret the communication.

The problem of scale-up and performance is the issue that makes cloud environments so attractive. The cloud will bring on assets as needed and retire them as needed. The trick is to maintain the security paradigm as we scale up.

A traffic cop (load balancer) monitors activity and posts a connection to an available instance. In this case all works out since the new instance has a unique name, end-point, and credentials with which to proceed. All of this, of course needs to be logged in a standard form and parameters passed to make it easy to reconstruct for forensics. We have shown a couple of threats that need mitigation where one eavesdrops on the communication and may actually try to insert himself into the conversation (man-in-the-middle). This highlights the importance of bi-lateral authentication and encrypted communications. The second highlights the need to protect caches and memory spaces.

IV. PROVISIONING OF APPLICATIONS

In order to make the first communication to a home page in the service environment, we will assume a web application (Enterprise Services Homepage [ESHP]) or a device application (Enterprise Application Store [EAPPS]) that can provide the user links to appropriate services. These links are complex in that they must contain the Request for SAML Token (RST), the URI of the IdP, and the URI of the target application. The first link to the ESHP will be provided as a widget on the Enterprise Standard desktop. This ESHP or EAPPS will bilaterally authenticate with the user and consume the user's SAML in order to provide the user an appropriate list of services.

Fig. 2. Enterprise Services Home Page

The initial web application will contain all of the information necessary to provide this to the user. To do so, the web application must have access to or contain a registry with the following information:

- An enumerated list of Authorized SAML Producers
  - Includes Public Key (for SAML consumption)
    - Includes URI of the SAML Producer (IdP)
- An enumerated list of web applications
  - Unique name of each application, or Traffic Handling Web Application (see discussion below)
  - The end point URI for accessing the applications or Traffic Handling Web Application (see discussion below)
  - A description of the service
  - The Access Control List (ACL) for the Service
  - The URI of the IdP for the user

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig. 3. Enterprise Application Store

The last element is used to eliminate services for which the user does not have authorization. This is done by matching the unfiltered SAML elements to the ACL in the last bullet. A match must occur (or the ACL is none) in order for the ESHP or EAPPS to provide a link to the service. When the match occurs, the ESHP will add the name, descriptive material, and the complex URI link to the home page.

All registered services are examined and when the list is compiled, the home page is sent to the user. A notional homepage is provided in figure 2. A notional device display for the EAS is presented in figure 3. The desktop or device icons must necessarily expire after a time period (say three months) to provide security against changes in status or privilege. When a change occurs, the authorization will fail, but the expiring of the icon will clean up the displays on the desktop or device.

Selecting an icon will invoke a link that will send an RST and a compound URI to the IdP for processing and connect the user with the end-point for bi-lateral authentication with a SAML token at the application. If the SAML is acceptable, a session is started between the user and the web application.

V. HIGH ASSURANCE ARCHITECTURE SCALE-UP

Scale-up to higher levels of users will require a number of different schemes. The most critical, since it involves every request, will be the STS. Example data needed for load-balancing calculations are provided below:

- Test data are still being developed; however, assume testing indicates 100 token requests can be satisfied in 1 second by the current STS (IdP).
  - Improved versions of the STS or processors may reduce these requirements
- Initial operational deployment
  - Requirement for 4 SAML tokens per second [1,000 users in 5 minutes]
  - Need 1 STS (IdP)
- Six months after initial operational deployment:
  - Requirement for 34 SAML tokens per second [10,000 users in 5 minutes]
  - Need 1 STS (IdP)
- One year after initial operational deployment:
  - Requirement for 334 SAML tokens per second [100,000 users in 5 minutes]
  - Need 5 STS (IdP) cluster – assumes a 20% throughput loss in clustering
- Two years after initial operational deployment:
  - Requirement for 1667 SAML tokens per second [500,000 users in 5 minutes]
  - Need 23 STS (IdP) – assumes a 25% throughput loss in multiple clustering.

Note that the STS is a trusted component and can be load balanced in the traditional manner, as shown in Figure 4.

In this configuration, the clusters share the same naming, PKI credentials and end-point identities. This is the only exception to the requirements for uniquely naming, PKI credentials, and end-points because the STS is the primary trusted component in the system. The need for load balancing of the SP is not anticipated at this time, since it is used for occasional SAML verification or SAML rewriting in the case of federation.

The second most likely element to need scale-up is the ESHP. This one is a bit trickier. Schemas that essentially extend the thread capabilities of the application may be employed. When the capacities of the thread architecture are exhausted and independent instance of the application need to be set up, the process requires care. The application or service is not a trusted element and is not exempt from the requirements for unique naming, credentialing, and end-points. Each independent instance must have its own name, URI, and credentials.

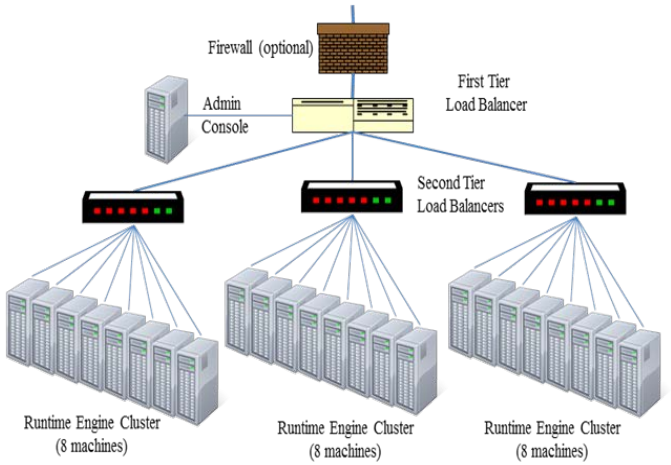

Fig. 4. STS Load Balancing

Further, each independent instance must be provisioned for in the attribute stores if it will make further service calls. Fortunately, all web applications and services can be handled in the same manner and the process is not dependent on the number of instances needed to handle the user request. A specific (x, where x corresponds to the web application or web service that needs balancing) Instance Availability Service (IAS_x) is set up as the end point for a request that needs to be load balanced. Note that this means that the end-point in the ESHP must be changed to the specific IAS_x, and the end-point in the widget for the ESHP must be changed to the IAS for the ESHP. The IAS_x needs to have the following information available:

- Number of independent instances of the web application
- Unique name of each independent instance of the web application

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Unique end-point of each independent instance of the web application
- (OPTIONALLY) Usage and load data for each of the independent instances of the web application

The user goes to the IdP with the address of the IAS, which he can have stored in his favorites, or executes the link on the ESHP (or in the case of ESHP from the desktop widget). The IdP then posts the SAML over HTTPS to the IAS. The IAS doesn't even need to read the SAML (authentication only, or identity-based access control), but would repost the SAML over HTTPS to the independent instance it calculates. It is then completely out of the way, just like the IdP is completely out of the way, and the user is in session with the instance of the web application. This process is shown in Figure 5.

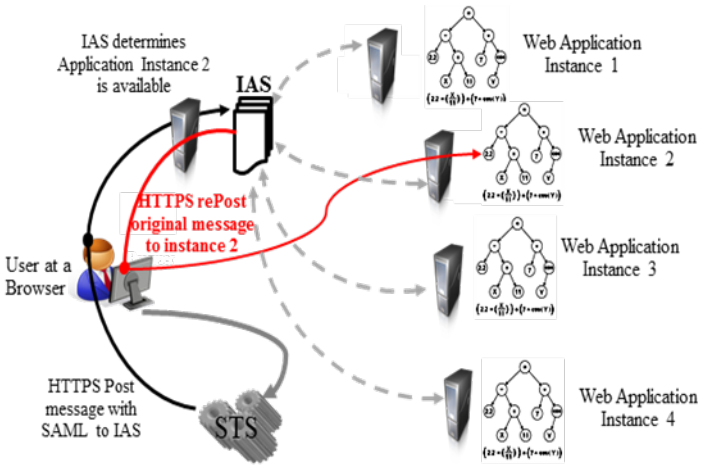

Fig. 5. High Assurance Load Balancing

The IAS (traffic director) monitors activity and posts a connection to an available independent instance. No communication with the user is required, much as the https interface to the IdP requires no communication with the user. The determination of an available connection can start simply (initially round robin) and become more sophisticated by monitoring activity at the independent instances and choosing the lowest activity instance (or even more sophisticated later by considering memory, CPU, and number of threads in each instance). In this case, all of the end-to-end processes work since the new independent instance has a unique name, end-point, and credentials with which to proceed. All of this, of course, needs to be logged in a standard form and parameters passed to make it easy to reconstruct for forensics.

Key management is complex and essential. When a new instance is required, it must be built and activated (credentials and properties in the attribute store, as well as end-point assignment). When a current instance is retired, it must be disassembled, and de-activated (credentials and properties in the attribute store, as well as end-point assignment). All of these activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics.

Rules for maintaining high assurance during scale-up:

1. *Shared Identities and credentials break the accountability paradigm.*
   - Each independent instance of a machine or service must be uniquely named [8] and provided a PKI Certificate for authentication. The Certificate must be activated while the virtual machine is in being, and de-activated when it is not, preventing hijacking of the certificate by nefarious activities.
   - The naming and certificates must be pre-issued and self- certification is not allowed. Each independent instance of a machine or service must have a unique end point. This may take some manipulation through the load balancing process but is required by attribution and accountability. This means that simple re-direct will not work. The one exception is the IdP of the STS, which is trusted software. Extensions of the thread mechanism by assigning resources to the operating system may preserve this functionality. The individual mechanism for virtualization will determine whether this can be accomplished.
2. *Each independent instance of a service must have an account provisioned* with appropriate elements in an attribute store. These must be pre-issued and linked to the unique name for each potential independent instance of a service. This is required for SAML token issuance.
3. *The importance of cryptography cannot be overstated,* and all internal communications as well as external communications should be encrypted to the end point of the communication. Memory and storage should also be encrypted to prevent theft of cached data and security parameters.
4. *Private keys must reside in Hardware Storage Modules (HSMs).* The security of the Java software key store does not meet high assurance criteria. Stand-up of an independent machine or service must link keys in HSM, and activate credentials pre-assigned to the service.
   - Stand-down of an independent machine service must de-link keys in HSM, and de-activate credentials pre-assigned to the service.
   - Key management in this environment is a particular concern and a complete management schema including destruction of session keys must be developed.
5. *Proxies and re-directs break the end-to-end paradigm.*
   - When end points must change, a re-posting of communication is the preferred method. There must be true end-to-end communication with full attribution. This will mean that communication must be re-initiated from client to server when an instance is instantiated, and it must have a unique end point, with unique credentials and cryptography capabilities.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

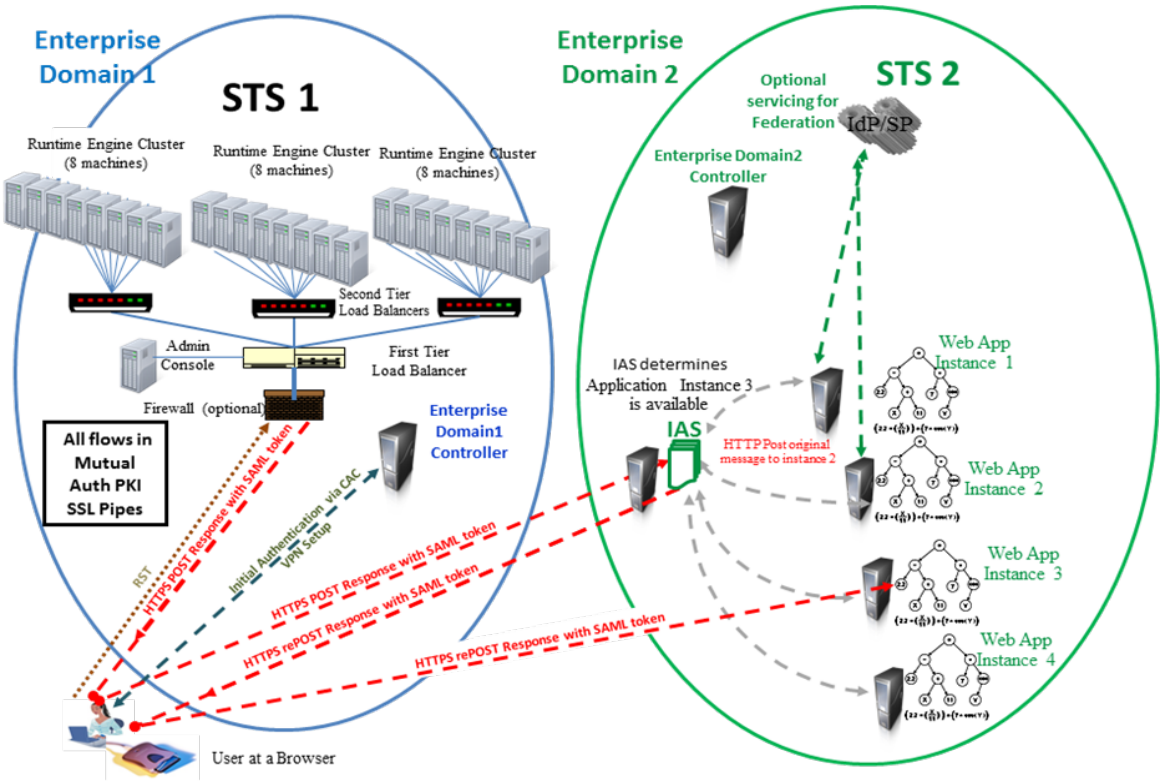

Fig. 6. Accessing a Web Application in a Scaled Up Environment

6. *All activities must be logged in a standard format with reference values* that make it easy to reassemble the chain of events for forensics.

VI. SUMMARY

We have reviewed the basic approaches to scale up in high assurance computing environments. Virtualization must be very carefully reviewed to ascertain if the security paradigm can be maintained. Extensions of the thread mechanism by assigning resources to the operating system may preserve this functionality. The individual mechanism for virtualization will determine whether this can be accomplished. Notably the extensive use of virtualization and redirection is severe enough that many customers who need high assurance have moved away from the concept of cloud computing. Figure 6 provides a summary of how a user addresses an individual web application in a scaled-up system. This processes described in this paper are part of a broad-scale, high-assurance enterprise stand-up, including high-assurance specification, and current implementation. Other aspects of these enterprise processes are described in [9-14].

REFERENCES

[1] Chandersekaran, C., 2009, Air Force Information Assurance Strategy Team, *Air Force Information Assurance Enterprise Architecture*, Version 1.70, SAF/XC. [Not available to all]

[2] Shibboleth Project, 2011, Available at http://shibboleth.internet2.edu/

[3] OASIS Identity Federation, 2005a, "Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0", Microsoft Corporation

[4] OASIS Identity Federation, 2005b, "WSE 3.0 and WS-ReliableMessaging", Microsoft White Paper, Available at http://msdn2.microsoft.com/en-us/library/ms996942(d=printer).aspx.

[5] OASIS Identity Federation, 2007, "WS-ReliableMessaging Specification" "WS-SecureConversation Specification", OASIS, March 2007

[6] OASIS Identity Federation, 2011a, *Liberty Alliance Project*, Available at http://projectliberty.org/resources/specifications.php.

[7] OASIS Identity Federation, 2011ba, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. (Accessed on 19 February 2011)

[8] Anonymous, 2010, Standard for Naming Active Entities on DoD IT Networks, Version 3.5

[9] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, The 1st International Multi-Conf. on Eng. and Tech. Innovation, "Cross-Domain Solutions in an Era of Information Sharing", Volume I, pp.313-318, Orlando, FL., June 2008.

[10] 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.

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

[12] Coimbatore Chandersekaran and William R. Simpson, 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, FL., April 2011.

[13] 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", 10pp, Orlando, FL., July 2011.

[14] 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.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)