Cloud Forensics Issues

William R Simpson and Coimbatore Chandersekaran

***Abstract*— Forensics is undertaken to find out exactly what happened on a computing system and who or what was responsible for it. This is done by a structured investigation while maintaining a documented chain of evidence. Cloud computing is emerging as an attractive, cost effective computing paradigm. The early offerings of cloud capabilities have not provided security, monitoring or attribution that would allow an effective forensics investigation. The 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 and forensics applications of cloud systems. The challenges of high assurance and the maintenance of a documented chain of evidence associated with cloud computing are primarily in four areas. The first is virtualization and the loss of attribution that accompanies a highly virtualized environment. The second is the loss of ability to perform end-to-end communications. The third is the extent to which encryption is needed and the need for a comprehensive key management process for public key infrastructure, as well as session and other cryptologic keys. The fourth is in monitoring and logging for attribution, compliance and data forensics. Our view of high assurance and the issues associated with web services is shaped by our work with DoD and the Air Force, but applies to a broader range of applications.**

***Index Terms* — Attribution, Cloud Computing, Forensics, IT Security, Standards, Monitoring, Virtualization.**

I. INTRODUCTION

CLOUD computing has come to mean many different things. To some, it is simply putting one's data on a remote server. However, in this paper, we utilize the definition provided by NIST [1]. They define five essential characteristics of any cloud computing environment:

1. On demand self-service,
2. Broad network access,
3. Resource pooling,
4. Rapid elasticity, and
5. Measured service.

It is important to note that multi-tenancy and virtualization are *not* essential characteristics for cloud computing.

Cloud computing is, at its core, a *service*. There are three primary models of this service. In the lowest level Infrastructure as a Service (IaaS), storage, computation, and networking are provided by the cloud provider to the cloud consumer. In level two of the cloud models, Platform as a Service (PaaS), all of the trappings of IaaS plus an operating system and perhaps some application programming interfaces (APIs) are provided and managed by the cloud provider. The highest service model is Software as a Service (SaaS), in which the cloud provider provides an end-user service such as webmail. The higher the service model, the more control the cloud provider has as compared to the cloud consumer. There are four different models for deploying cloud services. Primarily, they are public or private clouds. In a public cloud, the infrastructure--although generally not the data on it--may be used by anyone willing to agree to its terms of use. Public clouds exist off the premises of the cloud consumer. Private cloud infrastructure is used only by one organization. It may exist either on or off the organization's premises. There are two twists to these infrastructures. In a community cloud, a group of organizations with similar interests or needs share a cloud infrastructure. That infrastructure is not open to the general public. In a hybrid cloud, two or more cloud deployment models are connected in a way that allows data or services to move between them. An example of this would be an organization's private cloud that makes use of a community cloud during periods of high utilization.

Manuscript received February 2, 2014; revised March 1, 2014. 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)

II. CLOUD BENEFITS

Cloud computing benefits emerge from economies of scale [2]. Large cloud environments with multiple users are better able to balance heavy loads, since it is unlikely that a large proportion of cloud consumers will simultaneously have high utilization needs. The cloud environment can therefore run at a higher overall utilization, this may result in better cost effectiveness. In many cloud environments this balancing of resources is done by virtualization and the use of a hypervisor, offering resiliency and agility. In a large cloud computing environment, rather than having a number of information technology generalists, the staff has the ability to specialize and become the experts of their technical areas. With regard to information security, the staff can become even more specialized and spend more time hardening platforms to secure them from attacks. In the homogeneous cloud environment, patches can be rolled out quickly to the nearly identical hosts. Identically configured hardware elements are not a cloud requirement but do facilitate large-scale administration and focusing of expertise.

III. CLOUD WEAKNESSES

Cloud computing is not without its negatives. In cases where services are outsourced, there can be a loss of control. This can affect compliance with laws, regulations, and organizational policies. Cloud systems have additional levels of complexity to handle intra-cloud communications, scalability, elasticity, data abstraction, and more. To be available to cloud consumers, cloud providers may need to make their services available via the Internet, opening

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

interfaces that are subject to attack. And critically, many clouds allow multiple organizations simultaneous use of a single host and virtualization. If one tenant organization is compromised or malicious, it may be able to compromise the data or applications of the other organizations on the same host. The load balancing may use a single identity for all instances of a service whether it is virtual or real.

A. *Changes in the Threat Model*

There are clear differences in many of the threat scenarios as detailed below [3]:

1. Loss of governance (or visibility and/or control of the governance process)
2. Lock-in (threats may be present and locked into the cloud environment and shared among all tenants).
3. Isolation failure (e.g., hypervisor attack, lack of accountability – lack of distinction between virtualized instances of services).
4. Compliance risks (if provider cannot provide compliance evidence or will not permit audit by customer, lack of accountability)
5. Management interface compromise (and or inheritance of threats and/or malicious code from other users of the cloud).
6. Data protection (how does customer verify protection, lack of accountability)
7. Insecure or incomplete data protection and/or data deletion
8. Malicious insider (often the cloud insider is not vetted as well as the organizational insider, and insiders from other customers could bring in contagious viruses – see 5 above.)
9. Unprotected or ineffective key management for cryptography.

B. *Traditional Data Centers versus Cloud Computing*

Cloud computing relies on much of the same technical infrastructure (e.g., routers, switches, operating systems, databases, web servers) as traditional data centers and as a result, many of the security issues are similar in the two environments. The notable exception in some cases is the addition of a hypervisor for managing virtual machines. The Cloud Security Alliance's security guidance states "Security controls in cloud computing are, for the most part, no different than security controls in any IT environment. Cloud computing is about gracefully losing control while maintaining accountability even if the operational responsibility falls upon one or more third parties." While many of the controls are similar, there are two factors at work that make cloud computing different: perimeter removal and trust. With cloud computing, the concept of a network or information perimeter changes radically. Data and applications flow from cloud to cloud via gateways along the cloud perimeters. However, since the data may be stored in clouds outside the organization's premises or control, perimeter controls become less useful. In exchange for the lack of a single perimeter around one's data and applications, cloud consumers must be able to trust their cloud providers. A lack of trust in a cloud provider does not necessarily imply a lack of security in the provider's service. A cloud provider may be acceptably secure, but the novelty of cloud computing means that many providers have not had the opportunity to satisfactorily demonstrate their security in a way that earns the trust of cloud consumers. Trust must be managed through detailed Service Level Agreements (SLAs), with clear metrics and monitoring mechanisms, and clear delineation of security mechanisms [4].

IV. A PARADIGM FOR HIGH ASSURANCE

While the current implementations of cloud computing provide efficient and operationally friendly solutions to data computing and content distribution, they are not up to the challenge of high assurance.

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

- Banking industry enterprise.
- Defense industry applications,
- Credit card consolidation processes.
- Commercial point-of-sale processes.
- Medical -- privacy and statutory requirements,
- Content Distributor's -- rights in data, theft of content.

The attacks have been pervasive and often include previously unseen attack vectors and they 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 many threats including resistance to observation, intercept and masquerading. The web interface is the best way to provide access to many of its users despite this highly active threat environment. One way to maintain capability in this type of environment is to not only know and vet your users, but also your software and devices. 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 eaves dropping, ex-filtration, session high-jacking, data corruption, man-in-the-middle, repeat replay, masquerade, blocking or termination of service, and other nefarious behavior. Before we examine the challenges of cloud computing systems, let us first examine what high assurance architecture might look like.

A. *A Comprehensive Set of Tenets*

We have implemented twelve tenets that guide decisions in an architectural formulation for high assurance approaches [5]. These tenets are distinct from the functional requirements normally associated with specific software component(s) (e.g., a name(s) need to be unique or identities need to be established); they relate more to the goals of the solution that guide its implementation.

- The ***zeroth*** tenet is that the *malicious entities* are present and can look at all network traffic and may attempt to modify that traffic by sending 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 in their presence, although this does not exclude their ability to view some activity. Assets

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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 elegant 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. Its corollary in software design provides only information that is needed to a software segment or process.
- 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 registered and credentialed. By accountability we mean attribution with supporting evidence. Identity is a key attribute here and virtual elements must have unique identities. Without such an identity process and detailed logging, it is impossible to establish a chain of custody or do effective forensic analysis to investigate security incidents.
- Th**is *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*** 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. This means that for acquisition, functionality and compatibility are specified as opposed to specific reference to current implementations such as "must operate in a Microsoft forest" [6] 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. An example here is that data owners should implement sharing requirements even when the requirements come from "higher authority."
- The ***eighth*** tenet is *need-to-share* as overriding the need-to-know. Often effective health, defense, and finance rely upon and are ineffective without shared information.
- The ***ninth*** tenet is *separation of function*, this makes updates easier, isolates vulnerabilities and aids in forensics.
- The ***tenth*** tenet is *reliability*; it needs to work even if adversaries know how the process works. In setting up a large scale enterprise we need to publish exactly how things work and this should not create additional vulnerabilities.
- The ***eleventh*** tenet is to *trust but verify* (and validate). This essentially precludes the use of identity by self-attestation which is unverified.

B. Element of the High Assurance Architecture

In order to build an architecture that conforms to these tenets, there must be elements that ensure that they are built into the systems. In the architecture we describe, the basic formulation follows a web 2.0 approach and uses Organization for the Advancement of Structured Information Standards (OASIS) standards of security [7]. These elements are listed below:

Identity derives from *accountability*. Identity will be established by the requesting agency. To avoid collisions, 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 derive from *identity*, *reliability*, *trust but verify*, and *malicious entities*. 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 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.

Authentication derives from *accountability* and *malicious entities*. The requestor will not only authenticate to the service (not the server), but the service will authenticate to the requestor. The preferred method of communication is secure messaging, contained in Simple Object Access Profile (SOAP) envelopes.

Confidentiality in transit derives from *malicious entities* and *reliability*. All messages are encrypted for delivery to the recipient of the message.

Authorization derives from *accountability*, *malicious entities*, *lines of authority*, *trust but verify*, and *extensibility*. Authorizations will be through the use of SAML packages in accordance with the SAML 2.0 specification provided by OASIS [8].

V. CLOUD AND HIGH ASSURANCE

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

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

understand this, let's examine a security flow in a high assurance system.

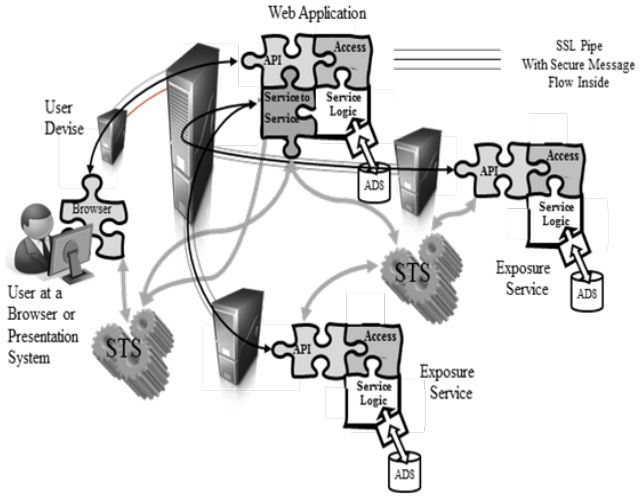

Fig. 1. High Assurance Security Flows

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 combines 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 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, an TLS 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 [9, 10] 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 and virtualization so attractive. The cloud will bring on assets as needed and retire them as needed. Let us first examine scale-up in the unclouded secure environment. We will show only the web application, although the same rules apply to all of the communication links between any active elements shown in Figure 1. The simplest form of dividing the load is to stand up multiple independent instances and divide users into groups who will use the various instances. Dependent instances that extend the thread capabilities of the server are considered single independent instances. Remember, all independent instances are uniquely named and credentialed and provisioned in the attribute stores.

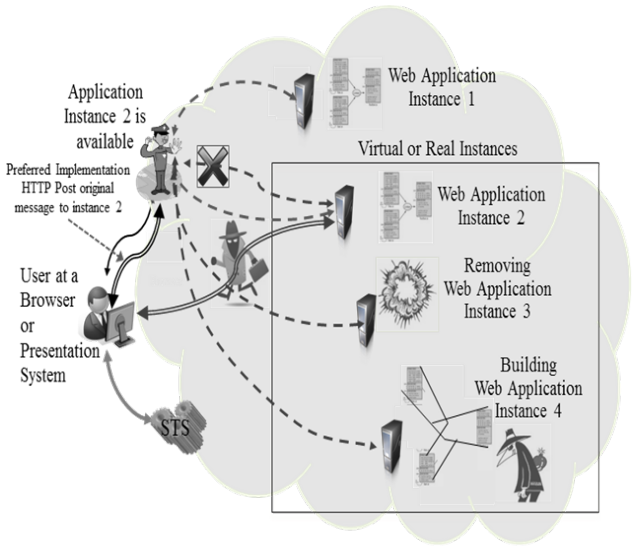

Fig. 2. High Assurance Virtualized Hypervisor Activity

A traffic cop (hypervisor) 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 is present on instance 4 and highlights the need to protect caches and memory spaces.

When a cloud environment runs out of resources for computing, it builds additional instances, some of these may be thread extension schemas, and some may be independent instances. The traffic cop here is often called a hypervisor and it keeps track of the instances and connections. Figure 2 shows notionally how this operation works. When thread capacity is saturated at the server, the hypervisor would nominally redirect the request to an independent virtual or real instance of the web application. If none exists, it will build one from elements in the resource pool as depicted in instance 4 on the chart. If the last requester signs off of an independent virtual or real instance (instance 3 in the figure), the hypervisor tears down the instance and places the resources back into the resource pool. This provides an efficient re-allocation of resources.

VI. ACCOUNTABILITY, MONITORING AND FORENSICS

The goal of computer forensics is to perform a structured investigation while maintaining a documented chain of evidence to find out exactly what happened on a computing system and who or what was responsible for it [11]. There are several steps that must be taken to preserve the state, if we are interested in a forensics reconstruction of the computing.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A. Accountability

In order to ensure accountability, the number of independent instances must be anticipated. Names, credentials and end points must be assigned for their use. The attribute stores and HSMs must be provisioned with properties and key to be used. The simple re-direct must be changed to a re-post loop as in Figure 2. The requester will then have a credentialed application to authenticate with bi-laterally and an end point for end-to-end message encryption. Key management is complex and essential. When a new independent instance is required, it must be built and 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. When a current independent instance is retired, it must be disassembled, and de-activated (credentials and properties in the attribute store, as well as end point assignment).

B. Monitoring

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. The same threats exist, and the same safeguards must be taken. In fact, in Figure 2 nefarious code is built right into the virtual or real instance 4, which underscores the need for trusted and verified software to do the virtualization, and protection of the resources while they are in the resource pool.

C. Knowledge Repository

The knowledge repository (KR) is a single integrated source of all information on the operation of the cloud. Instrumented agents feed the data base on a schedule or on demand. The knowledge base is where all information related to the enterprise SOA is stored. This will include the following:

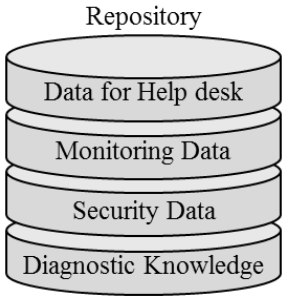

Fig. 3. Knowledge Repository

- Hardware/software current status from cloud or enterprise
- Current reports on test activities including response times, frequency of test, etc.
- Current reports of usage data from service agent monitors and service logs, including number of users, session times, response times, etc.
- Hardware/software historical data
- A list of current alerts for the entire enterprise
- Historical data on alerts
- Current monitoring records

Many of these feed status displays for the network monitoring and the enterprise support desk. They can provide a basis for real-time management and network defenses as well as forensics.

D. Forensic Tools

Tools are needed for KB query, correlation, and anomaly detection. Each tool should be functionally specified together with outputs and based upon the standard defined records described in the next section.

VII. STANDARDS REQUIREMENTS FOR CLOUD FORENSICS

Standards provide a basis for commercial applications as well as a market for developed tools and provide an interoperability process for generated files. Standards need to be developed, and are required for a number of cloud operations:

1. Standards for identity issuance of virtual objects. Current practice of overloading the identity makes the development of forensic strategies difficult or problematic. The standard should include credentialing and establishment of identity attributes.
2. Standards are required for a number of details associated with cloud monitoring. These requirements are based upon monitoring requirements derived for enterprise application and shown in Figure 4.
   - Standard lists of events that must be logged and alerted including event specific data (these may be separated into minimal, rich, and robust categories). The events should include all of the agility events such as stand-up and tear-down of services as well as credentialing and establishment of identity attributes.
   - A standard process for automatic escalation of monitoring under pre-defined circumstances.
   - Standard monitoring record content (that is separate from event specific content) including; Date/time, Record Number, Thread Number, Sequence Number, Active Entity Name, Event Name, ID of Service Requestor.
   - Standard identifiers for monitoring content (to assist upload to a centralized KR).
   - Standard processes for protection, access and integrity of monitor records.
   - Standards for continuous operations and recovery of the monitoring system and its records.
3. Standards for KR construction, KR update process, forensics tools and analysis processes.
4. Standards for protection, access and integrity of stored resources that will be used in agile virtualization.

VIII. SUMMARY

We have reviewed the basic approaches to clouds and their potentials for savings in computing environments. We have also discussed high assurance architectures and their requirements which provide direct challenges to the way cloud computing environments are organized. Notably the extensive use of virtualization and re-direction is severe enough that many customers who need high assurance and forensics capabilities have moved away from the concept of cloud computing [12 - 14]. We believe, however, that a precise statement of the high assurance and forensics requirements will lend themselves to solutions in the cloud computing environment, and expand the potential use of this technology. This work is part of a body of work for high assurance enterprise computing using web services. Elements of this work are described in [15-21].

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

All Records:
Record Number, Thread Number, Sequence Number, Active Entity Name, Event Name, ID of Service Requestor, Date/time, Other pertinent data (OP_i)

Event	Content
Start-up of the audit functions within the service	Event Name = " Start Up Audit", OP1= ID of Service Requestor, OP2 = Level of Audit
Shutdown of the audit functions within the service	Event Name = "Shut Down Audit", OP1= ID of Service Requestor
session initiation activities	Event Name = "Session Initiated", OP1= ID of Service Requestor
Session complete	Event Name = "Session Complete", OP1= ID of Service Requestor
Any exceptions due to Java or supporting programs	Event Name = "Exception", OP1= ID of Service Requestor, OP2 = Exception Message, Return to Next level Server Problem – call help desk (Sequence Number =SN)
Any known violations of security policy,	Event Name = "Session Complete", OP1= ID of Service Requestor , Return to Next level Server Problem – call help desk (Sequence Number =SN)
Authorization	Event Name = "Authorization", OP1 = ID of Service Requestor, OP2 = SAML, OP3 = Open Token
Authentication	Event Name = Authentication, OP1= ID of Service Requestor, OP2 = Requestor Cert DN , OP3 = Requestor Cert Public Key, OP4 = Requestor Cert Revocation Date, OP5 = Result of Cert Validation Check
Thread Establishment	Event Name = Thread Established, OP1= ID of Service Requestor, Op2 = Complete API Input
Service Request – Each	Event Name = Service Request, OP1= ID of Service Requestor, Op2 = Complete API Request
Service Response – Each	Event Name = "Service Response", OP1= ID of the Service Requestor, Op2 = Complete API return
	Health State Performance Monitors
Final Rollout Performance Data A program segment could be computational or waiting for inputs, etc. Timeouts and work around Timeouts may or may not occur	Event Name = "Performance", OP1= Program Segment Name 1 plus delta t1, OP2 = Program Segment Name 2 plus delta t2, OP3 = Program Segment Name 3 plus delta t3, OP4 = Program Segment Name 4 plus delta t4, OP5 = Program Segment Name 5 plus delta t5 Event Name = "Timeouts", OP1 = Timeout occurrence location, OP2 = Timeout delta t, OP3 = Workaround (e.g., cache data, default value, etc.)

Fig. 4. Monitoring Records

REFERENCES

[1] Peter Mell, Timothy Grance, NIST SP 800-145 Draft: Cloud Computing, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, January 2011, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

[2] Wayne Jansen, Timothy Grance, NIST SP 800-144 Draft: Guidelines on Security and Privacy in Public Cloud Computing, Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, January 2011, http://csrc.nist.gov/publications/drafts/800-144/Draft-SP-800-144_cloud-computing.pdf

[3] Daniele Catteddu and Giles Hogben, European Network Information Security Agency (ENISA), Cloud Computing Risk Assessment, November 2009, http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment

[4] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, December 2009, https://cloudsecurityalliance.org/csaguide.pdf

[5] Air Force Information Assurance Strategy Team, *Air Force Information Assurance Enterprise Architecture*, Version 1.70, SAF/XC, 15 March 2009. [Not available to all]

[6] *Windows Server 2003: Active Directory Infrastructure*. Microsoft Press. 2003. pp. 1–8 to 1–9. ISBN: 0-7356-1438-5

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

[8] OASIS 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.

[9] "Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0", Microsoft Corporation, 2005.

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

[11] SearchSecurity, Security Resources, computer forensics (cyber forensics), http://searchsecurity.techtarget.com/definition/computer-forensics

[12] Remarks-Debra Chrapaty, Corporate Vice President, Global Foundation Services, Microsoft Mgt Summit, Las Vegas, May 2008, http://www.microsoft.com/Presspass/exec/debrac/mms2008.mspx (accessed 19 February 2011).

[13] Bobbie Johnson, technology correspondent, guardian.co.uk, Cloud computing is a trap, warns GNU founder Richard Stallman, 29 September 2008, http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman (accessed 19 February 2011).

[14] Andy Plesser, Executive Producer, Beet.tv, Cloud Computing is Hyped and Overblown, Forrester's Frank Gillett.....Big Tech Companies Have "Cloud Envy", http://www.beet.tv/2008/09/cloud-computing.html, September 26, 2008 (Accessed on 19 February 2011).

[15] 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, FL., June 2008.

[16] 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.

[17] 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, FL., July 2009.

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

[19] 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, page 675-685..

[20] 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.

[21] 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.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)