OnTimeSecure: Secure Middleware for Federated Network Performance Monitoring

Prasad Calyam, Shweta Kulkarni, Alex Berryman, Kunpeng Zhu,
Mukundan Sridharan, Rajiv Ramnath, Gordon Springer
University of Missouri-Columbia, The Ohio State University, The Samraksh Company, USA;
Email: {*calyamp, gks*}*@missouri.edu*; {*kulkarni.85*, *berryman.15*, *zhu.249*, *ramnath.6*}*@osu.edu*; *mukundan.sridharan@samraksh.com*

***Abstract*—Multi-domain network monitoring systems based on active measurements are being widely deployed in high-performance computing and other communities that support large-scale data transfers. Security mechanisms such as policy-driven access to related federated Network Performance Monitoring (NPM) services are important to protect measurement resources and data. In this paper, we present a novel, secure middleware framework viz., "OnTimeSecure" that enables 'user-to-service' and 'service-to-service' authentication, and enforces federated authorization entitlement policies for timely orchestration of NPM services. OnTimeSecure is built using RESTful APIs and features a hierarchical policy-engine that interfaces with a meta-scheduler for prioritization of measurement requests when there is contention of users concurrently attempting to utilize measurement resources. We validate OnTimeSecure in a federated multi-domain NPM infrastructure by performing threat modeling and security risk assessments based on overall attack likelihood and impact factors.**

***Keywords*-multi-domain measurements, secure middleware; federated identity; entitlement service; enterprise access policy**

I. Introduction

Multi-domain network performance monitoring (NPM) systems based on active measurements using tools such as Ping, Traceroute, OWAMP and BWCTL [1] are being widely deployed in academia and industry. Data-intensive science and Big Data analytics demands for data transfers across small-to-large enterprises are driving the need for NPM systems. NPM frameworks can enable creation of "measurement federations" for collection and sharing of end-to-end performance measurements across administrative domains over the Internet. Collected measurements can be queried amongst federation members through interoperable web-service interfaces to mainly analyze network paths to ensure packet loss free paths and identify end-to-end bottlenecks. They can also help in diagnosing performance bottlenecks using anomaly detection [2], determining the optimal network path [3], or in network weather forecasting [4].

Examples of measurement federations that have recently evolved include perfSONAR [1], SamKnows [5] and M-Lab [6]. Measurement federation related standards-development efforts are on-going at Open Grid Forum (OGF), IETF IP Performance Metrics (IPPM), IEEE 802.1 ag, ITU-T Y.1731, and Metro Ethernet Forum (MEF) to foster interoperability and sustainability of measurement federations. Amongst the NPM frameworks, perfSONAR is a widely instrumented framework; based on the latest reports, there are ≈600 perfSONAR measurement points worldwide.

However, deploying the current implementations of perfSONAR into multi-domain measurement federations of enterprise networks requires establishment and enforcement of "measurement level agreements" [7] (MLAs), and also raises a number of security concerns. Measurement level agreements can be enforced for cross-domain measurement data collection and sharing through an appropriate "Resource Protection Service" to address measurement federation policies. In addition, there is a need to secure the perfSONAR measurement infrastructure for thwarting cyber attacks and for security audit compliance in enterprise networks. The cyber attacks could involve malicious take over of measurement points to launch DDoS attacks that consume vast amounts of bandwidth and disrupt services on federation related and other networks.

To avoid having barriers of wide-adoption (e.g., overheads in setting up federations and MLAs), the current recommendations and trends in implementation of perfSONAR are to have it in a default "totally open" mode to run tests on measurement points and view data within measurement archives. In this mode, the measurement points and data archives can be discovered by anonymous users through the Global Lookup Service [8] and there is minimal regulation imposed by restricting maximum probing bandwidth utilization for active measurement tools such as Ping, Traceroute, OWAMP and BWCTL. This totally open mode limits the security options for an enterprise and may result in undesired measurement tests and data shares within the domain's measurement infrastructure resources.

Consequently, enterprises that are concerned about open access to measurement resources resort to the other extreme option of "strictly closed" mode. In this mode, measurement points are not registered with the Global Lookup Service, and tests can only be scheduled between measurement points within the firewall by select intra-domain users and no measurement archive data is publicly accessible. Although this mode provides increased security, it limits cross-domain performance measurement collection and sharing that are essential for end-to-end monitoring and troubleshooting bottlenecks over the Internet.

In this paper, we address this gap and present our novel secure middleware viz., OnTimeSecure for perfSONAR based measurement federations. Owing to its architecture design, it enables perfSONAR to be integrated with popular federated authentication and authorization frameworks (e.g., Shibboleth-based [9]) and provides a *middle-ground* between "totally open" and "strictly closed" modes. The middle-ground approach seeks to effectively allow multi-domain monitoring while protecting and selectively restricting access to institutional measurement resources based upon intra-domain/inter-domain federation policies in measurement level agreements. OnTimeSecure features a hierarchical policy-engine and is

This material is based upon work supported by the Department of Energy under Award Numbers: DE-SC0001331 and DE-SC0007531 and the National Science Foundation under Award Numbers: ACI-1246001 and ACI-1245795. The views and opinions of authors expressed herein do not necessarily state or reflect those of the united states government or any agency thereof.

built using RESTful APIs [10] that are modular for extensibility, and are interoperable with perfSONAR standards based deployments. It uses API key authentication for various measurement functions such as: measurement point discovery, test initiation and measurement data query. The policy-engine also interfaces with a meta-scheduler we have developed in prior works [7] [11] for prioritization of measurement requests and conflict-free scheduling, while users concurrently attempt to utilize measurement resources.

We validate our OnTimeSecure middleware security robustness through an exemplar case study implementation of OnTimeSecure on a measurement federation testbed comprising of the following institutions: The Ohio State University, Ohio Technology Consortium and the University of Missouri. We configure perfSONAR in the “totally open” mode and in the “resource protected” mode through an ‘Entitlement Service’ within the testbed and perform a threat modeling and security risk assessment study based on overall attack likelihood and impact factors, following the National Institute of Standards and Technology (NIST) method [15] guidelines.

The remainder of this paper is organized as follows: Section II details the OnTimeSecure framework requirements and architecture considerations. Section III describes the user-to-service design and capabilities. Section IV describes service-to-service design and capabilities. Section V describes an exemplar case study implementation of OnTimeSecure on a federated testbed for security robustness validation. Section VI concludes the paper.

II. OnTimeSecure Requirements and Architecture

Figure 1 shows our perfSONAR NPM deployment envisioned within a content-delivery network enterprise (say Domain A). The deployment consists of a Central Intelligence System (CIS) that runs the broker service, which discovers, manages and controls a number of strategically placed Measurement Point Appliances (MPAs) in the core and at edges. The MPAs act as measurement end-points that host active measurement tools for end-to-end metrics (e.g., one-way delay, round-trip delay, jitter, loss, TCP/UDP throughput) and also can interface with other enterprise-related performance metric sources of system (e.g., encoder CPU, Memory), network (e.g., TCPdump) or application (e.g., Video Frame Rate).

The NPM system needs to support a large number of users’ monitoring objectives while also being compliant with federation policies for measurement resource access, and more importantly - being secure against cyber-attacks. The monitoring objectives differ depending upon enterprise user “Roles”. For example, a ‘Network Operator’ might want to schedule measurement tests for routine network-wide monitoring. Alternately, a ‘Power User’ of the network who regularly transfers large data sets over wide-area would want to be notified if there are any anomalies in network performance impacting data transfer throughputs. Further, a neighboring domain’s (say Domain B) ‘Regular User’ might want to schedule a measurement test to diagnose a performance bottleneck along an end-to-end network path traversing Domain A administered links (assuming Domain A and B are already part of a measurement federation i.e., they have shared each others’ measurement infrastructure topologies and have agreed user identity sharing policies).

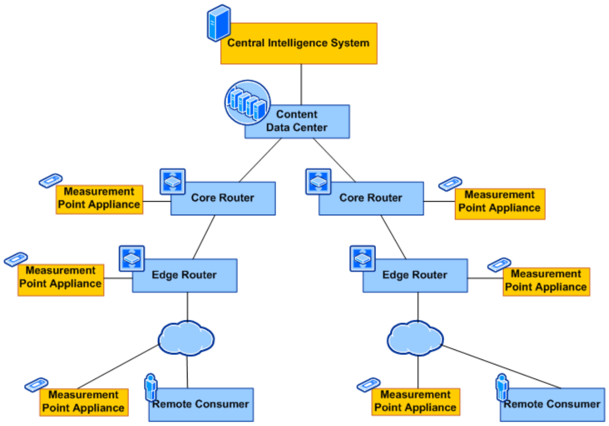
Central Intelligence System
Content Data Center
Measurement Point Appliance
Core Router
Core Router
Measurement Point Appliance
Measurement Point Appliance
Edge Router
Edge Router
Measurement Point Appliance
Measurement Point Appliance
Remote Consumer
Measurement Point Appliance
Remote Consumer

Fig. 1. Intra-domain perfSONAR deployment within a content-delivery network

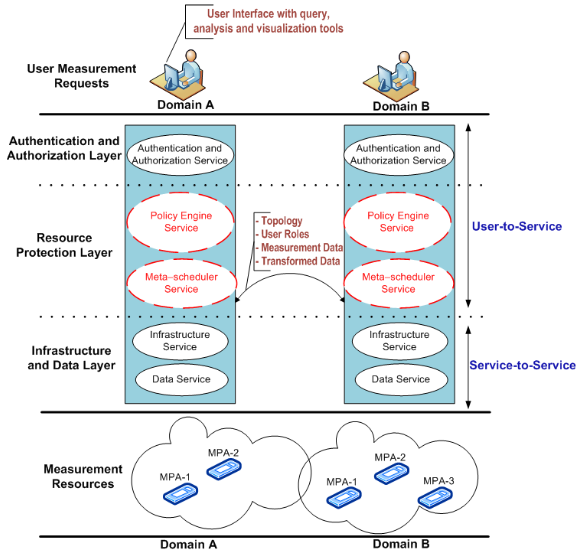

Fig. 2. OnTimeSecure layer extensions within the perfSONAR service-oriented architecture

In all of the above use cases, the NPM system (through the CIS) has to enforce the conflict-free scheduling and measurement level agreements and ensure security at the level of basic measurement functions such as: discover a measurement point, add a performance test to schedule upon user request, push the schedule to a measurement point, collect measurements into measurement archives, query the measurement archives for intra-domain performance analysis/visualization transformations, query measurement archive to share inter-domain performance reports, or notify intra-domain user of an anomaly event. Figure 2 shows our OnTimeSecure layer extensions for perfSONAR deployments to meet the ‘user-to-service’ and ‘service-to-service’ security requirements.

III. User-to-Service Design and Capabilities

A. Authentication and Authorization Layer

In order to determine whether a user is legitimate (authenticate) and thereafter grant appropriate privileges to access various measurement functions orchestrated by the CIS (authorize), the CIS interfaces with identity provider(s) (IdPs) and an “Entitlement Service” belonging to a measurement federation as shown in Figure 3. Our current implementation of OnTimeSecure interfaces the CIS with Shibboleth-based federations because it provides a solid platform for extensibility and wide adoption i.e., Shibboleth infrastructures are based upon Security Assertion Markup Language (SAML) [9], which

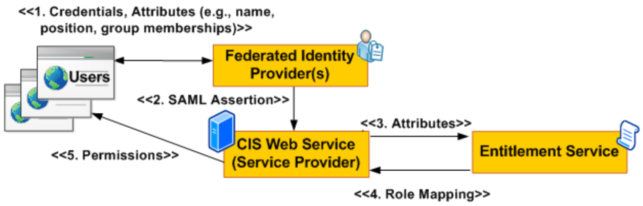
<<1. Credentials, Attributes (e.g., name, position, group memberships)>>
Users
Federated Identity Provider(s)
<<2. SAML Assertion>>
<<3. Attributes>>
CIS Web Service (Service Provider)
Entitlement Service
<<5. Permissions>>
<<4. Role Mapping>>

Fig. 3. Workflow of Federated authentication and authorization within OnTimeSecure

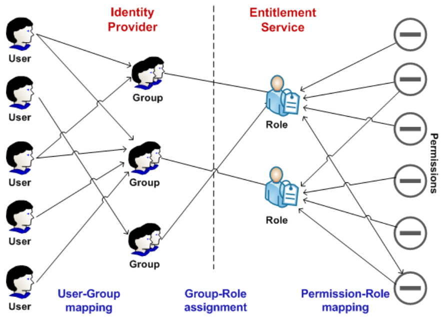

Fig. 4. Policy-engine design for user, group permission mapping

is an open-standard already being used within perfSONAR communities in academia and industry for exchanging authentication and authorization information in other federation use cases [12] [13]. Communities that use Shibboleth either have their own home-institution identity management system or rely on one or more external IdPs (e.g., OpenID). The CIS acts as a service provider (SP) for the perfSONAR users, and consumes the information provided by the IdPs and interfaces with a federation's separate Entitlement Service (ES) to grant user access to the various measurement functions.

B. Resource Protection Layer

The policy engine service serves measurement federation users and maintains information of "Groups" to allow subsequent policy mapping to "Roles" and "Permissions" as shown in Figure 4. Users can be a part of one or more Groups, and only certain Groups are permitted to have specific Roles (e.g., Network Operator, Power User, Regular User) that generally depend on institution-level functional status of users (e.g., student, faculty, engineering staff). Lastly, Roles can be assigned priority levels that are useful when there is contention due to multi-domain users concurrently attempting to utilize measurement resources in an intra-domain or inter-domain manner. The priority levels can be set to prioritize measurement requests based upon User Role (e.g., Network Operator Role users always have higher priority than Power Users). If the Permissions are for query or transformation of measurement data within archives, data retrieval and transformation services (e.g., anomaly detection service [14]) are invoked. However, if the Permissions are for initiation of new intra-domain or inter-domain measurement tests, then the policy engine directs the policy mapping output to a meta-scheduler service developed in our prior works [7] [11].

IV. Service-to-Service Design and Capabilities

In comparison to the user-to-service capabilities that deal with services directly interacting with users, the service-to-service design and capabilities in OnTimeSecure deal with mutual authentication between services that autonomously orchestrate with each other based solely on the identity of the service. Our service-to-service authentication scheme in OnTimeSecure is designed to provide a level of confidentiality in message exchanges between the NPM services. We remark that message authentication in current design of "totally open" perfSONAR mode does not exist due to the fact that every measurement test is scheduled individually by a user using a static web user interface (UI) on the MPA, and NPM services are not centrally managed as in the case of the CIS in the "resource protected" perfSONAR mode. In addition to confidentiality, there is a need to verify the authenticity of the REST requests being exchanged between the services using a two-way authentication scheme. The two-way authentication scheme ensures that an MPA will only accept a measurement schedule from a known CIS, and in return a CIS will only accept measurement results from a known MPA.

In the following, we present our service-to-service discovery scheme for initial pairing of the CIS and MPAs in which keys are securely exchanged, and are subsequently used to create digital signatures for confidentiality and authenticity of message exchanges between the services. The MPA Discovery involving pairing of the CIS and the MPAs is initiated through a web UI on the CIS such that only users with appropriate Permissions granted by appropriate Role assignments can add and configure new MPAs. The protocol followed in the MPA Discovery safeguards against attacks based on the introduction of compromised components (i.e., malicious MPAs) into the system. Figure 5 shows the MPA discovery protocol. The MPAs to be added are deployed at strategic points within the network and are set as being "discoverable". Following this, an authorized user can initiate the MPA Discovery protocol through the CIS web UI. The CIS holds a valid SSL certificate, allowing the CIS's identity to be verified by the MPAs using the normal X.509 certificate chaining model. The MPAs use a self-signed SSL certificate and a public RSA key to define their initial identity to the CIS. On initiation of a subnet scan from the CIS, an MPA responds with an AES Encrypted bundle containing its public key, and other pertinent state information (e.g., NAT status, software version, measurement functions supported). The CIS decrypts the bundle and presents the information to the requesting user for verification. After user approval of the MPA's identity, the CIS generates an encrypted bundle containing the 'APIKey'/'APISecret' pair that is sent and stored securely in the MPA using access control lists (ACLs) provided by the Linux file system in order to limit the readability of the key pair to the service account.

On completion of the above pairing of an MPA, the CIS would have shared a unique key pair consisting of an APIKey and an APISecret that represents the service identity of the newly added MPA. Every ensuing message and request sent between the MPA and CIS is digitally signed using this APISecret. The CIS stores the APISecret for all MPAs indexed by their corresponding APIKeys. The key pair is stored in plain text unlike user passwords and relies on typical permission schemes for protection; such a non-encrypted storage is a requirement for CIS operation, and by design, the key pairs are cheaply replaceable. The keys for each domain can be siloed into individual databases to limit information leakage in the event that a single key store is compromised. The APIKey element forms a part of the authentication header for a message sent by a service. On receiving a message from the MPA, the CIS retrieves the APISecret for the MPA using the APIKey and verifies the digital signature.

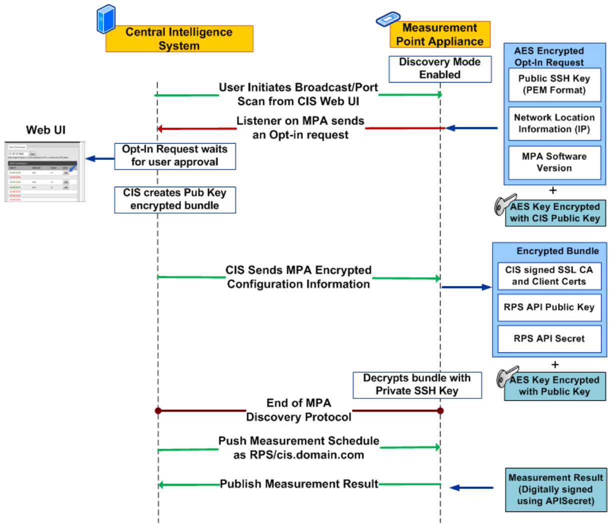
Central Intelligence System
Measurement Point Appliance
Discovery Mode Enabled
AES Encrypted Opt-In Request
Public SSH Key (PEM Format)
Network Location Information (IP)
MPA Software Version
+
AES Key Encrypted with CIS Public Key
User Initiates Broadcast/Port Scan from CIS Web UI
Listener on MPA sends an Opt-in request
Web UI
Opt-In Request waits for user approval
CIS creates Pub Key encrypted bundle
CIS Sends MPA Encrypted Configuration Information
Encrypted Bundle
CIS signed SSL CA and Client Certs
RPS API Public Key
RPS API Secret
+
AES Key Encrypted with Public Key
Decrypts bundle with Private SSH Key
End of MPA Discovery Protocol
Push Measurement Schedule as RPS/cis.domain.com
Publish Measurement Result
Measurement Result (Digitally signed using APISecret)

Fig. 5. MPA discovery protocol flow diagram

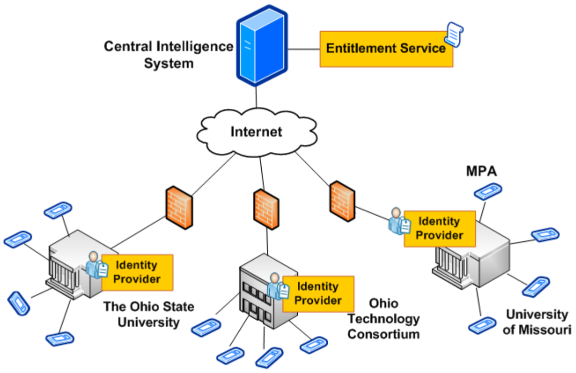

Fig. 6. Multi-domain NPM testbed for OnTimeSecure validation

V. OnTimeSecure Validation Case Study

A. Testbed Setup and Implementation

We setup an implementation of OnTimeSecure on a measurement federation testbed shown in Figure 6 comprising of the following institutions: The Ohio State University (OSU), Ohio Technology Consortium (OH-TECH) and the University of Missouri (MU). Each institution had 4 MPAs deployed at strategic points within their domains, and registered their MPAs to a common CIS using the secure MPA discovery protocol described earlier. All three institutions are part of the Internet2 InCommon Federation, which is the identity management federation for the US research and education community. More specifically, we registered OnTimeSecure as a certified service provider (there are only 9 other certified entities for 'Research and Scholarship' within Internet2 InCommon), and thus have made our implementation accessible to over 350 InCommon members in higher education, government, industry and research centers.

B. Threat Modeling and Security Assessment

To compare the security robustness of "totally open" perfSONAR (Open-pS) and "resource protected" perfSONAR (RPS-pS) configurations, we followed the National Institute of Standards and Technology (NIST) method [15] for conducting risk assessments. The risk calculation from a threat event is shown in Figure 7, and involves the following steps: (i) Assess the likelihood of threat occurrence on basis of probability of initiation and success, (ii) Assess the level of impact in event of a successful attack, and (iii) Calculate overall risk score as a combination of the likelihood and impact.

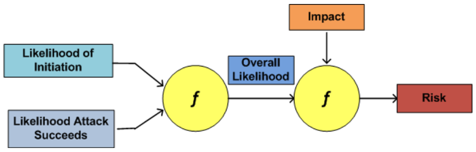

Fig. 7. Risk calculation using NIST method

We identified 59 possible threat events from the NIST guidelines that were potential security risks to an institution or measurement federation deploying a multi-domain NPM framework. Both, RPS-pS and Open-pS configurations were assessed on a semi-quantitative scale of 0-10 with 10 indicating high impact/likelihood, and the assessment results comparison is shown in Table I. We can see that RPS-pS successfully mitigates the risk from threat events by completely eliminating 'High' risks from potential threat events, while the number of threat events that could pose 'Moderate' risk is halved as compared to Open-pS. In addition, RPS-pS successfully limits risk to 'Low' risk scores for about 85% of possible threats. We remark that a "totally closed" perfSONAR mode that is restrictive and limits collaborative measurement (i.e., it only permits intra-domain measurement tests and does not register with the perfSONAR Global Lookup Service) can lower the number of threats posing moderate risk. However, there will always be threat events of 'Low' risk such as those caused by for e.g., insider-based threat events in the threat-model that can make even the "totally closed" perfSONAR mode deployment potentially vulnerable to cyber-attacks.

TABLE I
Risk assessment results showing perfSONAR security robustness improvement with OnTimeSecure

Risk Score	Open-pS	RPS-pS
Low	63%	85%
Moderate	32%	15%
High	5%	0%

VI. Conclusion

In this paper, we address an important problem of resource protection in multi-domain NPM deployments that has not been long-solved, especially in the perfSONAR community that has over 600 measurement point instances worldwide as part of explicit measurement federations. Our solution involves the "OnTimeSecure" middleware that enables 'user-to-service' and 'service-to-service' authentication, and enforces federated authorization entitlement policies for timely orchestration of NPM services. The novelty in OnTimeSecure is in its use of RESTful APIs and a hierarchical policy engine that interfaces with a meta-scheduler for prioritization of measurement requests when there is contention of users concurrently attempt to utilize measurement resources. Owing to its design, OnTimeSecure can be easily integrated within enterprises deploying perfSONAR and other NPM frameworks; it has the potential to transform how enterprises engage in multi-domain monitoring, while protecting and selectively restricting access to institutional measurement resources based upon intra-domain/inter-domain federation policies.

REFERENCES

[1] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, M. Swany, S. Trocha, J. Zurawski, "perfSONAR: A Service Oriented Architecture for Multi-Domain Network Monitoring", *Proc. of Service Oriented Computing, Springer Verlag, LNCS 3826*, pp. 241-254, 2005. (http://www.perfSONAR.net)

[2] P. Calyam, J. Pu, W. Mandrawa, A. Krishnamurthy, "OnTimeDetect: Dynamic Network Anomaly Notification in perfSONAR Deployments", *Proc. of IEEE/ACM MASCOTS*, 2010.

[3] S. Tao, K. Xu, A. Estepa, et. al., "Improving VoIP Quality through Path Switching", *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM)*, Vol. 4, pp. 2268-2278, 2005.

[4] B. Gaidioz, R. Wolski, B. Tourancheau, "Synchronizing Network Probes to avoid Measurement Intrusiveness with the Network Weather Service", *Proceedings of the 9th International Symposium on High-performance Distributed Computing Conference*, pp. 147-154, 2000.

[5] SamKnows - Worldwide Broadband Measurement Service - http://www.samknows.com

[6] M-Lab - Open Platform for Internet Measurement Tools Community - http://www.measurementlab.net

[7] P. Calyam, C.-G. Lee, E. Ekici, M. Haffner, N. Howes, "Orchestrating Network-wide Active Measurements for Supporting Distributed Computing Applications", *IEEE Transactions on Computers*, Vol. 56, No. 12, pp. 1629-1642, 2007.

[8] J. Zurawski, J. Boote, et. al., "Hierarchically Federated Registration and Lookup within the perfSONAR Framework", *Proc. of IFIP/IEEE Integrated Management Symposium*, 2007.

[9] R. Morgan, S. Cantor, et. al., "Federated Security: The Shibboleth Approach", *EDUCAUSE Quarterly*, Vol. 27 No.4, pp. 4-6, 2004.

[10] M. Masse, "REST API Design Rulebook", *O'Reilly Media ISBN: 978-1-4493-1050-9*, 2011.

[11] P. Calyam, L. Kumarasamy, F. Ozguner, "Semantic Scheduling of Active Measurements for meeting Network Monitoring Objectives", *Proceedings of International Conference on Network and Service Management (IEEE CNSM) (Short Paper)*, pp. 435-438, 2010.

[12] T. Barton, J. Basney, et. al., "Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy", *Proc. of PKI Research and Development Workshop*, 2006.

[13] W.Pugh, K.Austin, "Identity, Access Control, and VMware Horizon", *VMware Technical Journal*, 2012.

[14] M. Grigoriev, P. Demar, D. Eads, B. Tierney, J. Metzger, A. Lake, M. Frey, P. Calyam, "E-Center: Collaborative Platform for the Wide Area Network Users", *Proc. of Conferences on Computing in High Energy and Nuclear Physics (CHEP)*, 2012.

[15] R. Ross, "Guide for Conducting Risk Assessments", *NIST SP800-30-Rev1*, 2012.