NETWORK AND MIDDLEWARE SECURITY FOR ENTERPRISE NETWORK MONITORING

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By
Aravind Gopalakrishnan, B. Tech.
Graduate Program in Computer Science and Engineering

The Ohio State University
2012

Dissertation Committee:
Dr. Rajiv Ramnath, Advisor
Dr. Prasad Calyam, Co-Advisor
Dr. Jay Ramanathan
© Copyright by
Arunvind Gopalakrishnan
2012
ABSTRACT

Over the last decade or so, multi-domain network monitoring systems frameworks such as perfSONAR have been widely deployed in high-performance computing and other communities that support large-scale data movements. These frameworks allow end-to-end monitoring across domains such that performance measurements can be queried through web interfaces for interested parties to analyze the network paths for the purposes of detecting anomaly events and diagnosis of faults. perfSONAR, a web-services based infrastructure for collecting and publishing network performance monitoring data sets has made it easy for networking communities to solve end-to-end performance problems on paths crossing several networks and it has emerged as a popular network performance monitoring tool within organizations both in US as well as for communities within the European research networks. It consists of a set of services delivering performance measurements in a federated environment.

The current implementation of the perfSONAR services when hosted in an enterprise network environment in the form of E-perfSONAR measurement point appliance controlled by a central intelligence system has security limitations. It also does not support policy-based access to measurement data sets. The critical issues to be addressed are -

○ Ensure that the web services instrumented into the Central Intelligence system and the measurement point appliance are robust against cyber-attacks.
Allow a ‘selectively-open’ nature of view to the measurement data that are collected by the measurement points and also a policy-driven approach to scheduling of the services.

Therefore, there is a need to investigate network security issues and a need to support a policy-driven approach for Middle-ware security in E-perfSONAR deployments. The primary aim of the thesis is to identify and make steps towards resolution of network security problems of perfSONAR and also in coming up with an authentication and authorization policy that can be built into the E-perfSONAR framework. We refer to our solution as the ‘Resource protection service’ within the E-perfSONAR framework.

Looking at existing frameworks such as Globus toolkit and integration of authentication and authorization features in perfSONAR-MDM, a flavor of perfSONAR brought to us by GEANT, we propose a ‘Federated’ and a ‘Peer-to-Peer’ model of security that leverages the features of existing industry standards such as LDAP, Kerberos and Shibboleth technologies to perform authentication mechanism that fits E-perfSONAR architecture and also leverage the open, modular and distributed features of perfSONAR.

The suggestions made in this thesis towards the design of middleware security features can be utilized for the E-perfSONAR system to bring about a policy-driven approach to authentication, authorization and scheduling of the network monitoring services in a multi-domain environment. The results obtained from the thesis with respect to vulnerability testing of the web services can be extended to ensure that E-perfSONAR appliance will be a secure point of communication when deployed over an enterprise network.
Dedicated to Ma, Pa and Arjun.
ACKNOWLEDGMENTS

Firstly, I would like to thank Dr. Prasad Calyam and Dr. Rajiv Ramnath for giving me an opportunity to engage in the research work at the Ohio Supercomputer Center. I am very thankful to Dr. Prasad Calyam for giving me valuable inputs at various points of the project and for guiding me throughout the course of this thesis work. I also appreciate the inputs given to me by Dr. Rajiv Ramnath during the CETI weekly meetings and presentations. I would like to thank Dr. Jay Ramnathan for consenting to be on the thesis defense committee.

I would also like to thank Alex Berryman for his valuable assistance in brainstorming the ideas for the security architecture and his help in vulnerability testing of the E-perfSONAR equipment. I would like to thank Manojprasadh Dhanapalan and Shriram Ramamurthy for their assistance in integrating Resource Protection Service in the perfSONAR framework and Saravanan Mohan for his help in setting up a test environment on which we could deploy perfSONAR instances and test our ideas.

Finally, I would like to thank my parents (K S Gopalakrishnan and G Narmada Gouri), my brother (Arjun), my roommates (Manoj, Venmugil, Nitin, Saktheesh, Ashok), my sister (Aishwarya) and my friends (Christopher, Sinduja, Raghuram Onti, Kruthika, Rohit Pata, Kripa, Roopa) for being a constant source of support and motivation and making the last couple of years an unforgettable experience. A special note of thanks to Praveen, Padmini, Hanu and Amit Modi for helping me throughout my time here at OSU.
VITA

2010 ............................ B. Tech. Information Technology, Anna University, Chennai, India

2010-Present ..................... M.S. Computer Science, The Ohio State University, Columbus, USA

FIELDS OF STUDY

Major Field: Computer Science and Engineering
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td></td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td></td>
<td>ix</td>
</tr>
<tr>
<td>List of Figures</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Multi-domain Measurement Framework</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>perfSONAR</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1</td>
<td>About</td>
<td>2</td>
</tr>
<tr>
<td>1.2.2</td>
<td>perfSONAR Tools</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>E-perfSONAR</td>
<td>6</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Background</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Motivation</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>Problem Description</td>
<td>10</td>
</tr>
<tr>
<td>1.5.1</td>
<td>Execute Access control</td>
<td>10</td>
</tr>
<tr>
<td>1.5.2</td>
<td>Read Access Control</td>
<td>11</td>
</tr>
<tr>
<td>1.5.3</td>
<td>Write Access Control</td>
<td>11</td>
</tr>
<tr>
<td>1.5.4</td>
<td>Network Security Issues</td>
<td>11</td>
</tr>
<tr>
<td>1.6</td>
<td>Thesis Contributions</td>
<td>12</td>
</tr>
<tr>
<td>1.7</td>
<td>Related Work</td>
<td>13</td>
</tr>
<tr>
<td>1.8</td>
<td>Organization of this thesis</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Security Architecture</td>
<td>16</td>
</tr>
<tr>
<td>2.1</td>
<td>Middleware Security</td>
<td>16</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Federated Security Model</td>
<td>16</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>2.1.2 Peer-to-Peer Security Model</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>2.1.3 Authorization Model</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>3 Network Security</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>4 Experimental Evaluation</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>4.1 Metascheduler Implementation</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>4.2 Vulnerability Testing</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>5 Conclusion and Future Work</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Bibliography</td>
<td>52</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 OpenVAS test results based on CVSS scores</td>
<td>47</td>
</tr>
<tr>
<td>4.2 Ports that run vulnerable services</td>
<td>49</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 perfSONAR services architecture</td>
<td>3</td>
</tr>
<tr>
<td>1.2 E-perfSONAR vision</td>
<td>7</td>
</tr>
<tr>
<td>1.3 E-perfSONAR example federation structure</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Federated network structure</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Inter-domain authentication process</td>
<td>19</td>
</tr>
<tr>
<td>2.3 Shibboleth message flow</td>
<td>22</td>
</tr>
<tr>
<td>2.4 Peer-to-Peer authentication process</td>
<td>24</td>
</tr>
<tr>
<td>2.5 Kerberos message flow</td>
<td>27</td>
</tr>
<tr>
<td>2.6 Policy ontology tree</td>
<td>30</td>
</tr>
<tr>
<td>2.7 User ontology tree</td>
<td>31</td>
</tr>
<tr>
<td>2.8 Components of Resource Protection Service</td>
<td>33</td>
</tr>
<tr>
<td>2.9 Authorization mechanism in federated environment</td>
<td>35</td>
</tr>
<tr>
<td>2.10 Authorization mechanism in P2P environment</td>
<td>36</td>
</tr>
<tr>
<td>4.1 Current star topology in perfSONAR</td>
<td>42</td>
</tr>
<tr>
<td>4.2 Proposed topology in E-perfSONAR</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Metascheduler work-flow (Step 1)</td>
<td>44</td>
</tr>
<tr>
<td>4.4 Metascheduler work-flow (Step 2)</td>
<td>44</td>
</tr>
<tr>
<td>4.5 perfSONAR threats based on CVSS score</td>
<td>48</td>
</tr>
<tr>
<td>4.6 Threats in perfSONAR vs Threats in E-perfSONAR</td>
<td>48</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Multi-domain Measurement Framework

The Internet has become a vastly complex heterogeneous system that makes it hard for network operators, who utilize the data sets to perform operations such as analysis for anomaly detection and network weather forecasting and for anonymous users to view the data. Due to the extensive research work, several sub-communities have evolved that participate in a community-oriented measurement infrastructure. This has led to the development of Network monitoring infrastructures such as Nagios [10] and perfSONAR. perfSONAR provides a distributed, service-oriented architecture towards a global measurement framework. It is based on the decoupling of measurement services provided by network monitoring tools and differentiating measurement points along the network as described in [20]. Further abstraction layers have been added that provides an archive of collected data. This service-oriented architecture is detailed in [4] where the perfSONAR instances are deployed across LHC Tier1 and Tier 2 network backbones.
1.2 perfSONAR

1.2.1 About

perfSONAR is a joint project started by several R&E networks and interested partners to build a framework that enables network performance information to be gathered and exchanged in a inter-domain, federated environment. Consequently, perfSONAR enables gathering and sharing of this performance information to simplify management of advanced networks and facilitate cross-domain troubleshooting.

perfSONAR has several use cases such as collection and publication of latency data, collection and publication of achievable bandwidth results, publication of utilization data, publication of network topology data, diagnosing performance issues, and several others. perfSONAR has evolved over the years and currently supports SNMP monitoring as well and provides graphing interface for the data collected in the back-end interface. [11]

perfSONAR has found widespread acceptance in academic and enterprise network monitoring communities due to use cases such as - data transfer middleware solutions that can use perfSONAR to find a copy of a file, or to assist in determining the optimal network protocol to use for a given link. Network engineers could use perfSONAR to identify bulk data flow, policing of data flow by identifying packets as high/low priority, depending on its source or destination. Lastly, researchers find perfSONAR-enabled networks a convenient source of performance and topology information upon which further analysis can be performed to obtain intelligent data that provides insight into the issues present within the network. Some examples of the analysis that can be performed are anomaly detection and network weather forecasting. [20]

Figure 1.1 illustrates the design of current perfSONAR toolkit instance where
scheduling of services and viewing of data collected by running the services are accessed through a front-end GUI interface and data is maintained at a Measurement Archive (typically a SQL database) in the back-end.

Figure 1.1: perfSONAR services architecture
1.2.2 perfSONAR Tools

The perfSONAR-PS toolkit used in our experiments has three distinct features -

- Measurement Tools
- Diagnostic Tools
- Look-up Services

The perfSONAR toolkit consists of several measurement tools that facilitate active and passive measurements and it is packaged as a LiveCD distributable that uses the Knoppix Linux Distribution, a flavour of CentOS [17]. perfSONAR-BUOY is a service that encapsulates two other perfSONAR tools - OWAMP and BWCTL. We can define sets of latency and throughput tests to end measurement points for archiving and publishing. The web-service provided by the toolkit utilizes the Lookup Service to find other nodes capable of performing OWAMP and BWCTL tests and it allows for the users to set their own measurement set attributes. perfSONAR-BUOY exposes the data collected using a perfSONAR Measurement Archive (MA) and a graphical interface that pulls data from the back-end and furnishes it in the form of graphs to the end user.

**OWAMP** [18] is an implementation of OWAMP protocol [22] and is used to run active tests to collect one-way latency, loss and delay variation. One-way latency is useful for identifying and isolating the direction of performance issues and can also be used to look for routing problems as well as interface queuing.

**BWCTL** [14] is a tool that wraps around the popular Iperf [16] throughput testing tool. It allows ad-hoc throughput tests to occur on the same host as regular measurements without worrying about the issues of overlapping tests and garbled results. The perfSONAR toolkit GUI allows users to specify the periodicity and duration values of the tool instance and if TCP or UDP protocols are to be used.
**SNMP** Passive interface statistics that are delivered via SNMP [32], is a common indication of the network’s health. Metrics such as utilization and errors are particularly important when detecting performance and related problems. The pSNP Toolkit incorporates a Cacti [15] instance that can be configured to collect these interface metrics. The resulting Cacti Round Robin Database (RRD) of metrics is then stored in a perfSONAR Measurement Archive (MA) and made available to the user in graphical format.

**PingER** ‘ping’ based monitoring is a widely used tool by many wide area network monitoring services. ‘ping’ monitoring is useful as it is lightweight and only requires ICMP traffic to be allowed through a firewall. The perfSONAR PingER-MA supports the same set of features offered in the PingER project [33]. It is configurable using a web-based GUI (toolkit-PS) and it utilizes the perfSONAR Look-up Service to find other existing measurement nodes to which to run tests. PingER includes a perfSONAR MA interface and a graphical interface for publishing the end-to-end PingER test results.

perfSONAR also provides two other diagnostic tools in the form of NDT and NPAD. NDT allows for users to test the network path for multiple problems related to TCP buffer sizes and duplex mismatches. NPAD allows end users to test limited parts of the network backbone and find if there are isolated issues that would lead to bigger problems on the longer network path. Both the services are also registered with the perfSONAR lookup services to maintain uniformity.

The perfSONAR Look-up Services are central to the Service Oriented Architecture view of the framework as it allows for the services to be registered in an organization. The Look-up service allows for discovery of the services in a global or local network topology according to where the user is registered. The discovery services are constantly updated to provide newer nodes that have been registered in the topology. All
the service descriptions and information about the data a measurement point may collect are encoded in XML format.

1.3 E-perfSONAR

1.3.1 Background

E-perfSONAR is a collaborative approach undertaken by the Ohio Supercomputer Center and the Samraksh company towards implementing a proposed set of extensions to the current implementation of perfSONAR. As a consequence of the proposed extensions, the E-perfSONAR will allow network operators within enterprises to more effectively customize and configure perfSONAR instances for meeting monitoring objectives, enforcing measurement policies, and diagnosing performance related issues. The extensions are proposed based on the R&D efforts such as ‘OnTimeSample’ and ‘OnTimeDetect’ tools [1] [9] developed for perfSONAR communities. Much of the research work is conducted at the Ohio Supercomputer Center (OSC) and OARnet located at The Ohio State University (OSU). These tools attempt to bring into perfSONAR a programmable and extensible interface. The Samraksh company in collaboration with the team at OSU, is looking to provide commercially viable solutions for E-perfSONAR.

The E-perfSONAR software packages are intended to be delivered to enterprise end users as low-cost, small form-factor appliances. E-perfSONAR will be easily interoperable with already deployed perfSONAR instances and our aim is to maintain the open, modular and distributed principles of perfSONAR architecture that have contributed to the wide-adoption of perfSONAR among several academic communities worldwide.

The core E-perfSONAR components are- The Central Intelligence System
which encapsulates the Authentication, Authorization and Resource Protection modules and a **Measurement Point Appliance** which is used to run tests between nodes connected to the Central Intelligence System.

Figure 1.2: E-perfSONAR vision

Figure 1.2 outlines the E-perfSONAR big picture. perfSONAR relies on providing a Service Oriented Architecture and therefore, we propose an extension of the
architecture by integrating an authentication and authorization layer along with a resource protection scheme. These services have the responsibility of providing a base for the middle-ware security features that we need to bring about for E-perfSONAR.

1.4 Motivation

Measurements data collected by the perfSONAR tools is sensitive data for every organization and it is necessary to provide a way of controlling access to this information. Currently this data is freely available to any user who knows the location of any end point running the perfSONAR web services [3].

Also, perfSONAR’s current web-services interface does not provide a completely secure platform to work with. Some of the security issues that are currently prevalent in the toolkit are Cross-Site-Scripting, phishing through frames and link injection. There is a need to come up with viable solutions to mitigate these issues and be able to provide a secure base of operation for enterprises to work with.

perfSONAR provides a measurement framework that enables communities to run tests and share data that has been collected. However, this mechanism is implemented in a strict ‘Open’ or ‘Closed’ format.

- Communities either choose to publish the network statistics to the view of ‘all users’ or only to a select few ‘peer’ communities that are part of a federation.

- The fact remains that several communities are reluctant to publish their network utilization data.

This emphasizes the need to instrument a policy-driven approach so that data sets can be shared across multiple domains based on a federated list of communities that are willing to share their summary of the network statistics.
The primary aim of the Authentication and authorization Service (AS) is to provide a multi-domain policy driven approach towards resolving the problem of verifying whether the end user is valid and has substantial privileges to access the resource. The privileges can be based on policy-driven structure present in organizations.

Figure 1.3 presents the solutions proposed in the document to allow network domains that join the E-perfSONAR topological space in enterprise and research infrastructure to use an Authentication module to protect resources within their particular domain. Users are authenticated before they can access the resources required either to view collected measurement data or to carry out measurements. This service receives either authentication or authorization requests or both.

Figure 1.3: E-perfSONAR example federation structure
1.5 Problem Description

The fundamental aim of the security services in E-perfSONAR is to gain a middle ground between -

- A ‘completely-closed’ domain where the resources within the domain are shut-off from being accessed by the outside world through firewalls or access control lists.

- A ‘completely-open’ domain where the access to resource are not protected by any authentication mechanism. The current perfSONAR implementation allows users to register with a community that can be found using a global lookup service. Irrespective of the event that an organization decides to register on the global lookup service, the current implementation does not provide any security measures involved in protecting the access to data and resources.

The security features that need to be integrated into E-perfSONAR can be broadly classified into problems relating to the scheduling of services between nodes - either within a domain or across domains, and with respect to the viewing of collected measurement data. In this regard, we can place the problem characteristics into three categories -

1.5.1 Execute Access control

The current implementation of perfSONAR allows services to be scheduled to any machine if we know the IP address of the machine that runs a daemon of the service which we require. This gives rise to a scheduling problem where we need a dedicated Resource Protection Service to regulate the tests that are scheduled on each node. To solve this issue, we need to define a set of rules that specify the tools that can be
instantiated by a particular user. This classification of users can be based on roles defined by an organization.

1.5.2 Read Access Control

One of the major focus of security implementation in perfSONAR is that sharing of the data collected by the services must be regulated in a sophisticated manner among users in a multi-domain infrastructure. For this purpose, we must again define attributes for each user in the domain to provide fine-grained access control mechanism. Current implementation places no authentication or authorization scheme in place. Therefore, any user with information about IP address where services are hosted can view data collected by services at that node.

1.5.3 Write Access Control

The data collected within a particular domain may be shared across boundaries so that work may be shared in a collaborative environment. We can define access control mechanism for administrators who may wish to ‘pull/copy’ data from a ‘peer’ or an organization with which data sharing may be previously agreed upon.

1.5.4 Network Security Issues

By studying the audit reports generated by running Rational AppScan software against the perfSONAR toolkit front-end interface, we can conclude that current implementation of perfSONAR consists of issues related to cross-site scripting and phishing through frame attack vulnerabilities in the system. We need to provide solutions to firstly reduce the number of threats present in the framework and also suggest mechanisms through which we can mitigate the effect of any other threat present in the toolkit.
1.6 Thesis Contributions

In this thesis, we present a security architecture that is responsible for enforcing the middle-ware security policies in E-perfSONAR. To this end, we provide -

- A design of a Federated model of security for E-perfSONAR.
- A design of peer-to-peer model of security for E-perfSONAR.

Both features perform authentication of the user and measurement points either within a single domain or across multiple domains.

We have also integrated the Resource Protection Service (RPS) in perfSONAR and we validate its functionality in a real Department of Energy (DOE) community network. To this end, we provide -

- Script to detect changes in input configuration of meta-scheduler.
- Script to parse output of meta-scheduler.
- Script to generate new perfSONAR-services config files.
- Add functionality to automatically restart services.

We also analyze the threats and vulnerabilities present in the perfSONAR toolkit instance and provide results of running vulnerability testing software on the toolkit. The results help us to gain an insight into the threats present in the web application and the methods we can employ to mitigate the effects of the issues. For this purpose-

- Schedule openVAS test on network paths and on machines running perfSONAR services and obtain admissible score based on Common Vulnerabilities and Exposures (CVE) database.
1.7 Related Work

Several research and community based projects have aimed at providing a solution to the establishment of a global network performance monitoring framework. Some of the works such as AMP, RIPE [25], Archipelago [26] have tried to cover as much user space and be useful to the end users. Only Archipelago was able to overcome the problem of a centralized model of data collection and processing. Moreover, none of these different implementations were inter-operable. perfSONAR is based on inter-operable web services and it has aspects in common with grid software such as the Globus Monitoring and Discovery System (MDS) [23], which is able to summarize resources and federate with related monitors [27].

In the past, Allman et. al. described a ‘Community-Oriented’ network monitoring architecture in [5] that has much in common with the perfSONAR architecture. A major difference between the two systems is that their lookup service is based on OpenDHT, while perfSONAR is based on a hierarchical distributed service.

The idea of using semantics for resource allocation of computing services has been explored in the field of virtualization before as seen in [6] in which the authors provide an ontology driven scheme to allocate resources in a virtualized environment.

Ongoing research work towards integration of an anomaly detection scheme in perfSONAR [9] and a semantic scheduling policy based on ontologies [1] provide a solid foundation towards the resolution of the problems faced by the current perfSONAR framework. The semantic scheduling policy described in [1] proposes the idea of maintaining ontology of users in the domain and the resource ontology that preserves a service level agreement maintained by the several communities that instrument perfSONAR into their networks.
Previous work performed by GEANT community uses SAML profiles for authenticating and authorizing resources in multi-domain European research networks. The emergence of Shibboleth open-source framework that uses SAML to provide single-sign on, authentication and authorization capabilities and integration of Shibboleth for grid security in Globus toolkit(GridShib) [23] [28] provides further motivation for investigating the use of Shibboleth system for providing a federated security infrastructure in E-perfSONAR.

Work done by the Globus toolkit community provides ‘GridShib’ a name-mapping plugin for a Shibboleth IdP. Its purpose is to allow servicing of attribute queries from service providers based on the subjects ‘Distinguished Name’ [28]. Our approach adopts this mechanism to provide for authorization functionality by providing a mapping scheme between Federated Identity providers and the policy rules defined in the Resource Protection Service (RPS).

E-perfSONAR is a set of software packages that are extensions to the current set of services provided in perfSONAR. The extensions are based on the ‘OnTimeSample’ and ‘OnTimeDetect’ R&D efforts undertaken at OSU. These extensions allow network operators within enterprises to effectively customize perfSONAR instances for meeting monitoring objectives, enforcing measurement level agreements, and diagnosing performance bottlenecks in network paths that deploy the E-perfSONAR instance. These set of features are in place to bring about a level of programmability and extensibility to perfSONAR and make it better suited for enterprise networks.

These works provide us a solid foundation to build on the perfSONAR framework to integrate the necessary network and middle-ware security features in E-perfSONAR.
1.8 Organization of this thesis

The rest of the thesis is organized as follows-

Chapter 2 describes the security architecture with respect to the designs of a federated model and a peer-to-peer model of security and the features of Resource Protection Service (RPS) described in the OnTimeSample algorithm and its integration into the perfSONAR framework. The chapter also outlines the current implementations of middleware security in place and the architecture proposed for the E-perfSONAR project. Chapter 3 discusses briefly about the intrusion detection testing performed on the Linux based box (E-perfSONAR Measurement Point Appliance) to ensure that it does not become a vulnerability point in an enterprise network.

Chapter 4 details the experiment plan and the final integration of the Resource Protection Scheme in a DOE lab infrastructure. The chapter also showcases the authorization and authentication features implemented in perfSONAR framework and about the issues related to network security.

Chapter 5 summarizes the results of the thesis and discusses the scope of the future work.
CHAPTER 2
SECURITY ARCHITECTURE

2.1 Middleware Security

2.1.1 Federated Security Model

The aim of perfSONAR project was to create an inter-operable framework for data to be gathered and exchanged in a multi-domain, heterogeneous, federated manner. The previous work performed by eduGAIN developers takes into consideration the increasing trend of organizations to federate their management structure. Therefore, they defined mechanisms for bridging authentication and authorization requests from perfSONAR to the SAML-based infrastructures. An example federation structure provided by ANL is as shown in Figure 2.1.
Figure 2.1: Federated network structure
Another example implementation can be found in the E-Center initiative which integrates OpenID \cite{7} in their data management framework which provides a single-sign on mechanism for labs structured under the Department of Energy (DOE) federation.

We require to be able to perform an authentication between measurement points located across domains and also ensure that the user who instantiates the service has valid access privileges to run a service or view data sets. The user may be located at one domain while the service is located in a remote location or domain that is connected to the common federated identity.

Our model provides for cross-domain authentication features in perfSONAR. It aims to delegate authentication mechanism to a centralized server that encompasses multiple domains. Federated identity allows organizations using disparate authentication schemes to inter-operate thereby extending the services offered by each organization rather than revamping their structure. Federated identity also provides the advantage of single-sign on and thus reduces the number of passwords users have to remember \cite{19}. Since the advent of Federated Identity structure, there has been widespread interest from academia and enterprise organizations to integrate SAML based implementations in their interpretation of federated environments. The emergence of Shibboleth open-source framework that uses SAML to provide single-sign on, authentication and authorization capabilities provides us a solid platform to further investigate the use of Shibboleth system for providing a federated security infrastructure in E-perfSONAR.

The typical flow of messages during authentication phase proposed for E-perfSONAR is as follows-

- A user on domain A access resource on domain B.
- Users use federated authentication module to first mutually authenticate the measurement points with each other.
○ Upon successful authentication, access to the resource is granted to the user for scheduling purposes or to pull data for analysis.

○ If the user is requesting for a service to be run either from or to a measurement point located at the remote domain, then the SAML attribute assertion mechanism is set up between the two end-points (measurement points in this scenario) so that a scheduled test can be run.

Figure 2.2 displays the typical flow of messages for a request initiating from the user for accessing resource at a different domain.

Figure 2.2: Inter-domain authentication process
Design of Federated Security model using Shibboleth for inter-domain authentication:

The federated authentication module for E-perfSONAR utilizes the Shibboleth framework for providing access controls to resources located across domains. The Shibboleth architecture defines a set of message exchanges between an identity provider and a service provider to facilitate web browser single sign-on and attribute exchange. The core functional components, derived from the SAML domain model, are: an authentication authority and an attribute authority, along with single sign-on service [12] [13].

Identity Provider: An identity provider is an entity that authenticates principals and produces assertions of authentication and attributes information in accordance with the SAML Assertions [SAMLCore] and the SAML browser profiles [SAMLBind]. Each identity provider must be assigned a unique identifier (i.e., a ‘providerId’) [21].

Authentication Authority: The authentication authority issues authentication assertions about principals to service providers. Shibboleth does not specify how authentication of principals should be performed, but the authority works with the principal’s authentication service so that assertions about the authentication event are issued. The only specifically defined use of an authentication assertion in Shibboleth is with respect to the Browser/POST and Browser/Artifact profiles.

Attribute Authority: Shibboleth specifies the use of the standard SAML attribute request protocol to allow attribute sharing among IdP and SP. Such attribute exchange protocol is optional. In real authentication protocols, the service provider may need to know more about the authenticated user before access to a resource may be granted (authorization). This feature will be particularly useful in defining the roles of users so that we may be able to make fine-grained decisions on the privileges
each user may have in accessing the resource at a measurement point appliance. The
attribute authority is the module the processes the SAML request tag -

\[(\text{samlp}: \text{Request})\]

which contains messages containing the

\[(\text{samlp}: \text{AttributeQuery})\]

element. This service issues attribute assertions to service providers in a mutually
authenticated fashion. Implementations typically rely on SSL/TLS or SAML message
signatures to mutually authenticate the exchange.

**Single Sign-On Service:** A single sign-on (SSO) service is controlled by the
identity provider that receives and processes authentication requests from service
providers sent through web browsers. The SSO service initiates the authentication
process and redirects the client to the inter-domain service. Figure 2.3 illustrates the
message flow between Shibboleth components during inter-domain Authentication.

The steps involved in the authentication process are as follows

1. The client at domain B requests for a ‘target’ resource at the perfSONAR service
   provider located at domain A.

2. The Central Intelligence module delegates the authentication mechanism to the
   federated authentication service after verifying that the request for a service is
   inter-domain.

3. The client is redirected to the SSO service at the IdP. Three parameters that are
defined in this process are- the ‘shire’ parameter that specifies the location of
assertion consumer service (the SSO endpoint at the SP); the ‘target’ parameter
which is the final target that the client wants to access and the optional ‘time’
parameter that allows the IdP to detect redundant and unnecessary requests. The client now accesses the SSO service at the IdP and authenticates to the IdP. Once the user is verified, the IdP redirects to the client with the message that the principal has been verified.

4. The client issues a POST request to the SSO endpoint at the SP.

5. The SP parses the request, validates the signature to verify authenticity of the IdP and issues a security context. Now, control is passed to the attribute exchange protocol to verify the attributes of the user.

6. The attribute requester module creates a SAML SOAP message to the attribute authority at the IdP using the tag -

\[ \langle \text{samlp} : \text{Request} \rangle \]
It then encapsulates the subject handle and the attribute designator tag which is used to confirm the role of the user within the AttributeQuery.

7. The attribute authority module processes the request and returns required attributes to the requester module at SP.

8. The assertion consumer service filters the attributes, updates the security context and redirects the client to the target resource.

9. Future requests to access the resource are automatically granted as a valid security context exists.

2.1.2 Peer-to-Peer Security Model

While a federated model can be used for resource access and protection across several domains, a peer-to-peer authentication mechanism must be defined for ‘peer’ systems so that authentications can take place independently for systems that are within a single domain. Some of the common mechanisms of peer-to-peer authentication are: Unix passwords, ACL, NTLM, Kerberos.

In a peer-to-peer (P2P) network, peers communicate directly with each other to exchange information. P2P systems are characterized by being extremely decentralized and self-organized.

We propose an authentication model that authorizes each other peer in a P2P based environment. We apply Kerberos authentication mechanism to authenticate each other peer for strong authentication.

The typical flow of messages during the authentication process is as follows -

- A user on domain A access resource on same domain.

- Central Intelligence System uses peer-to-peer mechanism for authentication of users and measurement points within the domain
Upon successful authentication, client is redirected to the resource.

Figure 2.4 illustrates the proposed P2P authentication scheme for resource within a single domain.

Kerberos provides a credible solution to our problem as

- It is platform independent and therefore can fit in easily in academic and enterprise authentication structure.
Kerberos is an industry standard and is easy to integrate.

Kerberos is already in place. Kerberos is already integrated into most popular operating systems and many widely-used software applications.

Allows end users the benefit of single sign-on.

Provides scalability and suits federated/P2P environment. It supports separate administrative realms and cross-realm services that is used to extend authentication and authorization across organization boundaries.

Kerberos provides policy enforcement and audit-ability features. Kerberos administration is often integrated with other directory services and user administration software to provide for policy administration service.

**Design of peer-to-peer security model for intra-domain authentication using Kerberos:**

Kerberos works on the basis of ‘tickets’ to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. It is modeled primarily as a client-server model, and it provides mutual authentication, i.e. both the user and the server verify each other’s identity. Kerberos operates by encrypting data with a symmetric key. When working with the encryption key, the details are actually sent to a key distribution center (KDC), instead of sending the details directly between each computer. A hybrid model using Kerberos scheme for P2P authentication within a single domain has several benefits as opposed to a completely decentralized authentication mechanism [29].

The steps involved in the authentication process are as follows-

**Stage 1:**

1. The authentication service (AS) receives the request by the client and verifies
the authenticity of the client (a measurement point appliance in our case) using a simple database lookup of the user’s ID.

2. The ‘peer’ is verified and a time stamped user session is created. This session key is created such that only the peers can decrypt it.

3. The key is sent back to the client in the form of a ticket-granting ticket, or TGT. The TGT is used for authenticating the client for future reference.

Stage 2:

1. The client (peer A) requests to access resource of service provider server (peer B).

2. Peer A submits the ticket-granting ticket to the ticket-granting service (TGS) to get authenticated.

3. The TGS creates an encrypted key with a time-stamp, and grants client a service ticket.

4. Peer A now decrypts the ticket and then sends its own encrypted key to Peer B.

5. The service (peer B) decrypts the key, and makes sure the time-stamp is still valid. If it is, the service contacts the KDC to receive a session that is now returned to the client (peer A).

6. Peer A decrypts the ticket and initiates connection channel between the principals involved in the communication process.

The message flow during Kerberos Authentication phase is displayed in Figure 2.5

Some of the advantages of adopting the Kerberos model for P2P authentication are-
Figure 2.5: Kerberos message flow

- Scalability: Expert systems can track when new measurement points are added and can configure this new node for accessing its peers.

- Manageability: The expert-system can monitor the activity of the measurement points connected to it. This allows for better administrative capabilities for managing the peers.

- Load Balancing: Users do not have to authenticate every time to access resource due to SSO service

- Kerberos provides session time stamped ticket that provides continuous access for limited time period.

- The authentication module can exist on multiple nodes so that it can serve as a fallback in cases of failure
2.1.3 Authorization Model

Resource Protection Service

The Authentication and Authorization layer provide to the end user, access to measurement resources of multiple domains within an enterprise, based on the modes of access determined in the enterprise security and measurement-level policies. Enforcement of the enterprise policies on the measurement resources is performed in the Resource Protection Layer.

There have not been any previous efforts in the perfSONAR community to develop or integrate services for the Resource Protection Layer that particularly address needs of enterprise-specific deployments. The OnTimeSample prototype software developed at OSU describes a ‘policy-inference engine’ and ‘meta-scheduler’ service for policy management and enforcement of enterprise policies in the Resource Protection Layer. Our aim is to develop a distributed meta-scheduler within E-perfSONAR as a perfSONAR extension based on the OnTimeSample prototype software. This extension enables E-perfSONAR to provide a tool-specific resource broker with programmable measurement orchestration to:

1. Meet monitoring objectives accurately using strategies such as adaptive sampling taking into account timing constraints,

2. Perform concurrent execution whenever possible to increase number of measurement requests that can be handled network-wide, and

3. Work in tandem with the proposed authentication and authorization frameworks as well as a policy inference engine to enforce measurement-level agreements within enterprise federations

In the rest of the section, we will describe the OnTimeSample tool. The policy-inference service can be operated by an enterprise policy administrator who has access
to all the enterprise security and measurement-level policies. The policies are used
to construct ontology trees that capture the semantic relationships between different
entities of the enterprise. It is designed in a manner so as to allow easy additions,
modifications and deletions of policy rules. The ontology tree nodes represent rela-
tionships or associations among various users within a single domain and also show
how the domains are connected in a hierarchical fashion similar to the construction
of a federation. This mapping is useful when we integrate the Authentication and
Authorization modules with the Resource Protection Service.

When a new user measurement request arrives in an enterprise, an inference engine
processes the ontology trees correspondingly to determine the access control privileges
and relative priority of the new measurement request considering all the previously
scheduled measurement requests. This lends a dynamic nature to the working of the
meta-scheduler module.

The figures 2.6 illustrate how for example, policy ontologies can be constructed
within an enterprise comprising of multiple measurement domains.
Figure 2.6: Policy ontology tree
Enterprises typically are part of measurement federations comprising of multiple domains within the enterprise as well as other external domains. Given the fact that Internet performance monitoring is inherently multi-domain in nature (imagine multiple ISP domains connected to each other while end user tries to run measurements through them), measurement resources for a network path measurement involves both ‘intra domain’ resources and ‘inter domain’ resources. Based on such resource types, resource policies can be specified in the policy ontology that have been defined previously.
Also, domains within a federation could have common policies to co-operate with each other, and such policies could be specified in the policy ontology. Enterprise users can have IDs/names, and roles as an internal domain user or a federation user. User ontology example shown in Figure 2.7 can be used to capture such user IDs/names, roles and other detailed user preferences (i.e., sampling patterns, semantic monitoring objectives) that are part of measurement requests.

The user and resource policy ontologies are developed using the open-source protg-OWL [34]. The protg-OWL editor supports Web Ontology Language (OWL)and supports a knowledge-based framework. Semantic Web Rule Language (SWRL) is used extensively to run the rule-base in the inference engine while enforcing the enterprise security and measurement-level policies. SWRL is used because it is easily portable and extensible. For example, we can specify a rule such that intra-domain users get higher semantic priority versus the federation (external-domain) users.

The components of the Resource protection service are set up as shown in the figure 2.8.
Figure 2.8: Components of Resource Protection Service
In the ontology-based semantic meta-scheduler service illustrated in the figure 2.8, the priority calculator receives measurement requests (i.e., tasks to be scheduled) from end-users and uses the inference gained from ontology trees and a run-time solver to dynamically calculate the priority of each task. To capture the relative importance of the tasks, an initial-state and a run-state priority calculation is used. The initial state priority calculation is based on policy and user ontologies. The user roles, resource policy and sampling preference contribute towards initial state priority. Run state priority is set by the run-time solver based on oversampling penalty value obtained from the predictor scheme of the particular user. Final priority is calculated as the weighted difference between initial state priority and run state priority. Measurement requests are ordered based on decreasing initial state priority which is known apriori. The semantic scheduler module receives these ordered measurement requests and generates schedule which is sent to measurement points in perfSONAR, which in turn initiate the tools to sample performance data.

**Integration in E-perfSONAR**

The federated and peer-to-peer models provide a solid framework on which we can authenticate users and establish a trust relationship between the users either within a single domain or in a multi-domain environment. Once authentication is performed, we need a sophisticated model that can take authorization decisions based on the attributes associated with each user. To this end, we provide a novel solution that integrates directory structure along with a Resource protection scheme that provides an inference engine which takes fine-grained access control decisions based on-

- The level of user maintained in a directory structure within the domain.
- User attributes within a federated environment if access requests are cross-domain in nature.
The authorization mechanism for a Federated model is as shown in Figure 2.9. In this model, there is a mapping between the attributes specified in the SAML profiles used during Shibboleth scheme and the rules (ontology based rule file) maintained by the Resource Protection Service tool. The mapping could be modeled such that user attributes are more generic when placed in a SAML profile as compared to attributes placed within an individual domain and the intra-domain attributes can over-ride the more generic versions as and when necessary.

Figure 2.9: Authorization mechanism in federated environment
The authorization mechanism for a peer-to-peer model is shown in Figure 2.10. This model illustrates the interactions between the LDAP module and the Resource Protection Scheme wherein the policy levels are maintained and schedule is generated based on the structure of LDAP directory information.

Figure 2.10: Authorization mechanism in P2P environment
CHAPTER 3
NETWORK SECURITY

Both the components of E-perfSONAR as described in Chapter 1, when placed in an enterprise network, must not be an easy entry point of attack for malicious agents trying to force their way into the network. The main objective of the vulnerability testing phase is to identify, investigate and find steps towards the resolution of network security problems present in perfSONAR and to provide insight into which ports need to be looked at so that network administrators have better data to work with while mitigating the effects of the threats.

To verify that the components do not pose any significant threats to the network, we use the industry standard Common Vulnerability Scoring System [31] to evaluate the initial deployment of E-perfSONAR Measurement point appliance on the test network at the Ohio Supercomputer Center.

The Common Vulnerability Scoring System (CVSS) provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. Its quantitative model which provides for a mathematical model for calculation of base, temporal and environmental impact scores ensures accurate measurements while enabling users to see the underlying vulnerability characteristics that were used to generate the scores. Thus, CVSS is adopted on a wide-scale as a standard measurement system for industries, organizations, and governments that need accurate vulnerability impact scores. The National Vulnerability Database (NVD) provides CVSS scores
for known vulnerabilities. CVSS is composed of three core metric groups: Base, Temporal, and Environmental, each consisting of a set of metrics. These metric groups are described as follows-

1. Base: Represents the fundamental characteristic of a vulnerability that is constant over time and user environments.

2. Temporal: Represents the characteristics of a vulnerability that change over time but not among user environments.

3. Environmental: Represents the characteristics of a vulnerability that are unique to a particular user’s environment.

The National Vulnerability Database does not currently provide temporal scores. However, NVD does provide a CVSS score calculator to allow you to add temporal data and to calculate environmental scores.

**NVD Vulnerability Severity Ratings:** NVD provides severity rankings of ‘Low’ ‘Medium’ and ‘High’ in addition to the numeric CVSS scores but these qualitative rankings arise from a simple mapping from the numeric CVSS scores:

1. Vulnerabilities are labeled ‘Low’ severity if they have a CVSS base score of 0.0-3.9.

2. Vulnerabilities are labeled ‘Medium’ severity if they have a base CVSS score of 4.0-6.9.

3. Vulnerabilities are labeled ‘High’ severity if they have a CVSS base score of 7.0-10.0.

For obtaining the vulnerabilities in the system and to be able to compare the threat levels of perfSONAR toolkit installation against the E-perfSONAR test deployment,
we use BackTrack [35]. BackTrack is a Linux-based penetration testing arsenal that aids security professionals in the ability to perform assessments in a purely native environment dedicated to hacking. The penetration distribution has been customized down to every package, kernel configuration, script and patch solely for the purpose of the penetration tester.

BackTrack provides a Vulnerability assessment tool called ‘OpenVAS’ [30]. The Open Vulnerability Assessment System (OpenVAS) is a framework of several services and tools. The fundamental tool provided through a service-oriented architecture is the OpenVAS Scanner. The scanner efficiently executes the actual Network Vulnerability Tests (NVTs) which are served with daily updates via the OpenVAS NVT Feed or via a commercial feed service. The remainder of the section briefly describes the openVAS components and their uses.

The OpenVAS Manager controls the Scanner using the OTP (OpenVAS Transfer Protocol) and itself offers the XML-based, stateless OpenVAS Management Protocol (OMP). The Manager also controls a SQL database (sqlite-based) where all configuration and scan result data is located.

Currently, two options for OMP clients are available-

- **The Greenbone Security Assistant** (GSA) is a web service offering a user interface for web browsers. GSA uses XSL transformation style-sheet that converts OMP responses into HTML.

- **The Greenbone Security Desktop** (GSD) is a Qt-based desktop client for OMP. It runs on various Linux, Windows and other operating systems.

OpenVAS CLI contains the command line tool ‘omp’ which allows to create batch processes to drive OpenVAS Manager and the OpenVAS Administrator acts as a command line tool or as a full service daemon offering the OpenVAS Administration
Protocol (OAP). The most important tasks are the user management and feed management. GSA users with the role ‘Admin’ can access the OpenVAS Administrator functionality.

The experimental results obtained from running the OpenVAS scanner is documented in the Experimental Evaluation section.
CHAPTER 4
EXPERIMENTAL EVALUATION

4.1 Metascheduler Implementation

Figure 4.1 shows current architecture of Scheduling services running perfSONAR instances. Identical perfSONAR deployments are made on all the nodes and instances of tools are scheduled from a central node to any of it’s available neighbors.

We now detail the plan to integrate customized E-perfSONAR instances of the policy-inference and meta-scheduler services within an example DOE lab site for functional validations. We use the E-Center initiative for this purpose. The E-Center is the enterprise implementation of perfSONAR within the DOE community to orchestrate and manage DOE enterprise user requests for network performance measurements [8]. Figure 4.2 shows an example use case of the E-perfSONAR’s Resource Protection Service in E-Center. In this model, we place the Resource protection service at the central node which has the responsibility of producing policy-compliant and timely-ordered set of tool instances.

The work plan of the Meta-Scheduler module that implements the policy inference engine and produces an output of a timely-ordered prioritized set of schedule, is as follows -

1. Policy-inference engine

   - Runs on test measurement point.
Figure 4.1: Current star topology in perfSONAR

- Produces policy ordered and prioritized list of services to be scheduled.
- Enforces multi-domain policy rules and provide prioritization of users who instantiate services.
- Provide scalable framework where addition/removal of services does not hinder currently running services.

2. Initiate tests between the ‘Test Measurement Point’ node and ‘Peer Measurement Point’ nodes based on the output of Meta-scheduler

3. Regenerate perfSONAR specific config files

4. Restart services
5. Collect data using perfSONAR MA and view graphs using perfSONAR toolkit GUI. The working is illustrated in Figures 4.3 and 4.4.
Figure 4.3: Metascheduler work-flow (Step 1)

INPUT CONFIG FOR METASCHEDULER

OUTPUT OF POLICY-ORDERED PROCESSES

Figure 4.4: Metascheduler work-flow (Step 2)

FEEDING NEW INPUT

REORDERING OF MEASUREMENT TOOL PROCESS BASED ON PRIORITY
The Resource protection service runs a policy inference engine whose basic functionalities are

- Produces policy ordered and prioritized list of tools to be scheduled
- Meets network performance monitoring needs of federated environment
- Enforce multi-domain policy rules and provide prioritization of users who instantiate services
- Defines ontology of users based on the policies described in OnTimeSample paper
- Can be extended to fit Argonne National Lab (ANL) and Stanford Linear Accelerator Center (SLAC) views of user-level policies
- Provide scalable framework where addition/removal of services does not hinder currently running services

**Features of Resource Protection Scheme:** The following list summarizes the features of the Resource protection scheme-

1. Allows a measurement federation to provide better mechanisms of resource protection by defining an ontology of users and their domain level access

2. Does not disturb previously scheduled tests

3. Allows prioritization of users:

   - Network-operator
   - Power-user
   - Regular-user
4. Allows splitting up of domains

- Inter-domain
- Intra-domain
4.2 Vulnerability Testing

BackTrack, an open source Linux distribution that is tailored for vulnerability testing was used to run tools at the IP addresses where the perfSONAR toolkit was deployed and at the location where the E-perfONAR Measurement Point Appliance was deployed. The results are as shown -

Table 4.1: OpenVAS test results based on CVSS scores

<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
<th>False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

The results of running the openVAS test to identify which ports are exhibiting the vulnerabilities provides us further insight to be able to mitigate the issues present. The results are collated in the table 4.2.
Figure 4.5: perfSONAR threats based on CVSS score

Figure 4.6: Threats in perfSONAR vs Threats in E-perfSONAR
<table>
<thead>
<tr>
<th>Service (Port)</th>
<th>Threat Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntp (123/udp)</td>
<td>High</td>
</tr>
<tr>
<td>palace-4 (9995/tcp)</td>
<td>Medium</td>
</tr>
<tr>
<td>general/tcp</td>
<td>Low</td>
</tr>
<tr>
<td>http (80/tcp)</td>
<td>Low</td>
</tr>
<tr>
<td>https (443/tcp)</td>
<td>Low</td>
</tr>
<tr>
<td>osm-appsrvr (9990/tcp)</td>
<td>Low</td>
</tr>
<tr>
<td>ssh (22/tcp)</td>
<td>Low</td>
</tr>
</tbody>
</table>
CHAPTER 5
CONCLUSION AND FUTURE WORK

Conclusion: To conclude, the main objective of the thesis is to identify, investigate and make steps towards the resolution of network security problems of perfSONAR and also in designing an authentication and authorization policy framework that can be built into the E-perfSONAR system. We refer to our solution as the ‘Resource protection service’ within the E-perfSONAR. The bulk of the work is based on experiments conducted to improve network security and middle-ware security, and validation of the work ability of the ‘Resource Protection Service’ by an actual integration in an enterprise built around DOE National Labs.

The perfSONAR toolkit services were also investigated using BackTrack penetration testing software and the results have been documented. The results obtained indicate a decrease in the number of vulnerabilities found between perfSONAR and the E-perfSONAR measurement point appliance being developed.

Future Work: Research work related to QoS management [2] shed light on the various methods that can be used to resolve conflicts in policy-driven approaches. We can look at future work in this respect. Configuration analysis and testing frameworks introduced in [2] focus on the policies that were instrumented in DiffServ networks and propose diverse concepts such as policy grammar model and Binary Decision
Trees that model PHB (Per hop behavior). The ideas can be extrapolated towards policy-conflict resolution which has applications in the resource protection scheme central to our problem. The rules for policy-conflict-resolution is applied to the ontologies described using languages such as SWRL and OWL [34] which adds value to the security structure of perfSONAR.

Current work performed by the Globus toolkit community provides support to Plug-gable Authentication Modules in Unix systems to allow the flexibility for the end user to follow authentication scheme of their choice. Such a mechanism would be beneficial for E-perfSONAR as it tries to cater to various enterprises with disparate authentication systems.

Federated Authentication systems are currently in place that use SAML profiles and Shibboleth for their authentication. But work is being conducted to provide a fault-tolerance mechanism for the Identity providers in a Federated framework. Integration of these features in E-perfSONAR would make it more robust and more marketable.
BIBLIOGRAPHY


[16] Iperf. [http://dast.nlanr.net/Projects/Iperf/](http://dast.nlanr.net/Projects/Iperf/)


[34] protg-OWL. [http://protege.stanford.edu]