Location Aware Self-Adapting Firewall Policies

TUGKAN TUGLULAR
Department of Computer Engineering
Izmir Institute of Technology
Gulbahce Koyu, Urla, Izmir
TURKEY
tugkantuglular@iyte.edu.tr http://www.iyte.edu.tr /~tugkantuglular/

Abstract: - Private access to corporate servers from Internet can be achieved using various security mechanisms. This article presents a network access control mechanism that employs a policy management architecture empowered with dynamic firewalls. With the existence of such an architecture, system and/or network administrators do not need to reconfigure firewalls when there is a location change in user settings, reconfiguration will be automatic and seamless. The proposed architecture utilizes dynamic firewalls, which adapt their policies according to user locations through the guidance of a policy server. This architecture is composed of a VPN client at user site, a domain firewall with VPN capabilities, a policy server containing a policy decision engine, and policy agents residing in dynamic firewalls, which map policy server decisions to firewall policy rules, at server site.

Key-Words: - Network Access Control, Firewalls, Firewall Policies, VPN, Location Awareness, XACML, Policy Agents.

1 Introduction

Maintaining security on their networks is critical for all corporations. One primary tool that every network needs is access control – the ability to carefully define and enforce which users have what type of access to specific applications, data and services [12]. When the network is contained within a single building, the problem is generally handled by a firewall. For corporate networks, which involve multiple interconnected segments distributed across various locations, the problem is hard and beyond node level configuration. To manage this kind of heterogeneous, distributed, and dynamic networks, system administrators want to focus on policy level management instead of taking their time for node level configuration [28]. Therefore, utilization of policy-based network management systems for the large scale networks is suggested [7].

Policy-based management approach will help to manage information technology risks more efficiently because of the minimization of human involvement, hence less risk exposure, in the process of network security management. According to Corporate Information Technology (IT) Risk Management Model [24], policies on management level and procedures on functional level are necessary for a successful risk management. The risk management model matches the approach followed in this article.

Policy-based management has become a promising solution for managing enterprise-wide networks and distributed systems. These are typically large-scale systems which require management solutions that are both self-adapting and that dynamically change the behavior of the managed system [6]. The main motivation for the recent interest in policy-based services, networks and security systems is to support dynamic adaptability of behavior by changing policy without recoding or stopping the system [6]. This implies that new mechanisms, which dynamically update the policy rules interpreted by distributed entities to modify their behavior, should be integrated to existing architectures. The ability to dynamically change, distribute, and enforce these policies is the key to robust corporate network security management [23].

In corporate network environments, where locations of users change frequently, it is desired to have network access control mechanisms that are able to adapt to these changes. This will provide users with more seamless, easy use services. The paradigm of tailoring applications, services, communication and connectivity to the user's current situation and needs, is referred to as context awareness [9]. The goal of location aware user based network access control approach is to develop an architecture that enables the network access

control policy to be adapted to user context information, specifically to the location of the user.

This article addresses the challenge of managing and enforcing a corporate network access control policies with respect to user and his/her location. An important aspect of this architecture is that network security policy management and enforcement are externalized from applications. This reduces effort and complexity related to maintaining and enforcing organizational policy, especially when information and related policies are distributed across multiple network segments [3].

This article presents a network access control approach based on the XACML [19] standards, which will be used for expressing access control policies based on authorization attributes and statements. Similar approaches exist in the literature [16], [15], and [8]. One of the reasons to choose XACML as the access control policy language for the architecture is that there is an implementation to support the XACML specification, known as Sun java XACML implementation [25]. It is possible to extend this open-source implementation to adapt it to context-aware access control [9]. In this article, introduction of XACML implementation is followed by the demonstration of the way to extend it for location aware user based network access control usage.

For the enforcement of location aware user based network access control policies, the focus will be on firewalls as the network access control mechanisms. The challenge is for firewalls to know the right reconfiguration so that the appropriate security policies are upheld preventing illegitimate users from gaining access [4]. The architecture presented in this article also focuses on the configuration management of firewalls so that organizational network security policies can be enforced. Building a management layer for firewalls using policies enhances its value by making the management of the firewall technologies highly programmable and easily adaptable to different environment and security requirements [27]. Relevant network access control policies will be disseminated to the firewall policy agents so that they implement them on their firewalls [22].

The rest of this article is structured as follows. Section 2 provides an overview of related work. Section 3 describes requirements of a location aware self-adaptive firewall policy management architecture. The proposed architecture is explained in Section 4. This section includes main components of the architecture, the information model utilized, dynamic configuration scenario, policy decision making process, and policy adaptation process. A prototype implementation is presented in the following section with the operational values measured. Finally, the article concludes with some remarks and future directions.

2 Related Work

Yoshioka et al. proposed a policy based automatic network configuration mechanism that makes the configuration of segmented networks inconsistency-free [28]. In their proposed mechanism, a policy describes the types of events and configuration messages a network element is to accept, and that policy is registered on a central policy server by each network element in the initialization phase of the element. After the initialization, all events and messages are sent to the policy server, and the server forwards them to appropriate network elements considering the registered policy so that they can reconfigure themselves automatically. With the introduction of a policy server, network administrators do not have to do anything to the network configurations, because policy server adjusts the configuration according to the changes in the environment automatically. They claim that this feature also expels inconsistencies.

There is another work on policy based configuration of network elements using SNMPv3 [18]. In their work, Omari et al. aimed to integrate the policy concept into the SNMPv3 framework. They proposed a set of rules to map authorization policies to the VACM (View Based Access Control Model) standardized as part of the SNMPv3 management framework. Within their framework, Policy attributes are maintained in a configuration database local to the SNMPv3 entity and a new application is incorporated into the SNMPv3 entity to perform the mapping. This will ultimately allow manager and management applications to enforce enterprise authorization policies independently of the security model(s) implemented by SNMPv3 entities.

One other research on policy based configuration of network elements is presented by Kim et al. [13]. Their paper describes a technique relative to a policy based network security management system managing network security devices centrally. They presented a method using legacy devices that is already arranged and used at network within the Policy Framework. They claim that their method allows people to manage network security devices with small cost and effort, centrally and efficiently.

Various frameworks, systems and architectures have been proposed providing context aware access

control and/or policy enforcement. Some of them are closely related to the work presented in this paper. Lymberopoulos et al. presented a framework for specifying policies for the management of network services in their paper [17]. Their framework supports automated policy deployment and flexible event triggers to permit dynamic policy configuration. In their paper, they focused on solutions for dynamic adaptation of policies in response to changes within the managed environment. They claim that policy adaptation includes both dynamically changing policy parameters and reconfiguring the policy objects, of which application to access control for network services is discussed in their paper as well.

Kapsalis et al. proposed a context-aware, access control architecture, in order to support fine-grained authorizations for the provision of e-services, based on an end-to-end web services infrastructure [11]. In their proposal, access permissions to distributed web services are controlled through an intermediary server, based on a role-based access control model, which incorporates dynamic context information, in the form of context constraints. They claim that a high level of abstraction of the physical environment is achieved by using the concepts of simple and composite context conditions. Also, their paper proposes adequate mechanisms for updating context dynamically. They concluded with a presentation of an example use case.

Han surveyed the recent researches about context awareness and context-aware security. Based on the survey results, the requirements of CASPEr system are analyzed and using the knowledge from analysis and up-to-date technologies the CASPEr model is designed. The CASPEr (Context Aware Security Policy Enforcement) model focuses on introducing context awareness into security mechanism to let the enforcement of security policy dynamically adjusted according to the changes of context information, consequently providing dynamical, advanced and seamless security setting. Furthermore, this model is applied on access control to let this security mechanism to be context aware [9].

Lorch et al. presented XACML, a standard access control language, as one component of a distributed and inter-operable authorization framework in their paper. Several emerging systems which incorporate XACML are discussed and using these discussions they illustrate how authorization can be deployed in distributed, decentralized systems. At the end of their paper, they presented some new and future topics to show where this work is heading and how it will help connect the general components of an authorization system [15].

Giordano et al. proposed a visual policy definition hierarchy and a tool for specifying and implementing access and security policies based on the RBAC model to support network, system, and/or platform administrators in the management of resources. They represented those policies in XACML [8].

Tuglular et al. proposed an architecture for XACML based management of firewall configurations in large enterprise networks. The goal of their architecture is to allow administrators and end-users to manage their firewalls, while enforcement of organizational policy is ensured to prevent unacceptable traffic gaining access to the private network domain. They suggested a method to represent firewall policies using XACML for the platform independent management of distributed firewalls [26].

3 Requirements

The corporate network environment can be modeled as a network composed of multiple interconnected network segments distributed across various locations. There is an authentication server at each location, where corporate users are required to be identified and authenticated first, and then they open Virtual Private Network (VPN) connections to other locations, or sites, to access servers at those sites. Servers are maintained in a demilitarized zone (DMZ) at each site, which are protected by a domain firewall as well as an additional dynamic firewall. There is a policy server in the DMZ, which is responsible for automatic reconfiguration of dynamic firewalls so that a user logged into a domain, or a site, can access a server maintained at another site if he or she is authorized to do so. Each of these network segments, locations, sites, or domains, is protected with a domain firewall as mentioned above. The overview of this corporate network model is shown in Fig.1.

In this model, a user is able to move from one domain or location to another and a server can provide services from any domain. After authentication, VPN client application residing in user's laptop triggers the policy server, which reconfigures all the dynamic firewalls on the route of servers that the user is authorized to access. After automatic reconfiguration of firewalls, the user can access corresponding servers.

A typical corporate network system consists of heterogeneous networks and servers running a

variety of applications and offering services to a large number of users. The complexity of the managed systems results in high administrative costs and long deployment cycles for business initiatives, and imposes three requirements on their management systems. Although these requirements have long been recognized their importance is now becoming increasingly critical [5]: (i) Management must be distributed in order to be scalable and so that it is possible to cope with the size of enterprise networks. (ii) Management procedures must be automated to reduce administrative costs. Manual management is expensive, furthermore the effort and time needed for management increases exponentially as a system expands. (iii) As networked systems are increasingly driven by changing business requirements, management also needs to be dynamic and flexible to deal with the evolution of the systems being managed.

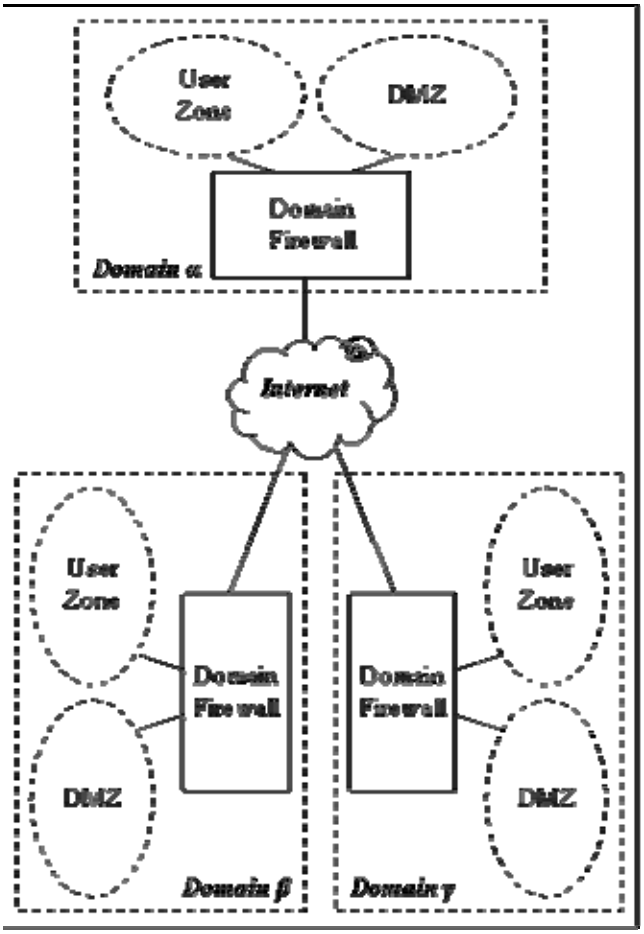

Fig.1 Corporate network model enables automatic reconfiguration of firewalls

The requirements, which are also valid for a network access control system, can be facilitated with policy-based management approaches where the support for distribution, automation and dynamic adaptation of the behavior of the managed system is achieved by using policies. The main benefits of using policies are improved scalability and flexibility for the managed system. Scalability is improved by uniformly applying the same policy to large sets of subjects and objects, while flexibility is achieved by separating management from the implementation. In this way, policies can be changed dynamically, it is possible to change the behavior of a system, without modifying its implementation or interrupting its operation [5].

Assuming dynamic changes in user behavior and working with a network security management willing to adapt to those changes, the desired network access control architecture would have the following requirements [20]:

- Flexible policy definition: ability to define policies that are either general-purpose or specific to particular user behaviors.
- Highly secure enforcement: policy enforcement mechanism must be based on well-grounded security techniques.
- Conductive to segmented network environment: policies must be enforced on any network segment. Furthermore, policies and enforcement mechanisms must permit clients to move among network segments.

It is necessary and possible to make the network security to adapt to the changes in user behavior, or context. The context information such as location of the user is required for automatically enforcing the network security policy. That means that the network access control configuration should be able to adapt to new user context. In addition, the security configuration process should be invisible to users. Users should not be bothered by applying security mechanisms before or during their activities. Therefore, the proposed architecture should have a mechanism to achieve the dynamic auto-configuration of network access control policies, or simply firewall policies. This architecture should be able to input user location information (i.e. user residing on specific host, domain, and network) and use it to trigger policy decision making process.

In summary, the location aware self-adaptive network access control architecture should be able to satisfy the following requirements, which are similar or analogous to requirements delineated in [10] and [9]:

- The users must be authenticated before receiving access to servers.
- The users should be able to have access rights on different servers.

- The users should be able to access servers several times in a session.
- The users should be able to login to the corporate network from different network segments.
- Several network protocols should be available to the users to obtain services from various servers.
- The architecture should be invisible to the users of the corporate network.
- The architecture should be able to provide logged-in users with a connection infrastructure to servers they are authorized to access.
- Servers must be protected from illegal access.
- Users and servers should have previous registration for any access operation to be performed between them.
- The administrators should be able to register users with their access rights and servers with their deployment information.
- The administrators should be able to define network topology and firewall deployments.
- Organizational network access control policies should have precedence over user access control policies. For instance, if a server is not allowed to be accessed from a network segment and a user who has access right to that server logs in to that network segment, then the user is not allowed to access that server.

4 Architecture

Traditional security systems lack adaptive security policies and enforcement mechanisms [21]. In the non-adaptive setting the set of policies is chosen in advance, before the server connection request is received. The adaptive policy enforcement architecture presented in this article selects the appropriate policies during the course of VPN connection operation based on the current location of the user. This architecture employs a Policy Server (PS) and a number of Policy Agents (PA). Each domain has a policy server to adapt network access policies and each firewall in the corporate network has a policy agent to modify its policy. The whole architectural picture is given in Fig.2.

4.1 Dynamic Configuration Scenario

The main principle of this architecture can be defined as centralized policy decision and distributed reconfiguration. In this semi-distributed architecture, the domain firewall initiates firewall configuration requests, then policy server makes network access control policy decisions and communicates with firewall agents to distribute necessary policy changes so that firewalls are reconfigured automatically.

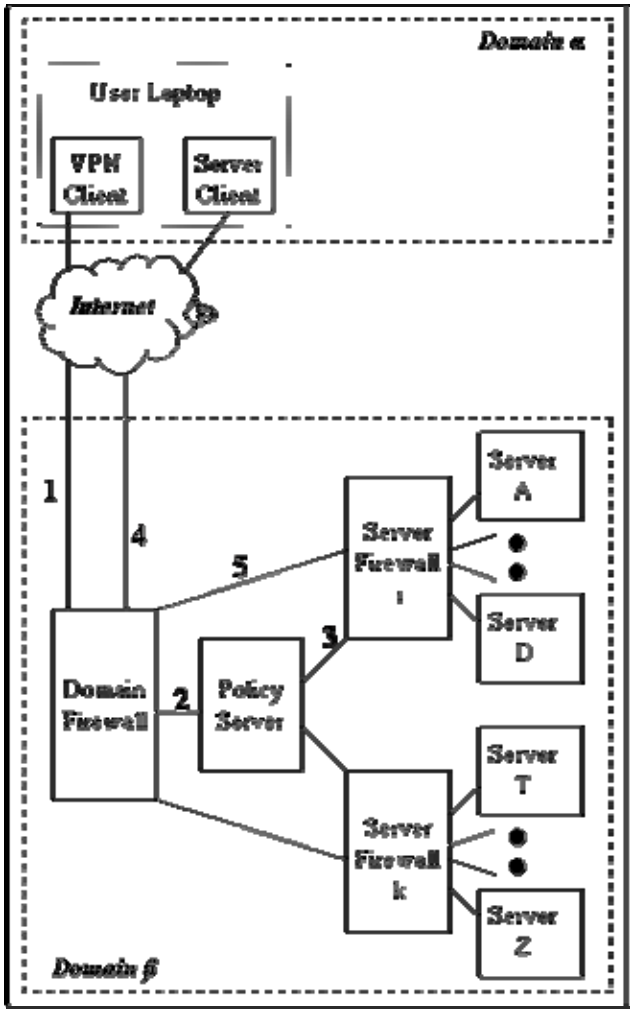

Fig.2 Location aware user based network access control architecture

On-demand configuration requires the following algorithm [2], which assumes pre-determined policy is already deployed on firewalls. All the steps are performed automatically:

i. adapt policy to current user location
ii. deploy adapted policy
iii. enforce adapted policy on firewall(s)
iv. monitor user location changes
v. return to step *i* when a change is detected

The above algorithm becomes operational on the proposed architecture as follows. First, VPN client opens a VPN connection to the domain firewall, so that domain firewall becomes aware of the users location and informs the policy server as shown as step 2 in Fig.2. Depending on the user access rights,

policy server modifies high level network access control policy and deploys this adapted policy to corresponding firewalls in step 3. The policy agent residing on the corresponding firewall maps high level policy to machine dependent low level policy, which is actually the firewall policy (for some implementations it is actually a file). Starting from that point on, server client application, such as a web browser, can connect to servers that the user has access rights. When the VPN connection is broken the domain firewall notifies the policy server, so that it replaces the adapted policy with the original network access control policy and deploys to the corresponding firewalls.

4.2 Policy Server

In the core of this architecture lies the policy server which manages user location based access control policies. The following components, which are shown in Fig.3, are identified within the policy server: Configuration Request Point (CDP), Policy Decision Point (PDP), User access rights Policy Information Point (UPIP), Network access control Policy Information Point (NPIP), Topology Information Point (TIP), and Configuration Deployment Point (CDP).

CRP is the entry point of policy change requests. PDP is the core of the policy server. It evaluates applicable policies, i.e. user access control policies, and renders policy decisions against network topology information to formulate new network access control policy. Policy and topology information are retrieved from UPIP, NPIP and TIP. CDP is responsible for enforcing the policy decisions through configuration change obligations.

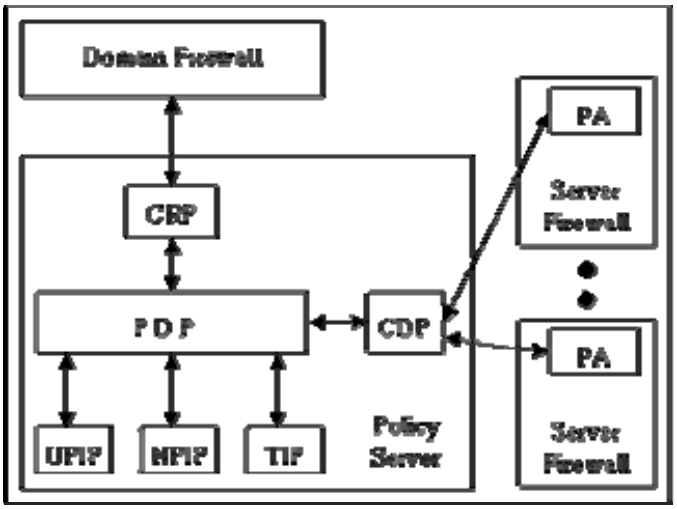

Fig.3 Policy server components

4.3 Policy Decision Making Process

This process is executed where the policy server makes policy decisions according to network and user policies using topology information. The whole process is described as follows: First, CRP sends a PolicyDecisionRequest message to PDP. Then PDP sends a PolicyRequest message to UPIP to obtain the user access rights policy for the user. UPIP returns a PolicyResponse message that includes user access rights information objects to PDP. In the next step, based on the information from UPIP, PDP sends TopologyInformationRequest message to TIP to obtain necessary topology information for the policy decision. Following this step, PDP requests from NPIP the policies of firewalls that are on the route to the servers, which user has rights to access. Then PDP evaluates policy statement using the information collected from UPIP, TIP, and NPIP. As a last step, PDP sends an XACML PolicyDecision to CDP.

4.4 Information Model

All the information points, namely UPIP, TIP, and NPIP, that exist within the policy server store their information in XML format. For UPIP and TIP, custom XML format is used whereas XACML specification is followed for NPIP. The details of each information structure are given below.

UPIP holds user access rights with respect to servers, for which the proposed DTD (Document Type Definition) is shown in Fig.4. It simply represents which user can access which servers. The representation indicates that a user can access any number of servers, which is a valid assumption. Users are recognized with their VPN client IDs whereas servers are identified with their IP addresses.

```
<?XML version="1.0"?>
<!DOCTYPE Access_Rights [
   <!ELEMENT Access_Rights (User, Server*)>
   <!ELEMENT User>
   <!ATTLIST  User VPN_Client_ID CDATA
#REQUIRED>
   <!ELEMENT Server>
   <!ATTLIST  Server  IP_AD CDATA #REQUIRED
                      Port CDATA #REQUIRED
                   Protocol CDATA #REQUIRED>
]>
```

Fig.4 A DTD for user access rights representation

TIP stores topology information mapping, which firewall protects which servers. Therefore, a topology element consists of one or more firewall

elements and a firewall element consists of one or more server elements. A firewall element has only an IP_AD (the IP address of the firewall) attribute, whereas a server element has three attributes: IP_AD (the IP address of the server), the Port (the port number it serves from) and the Protocol (the protocol it uses to serve). To represent such information, a DTD for topology representation is proposed here and presented in Fig.5.

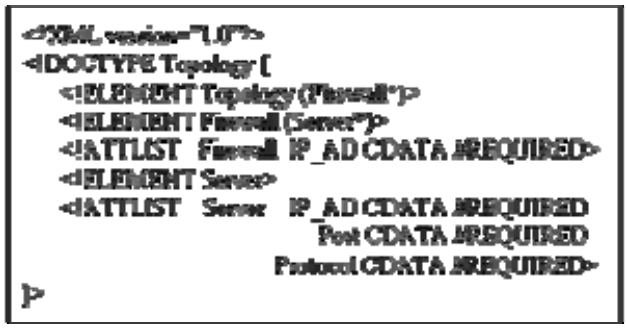

Fig.5 A DTD for topology representation

NPIP is responsible for holding network access control policies. As mentioned above, Tuglular et al. developed a method to represent high level firewall policies using XACML [26]. XACML is suitable as policy description language because it is able to represent relations among firewall policies and to embed other information related to a firewall into its policy such as; firewall IP address, agent identification, implementation independent representation of policy rules and rule combining algorithms. The same method is used here for NPIP to store firewall policies.

The XACML PolicySet element holds all the policies of firewalls deployed in the domain. The XACML Policy element encapsulates all the rules of a firewall policy and the XACML Rule element represents a single rule of a firewall policy. The order and action fields of a policy rule are represented as the attributes of XACML Rule element. The protocol information is stored in the XACML Rule Condition element. The src_ip and src_port information are held in a Subject element of the XACML Rule Target element. Similarly, the dst_ip and dst_port information are held in a Resource element of the XACML Rule Target element. The implementation independent representation of a rule is stored in the XACML Rule Description element.

A part of the XACML file is illustrated in Fig.6 given in Appendix. The part taken from the file shows the XACML representation of rule number 3 of the firewall number 2. Examples on the XACML elements explained above can be found in Fig.6. Furthermore, it is important to note that IP addresses and port numbers are represented with regular expressions.

4.5 Policy Agent

Policy agent plays an important role in reconfiguring firewall according to the policy decision made by the policy server. In other words, firewall adaptation to the new context is enabled by the policy agent. In this article, firewall policies are adapted by using one of the ways explained by Lymberopoulos et al. [17], which is the adaptation by dynamically changing the rules of a firewall policy to specify new configuration for the run-time of firewall. This policy adaptation process is sometimes referred as mapping high level policies to low level policies in the literature [1], [18]. In the proposed architecture, high level policies represented in XACML are mapped to machine dependent firewall rules with the help of policy agents.

4.6 Policy Adaptation Process

This process runs after the policy decision making process finishes and an adaptation requirement appears. The whole process is described as follows: When PDP sends a PolicyDecision message to CDP, CDP decodes this message and prepares a ConfigurationObligation messages for the firewall agents. CDP sends ConfigurationObligation messages to the firewall agents, which are within the scope of automatic reconfiguration. Firewall agent decodes the ConfigurationObligation message, updates firewall policy accordingly and signals the firewall to load the new policy file. When firewall finishes loading the new policy file, it signals back to firewall agent indicating that new configuration is up and running. Firewall agent sends a ConfigurationUpdated message to CDP. When all firewall agents return back to CEP with ConfigurationUpdated messages, CDP sends a ReconfigurationCompleted message to PDP.

5 Implementation

The security officer or system administrator responsible for policy management needs a management console, where she/he can administer firewalls, servers and users and their interrelationships. A prototype implementation of policy server management console is shown in Fig.7. This console is designed in such a way that

when an item from firewalls, servers, or users categories is selected, the console shows interrelated items from other categories. For instance, as seen in Fig.7 firewall FW1 protects servers Server2, Server3, and Server4. When a server item is chosen, the firewall that protects that server and user(s) that has access rights to that server will be highlighted.

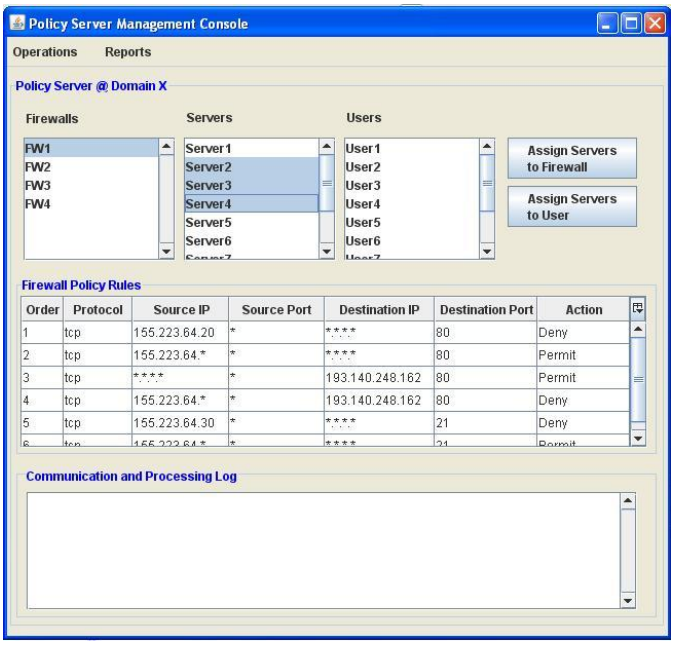

Fig.7 A prototype implementation of policy server management console

Same management console can be used to assign servers to a firewall and to assign servers to a user via the buttons on the right-hand side of the console. The prototype also includes a panel for firewall policy rules. When a firewall is selected, its rules can be seen in this panel. The last panel implemented in the prototype is the communication and processing log panel, which displays category item operations (i.e. adding or deleting a server), assignment operations (i.e. assigning a server to a user), firewall rule operations (i.e. updating a rule), and policy mapping operations (i.e. modifying a machine dependent firewall policy or configuration file).

A domain setting with 10 servers and 2 dynamic firewalls, each protecting 5 servers, is selected for the experiment. Each firewall is assumed to have 10 rules. While keeping these variables constant, the number of users trying to reach to the servers of the domain under consideration are let to range from 100 to 500 with 100 increments. Their user access rights policies are randomly generated. In other words, the number and order of servers that a user is authorized to access is determined randomly. The average processing time values for a policy decision operation is illustrated in Fig.8. a policy decision operation starts with PDP receiving a PolicyDecisionRequest message from CRP and ends with PDP sending a PolicyDecision message to CDP.

The experiment is repeated for 100 times for each user group. The values presented in the graph are the average of 100 values obtained for each group in the experiment.

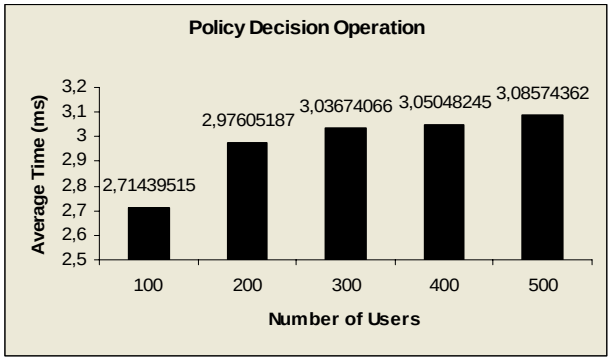

Fig.8 Average processing time values for a policy decision operation

The experimental setup, with 10 servers and 2 dynamic firewalls in each domain and the number of users ranging between 100 and 500, is assumed to be realistic for a corporation. Even for 500 users, the average processing time for a policy decision operation is around 3 ms for an ordinary computer.

6 Conclusion

This article describes a location aware user based network access control architecture which provides automatic reconfiguration of firewalls so that a corporate user requesting a VPN connection from any domain will have network access control mechanisms updated with respect to his/her server access rights when the VPN connection procedure is completed. The proposed architecture utilizes network access control and user access rights policies as well as network topology information to make a policy decision and then this decision is mapped to firewall policies through firewall agents. The whole firewall policy adaptation process is invisible to both users and servers.

The proposed architecture has a lot of features that requires further development. One important future work is to include a secure policy exchange protocol between the policy server and policy agents that provides authentication, data integrity, confidentiality and access control. Moreover, policy decisions made on location awareness can be extended to encapsulate not only access to servers

but also type of access to servers. Location awareness can be incorporated with temporal awareness, so that they both can effect policy decisions.

Although XACML 2.0 covers both core and hierarchical RBAC model, there is no publicly available open source XACML 2.0 implementation. Therefore, the access control policy mechanism is not based on a RBAC model. When XACML 2.0 implementation is available, the proposed architecture will be extended to implement both RBAC models. In addition, the future focus will not only be on firewalls but also on proxies and the architecture is planned to cover privacy protection policies through NAT and VPN. Real adaptable systems adjust themselves through feedbacks from their environment. The types of feedback to be collected from network access control environment is another research area to work on.

The lack of privacy may become the major pitfall of these context-sensitive systems [14]. Therefore, location-aware platforms should provide some privacy protecting functionality. They should not only focus on protecting the current position but also on the granularity of the protection means. The proposed architecture can be improved to satisfy privacy concern.

References:

[1] A. Apvrille and M. Pourzandi, XML distributed security policy for clusters, *Computers & Security*, Volume 23, Issue 8, December 2004.

[2] F. Barrère, A. Benzekri, F. Grasset, R. Laborde, and B. Nasser, Automated inter-domain security policy generation, *Proceedings of the 11th Annual Workshop of HP OpenView University Association*, June 2004, Paris, France.

[3] P. Charles and B. Daly, Rule Based Infrastructure: A Design and Runtime System for Enabling XML Schema Driven Applications, available at http://rosetta.sims.berkeley.edu:8084/docs/RuleBasedInfrastructure.html.

[4] J. Burns, A. Cheng, P. Gurung, S. Rajagopalan, P. Rao, D. Rosenbluth, A.V. Surendran, and D.M. Martin, Jr., Automatic Management of Network Security Policy, *Proceedings of DARPA Information Survivability Conference & Exposition II, Anaheim*, CA, USA, 2001.

[5] N. Damianou, *A Policy Framework for Management of Distributed Systems*, PhD Thesis, Imperial College, London, March 2002.

[6] N. Damianou, A. Bandara, M. Sloman, and E. Lupu, A Survey of Policy Specification Approaches, April 2002, available at http://www.doc.ic.ac.uk/~mss/MSSPubs.html.

[7] P. Flegkas, P. Trimintzios, G. Pavlou, and A. Liotta, Design and implementation of a policy-based resource management architecture, *Proceedings of IFIP/IEEE Eighth International Symposium on Integrated Network Management*, 2003.

[8] M. Giordano, G. Polese, G. Scanniello, and G. Tortora, Visual modelling of role-based security policies in distributed multimedia applications, *Proceedings of IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004.

[9] Y. Han, *Context Aware Security Policy Enforcement: CASPEr*, Master's Thesis, Technische Universiteit Eindhoven, 2005.

[10] V. Hays, M. Loutrel, and E.B. Fernandez, The Object Filter and Access Control Framework, *Proceedings of PLoP 2000*, Monticello, Illinois. 2000.

[11] V. Kapsalis, D. Karelis, L. Hadellis, and G. Papadopoulos, A context-aware access control framework for e-service provision, *Proceedings of (ICIT 2005) IEEE International Conference on Industrial Technology*, 2005.

[12] R. Kay, Analysis: XACML, 31/12/2003. available at http://www.computerworld.com.au/index.php? id=997396262&fp=16&fpid=0.

[13] G. Kim, H. Bang, J. Na, and J. Jang, The Development of Policy Proxy Agent in Policy-Based Network Security Management System, *Proceedings of the 5th WSEAS Int. Conference on Information Security and Privacy*, Venice, Italy, November 20-22, 2006.

[14] P. Langendoerfer, and K. Piotrowski, More Privacy in Context-aware Platforms: User Controlled Access Right Delegation using Kerberos, *Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers*, Tenerife, Spain, December 16-18, 2005.

[15] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shah, First experiences using XACML for access control in distributed systems, *Proceedings of the 2003 ACM workshop on XML security*, Fairfax, Virginia. 2003.

[16] G. Lopez, A.F. Gomez, R. Marin, and O. Canovas, A Network Access Control Approach based on the AAA Architecture and Authorization Attributes, *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[17] L. Lymberopoulos, E. Lupu, and M. Sloman, An Adaptive Policy-Based Framework for Network Services Management, *Journal of Network and Systems Management*, Volume 11, Issue 3, September 2003.
[18] S. Omari, R. Boutaba, and O. Cherkaoui, Policies in SNMPv3-based Management, *Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management*, Boston, MA, USA. 1999.
[19] OASIS, eXtensible Access Control Markup Language (XACML), available at http://www.oasis-open.org/committees/xacml/.
[20] S. Payette and C. Lagoze, Policy-Carrying, Policy-Enforcing Digital Objects, *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries, LNCS*, Vol. 1923, 2000.
[21] T. Ryutov and C. Neuman, The Specification and Enforcement of Advanced Security Policies, *Proceedings of Third International Workshop on Policies for Distributed Systems and Network*, 2002.
[22] M. Sloman and E. Lupu, Security and management policy specification, *IEEE Network*, Volume 16, Issue 2, March, 2002.
[23] T.J. Smith and L. Ramakrishnan, Joint Policy Management And Auditing In Virtual Organizations, *Proceedings of Fourth International Workshop on Grid Computing*, 2003.
[24] M. Spremic, and M. Popovic, Towards a Corporate IT Risk Management Model, *Proceedings 6th WSEAS International Conference on Information Security and Privacy*, Tenerife, Spain, December 14-16, 2007.
[25] SUN, Sun's XACML Implementation, available at http://sunxacml.sourceforge.net/.
[26] T. Tuglular, F. Cetin, O. Yarimtepe, and G. Gercek, Firewall Configuration Management Using XACML Policies, to be presented in *13th International Telecommunications Network Strategy and Planning Symposium*, Sep. 28 – Oct. 2, 2008, Budapest, Hungary.
[27] A. Virmani, J. Lobo, and M. Kohli, Netmon: network management for the SARAS softswitch, *Proceedings of IEEE/IFIP Network Operations and Management Symposium*, 2000.
[28] T. Yoshioka, T. Igakura, and T. Tonouchi, Policy-based Automatic Configuration of Network Elements in Separate Segments, *Proceedings of Asia-Pacific Network Operations and Management Symposium*, Oct. 2003.

Appendix

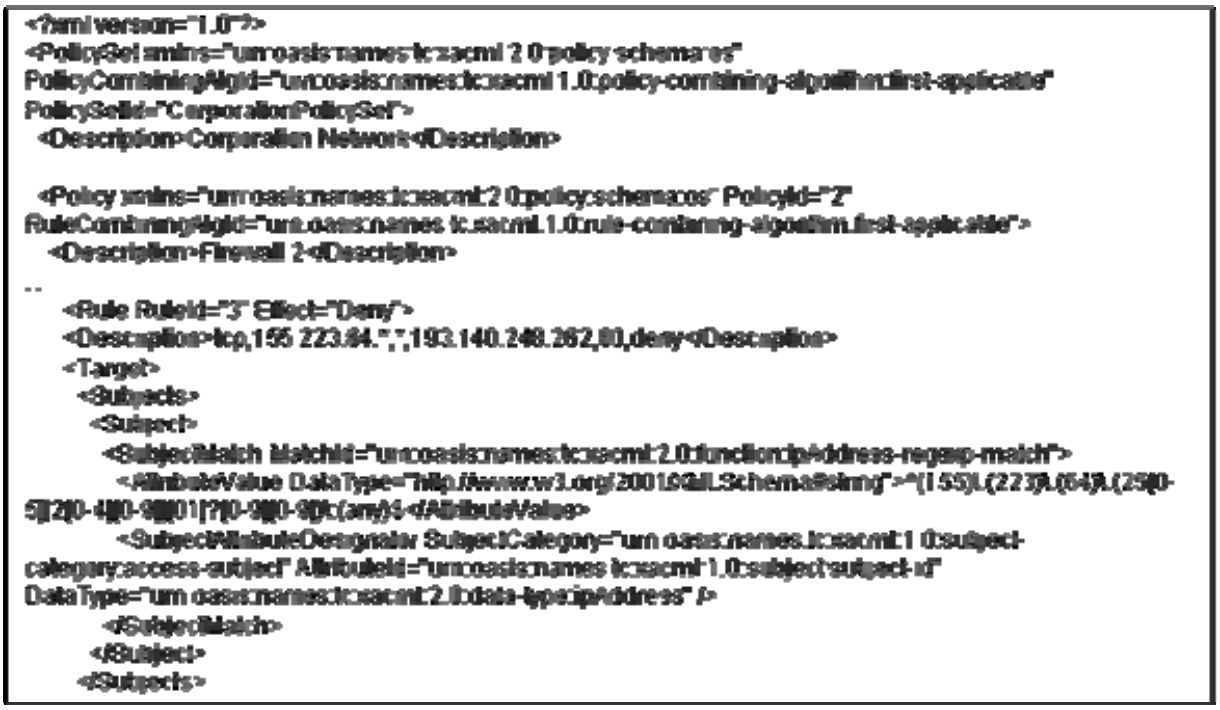

Fig.6 Example XACML representation of a firewall policy rule

```
<Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:2.0:function:ipAddress-regexp-match">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">^(193)\.(140)\.(245)\.(252):(80)$</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="urn:oasis:names:tc:xacml:2.0:data-type:ipAddress" />
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <SubjectAttributeDesignator AttributeId="protocol"
DataType="http://www.w3.org/2001/XMLSchema#string" />
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">tcp</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
..
 </Policy>
..
</PolicySet>
```

Fig.6 Example XACML representation of a firewall policy rule (continued)