Enterprise firewall virtualization design

Ari Sujarwo[1, *] and *Jefferson* Tan[2]

[1]*Universitas Islam Indonesia, Jalan Kaliurang Km.14,5, Sleman, Indonesia*
[2] *Monash University, 900 Dandenong Rd, Caulfield East VIC 3145, Australia*

Abstract. Grid computing could consists of thousand of computers across the world with the role as the resource provider or Grid users. This conditions make a challenge for engineer to keep the Grid secure but at the same the Grid must be easily joined by clients and providers. This paper aims to propose a method to control enterprise firewall which is basically a static engine to be a dynamic engine to cope with the Grid needs. At the end of the paper, the results shows that this method is works and was tested using a real enterprise firewall.

1 Introduction

This paper proposes the design of *enterprise firewall virtualization* for Grid. The enterprise firewall virtualization is a composite system that is designed to manage a firewall via a firewall agent for Grid computing. The Grid needs a more advanced security system to filter the incoming traffic than the current security system. Currently, the Grid computing system is secured by a basic certificate-based authentication system and a static firewall. This static firewall possibly raises an open pinhole problem caused by several ports, which is configured to remain open. Globus recommended many ports to remain open when the Grid in operation: as much as 10 for each connecting client [1]. These open ports could be the way attackers compromise the security of Grid.

Several research projects have been conducted to propose an idea to solve such a problem for example: Port knocking approach [2], transparent firewall virtualization [3], and a firewall virtualization that works based on RADIUS [4].

The concept of firewall virtualization itself is the basis for the work Tan in the Romulus project [5]. Before Romulus, Tan [6] had worked on Remus, which is a firewall traversal method based on tunneling over SSH protocol. The project found that tunneling over SSH introduced performance degradation compared to the direct connection caused by the encapsulations overhead during tunneling process [6]. The latency was increased about 25% of the direct connection latency. Based on that fact over Remus, this paper try to propose a technique that might be another approach than Remus. Here in this project, another model based on Romulus is proposed to solve the problem by managing an enterprise firewall based on the client's details that can be identified during the authentication, authorization, and accounting (AAA) processes.

Several components are involved in this model. The firewall agent which manages the firewall consists of an authentication server and some other components. This authentication server validates the credentials of each incoming user. Moreover, the user details are extracted from the AAA logs. Another component formulates and pushes the appropriate firewall rules out to the firewall.

2 Firewall Virtualization System Architecture

As briefly explained above, this system consists of several cooperating components. Each of them has a responsibility to support the entire system in meeting its aim. Fig 1 shows the steps on how the system works, while Fig 2 shows the network architecture.

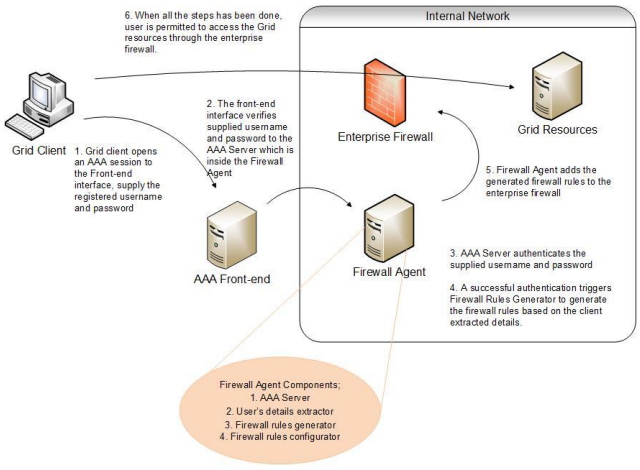

Fig 1. Firewall virtualization diagram

* Corresponding author: ari.sujarwo@uii.ac.id

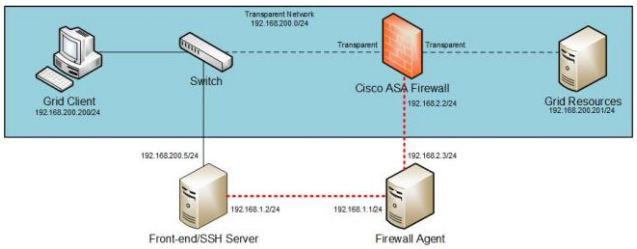

Fig 2. Network architecture

As shown on the figure, the authentication process has a role to validate the incoming user. The incoming user which provides a credential must be matched with the known user credentials to ensure that this is a valid credential holder [7]. It is essential to identify Grid users. Since the firewall is dropping all incoming data to the Grid by default, a *pinhole* is needed to enable the authorized client to access the Grid resources. The firewall rule that is used to open the pinhole is generated based on the client's details. This situation makes successful user identification compulsory. Once a client is authenticated, the following phases are authorization and accounting. The client's details from the AAA phase are then extracted to build the pinhole opening rules. After the firewall rule is added to the firewall, the Grid client is allowed to access the Grid resources. The authentication system explained above is used to match the credential. However, this authentication system needs a front-end to let user enter the username and password. Among authentication protocols, there are some *front-end* components available based on the *web interface* and *command line interface (CLI).*

Selecting an appropriate front-end depends on the features of the authentication engine itself. Some AAA protocols have many plug-ins developed to work with, for example, a combination of secure web interface and AAA as implemented in the previous research [4]. Other AAA systems may use simpler applications as the front-end. This kind of AAA can use a common operating system console as front-end for users to supply the credentials. The console-based front-end applies to a server which provides command line-based remote access for a Grid user. The basic function of this server is to enable users to get in to a server in secure mode then this occasion written to the accounting logs.

Logs are crucial. They record almost all activities happening within the system. There are so many logs in an operating system that each of them tells of specific events regarding the configured services. The most highlighted log in this project is accounting log. This log records when a Grid user starts and stops the session after being authorized. This log provides each user's IP Address and username to be used to generate firewall rules. The agent manages the firewall by configuring firewall rules based on user identity so that the identification of the user is the most crucial step. First, firewall agent must be able to get the user details from the log. As explained above, the log contains the essential information relating to authenticated users such as IP address and username. Those details are extracted and then used to generate the firewall rules which are the second function of this component.

To extract the user details from the log, the generator will search for keywords that contain relevant information. The user's source IP address is used as the source address parameter in the firewall rule, and then the username is used as the firewall rule's name. However, since the Grid resource has a fixed address, the destination address in the firewall generation would use a single IP address.

3 Enterprise Firewall

An enterprise firewall is typically used to protect a large production system. Such a large organization needs a powerful security device which protect their system professionally and is strong enough to filter any incoming or outgoing data. Grid infrastructure today is no longer a prototype system. It has been long considered a production system which provides users substantial computing resources. It is no longer a simple system which serves a few users, but thousands of users are now connected to the Grid, wherever it is. A simple one layer security is no longer suitable to Grid infrastructure needs.

The firewall performs traffic checks during its operation. The firewall drops all traffic except the packets which are sent by an authenticated user. The traffic which comes from an authenticated user match with a rule or a policy implemented by the firewall. The firewall rules contain platform-specific syntax to allow the authenticated client's traffic to pass the firewall. Up to now, the manufactured firewalls work based on statically configured rules. An administrator must get into the firewall, and then put some rules to implement what is desired.

However, to make a firewall more dynamic to allow traffic based on user needs, the firewall must be managed by the agent. The firewall is configured to enable client's traffic to pass the firewall. The firewall rules that the firewall agent prepares needs to be configured and activated on the firewall. Once the firewall is configured with the new rules, the matched traffic are permitted to pass the firewall.

The firewall virtualization is designed to be fault tolerant. As normal equipment, the Grid components always have risks of system failures for various reasons. The recovery strategy is focus on the management of the enterprise firewall. When the firewall goes down, the current firewall configuration might be lost. Since the firewall usually works in a dynamic way, almost all access control lists rules are configured remotely by the firewall agent. The firewall has only the basic pre-configuration that contains a basic communication configuration that enables the firewall to be accessed remotely by the firewall agent, and basic and static access control lists rules that drop all incoming packets unless allowed by the agent.

3.1 Firewall Pinhole Opening Steps

In the authentication procedure, Client sends a request to open firewall. To do so, the client must run two steps in

his client machine: open an AAA session to the front-end interface, and supply the username and password, and then open the Grid client application. The AAA front-end allows the user to enter his credential, at the time, when the firewall is still dropping the packets (Fig 3 Step 1). In the background, the supplied username and password (Step 2) is authenticated by the AAA server inside the firewall agent host (Step 3). The AAA server validates the incoming credentials with the database. If the authenticating user is identified as a valid user, the AAA server grants the user specific rights as an authenticated user. Otherwise, the AAA server rejects the authentication request and writes the incident to the log file.

A successful authentication process brings the user to the authorization process, which is a granting step to give the user rights for a further step. The next step is accounting. The username and the IP address from which users are logged-in from, are recorded in the accounting log. This accounting event triggers the user's detail extractor to obtain all the details and runs the firewall rules generator to create the firewall rules. After the firewall rules have been generated, the agent opens the management session to the firewall. The firewall agent adds the generated firewall rules to the enterprise firewall to achieve the opening of the desired firewall ports (Step 4). Step 5 is the step when users are able to access the Grid resources.

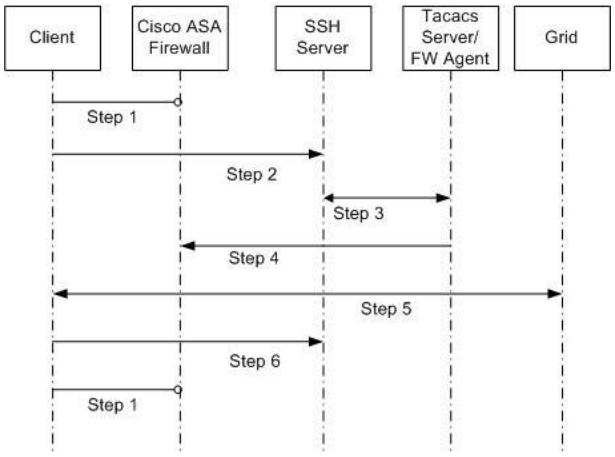

Fig 3. Pinhole opening steps

3.2 Ending Firewall Pinhole Opening Session

The pinhole is open during the Grid user activity and can close at any time when the Grid user wants. This firewall virtualization has a simple design. It makes it convenient for the Grid user to authenticate himself to the system. The way to end the session is designed in a simple way as well. The user just needs to type **exit** on the opened AAA session. This command sends the message to the AAA server to close the session for this user (Fig 3 Step 6). The AAA server will then record the timestamp of when the session closed, and triggers the firewall agent to remove the user-related firewall rules from the enterprise firewall.

3.3 Cisco ASA with Clogin

This section proposes the design and implementation details of enterprise firewall virtualization which manages a Cisco ASA enterprise firewall unit by a firewall agent. Several prepared components work to build such a firewall agent. The firewall agent consist of an authentication server which validates each of incoming user's credential, couple of scripts which work to fetch user details from accounting log file, and a mechanism to write the firewall rules to the firewall engine.

The Cisco ASA firewall is placed as a security check point between client and the Grid system which prevents unauthorized access by client. A client will need to pass the firewall through the open pinholes before it is able to send any data to the Grid. The pinhole itself opened by Clogin. Clogin is an application that enables an application or a system communicate with a Cisco device interactively. Clogin installed within the firewall agent machine, and the only established communications are with the Cisco ASA firewall and the SSH Server.

Clogin has a mechanism to configure a Cisco device remotely from a Linux operating system where it is installed. Based on the Ubuntu manual, Clogin is explained to be an *Expect script*, which is a script that automates a logging activity and configure a device. Clogin works in two steps: logging in to the Cisco device and executing the written script that contains new device instructions. To be able to log into a Cisco device, the Clogin reads a configuration file which contains credentials to enter the Cisco device. This file is a hidden file, which is secretly secured from unauthorized access other than Clogin. Several preprocessing are applied to this file before the Clogin ready to be used.

3.4 Firewall Rules for Grid

Based on the Globus toolkit document [1], a Grid needs at least 10 ports to be open per simultaneous user to let the functions of Grid work. Considering this condition, the default firewall settings which are mostly static will either conflict with Globus recommendations or make the firewall always to open on those ports.

In the Globus-toolkit, among those 10 recommended opened ports, there are three ports which open for Grid basic services: *Grid resources Allocation and Management (GRAM)* which work over port 2119 and 8443, and *GridFTP* which needs port 2811. Those ports will be facilitated by firewall agent to be opened.

4 Performance Evaluation

The performance of firewall virtualization was measured by two different tests. The first test was measuring the consumed time when a Grid user was requesting a pinhole opening. The second test was measuring the performance of the clogin script to generate the firewall rules and reconfigure the firewall.

There are only two steps of test which shows that a hundred percent of service has been delivered: when there is only one user requests and a maximum of a couple users request at a time. This means that before any further investigation conducted in the future to resolve the problem, these tests limit maximum of two user can send

requests at a time to get the system works properly. Results of the tests can be seen in Fig 4 and Fig 5 below.

Figure 4 shows the performance results of the new approach. The graphics shows that the system works with some notes:

1. The approach send firewall rules by Agent to the enterprise firewall. Since the rules were not preprocessed (Figure 5), and delivered as is, it makes packet data massively moved by Agent when user increased.
2. The massive amount of rules generated by Agent reduce the preformance of system delivery.
3. Figure 6 and 7 shows how the system tested in security test environment to show pinhole were successfully opened and closed. It allows data moved while open and reject data while port closed.

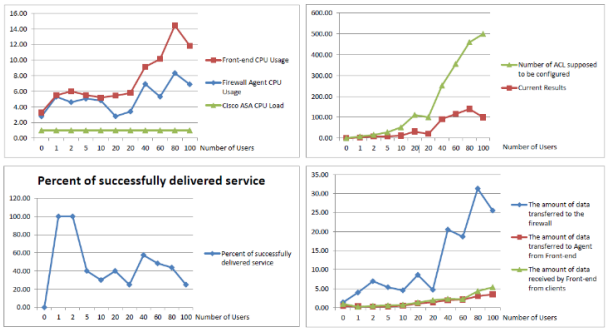

Fig 4. Stress test results

```
ciscoasa# Trying 192.168.2.1...
telnet: Unable to connect to remote host: Connection refused
spawn conf t
ciscoasa(config)#Start akses dari user: griduser11 via: ssh alamat ip: 192.168.200.200
 ssh -c 3des -x -l rancid ciscoasa
  access-list griduser1 remark Rule(s) for user griduser1
ciscoasa(config)#  access-list griduser1 permit tcp 192.168.200.200 255.255.255.255 any eq 2811
ciscoasa(config)#  access-list griduser1 permit tcp 192.168.200.200 255.255.255.255 any eq 8443
ciscoasa(config)#  access-list griduser1 permit tcp 192.168.200.200 255.255.255.255 any eq 2119
ciscoasa(config)#  access-group griduser1 in interface outside
Tue Jun 18 11:18:17 EST 2013
```

Fig 5. Clogin controls the firewall

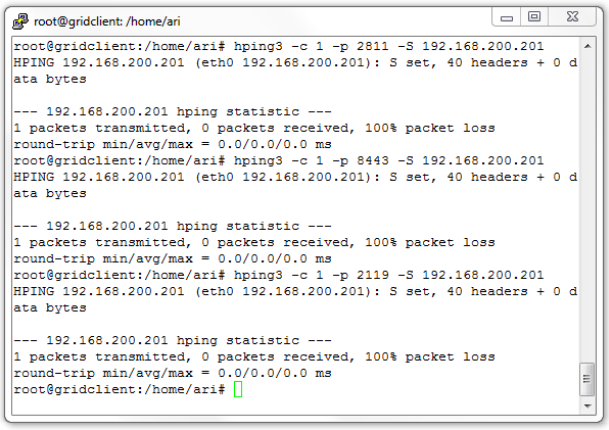

Fig 6. Port detected as closed before pinhole opening

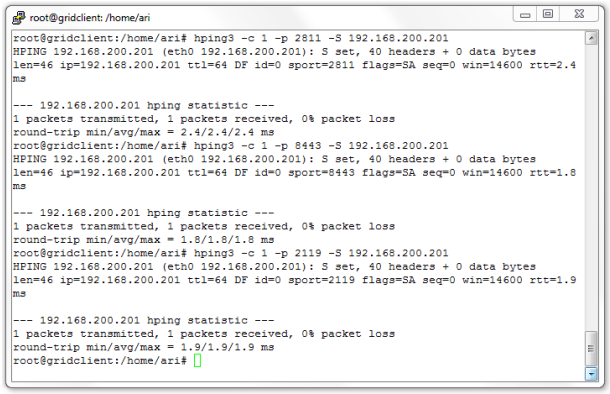

Fig 7. Port detected as open after pinhole opening

5 Conclusion

This paper shows the enterprise firewall virtualization model which worked as a group of security elements. The firewall virtualization infrastructure contained an enterprise firewall device which separated the network of the Grid resources from the outside, a firewall agent which offered firewall management based on Authentication, Authorization, and Accounting (AAA), and a front-end machine which accepted credentials which were supplied by Grid clients. Firewall virtualization was proposed as an advanced security solution for Grid infrastructure, and offered a controllable firewall which is located in front of the Grid.

References

1. V. Welch and O. Mulmo, Usg. Glbs. Tlkit. Frwlls (2006)
2. W. Lian, Prt. Knckng Auth. Mechm. Grd. Svc. Mon. Uni (2011)
3. G. Vilcans, Transp. Frwll. Virt. Grd. Cls. Mon. Uni (2011)
4. Mahjoubi, P. RAD. Frwll. Virt. Grds. Mon. Uni (2010)
5. J. Tan, D. Abramson, and C. Enticott, A virt. Con. lyr grds. E. Sc (2009).
6. J. Tan, D. Abramson, and C. Enticott. REMUS: a rer. and mult. Sys. Grd. Con. Acr. Frwlls. J. Grd Com., vol. 7, no. 1, pp. 25–50, (2009)
7. C. Metz. AAA prot. Auth. Autho. Acc. Int. IEEE Internet Comput., vol. 3, no. 6, pp. 75–79 (1999)