A Proposed Approach for Management of Multiple Firewalls Using REST API Architecture

Mohamed E. Elhamahmy[1] , Mohamed M.A. Elgazzar[2], Abdel-Hamid M. Emara[3]

[1]*Dr. Information Technology,, Cairo, Egypt.*
[2]*Asso.Prof, Higher Institute of Computer Science and Information Systems, Fifth Settlement, New Cairo, Egypt.*
[3]*Department of Computers and Systems Engineering, Faculty of Engineering, Al-Azhar University, Cairo, Egypt.*
[4]*Department of Computer Science, Taibah University, Medinah, Kingdom of Saudi Arabia.*

Abstract *— a lot of work has been done on managing firewall policy anomalies. There are tools have been proposed to assist subordinate on these considers, for example "Policy Advisor tool". In any case, it subordinate on the addition of policy rules manually into the firewall device. There's a genuine require for a tool that acquire the firewall policy rules in real-time without affecting its configuration. There are moreover devices created by firewall sellers created for firewall frameworks to work on the administration of their appliances as it were and so does not work with the devices created by other companies. There is a need for a tool to manage different firewalls. In this paper, a new approach is proposed for acquiring a firewall policy objects in real-time. Then present the obtained policy rule-set in a visualization and usability perspectives using REST API architecture. We also built a practical tool FPM (Firewall Policy Manager) based on the proposed approach. It is an assistant tool for the administrator to connect centrally to each firewall in the network and get its policy rules safely without changing its configuration. It also provides a visual interface for assisting configuration auditing tasks at the standpoint of visualization and usability. A proposed test environment based on virtual machine technology for testing the FPM against the most commonly used firewalls is described as well. FPM is tested in a real environment and the results were satisfied by the administrator. The proposed approach is the step towards investigating the different firewall policy rules using web services for anomaly detection and correction. However, in this paper, it is focused on the policy rule acquisition in real-time using the proposed FPM tool.*

Keywords - *Firewall; rule; policy; API; REST API*

I. INTRODUCTION

Nowadays, Firewalls manage huge traffic across enterprise networks. The security policy implemented in these firewalls consists of hundreds of rules and objects. The firewall rules included groups of servers, user machines, sub-networks in the data center and network of company branch offices that have one or more Demilitarized Zone (DMZ) [1]. Many firewalls assumed by the same corporate to protect its perimeters [2]. The firewalls rules and policies need to be frequently changed and modified because of the dynamic needs of its organization business [3]. On the other hand, there is a high demand for a single tool, brand independent, to manage different firewalls. The required tool should stands for usability and visualization to assist administrators to review the policy rules for multiple protection devices using one tool and conveniently style. On the other hand the probability of policy rule anomalies may be increased. This may lead to network security threats [6]. These multiple firewalls assumed to be different brands so that, it thus becomes hard to centrally manage these firewalls [4]. The traditional updates of its policy rules using command line interface (CLI) require human intervention and may be error prone [5]. There are web based tools to manage the recent firewalls. However, these tools accompany a dedicated product exclusively, and couldn't be used for accessing other different versions or brands.

A. Problem statement

There is a need for an effective firewall management tool [2]. This would help administrators in keeping the firewalls they have, refreshed and their approach rules reliable. Notwithstanding keep away from monotonous works of dealing with every one of the firewalls exclusively. As, these days, the most commonly used firewall vendors making a web based tool available for editing their firewall products. Besides, much of the time changes in corporate necessities and the progressions required in arrangement tenets to mirror the corporate needs are redundant undertakings. Consequently, a compelling firewall the board instruments are required so as to mechanize the dull errands and to spare both time and assets [5]. Thus, in this paper, it is intended to essentially propose the

required tool for helping the administrator in administering multiple firewalls to the extent of usability and observation. The proposed tool should help the administrators in upgrading their firewall policy rules. It stays away from monotonous works of dealing with every firewall utilizing its packaged tools solely.

B. Scope of work

In this paper, we have focused on a proposition to implement a multi-layered model to recover policy rules from numerous firewall gadgets. Just as building a down to earth device dependent on the proposed introduction. At that point test this device with some regular firewall equipment by building a suitable testing condition. It ought to permit adding firewalls to the current rundown. It likewise encourages filtering rules for the anomalies. Lastly a genuine trial of the proposed device in a server farm through the framework administrator.

II. RELATED WORKS

There are many studies focused on firewall management. In [2], they aimed to identify existing solutions that help professional and non-professional users to create and manage firewall configuration files and to analyze the proposals in respect of usability. They also proposed taxonomy of existing solutions as well as a synthesis and in-depth discussion about the state of the art in firewall usability. Over the last years, there have been many tools for managing firewall policy rules [1-5]. However, the method of acquiring policy rules was not presented.

In [3], the researchers aimed to define a required firewall management tool from different suppliers that would reduce the work of frequent companies, and be a reliable alternative to the tools that accompany each product exclusively. The study involved the collection of the required tool features from network administrators, as well as a survey of previous studies and the most frequently used commercial products such as Tufin and Algosic. So they fill in as the design features of the ideal tool. However, their investigation did exclude the real development of the firewall the executive device nor proposed an approach to gain the policy rules from various firewall gadgets. They additionally inferred that there an exceedingly expanded requirement for that tool and they depicted unequivocally the prerequisites that ought to be indicated by that tool. In this paper, it is proposed a graphical user interface (GUI) based tool that can acquire the security rules from various sorts of firewalls through web service architecture - so as to facilitate testing and recognizing anomalies and prescribe the remedial activity so as to determine exact security policy. Another reference [6], in their study, they proposed an analysis tool to visualize segment-based firewall rules to facilitate verification of the current control condition. The proposed tool analyzes the control conditions of packets automatically, thereby eliminating the need for manual inspection as before, and displays the conditions with a visualization model to allow them to be easily verified. However, they did not focus on how to acquire the policy from multi-firewall. Most of them involve some versions of CLI scripting since nearly all firewall features and functions are available in this manner [5]. Perl and Expect scripts are the common scripting languages in use for managing traditional firewalls [6]. Historically the traditional firewalls have been a closed platform in the sense “Syslog” has been the only choice for event management. Recently, fully functional virtualized firewalls were introduced [7]. Combined with this virtualized offering and the increase in Software Defined Networking (SDN), and “Network Function Virtualization (NFV)”, the next logical step was to develop an API to manage firewalls [8]. This gives administrators a very flexible platform where common elements could be created and re-purposed to orchestrate and automate deployments [9].

III. THE FIREWALL POLICY MANAGEMENT

A. Firewall policy rules

A firewall is a security system that expected to protect private network edges. It controls the approaching and active network traffic by sifting the information bundles as per requested principles to decide if they ought to be permitted or denied.

<Order><Prot.><Src. IP.><Src. Port><Dst. IP.><Dst. Port><Action>

FIGURE (1): THE POLICY RULE STRUCTURE

The network traffic is matched against a list of firewall rules in a sequence, or chain, from first to last. More specifically, once a rule is matched, the associated action is applied to the network traffic. The ordered rules are predefined according to the security requirements of the private network owner or organization. It includes the rule number in sequence order, as shown in the above Figure (1). Each rule has an action value which may be either (Allow) or (Deny).

For example, the rule “R1” may be as following:

<R1><TCP><ANY><ANY><ANY><ANY><ALLOW>

It permits all TCP traffic from any source IP address, using any port to pass into any destination IP of any port. This is the most exceedingly awful kind of access control rule. It repudiates both of the security ideas of denying traffic as a matter of course and the primary of the least benefit. The destination port ought to be

constantly indicated, and the destination IP address ought to be determined when down to earth. The source IP address ought to be determined except if the application is worked to get customers from the Internet, for example, a web server. A decent guideline would permit any incoming TCP traffic from any address using any port to access the web server named (WEB-SERVER1) through its HTTP (Hyper Text Transfer Protocol) ports as following:

<R1><TCP><ANY><ANY><WEB-SERVER1><HTTP><ALLOW>

Configuring security rules to uphold traffic governs in a system can be moderately simple yet requires watchful thought. There are several best practices to use when defining an effective firewall policy to optimize policy configuration. These days, it's wont to have multiple firewalls of various brands to secure a network. The contradiction between some rules on different firewalls additionally to those on the same firewall delineated a security issue. The traditional access of the firewall configuration using the Telnet protocol or CLI did not allow the administrator to review the policy rules [5]. In addition to the necessity for using an accompany firewall tool to handle every firewall policy rules representing a burden on the network administrator.

B. Web services (WS)

WS are the alternative for the CLI. They may be either Simple Object Access Protocol (SOAP) or Representational State Transfer (REST). SOAP is an Extension Markup Language (XML) based protocol for accessing WS. It is a platform independent and language independent as well. It also allows interacting with other programming language applications. REST is an architectural style, not a protocol, as a web service, it is faster than SOAP because there is no strict specification like SOAP [10]. It consumes less bandwidth and resources as well. REST WS are both language and platform independent. They can be written in any programming language and executed in any platform. In comparing with SOAP, REST WS can use SOAP WS as the implementation. However, SOAP cannot use REST because it is a protocol. One more advantage of using REST API architecture, it permits different data formats such as Plain Text, Hyper Text Markup Language (HTML), XML, and Java Script Object Notation (JSON). Therefore, REST is more preferred than SOAP in implementation of the proposed tool because it is easier, faster and an architectural style.REST API Design was defined by Dr. Roy Fielding in his 2000 doctorate dissertation [11]. It is notable for its incredible layer of tractability. Since data is not tied to methods and resources, REST has the ability to handle multiple types of calls, return different data formats and even modification structurally with the correct implementation of hypermedia [10].

C. Visualization and usability

The objective is to enable administrator to construct mental pictures of the most scattered pieces of firewall policy rules designs using GUI. The perceptions accumulate data that are dissipated over the distributed firewalls over the network and present them to the administrator in a setting proper manner [6]. Comparing to the traditional methods of accessing the firewall configurations such as using Telnet or secure shell (SSH) applications did not satisfy either of usability or visualization [12]. Therefore, one of the most important determinants of the design of the proposed approach is to rely on a visual interface and always maintain a copy of the policy rules.

The rest of this paper is organized as following, the next section describes the proposed model for managing multiple firewall devices, including building a practical tool based on this approach, and then building an appropriate test environment for testing the practical tool named FPM. Then, the following section examines the test results including the results of a real test in one data center using the proposed tool. The results are then discussed and the conclusion and future work presented.

IV. THE PROPOSED APPROACH

A layered framework is a framework involved layers, with each layer having a particular usefulness and obligation. Each layer has its very own obligations, with the models containing how the information ought to be framed, the controller concentrating on the approaching activities and the view concentrating on the yield. Each layer is independent yet additionally collaborates with the other.

In REST API design, a similar standard remains constant, with various layers cooperating to construct a chain of command that makes a progressively adaptable and secluded application. Based on the properties and parameters of a firewall design tool in [3] a multi-layers based model was proposed to achieve these determinants.

The trend towards a multi-layered model is also consistent with REST API architecture. To connect with the firewall device using WS it should ask for the authentication first. The handling is maintained through secure web protocol (HTTPS). Then the configuration files should be issued by the authenticated user. The firewall then responds by sending the policy rules as a JSON file. Then the obtained files should be prepared for presenting through a GUI standing for visualization and usability.

A. The proposed model

As sown in Figure (2), the proposed approach for managing the firewall security policy is composed of a four-layered conceptual model.

The proposed model should stand for usability as well as visualization. Data acquisition is the preliminary step in visualization [1]. So, the data acquisition layer is that layer to interact with the firewall asking for its configuration.

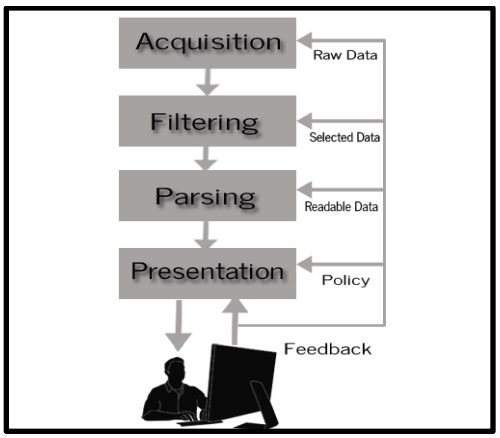

FIGURE (2): THE PROPOSED FIREWALL MANAGEMENT APPROACH

The firewall policy rules are the data of interest in this scope. The policy rules are obtained from the firewall by the administrator. He should use suitable credentials for firewall authentication access. Then the suitable command should be issued for acquiring the policy rules. The second layer is the filtering. Data filtering is used to extract the portion of data of interest to be visualized. The third layer is the parsing layer. Data parsing is used to transform selected objects extracted from the returned files into readable data. The last layer is the presentation layer. It is aimed to represent the obtained firewall policy rules in GUI. The presentation step is used to bring realism in data.

B. FPM tool based on the proposed approach

An implementation of a tool based on the proposed model is demonstrated. It is a tool to be used by the administrator in a network. The (FPM) tool aimed to assist the administrator for the central management of multi-firewalls (may be of different vendors). It stands for design principals of usability and visualization. It is safe as it just retrieves a copy of the firewall configuration without changing on the resources. It uses WS instead of CLI. The data acquisition of firewall policy rules issues API scripts over the HTTP protocol to authenticate then asking for the firewall policy. It returned with its policy on a form of JSON file. JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types.

As appeared in the Figure (3), the proposed model depends on a few layers. The first layer in which the demand for the required firewall configuration files is issued from the system overseer. At this stage the gadget IP address, username and password must be controlled by the system administrator for confirmation reason. At that point once the validated pass checked, the administrator utilized the FPM should start the reasonable (HTTP) asks. The HTTP asks for (Get/Post) are started to get to the focused on firewall. At that point, the firewall items ought to be returned in a type of JSON record. The JSON document ought to be passed to the appropriate parser module. The parsing of returned records varies as indicated by the firewall item demonstrate.

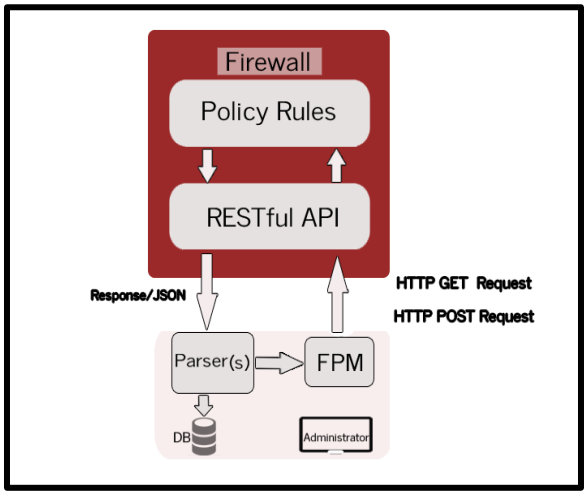

FIGURE (3): THE PROPOSED MODEL OF FPM

The parser organized the information important to make it proper for both introducing through the FPM tool, and to be spared in a database for later on check and reviewing reason. The parsing layer is vital as the treatment of the returned JSON document varies relying upon the firewall model/vendor just as dependent on its working operating system form also.

C. Building a test environment for evaluating FPM tool

In the wake of building a practical tool based on the proposed approach for firewall management called FPM we have to test this tool. Four firewall gadgets have been chosen from various providers, all of which

supported by REST API architecture. For example, Cisco, Juniper, Palo Alto, and FortiGate are chosen in type of virtual machines (VM) for assessing the proposed FPM tool. It is meant to locate a virtual machine to speak to the firewall that underpins the REST API architecture. As appeared in Figure (4), the four models are chosen identified with various vendors. The firewall show Cisco Adaptive Security Appliance (ASA) series v9.4 is added to the proposed tool FPM. In a word, Cisco ASA is a security gadget that joins firewall, antivirus, interruption aversion, and virtual private system (VPN).

Cisco series firewalls version 9.4(x) and higher are included the REST API architecture. The second firewall gadget added to the proposed model FPM is FortiGate firewall operating system (FortiOS) version 5.4.1. FortiGate is a brand that creates and showcases digital security programming, apparatuses and administrations, for example, firewalls, hostile to infection, interruption anticipation and endpoint security. FortiGate series version 5.4.1 and up are supported by REST API design. Juniper Network Operating System (JUNOS) Series Services Gateways (SRX) version SRX-15.1X49-D154 is added to the FPM too. JUNOS is the FreeBSD-based working framework utilized in Juniper Networks equipment switches and firewalls. SRX (Series Service Gateways) depend on JUNOS demonstrated working framework which conveys security and propelled assurance administrations. The chose adaptation that upheld REST API is added to the proposed instrument FPM. The last firewall gadget added to the proposed instrument is Palo Alto operating system (PAN-OS) version 8.0.0. Skillet operating system is the product that runs all Palo Alto Networks cutting edge firewalls. The Palo Alto firewall models of rendition 8.0.0 or more upheld REST API.

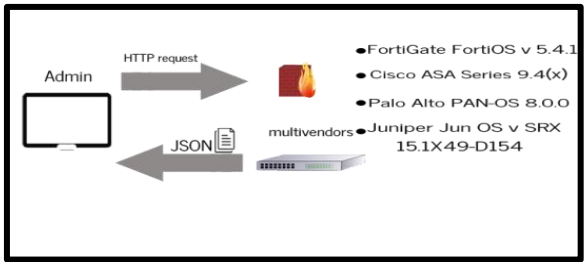

FIGURE (4): FIREWALL POLICY ACQUISITION

The firewall models shown in Figure (4) are available as VM as well. It facilitates implementation of test environment to evaluate the proposed tool FPM. In addition to these selected devices are the most commonly used firewalls, all of their models and above are supported REST API. The REST API architecture uses HTTP methods for connecting these firewall devices. Figure (5) shows the HTTP methods (GET, PUT, POST and DELETE) that are supported by REST.

The basic REST design principle uses the HTTP protocol methods for typical CRUD operations as shown in Table (1).

FIGURE (5): CLIENT HTTP REQUEST TO THE REST API URL

GET requests are used for retrieving resource data and not to modify it in any way. GET requests do not change the state of the resource, these are said to be safe methods as well [11].

TABLE I. THE HTTP METHOD AND ITS RELATIVE CRUD

HTTP Method	CRUD	Response Code
POST	**C**reate a resource	201 (Created),
GET	**R**etrieve a resource	200 (OK),
PUT	**U**pdate a resource	200 (OK) or 204 (No Content)
DELETE	**D**ELETE a resource	200 (OK). 404 (Not Found),

The other HTTP methods such as (PUT and DELETE) allow user agents to update/delete the resource on the server and so should be used carefully. In this paper, it is focused on using the safe request methods. The main target is using the proposed tool for firewall policy rules acquisition.

D. FPM tool based on the proposed approach

An implementation of a practical tool based on the proposed model is demonstrated. It is a tool to be used by the network administrator and it is named FPM. The (FPM) tool aimed to assist the administrator for the central management of multi-firewalls (may be of different vendors) in real-time. It stands for design principals of usability and visualization. It is safe as it just retrieves a copy of the firewall rule set without changing on the resources. It uses windows services (WS) instead of command-line interface (CLI), and provide a graphical user interface (GUI) for a sophisticated of firewall policy objects presentation. The firewall policy rule-set acquisition process is

maintained by the administrator. The administrator first uses the appropriate credentials to authenticate the use of the determined firewall device. Then using the FPM tool to issue API scripts over the HTTP protocol to acquire the firewall policy. It returned with its policy rule-set on a form of JSON file. JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types. The obtained file is processed and parsed in order to get a visual firewall rule set. Then the returned rule set should be investigated. To evaluate the proposed tool named FPM, it is required to have firewall devices that supported the REST API architectures. The virtualization technology and cloud environment may be used for testing the firewall configuration [13]. The virtual machine (VM) of firewalls from different vendors such as "Cisco Adaptive Security Appliance (ASA) series v9.4" is added to the proposed tool FPM. In a word, Cisco ASA is a security device that joins firewall, antivirus, interruption aversion, and virtual private system (VPN). Another firewall device is added to FPM tool is Palo Alto device with its operating system (PAN-OS) version 8.0.0. Skillet operating system (PAN-OS) is the product that runs all Palo Alto Networks cutting edge firewalls. The Palo Alto firewall models of rendition 8.0.0 or more upheld REST API. One more firewall device is added to the FPM tool is Juniper Network Operating System (JUNOS) Series Services Gateways (SRX) version SRX-15.1X49-D154 is added to the FPM too. JUNOS is the FreeBSD-based working framework utilized in Juniper Networks equipment switches and firewalls. SRX (Series Service Gateways) depend on JUNOS demonstrated working framework which conveys security and propelled assurance administrations. The last firewall added to the FPM tool in this paper is "FortiGate v4.5.1". These virtual machines represented the most known firewall devices are commonly used nowadays. Each one of them is studied well, to learn how to add the device to the proposed FPM tool and how to send and receive the required data based on the proposed approach. It has been found that each of them has a different form in the handling of policy objects. For example, all the security objects of the Juniper firewall are retrieved once. However, the other three devices (Cisco, Palo Alto and ForiGate) have multiple objects that returned by multiple JSON file queries.

After some trials, we have decided to upgrade all applications to ".NET 4.5". The .NET Framework 4.5 is highly compatible with applications that are built with earlier .NET Framework versions, except for some changes that were made to improve security, standards compliance, correctness, reliability, and performance. In our proposed tool, the administrator used the appropriate credentials in order to login the required firewall device inside the network, then and after obtaining the required policy objects he disconnected. This upgrade selects the version of the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol to use for new connections; existing connections aren't changed.

An example REST call that authenticate a firewall device that supports REST API architecture at IP address <firewall-IP> is:

RestClient (https://<firewall-IP> + <Logincheck>);

An example REST call that provides the appropriate credentials to the proposed tool (FPM) at IP address <firewall-IP>, user name and secret key is shown below:

var client = new RestClient(https://+ IP+
"/logincheck");
Client.Authenticator = new SimpleAuthenticator
("username", User, "secretkey", PW);
var request = new RestRequest ()
{ Method = Method.Post } ;
var v = client.Post (request) ;

An example REST call that provides data of a firewall device by its dedicated IP-address <firewall-IP> in JSON to the proposed tool (FPM).

```
"http_method": "GET",
"results": [
{
        "policyid":1,
        "q_orrigin_key":"1",
        "name":"Ahmed",
"uuid":"bdd275c8_6218_51e7_e5d3_3c458e379603",
"srcaddr": [
{
        "name":"192.111.1.4",
        "q_origin_key":"192.111.1.4"
}
]
"action":"accept",
}
].
```

As shown in above code, UUID (Universal Unique Identifier) is a 128-bit number used to uniquely identify some objects. The objects are units of code that are eventually derived from the process or entity on the network. Depending on the specific mechanisms used, a UUID is either guaranteed to be different or is, at least, extremely likely to be different from any other UUID generated until certain predefined value. The next step after obtaining the JSON file of the firewall policy objects, it should be serialized into data table as shown in Figure (5). A DataTable is an in-memory

representation of a relational database table. It can define columns of specific data types and indeed uphold uniqueness and foreign key limitations. This part clarifies how to login the firewall device in real time, authenticate using the appropriate credentials and acquire the firewall object policies using web architecture HTTP method "GET", over a DataTable to JSON in ASP.NET C#. In other words, how to serialize a DataTable to a JSON cluster in C#.

```
Public DataTableGetFortiPolicydatatable()
{
        RootObjectver = GetFortiPolicy ();
        DataTable table = new DataTable();
        Table.Columns.Add("Seq");
        Table.Columns.Add("Name");
        Table.Columns.Add("Source");
        Table.Columns.Add("Destination");
        Table.Columns.Add("Service");
        Table.Columns.Add("Action");
        Table.Columns.Add("NAT");
        Table.Columns.Add("Log");
Foreach (Result t in ver.results)
{
        var values = new object[9];
        values[0] = t.policyid;
        values[1] = t.name;
        values[2] = srcaddr[0].name();
        values[3] = t.dstaddr[0].name();
        values[4] = t.schedule;
        values[5] = t.service[0].name();
        values[6] = t.action;
        values[7] = t.nat;
        values[8] = t.logtraffic;
table.NewRow ();
table.Rows.Add(values);
}
Return table;
}
```

The GridView control may be included wealthy and flexible control utilized to acknowledge, show, and edit data on a web page. So the next step is binding the DataTable into GridView as shown in next example:

```
If  (Request.QueryString["id"] != null
{……..
DataTabledt = new DataTable();
      If (type ==
DeviceAdapter.Device_Type.FORTINET_FORTIGAT
E)
      {
        If (Request.QuerySrting["type"] == "Policies")
         dt = par.GetFortiPolicydatatable ();
         else if (Request.QuerySrting["type"] ==
"NATRules")
dt = par.GetFortiNATdatatable ();
else if (Request.QuerySrting["type"] == "NATRules")
dt = par.GetFortiNATdatatable ();
else if (Request.QuerySrting["type"] == "NATRules")
dt = par.GetFortiNATdatatable ();
else if (Request.QuerySrting["type"] == "NATRules")
dt = par.GetFortiNATdatatable ();
      }
}
```

E. Evaluation of the Performance

To test the response time of each firewall device, the experiment is run on the available personal computer. The implementation of the proposed work is maintained by a laptop system type x64-based PC. The operating system's name is Microsoft Windows 10 home edition – version: 10.0.17134-build 17134. Processor Intel(r) core(tm) i5-7300hq central processing unit is 2.50 GHz, 2501 MHz, four core(s), four logical processor(s). Installed physical memory (ram) is eight GB. Total physical memory is 7.87 GB. Available physical memory: 1.64 GB. Total virtual memory: 19.4 GB. Available virtual memory: 3.08 GB. Page file space: 11.5 GB. As shown inFigure (6), the virtual machine represents FortiGate firewall v5.4.1 has a good time response to deliver its JSON files that contain its configurations. Cisco ASA v9.4 virtual machine consumed more time (about 890ms) to respond with its Network Address Translation (NAT) configuration.The virtual machines technology and cloud environment may be used for testing the firewall configuration [13]. The virtual machines of firewalls from different vendors such as, Cisco ASA series v9.4, Palo Alto PAN-OS v8.0.0, Juniper JunOS v SRX 15.1X49-D15.4 and Fortigate v4.5.1 are used for testing purpose. Each one of them is studied well, from the point of view of acquiring its configurations and the structure of its policy rules.

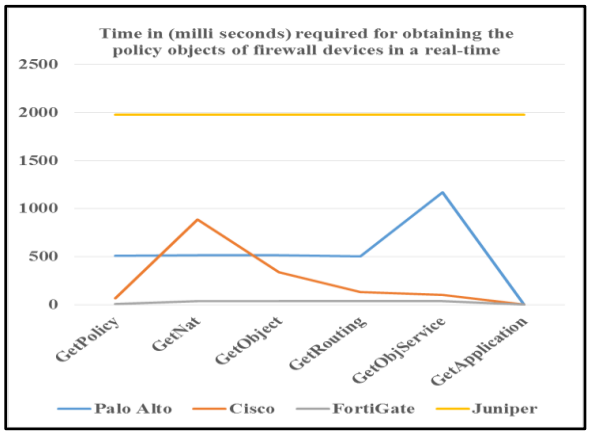

Figure (6): The response time in (mille second) of the virtual machines

As shown on the Figure (6 and 7), the FPM tool is tested against the previously mentioned firewall devices in a test environment using virtual machines. Figure (6) shows the response time in milliseconds for each used firewall device. Figure (7) shows the size of the obtained policy objects in (bytes). The time is measured starting from the tool try to login into each device until the device reply with its policy objects in a form of JSON file. The virtual machine represents Fortigate firewall v 5.4.1 has a fixed time response to reply with its policy objects. Cisco ASA v9.4 virtual machine consumed more time to respond with its NAT configuration.

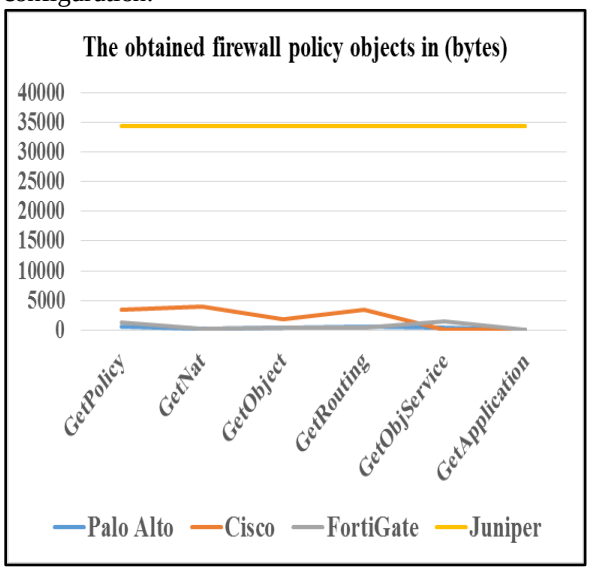

Figure (7): The response time in (mille second) of the virtual machines

The virtual machine of Palo Alto firewall v8.0.0 has a longest latency in replying with its configuration relative to the others. The virtual machine of Juniper vSRX 15.1x49 has a different configuration object named (Application), and it replied with affixed time about 1976 milliseconds for each configuration file response.

F. The real environment results

Subsequent to adding firewall equipment to the proposed tool and testing on four firewall gadgets through VM, we went to attempt the proposed tool in a genuine situation.

A server farm identified with one Egyptian college was chosen and we acquainted the proposed instrument FPM with the system administrator and requested a firewall gadget to be evaluated. It was FortiGate

version 4.5.1 managed security service (MSA). It is named (MSA Forti).

FIGURE (8): ADD NEW FIREWALL TO THE FPM TOOL

The required username, password, and IP addresses are controlled by the administrator so as to associate the "MSA Forti" firewall utilizing the FPM tool as shown in Figure (8). It was perceived specifically in light of the fact that a similar kind of firewall was predefined to the FPM instrument previously. After passing the authentication phase, the HTTP request was initiated, and then the firewall gadget reacted with sending back its configurations in type of JSON files.

The proposed tool (FPM) processed files and extracted the data of interest (DOI). These DOI includes the policy rules, NAT, the routing Table, object addresses and object service. These objects were processed for viewing through a visual interface and in a way that helps the network administrator review the policy rules as shown in Figure (9). As appeared in Figure (9), in the left pane of the screen are the firewall devices that previously included. By choosing the firewall device name, you will find that the arrangement objects have been appeared at the base of thedevice.

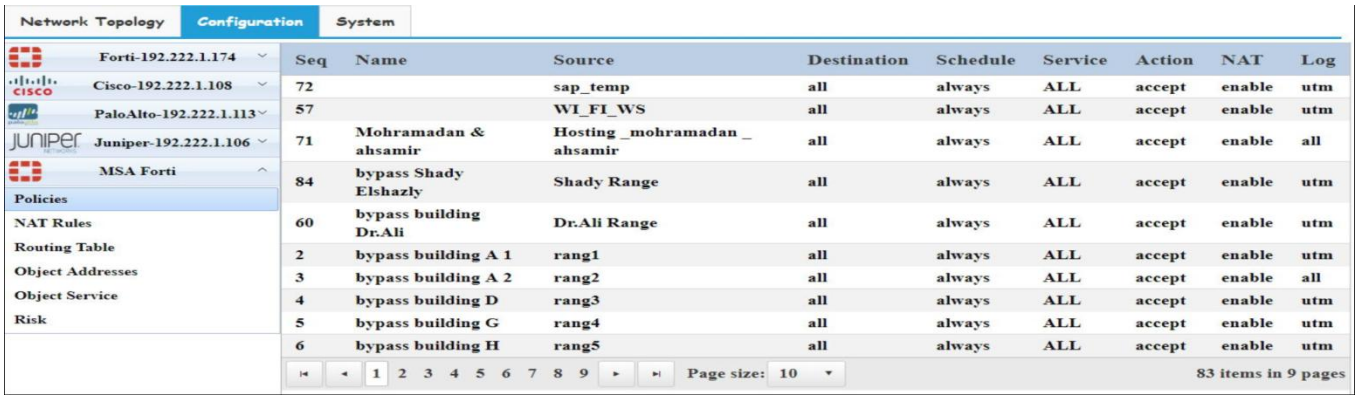

Figure (9): The returned real-time Fortigate firewall policy rule-set

As appeared on the Figure (9), there were 83 rules that can be perused in nine pages. The displayed page (1) contains the policy rules which displayed in a tabular form. The first row on the Table includes the column names. The FortiGate machine is named likewise Unified Threat Management (UTM). Empowering log security occasions will just show up traffic log coordinate UTM profile characterized. Notwithstanding, (ALL) or (log all sessions) will incorporate traffic log both match and non-coordinate UTM profile characterized.

Name	Type	Details	Interface	Visibility
*.live.com	ipmask		0.0.0.0 0.0.0.0	enable
Adobe Login	ipmask		0.0.0.0 0.0.0.0	enable
Camera	ipmask		172.20.0.0 255.255.240.0	enable
Cisco core	ipmask		192.168.104.2 255.255.255.255	enable
Class Building D	iprange		10.10.129.1 10.10.129.60	enable
Class Bulding A	iprange		10.10.65.2 10.10.65.50	enable
Class Bulding G	iprange		10.10.193.2 10.10.193.60	enable
Class Bulding H	iprange		10.11.1.2 10.11.1.30	enable
Class Bulding L	iprange		10.11.65.2 10.11.65.25	enable
DC_DNS1	iprange		192.168.1.101 192.168.1.103	enable

1 2 3 4 5 6 7 8 9 10 ... Page size: 10 — 114 items in 12 pages

FIGURE (10): THE OBJECT ADDRESSES OF MSA FORTI FIREWALL

The system get to interpretation (NAT) controls likewise were come back from the (MSA Forti) firewall. It has three segments, and 30 things showed in three pages. It included routed IP addresses so it isn't like to distribute the Figure. The "Routing Table" was discovered unfilled, as the administrator clarified that it isn't should have been designed yet. So, the next item was (object addresses). It was found 114 items displayed in 12 pages as shown in Figure (10). The (Object Service) Table was found in the JSON file and displayed as shown on the next Figure (11). It contains 21 items displayed in three pages. The system administrator was happy with the execution of the FPM tool. The administrator clarified that he has 20 distinctive firewall devices models. Each firewall model has a tool that exclusively manages it. So having one tool that can deal with numerous models of firewall is great. Likewise, the manner in which the approach rules are shown and kept up is critical for the system administrator. He also recommended with adding filters in order to display the columns is important to make it less demanding to survey numerous strategy rules.

Name	Category	Details	IP/FQDN	Show in Service List
LDAP	Authentication	TCP / 389	0.0.0.0 /	enable
RADIUS	Authentication	UDP / 1812 1813	0.0.0.0 /	enable
KERBEROS	Authentication	TCP / 88 464	0.0.0.0 /	enable
LDAP_UDP	Authentication	UDP / 389	0.0.0.0 /	enable
IMAP	Email	TCP / 143	0.0.0.0 /	enable
IMAPS	Email	TCP / 993	0.0.0.0 /	enable
POP3	Email	TCP / 110	0.0.0.0 /	enable
POP3S	Email	TCP / 995	0.0.0.0 /	enable
SMTP	Email	TCP / 25	0.0.0.0 /	enable
SMTPS	Email	TCP / 465	0.0.0.0 /	enable

1 2 3 4 5 6 7 8 9 10 ... Page size: 10 105 items in 11 pages

FIGURE (11): THE (OBJECT SERVICE) OF (MSA FORTI) FIREWALL

The required filters have just been included as appeared in Figure (12). The policy rules are filtered to detect the risk rules. It is a risk to use generalization rules, by allowing to (all) source addresses and (all) destination addresses (all) services, for example.

The specified values should be identified by the source or destination addresses to avoid one of the most rule anomalies. These filters also make it easier to go directly to certain rules. As it displayed there are many pages of policy rules so it is hard to browse them one by one for searching certain rule.

NAME	ORDER	SOURCE_IP	SOURCE_PORT	DESTINATION_IP	DESTINATION_PORT	PROTOCOL	ACTION
		0.0.0.0	ANY	0.0.0.0	ALL		
From Port A To Dokki ()	8	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
From Dokki To Port A ()	10	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
(allow youtube)	11	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
()	17	0.0.0.0	ANY	0.0.0.0	ALL	ALL	deny
(default)	18	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
allow viber (allow viber)	19	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
MSA Users (default)	20	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
Allow MSA Site Only (Block All)	22	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
Allow youtube (allow youtube)	23	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
Allow Facebook Video (Allow Facebook video)	26	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
Internet for Servers VLan (Allow_everything_APP)	32	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
()	33	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
All users to servers ()	34	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
From Dokki To Servers ()	35	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept
()	36	0.0.0.0	ANY	0.0.0.0	ALL	ALL	accept

1 2 Page size: 15 22 item

FIGURE (12): THE (RISK) DETECTED IN POLICY RULES OF (MSA FORTI) FIREWALL

In Figure (12) it is noticed that the estimation of (0.0.0.0) that showed any IP address has been doled out to numerous guidelines in both if their source and destination addresses. The esteem (ANY) is doled out to the source port. The esteem (ALL) is allocated to the destination port. At that point, the policy rules are sifted and brought about 22 rules. These came about standards speaking to various activities for the system traffic that originate from any source address originates from all source ports and targeted on any destination address through any destination port. It is fundamental for the system administrator to survey these rules as it is considered as high risk rules. They are positively fair or should be changed so as to show signs of improvement execution.After excluding high risk rules, the remaining rules are examined to detect anomalies with the splitting of these rules if there is an overlap between them as shown previously.

V. DISCUSSION

The proposed tool FPM is executed dependent on the proposed approach. The REST API architecture is

utilized as WS. It has fulfilled better outcomes from perspective of availability, effortlessness, representation and convenience. The test condition was actualized utilizing VM of various firewall gadgets. It was easy to add firewall gadget to the FPM device. The response time of associating the utilized firewall gadgets as appeared Table (II) was acknowledged by the executive. The assessment of the proposed tool in genuine server farm and association with the FortiGate gadget was straightforward and simple work due to beforehand including of a similar model. Adding channels to the segments spoke to the fields of the strategy rules were compelling. It encourage to distinguish the high complex controls in simple way.

The proposed FPM tool utilizes the HTTP ask for (POST/GET) techniques so as to secure the firewall configuration. Be that as it may, utilizing the channels make it simple to recognize some policy rule anomalies. It urges to contemplate the rule anomalies discovery on the future work.

VI. CONCLUSION AND FUTURE WORK

The traditional approach for managing the firewall policy rules consolidate using CLI isn't simple and might be blunder inclined. A few firewall management tools that are supporting the visualization are accompanied with certain firewall models only. Thus it cannot be used with another particular firewalls. It suggests that the managers should need to use different gadgets to manage the firewalls that are found on the framework. In this paper, an approach using REST API is proposed to acquire policy from multiple firewalls. It is preferred to use REST API as WS rather than using SOAP. REST is a building style, not a protocol, as a web administration, it is quicker than SOAP in light of the fact that there is no strict detail like SOAP. It devours less transfer speed and assets too. REST web administrations are both language and stage autonomous. They can be written in any programming language and executed in any stage.

FPM is a proposed tool dependent on the proposed approach. It is consented with the literature to fix the requirement of brought together tools for overseeing distinctive firewalls. FPM keeps up the procured firewall approach represents both ease of use and perception. A test domain is implemented utilizing four distinctive firewall gadgets in type of VM. The most widely recognized utilized firewall items from different vendors (Cisco, Palo Alto, Juniper, and FortiGate) are utilized for test purposes. In any case, numerous different items that upheld the REST API, from various models, might be added to the FPM apparatus. The FPM is tried likewise in a genuine Data Center of an Egyptian college, using the FortiGate firewall. The outcomes were accepted from the perspective of convenience and representation.

Acknowledgment

We are gratefully thanking Mr. George Ezzat, Eng. Amjad Mohamed from IP Protocol Company for their technical advice. An appreciation for Eng. Adaham Mohamed is a must for his great support.

References

[1] Kumar, S. (2016). A Review of Recent Trends and Issues in Visualization. International Journal on Computer Science and Engineering (IJCSE), 8(3), 41–54.

[2] Iwaya, L. H., Voronkov, A., Martucci, L. A., Lindskog, S., and Fischer-Hübner, S. (2016). Firewall Usability and Visualization : A Systematic Literature Review (Karlstad University Studies). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-44688

[3] Voronkov, A., Iwaya, L.H., Martucci, L.A., &Lindskog, S. (2017). Systematic Literature Review on Usability of Firewall Configuration. ACM Comput. Surv., 50, 87:1-87:35.

[4] Windmüller, S. (2011). Offline Validation of Firewalls. 2011 IEEE 34th Software Engineering Workshop, 36-41.

[5] Martínez A. Yannuzzi M. López J. Serral-Gracià R. Ramirez W. (2015). Applying information extraction for abstracting and automating the CLI-based configuration of network devices in heterogeneous environments.

[6] Kim H., Ko S., Kim D. S. and Kim H. K. (2017). Firewall ruleset visualization analysis tool based on segmentation. IEEE Symposium on Visualization for Cyber Security (VizSec), Phoenix, AZ, pp. 1-8. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8062196&isnumber=8062166

[7] Tran, T., Al-Shaer, E., &Boutaba, R. (2007). PolicyVis: Firewall Security Policy Visualization and Inspection. LISA.

[8] X. Wang et al. (2018). PNPL: Simplifying programming for protocol-oblivious SDN networks. Computer Networks. 147, 64–80.

[9] Antonio JesúsFernández-García, Luis Iribarne, Antonio Corral, Javier Criado, James Z. Wang. (2018). A flexible data acquisition system for storing the interactions on mashup user interfaces. Computer Standards & Interfaces, Volume 59, Pages 10-34, ISSN 0920-5489. International Journal of Scientific & Engineering Research, Volume 6, Issue 2, February-2015. ISSN 2229-5518.

[10] R. Richards. (2006). Representational State Transfer (REST) in "Pro PHP XML and Web Services". In: Springer Publishing, Chap. 17, pp. 633–672.

[11] Roy Thomas Fielding and Richard N. Taylor (2000). Architectural styles and the design of network-based software architectures, University of California, Irvin.

[12] C. Pautasso et al., (2008). REST Web Services vs. Big Web Services: Making the Right Architectural Decision, Beijing China.

[13] GladmanJekese, R.Subburaj Professor, ChiedzaHwata (2015). Virtual Firewall Security on Virtual Machines in Cloud Environment. International Journal of Scientific & Engineering Research, Volume 6, Issue 2, February-2015. ISSN 2229-5518.

FIRST AUTHOR: Dr. Eng. Mohamed EzzatElhamahmy is a cyber-technology expert. He is from Egypt. He has received his Ph. D. in computer networks security from Faculty of Computers and information, Cairo University, 2011. He has received his B.Sc. in Computer Engineering from MTC, Cairo in 1989. He received his M. Sc. in Computer Engineering, Faculty of Engineering, Al-Azhar University, Cairo in 2001. His past research was in building text to speech synthesis system in Arabic. His current research interests are Network Security and Machine Learning.

SECONDAUTHOR: Ass.Prof.DrMohamed.M.A.Elgazzar, He received the B.Sc.degree in Electronic and Communications from MTC, Cairo, Egypt, in 1984 and M.Sc.degree in Computer NetworksfromFaculty of Engineering,Cairo University,Egypt, in 1991 and the Ph.D. degree in Computer Networks at Faculty of Engineering, from Cairo University, Egypt, in 1995.He has experience of 33 years which includes both academic and research.He is currently an Associate Professor in Computer Science Department at Higher institute of Computer Science and information systems fifth Settlement, New Cairo, Egypt. His research interests Computer Networks, Machine Learning, and Network Security. He published several research papers.

Third AUTHOR: Abdel-Hamid Emarareceived the B. S., M. S., and Ph.D. degree in computers engineering from Al-Azhar University in 1992, 2000, 2006, respectively. He works in Computers and Systems Engineering Department at Faculty of Engineering, Al-Azhar University,Cairo, Egypt. He has experience of 12 years which includes both academic and research. He is currently an Assistant Professor in Computer Science Department at the Taibah University, Al Madinah Al Monawarah, KSA. His research interests educational data mining, Machine Learning, Arabic text mining, and intelligent, and adaptive systems. He published several research papers.