Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPROVEMENT OF ALGORITHM FOR UPDATING FIREWALL POLICIES

[1]Z.KARTIT, [2]H.KAMAL IDRISSI, [3]A.KARTIT, [4]M.EL MARRAKI

[1,2,3,4] University Mohammed V – Agdal Rabat, Faculty of Sciences, LRIT
4 Avenue Ibn Batouta. BP 1014 RP 10006 Rabat, Morocco
E-mail: [1]z_kartit@yahoo.fr

ABSTRACT

Security issues are becoming more critical in network systems. Firewalls offer an important defense and protection for network, permit to strengthen security aspect. Firewalls are network devices or programs which enforce an organization's security policy, permit to control and monitor the traffic flow of network; they are installed between networks or hosts that employ differing security postures. Generally firewalls were deployed at network perimeters in order to provide some measure of protection for internal hosts. Different firewalls support different policy editing commands. The set of policy editing commands that a firewall supports is called its policy editing language. In [1], the authors provide deployment algorithm for type II language. This paper aims to develop an efficient algorithm for the updates of the security policy. Our proposal is considered improved type II edition policies algorithm. Although the proposed algorithm in [1] gives correct results, but it has severe shortcomings of security in the implementation of the new policy of security, in addition, this algorithm has a high degree of complexity. We will propose an algorithm that will address these two weaknesses of the old algorithm, which allows us to gain in terms of security and complexity.

Keywords: *Firewall, Security Policy, Policy Deployment, Network Security.*

1. INTRODUCTION

A firewall is software or hardware-based network security system that controls the incoming and outgoing network traffic by analyzing the data packets and determining whether they should be allowed through or not, based on applied rule set [2].

in other way, a firewall can be considered as a system or group of systems (router, proxy, or gateway) that implements a set of security rules to enforce access control between two networks to protect "inside" network from "outside" network. The use of firewalls is still a very important step. It may be a hardware device or a software program running on a secure host computer. In either case, it must have at least two network interfaces, one for the network it is intended to protect, and one for the network it is exposed. It can monitor and possibly block the flow of data by analyzing the information contained in the data stream. A firewall sits at the junction point or gateway between the two networks, usually a private network and a public network such as the Internet. The firewall provides control in both directions, firstly it allows blocking and traces attacks or suspicious connections may originate from viruses or others. Secondly, a firewall is used in many cases also to prevent the uncontrolled leakage of information to outside.

Firewalls are host-resident security software applications that protect the enterprise network's servers and end-user machines against unwanted intrusion. They offer the advantage of filtering traffic from both the Internet and the internal network. This enables them to prevent hacking attacks that originate from both the Internet and the internal network. This is important because the most costly and destructive attacks still originate from within the organization. They offer several important advantages like central management, logging, and in some cases, access-control granularity.

These features are necessary to implement corporate security policies in larger enterprises. Policies can be defined and pushed out on an enterprise-wide basis. Several firewalls deployed Policies containing 10K rules are not uncommon in commercially deployed firewalls, and we have seen

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a firewall configured with 50K rules. Manually configuring such policies has clearly become mission impossible even for guru network administrators. These rules in general [2] are: (i) accept a connection (enabled), (ii) blocks a connection (deny).

This paper aims to analyze the algorithm "ENHANCED-Greedy-two-Phase Deployment" provided in [1] and show that this algorithm has serious flaws. We present an improved safety formalization that can be used as a basis for formulating safe deployment strategies. We provide most-efficient and safe algorithm for Type II languages.

2. SECURITY POLICY

Generally, Security policy is a definition of what it means to be secure for a system, organization or other entity. For an organization, it's a defined action plan to preserve the integrity and security of computer resources (data, OS, applications, hardware ...) in an open network. It reflects the strategic vision of IT decision-makers of the organization (SME, industry, administration, state ...).

Particularly for firewall, its principle for operation is simple; it is a set of rules defined by an administrator based on the principle: everything that is not explicitly allowed is prohibited, which means that these rules are part of the configuration firewall must allow or dismiss an action or a data stream in order to establish or block a connection.

It addresses the constraints on behavior of its members as well as constraints imposed on adversaries by several mechanisms such as ACL. For systems, the security policy addresses constraints on functions and flow among them, constraints on access by external systems and adversaries including programs and access to data by foreign users.

If it is very important to be secure, then it is important to be sure all of the security policy is enforced by mechanisms that are strong enough. There are many organized methodologies and risk assessment strategies to assure completeness of security policies and assure that they are completely enforced. In information systems, policies can be decomposed into sub-policies to facilitate the allocation of security mechanisms to enforce sub-policies. A top level security policy is essential to any serious security scheme and sub-policies and rules of operation are meaningless without it.

3. FIREWALL BACKGROUND

The network is composed, generally, of three areas. These areas are; the internet, the demilitarized zone (DMZ) and the intranet. These three zones have different levels of trust. The DMZ contains typically common public services (such as web sites, DNS server, FTP server… as schematized in figure 1) and remains available to hosts from the Internet.

A firewall is generally placed at the borderline of the network to act as the Access Controller for all incoming and outgoing traffic (figure 1). It's basically the first line of defense for any network

Firewall policy permits to deny a certain service to a list of specific hosts which are considered malicious.

A firewall controls traffic by examining the contents of network packets. Five packet fields are most commonly used for traffic filtering: *protocol type, source IP address, source port, destination IP address, and destination port* (Figure 2).

The filtering decision is based on a firewall policy defined by network administrator. A firewall policy is an ordered list of rules. A firewall rule r specifies an action, typically accept or deny, for the set of packets matching its criteria. It is possible to use any field of IP, UDP, or TCP headers [3].

Any field in packet's header can be used for the matching process. However, the same five fields are most commonly used. In a packet, each of these fields has an atomic value. If all the fields of a packet p match with the corresponding fields of a rule r, then p is accepted or rejected according to the decision field of r. If p does not match to any rule in policy, then the default match-all rule is applied [1].

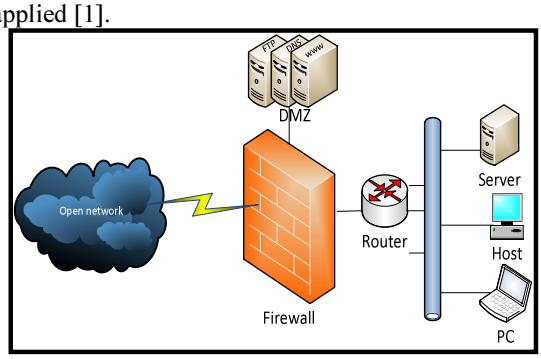

Figure 1 : Firewall Architecture

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. POLICY DEPLOYMENT

4.1 Definition

Policy deployment is the process by which policy editing commands are issued on firewall, so that the running policy is replaced by a target policy. The deployment should be performed in online mode in order to keep applications available; especially for the applications that treat a critical data stream and in real-time such as VoIP and online e-commerce.

a. permit HTTP 192.168.1.1 /24 80
b. deny IP 10.2.1.0/24 any
c. permit TCP 172.20.0.0/16 any 123
d. deny FTP 10.1.2.0/24 76.54.32.1
e. permit TELNET 10.0.0.0/8 any

Figure 2 : Example Of Policy Deployment

4.2 Policy deployment characteristics

In order to permit to firewall to accomplish its mission as it should, the policy deployment should have following characteristics [3]: correctness, confidentiality, safety, and speed.

Correctness: Is an essential requirement for any deployment, a deployment is considered correct if, starting from an initial policy; we arrive at the target policy successfully. After implementation, the target policy becomes the running policy.

Confidentiality: The communication between firewall and a management tool must be secured because of the sensitive nature of informations transmitted during the deployment. A lack of confidentiality may affect safety; a hacker can see the security policy implemented in the firewall and to detect possible security breaches. Confidentiality can be achieved by using encrypted communication protocols such as SSH [4] and SSL [5].

Safety: We can say that the deployment is safe if no security hole is introduced and no legal traffic is denied, and no illegal packet is accepted at any moment during the deployment. A temporary security hole permits malicious traffic to pass through the firewall and this may cause serious damage to the network infrastructure. Interdiction of legal traffic during deployment may interrupt critical data stream and may causes serious losses. This is like inflicting a self-DoS attack that is intolerable in mission-critical networks, even for a short time [6]. Deployment safety is particularly important in cases where many changes are to be made to a large firewall policy. In such cases, a deployment can last up to several minutes, which may provide sufficient opportunity to a malicious party to exploit vulnerability. Fast spreading worms, such as Slammer [7] and Conficker [8], can infect millions of systems across the globe within minutes. A skilled hacker can use automated tools to continuously probe for vulnerabilities and instantly exploit these as they appear during an unsafe deployment.

Speed: It's very important that the policy of the firewall should be deployed in very short time, to avoid any suspicious traffic. So that the desired state of affairs is achieved as quickly as possible. Slow deployments are unpleasant for users and can also raise security issues: very often a firewall policy change needs to be deployed immediately to close a hole for illegal access or to open access for highly urgent traffic. That's why the use of a little number of commands is very required to reduce the complexity and so the running time of the algorithm.

4.3 Policy editing languages

A network administrator or a management tool issues commands on firewall to transform the running policy R into the target policy T. The set of commands that a firewall supports is called its policy editing language. Typically, a firewall uses a subset of the following editing commands [9]:

(app r), (del r), (del i), (ins i r) and (mov i j).

Policy editing languages can be classified into two representative classes [1]: Type I and Type II.

4.3.1 Type I editing

Type I editing supports only two commands, append and delete. Command (app r) appends a rule r at the end of the running policy R, unless r is already in R, in which case the command fails. Command (del r) deletes r from R, if it is present. As Type I editing can transform any running policy into any target policy [1], therefore it is complete. Older firewalls and some recent firewalls, such as FWSM 2.x [10] and JUNOSe 7.x [3], only support Type I editing.

4.3.2 Type II editing

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Type II languages allow random editing of firewall policy. It supports three operations: (ins i r) inserts rule r as the ith rule in running policy R, unless r is already present; (del i) deletes ith rule from R; (mov i j) moves the ith rule to the jth in R position. Type II editing can transform any running policy into any target policy without accepting illegal packets or rejecting legal packets [1], therefore it is both complete and safe. It is obvious that for a given set of initial and target policies, a Type II deployment normally uses fewer editing commands than an equivalent Type I deployment. Examples of Type II editing firewalls include SunScreen 3.1 Lite [11] and Enterasys Matrix X [12].

4.4 Deployment efficiency

A deployment is most-efficient if it utilizes the minimum number of editing commands in a given language, to correctly deploy a target policy on a firewall. Therefore for a given deployment scenario, the most-efficient Type I deployment uses the minimum number of append and delete commands, similarly a most-efficient Type II deployment uses the minimum number of insert, delete and move commands. Usually a policy editing command takes constant time, and the variation in deployment time is negligible for different types of commands. Therefore, the most-efficient deployment minimizes the overall deployment time. Deployment efficiency for Type II languages is discussed in more detail in Section 4.

5. TYPE II DEPLOYMENT

The Type II deployment allows for random modification of a running policy. Therefore, for a given set of I and T, a safe Type II deployment usually utilizes less editing commands than an equivalent Type I deployment.

5.1 Old version of the algorithm

The algorithm that was proposed in [1] named ENHANCED- Greedy- 2-Phase Deployment (see Algorithm 2) is considered an improvement of the algorithm in [3] Called Greedy-2-Phase Deployment.

Algorithm 2: ENHANCED-Greedy-2-Phase Deployment

```
1. ENHANCEDTwoPhaseDeployment (I, T) {
2. /* algorithm to calculate a safe type
II deployment */
3. /* to transform firewall policy I into
T */
4.
5. /* Phase 1: insert and move */
7. for t←1 to SizeOf(T) do
8. if T[t] ∉ I then
9. IssueCommand(ins t T[t])
11. else
12. IssueCommand( mov IndexOf(T[t] , I)
t)
13.
14. /* Phase 2: backward delete */
15. for i←SizeOf(I) down to 1 do
16. if I[i] ∉ T then
17. IssueCommand( del i )
18. }.
```

The execution of this algorithm, with two different examples of I and T, gave:

Case 1:
I = a-b-c
T = c-b-a
R = c-b-a

Proof:

1- t=1 ; indexof(T(t)=c,I)=3 ; move(3,1) ; R0= c-b
2- t=2 ; indexof(T(t)=b,I)=2 ; move(2,2) ; R1= c-b
3- t=3 ; ins(T(t),3) ; R2= c-b-a

Figure 3 : ENHANCED-TwoPhaseDeployment running for case 1

Case 2:

I = A-M-C-L-K-E
T = L-C-E-M-B-D-F-K
R = L-C-E-M-B-D-F-K

Proof:

1- t=1 ; indexof(T(t)=L,I)=4 ; move(4,1) ; R0= L-M-C-K-E
2- t=2 ; indexof(T(t)=C,I)=3 ; move(3,2) ; R1=L-C-K-E
3- t=3 ; indexof(T(t)=E,I)=4 ; move(4,3) ; R2= L-C-E
4- t=4 ; T(t)=M ins(M,4) ; R3= L-C-E-M
5- t=5 ; T(t)=B ins(B,5) ; R4= L-C-E-M-B
6- t=6 ; T(t)=D ins(D,6) ; R5= L-C-E-M-B-D
7- t=7 ; T(t)=F ins(F,7) ; R6= L-C-E-M-B-D-F
8- t=8 ; T(t)=K ins (K,8) ; R7= L-C-E-M-B-D-F-K

Figure 4 : ENHANCED-TwoPhaseDeployment running for case 2

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We can clearly see that this algorithm has two serious flaws:

1- First, this algorithm contains a gap of security when updating the security policy, for example in t=3 in figure 4, the running policy R2 contain only a few rules of the initial policy I (only 3 rules in the 06 rules of initial policy), so deployment does not meet the safety criterion. This approach of policy deployment can temporarily open a network to unwanted traffic and prohibit desired traffic; an illegal access can use a hole of security in this sensitive moment.

2- Second, the complexity of this algorithm is quadratic and it is about $O(n^2)$. This will have an impact on the speed for Deployment, and will last longer during the update. This is undesirable because the firewall may be in the unstable state of security at the time of the update; this issue has already been discussed in the first drawback.

5.2 Our algorithm for type II deployment

The above problems motivate us to provide a correct, safe and efficient algorithm, called IMPROVED-TWOPHASEDEPLOYMENT to calculate a safe type II deployment for policies I and T .This algorithm got also two phases; In the first phase, the algorithm inserts the rules of T at the beginning of the running policy R. In the second phase, all rules of I are deleted starting from the last rule. This is described in our Algorithm (see Algorithm 2.1 **IMPROVED-Greedy-2-Phase Deployment)**.

Algorithm 2.1: IMPROVED-Greedy-2-Phase Deployment

```
1. IMPROVEDTwoPhaseDeployment (I, T) {
2. /* algorithm to calculate a safe type II
deployment */
3. /* to transform firewall policy I into T */
4.
5. /* Phase 1: insert new policy */
7. for t←1 to SizeOf(T) do
8.   {
9.       IssueCommand(ins t T[t])
10.  }
11. /* Phase 2: delete old policy */
12. for i←(SizeOf(T)+ SizeOf(I)) down to
(SizeOf(T)+1)  do
13.  {
14.      IssueCommand( del i )
15.  }.
16. }.
```

The execution of this algorithm, with two different examples of I and T, gave:

Case 1 :
I= A-C-D
T=B-A-E-D
R=B-A-E-D

Proof:
1- t=1 ; ins(1, B) ; R1=B-A-C-D
2- t=2 ; ins(2, A) ; R2=B-A-A-C-D
3- t=3 ; ins(3, E) ; R3=B-A-E-A-C-D
4- t=4 ; ins(4, D) ; R4=B-A-E-D-A-C-D

5- i=7 ; del(7) ; R5=B-A-E-D-A-C
6- i=6 ; del(6) ; R6=B-A-E-D-A
7- i=5 ; del(5) ; **R7=B-A-E-D**

Figure 5 : IMPROVED-TwoPhaseDeployment running for case 1

Case 2 :
I= A-K-C-L-M-D
T=B-F-A-E-D-K
R=B-F-A-E-D-K

Proof:
1- t=1 ; ins(1, B) ; R1=B-A-K-C-L-M-D
2- t=2 ; ins(2, F) ; R2=B-F-A-K-C-L-M-D
3- t=3 ; ins(3, A) ; R3=B-F-A-A-K-C-L-M-D
4- t=4 ; ins(4, E) ; R4=B-F-A-E-A-K-C-L-M-D
5- t=5 ; ins(5, D) ; R5=B-F-A-E-D-A-K-C-L-M-D
6- t=6 ; ins(6, K) ; R6=B-F-A-E-D-K-A-K-C-L-M-D

7- i=12 ; del(12) ; R7 =B-F-A-E-D-K-A-K-C-L-M
8- i=11 ; del(11) ; R8 =B-F-A-E-D-K-A-K-C-L
9- i=10 ; del(10) ; R9 =B-F-A-E-D-K-A-K-C
10- i=9 ; del(9) ; R10=B-F-A-E-D-K-A-K
11- i=8 ; del(8) ; R11=B-F-A-E-D-K-A
12- i=7 ; del(7) ; R12=**B-F-A-E-D-K**

Figure 6 : IMPROVED-TwoPhaseDeployment running for case 2

When we apply the new version of the algorithm it's clear that the result given is correct, this algorithm includes the following changes:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1- Remove the test's instruction (if T[t] ∉ I then) and (if I[i] ∉ T then) in the old algorithm. This measure will increase the speed of deployment and therefore reduce the period of updating. It means that we have won in terms of performance, making this algorithm complexity linear.

2- Remove the operation *mov*, since we proceed with the insertion of the new policy T at the beginning of the running policy, this measure allows us to keep all the rules of the old policy safety during the implementation of the new policy. the removal of the old rules do not begin until the total insertion of the new policy T. This transitional stage does not affect the security of our network and do not expose it for vulnerabilities.

6. CONCLUSION

Firewall policy deployment safety is a new area of research, and is an actuality subject. Several approaches have proposed strategies in order to update a policy while respecting the safety and efficiency criteria, but fail to provide an effective solution, which gives good results in all cases. Most of those approaches contain critical errors and can temporarily open a network to unwanted traffic and/or interrupt network services by prohibiting legal traffic during a deployment.

In this paper, we have provided for type II policy editing languages a linear, safe and efficient algorithm called IMPROVED-TWOPHASEDEPLOYMENT.

In future work, we plan to work on the second algorithm called SANITIZEIT to improve it.

REFRENCES:

[1] A. Kartit, Z.Kartit and M. El Marraki "Towards a new Algorithm more efficient for updating Firewall Policies", Journal of Theoretical and Applied Information Technology, Volume 59, n°2, pages 250 – 254, 20th Janouary 2014.

[2] http://en.wikipedia.org/wiki/Firewall_(computing).

[3] C. C. Zhang, M. Winslett, and C. A. Gunter. "On the Safety and Efficiency of Firewall Policy Deployment". In SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pages 33–50,Washington, DC, USA, 2007. IEEE Computer Society.

[4] T. Yl¨onen. SSH: secure login connections over the internet. In SSYM'96: Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography, pages 4–4, Berkeley, CA, USA, 1996. USENIX Association.

[5] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. In WOEC'96: Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce, pages 4–4, Berkeley, CA, USA, 1996. USENIX Association.

[6] Zeeshan Ahmed, Abdessamad Imine, and Michael Rusinowitch1"Safe and Efficient Strategies for Updating Firewall Policies" INRIA Nancy Grand Est. INRIA ARC 2010 ACCESS and FP7-ICT-2007-1 Project No.216471 AVANTSSAR. http://www.avantssar.eu/pdf/publications/trustbus10.pdf

[7] F-Secure. Malware information pages: Slammer,http://www.f-secure.com/v-descs/mssqlm.shtml

[8] F-Secure. Malware information pages: Worm:w32/downadup.al, http://www.f-secure.com/v-descs/worm_w32_downadup_al.shtml

[9] Z. Ahmed, A. Imine, and M. Rusinowitch. "Safe and efficient strategies for updating firewall policies". *Trust, Privacy and Security in Digital Business, 2010,pp. 45-57.*

[10] Bezzazi, F and Kartit, A and Marraki, M El and Aboutajdine, D, "Optimized strategy of deployment firewall policies", *Second International Conference on Innovative Computing Technology (INTECH), IEEE 2012, pp. 46-50.*

[11] M. Englund. Securing systems with host-based firewalls. In Sun BluePrints Online, September 2001.

[12] Entrasys Matrix X Core Router. http://www.entrasys.com/products/routing/x/