Efficient Method for Inferring a Firewall Policy

Hyeonwoo Kim, Hongtaek Ju
Dept. of Computer Engineering
Keimyung University
Daegu, Republic of Korea
{ hwkim84, juht }@kmu.ac.kr

Abstract—**We propose a framework which infers the policy of firewall deployed in the Internet access point and computer system. The proposed methodology shows how to infer a firewall policy from restricted probing packets, using consecutive characteristics of the IP address and TCP/UDP port number. We also show the experimental results and the performance of the proposed method.**

Keywords- Firewall; Firewall policy; Network Fingerprinting

I. INTRODUCTION

The Internet is exposed to many kinds of threats frequently. The attackers make the target network unable to provide normal services using bad traffics. Therefore, most corporations and organizations secure their internal hosts by firewall which block the traffic from outside to inward and vice versa. The principal purpose of the firewall is to inspect the packets coming from or going to outside. It passes or blocks them by the result. Firewall is distributed two types. One is installed in the point the internet connects, and the other is set on personal computer systems. We name the former Internet firewall and the latter host firewall.

If attackers know a firewall policy of a target network in outside, they can perform the easy and effective attack. It is to say that, if attackers uses packets only comprised of traffics permitted by the firewall policy, they can execute an effective attack on the target network. On the contrary, if administrators define their firewall policy in a difficult way, they can defend attack effectively from outside. In addition to the internet security aspect, inferring an internet firewall policy can improve performance of distributed process application or peer-to-peer application because both the processes depend on host availability or accessibility in virtual network. Inferring a firewall policy is the key to reach those results. It is also helpful to practice network fingerprinting with effect; network fingerprinting is the way to distinguish the target network from others. The generally used networks are about understanding the host availability distribution or network connection information of a target network. Those elements can be gathered by sending test packets to the target network. If you already know the firewall policy, you can exclude the unnecessary test packets and generate the effective network fingerprint. In other words, if gathered information of a firewall policy is correct, that firewall policy can be used for network fingerprint.

In this paper, we propose an efficient method for inferring a firewall policy. As mentioned, Firewall policy inference for the constructive as well as for Internet security. Firewall policy is not easy to infer. The simplest firewall policy discovery method is, infers a firewall policy according to result to send all possible probing packets. But, this method is impossible due to the many test packet needed. Therefore, it is required to discover a method, where firewall policy sends only limited probing packet.

The Internet firewall policy is defined by the network administrator based on security policy of each corporate. The firewall policy consists of several set of rules, each rule classifying a condition and an action. The condition is defined as an element to determine whether to permit network transmissions after checking inbound/outbound packets or not and the action is defined as an element to perform filtering packets according to a decision based on the condition. When the packets matches all of these condition, they will be allowed to pass the firewall. In general, firewall policy consists of following policy.

1) Default deny rule: The default rule of firewall policy uses a deny rule. This rule is to deny all packets that are not included in the permission rules. It may not be explicitly contained in the firewall policy.

2) Permission rules: Permission rules are consists of conditions on packets to pass the firewall. When the packet arrives on the firewall, at least one the condition is checked and action will be taken accordingly.

3) Priority of rule: The firewall policy exists priority between rules..Priorities between the rules determine whether to pass the packets or not upon applying the rule.

Table 1 shows an example of firewall policy defined according to policy construction method and it is the common format of firewall policy.. Rule R1 and R2 in a local network consists of IP address 192.168.10.1 to 254 and allows the external web connection. Rule R3 and R4 is only to allow internal web connection in a local network, manage web server of IP address 192.168.10.10. This method is more secure than another methods used often [9]. Many firewall policies consists of this method. The firewall policy in which the default rule is 'accept all' and followed by deny rules is not generally used. DDoS attack would be more efficient if we know the rule of firewall. In other words, DDoS attack is more efficient when all the allowing rules are applied, this is because the firewall calculation is would be more to block packet. As the above example, if DdoS attack with packets except web traffic against the firewall; these packets are the most expensive ones on

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0028239).

TABLE I. AN EXAMPLE OF FIREWALL POLICY

Rule	Direction	Protocol	Src IP	Src Port	Dst IP	Dst Port	Action
R1	Outbound	TCP	192.168.10.*	1024:65535	Any	80	Permit
R2	Inbound	TCP	Any	80	192.168.10.*	1024:65535	Permit
R3	Inbound	TCP	Any	1024:65535	192.168.10.10	80	Permit
R4	Outbound	TCP	192.168.10.10	80	Any	1024:65535	Permit
Default	Any	Any	Any	Any	Any	Any	Deny

performance. Priorities between rules can be calculated by taking time and packets delays into response packet. But, this accuracy has difficulty to high reliability.

In this paper, we propose a method which infers a firewall policy. Inferring a firewall policy is semantically equivalent to the policy of firewall deployed in the Internet access point from out of target network. To discover firewall policy of target network, we generate probing packets, send them and infer the policy by analyzing response packets. First, we assume that the default deny rules exists, add to the allow rule sequentially in policy. If we receive a response packet for sent probing packet, add to the allow rule about response packets header information. To add an allow rule effectively, determine next probing packet based on the firewall response. We need to select probing packets so intelligently it effectively infers the firewall policy. It is very important how to select probing packets. The firewall policy for target network should be constructed by sending and receiving packets repeatedly. At last constructed firewall is semantically equivalent to the policy of firewall deployed in target network. If many of packets sent to target network during this process, there is a concerned that these packets may be recognized as an Internet attack like IP address or port scanning. Therefore, packets are needed to be selected intelligently. Intelligently selected probing packet are sent to firewall, we can construct permission rule of the firewall according to response of packet for sent packet.

Such method for inferring a firewall policy has been used in previous work. In FireCracker [2], they solved the problem to infer the policy by converting space searching problem using several space searching algorithm. And they evaluated the effectiveness of each algorithm. If they are considered only source IP and source port in the rule R1 of Table 1, the problem of searching the rule R1 is interprets the following. Search space is interpreted with a 2-dimensional plane for outbound traffic. X axis is 192.168.10.0 to 192.168.10.255. Y axis is 0 to 65535. They are search rectangle representing the conditions of permitting packet in 2-dimensional (i.e., (192.168.10.0:1024), (192.168.10.255:1024), (192.168.10.255: 65535), (192.168.10.0:65535)). Of course, FireCracker wasn't exactly 2-dimensional. They launched space search in 6-dimensional considering all of fields. Each dimension was corresponded to each field values used for inferring a firewall policy. If you use a field value of n, it represents *n*-dimensional space. Each rules in the firewall policy within *n*-dimensional space represented by 1 to *n*-dimensional shape. Thus, shapes in the space are represented, indicating permit space. FireCracker is used for various space search algorithm for searching permitted space. The firewall policy is inferred by searching permitted space to packet in *n*-dimensional space using space search algorithm.

But, FireCracker was considering only Internet firewall. In general, the Internet firewall was deployed in the Internet access point. It provides comprehensive policy about internal network. A rule of Internet firewall policy defines consecutive values for IP address, port number and etc. Therefore, space search algorithm proposed in FireCracker can infer a firewall policy with proper accuracy about Internet firewall. The firewall policies proposed in FireCracker is represented as a hyper-rectangle in the multi-dimensional space.

In this paper, we are proposing a method improved than FireCracker. Approach for problem of space search is the same as the FireCracker. But, we were proposing method for inferring a firewall policy to the problem of searching the shape of the 1 to *n-1*-dimensional rather than 1 to *n*-dimensional shape in *n*-dimensional space. We investigated firewall policies in the actual network. As a result, the firewall policies are represented as shape of 1 to *n-1*-dimensional. From *n*-dimensional space, can easily search shape of 1 to *n-1*-dimensional than search shape of 1 to *n*-dimensional. Thus, our proposed method were effective than the FireCracker.

The firewall policies are represented by shapes of 1 to *n-1*-dimensional. The firewall policy was reflected in the Internet firewall as well as in the host firewall and also host is live or not, for inferring a firewall policy externally. In the current Internet, the firewall is deployed in the Internet access point. Furthermore software firewall is performing by Operating System (OS) at internal host of Internet. The FireCracker is focused on inferring an Internet firewall. If the packet sent from source host to destination host, even if they pass the Internet firewall, they are blocked by Internet firewall on each host. In this case destination host does not receives the response packet. Therefore, we combine the firewall policy considering an Internet firewall and host firewall for inferring an accurate a firewall policy. We verified the inferred a firewall policy by applying actual network proposed method, and we measured accuracy of the inferred a firewall policy and efficiency of method for inferring a firewall policy.

In section 2, we provide a simple overview of the related work. Section 3 describes the theoretical background for inferring a firewall policy. In section 4, we describe details the that proposes method for inferring a firewall policy. In section 5, we show the experimental results and the performance of the proposed method. Section 6 has plan about idea to get through experiment and future work.

II. RELATED WORK

Firewalk is a detection tool which finds the packets passed through the network firewall or the Internet router [3]. Initially, it probes the routing path to the firewall or router. This is possible to send packets by increasing TTL values and to analyze response packets like Traceroute. Then, Firewalk sends the packets with TTL values which adds the length of the found routing path to one. These packets are passed through or rejected by the firewall or router. The packets which passed through the firewall or router triggers ICMP time exceed messages by next network device to sender. Rejected packets are dropped and do not trigger any response. The Firewalk finds the packets which are passed through the firewall or router continuously by changing Internet protocol, destination IP address, and port. But this tool does not provide the intelligent scanning method. The limitation of this method is, which is efficient only for small searching space because it sends packets to all of destination address and port. Also, since most of the firewall and router, filters the ICMP packets, Firewalk cannot guarantee to work exactly always. However Firewalk provides the method to separate Internet and host firewall policy. In this paper, we distinguish between Internet and host firewall using this method.

The firewall analysis tool analyzes and verifies the policy of the deployed or pre-deployed firewall. It loads the firewall policy which describes the low-level language, gets input about the network topology and verifies the firewall policy. Also, it can help those administrators to make the effective and efficient policy. Alain Mayer et al. developed the firewall analysis tool named FANG[4]. This tool reorganizes the policy using network topology and configuration of the firewall. If an administrator queries to FANG, it displays analysis result based the firewall policy. Administrators find out wrong rules of the policy effectively. Later FANG was upgraded to generate specific queries automaticity. The firewall analysis tool is not our goal because it can discover the policy using network topology. But understanding the firewall and the method to reorganize the policy can be used to infer the rules of the policy one by one and it is our future work. Other tools aims at finding out about conflicts in the policy [6, 7, 8]. Ehab S et al. proposed the method to discover conflicts of the policies among distributed firewalls. Mohamed G et al. proposed the method to optimize the policy of a deployed firewall. In this paper, we are not concern with optimizing the policy. We have the plan to optimize the inferred policy by using related work.

Previous work to infer the firewall policy was performed by Tagrid Samak et al [1, 2]. They solved the problem by inferring the policy by converting space searching problem using the novel framework named FireCracker. As mentioned, one of filtering conditions consisted of Direction, Protocol, Source IP address, Source Port, Destination IP address and Destination Port fields. Action is determined whether to permit network transmissions or not. FireCracker searches multi-dimensional shapes in 6-dimensional space which consisted of six condition fields. The defaults deny rule covers whole 6-dimensional space. Multi-dimensional shapes correspond with each permission rules. Rule R1 in Table 1 correspond with a rectangle, which is searched by using Region Growing algorithm. Split-and-Merge and Genetic algorithm were proposed, too [2]. As results of the experiment, they proved that Region Growing algorithm is efficient and better than other algorithms.

Our approach is similar to Tagrid Samak's approach which uses space searching, but we are concerned with searching 1~*n-1* dimensional shapes in *n*-dimensional space, instead of *n*-dimensional shapes. For instance, we search 1-dimensional shapes such as lines, 2-dimensional shapes such as rectangles in 2-dimensional space. By decreasing 1-dimension, Searching space was permitted efficiently.

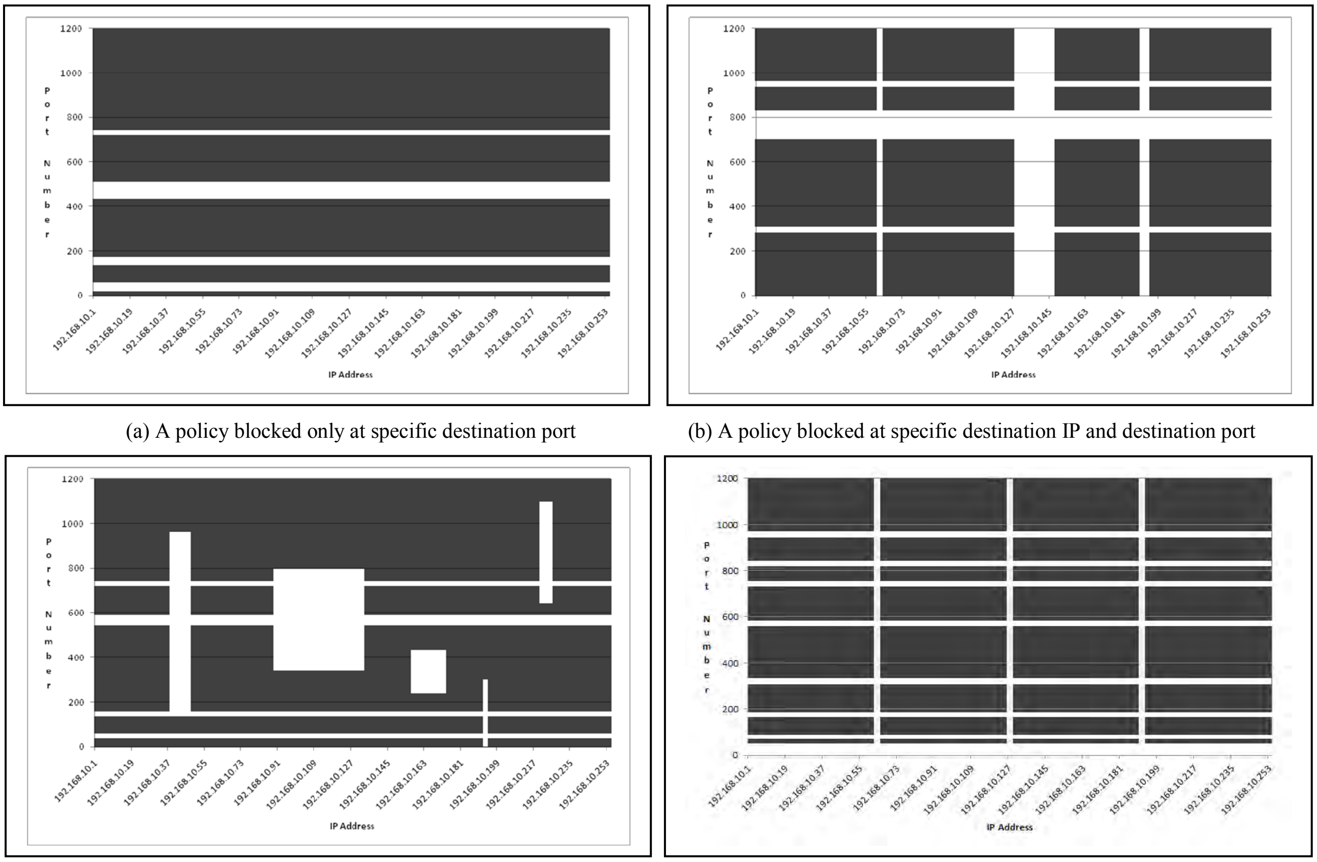
III. THEORETICAL BACKGROUND

As already mentioned, we changed the inferring a firewall policy problem that converts polygon from n dimension to 1~1-n dimension. For example, we search a line from 2-dimension to 1-dimension. Previous work searched rectangle from 2-dimension to 2dimension. Validity of our method is proved a thought of previous work. In addition, this paper suggests effective method of proposed space search. In this section, we explain inference results are more useful than previous work.

Direction fields of firewall policy are only two types for inbound and outbound, so search space is small. In the case of protocol, we consider to restrict protocol about TCP, UDP, ICMP, IP, SNMP, RTP and etc. As a result, search space is small. Primary meaning of the fields of search space are source IP, source Port, destination IP and destination Port.. When a fire wall policy is constructed using four fields, permission rules are comprised of operating server which allows access from internal to external.

The characteristics of the firewall policy should be constructed so that, it's should be smaller than dimensional shapes of the search space. In other words, external access which has specific IP, replaces restriction and effect of restriction decrease on one dimension on. As per the discussion made on the server restriction, same effect is appeared by specific server restriction on connection, which allows the access from internal to external. For Example Table 1 shows the effect of restriction of rules R1 and R2 on destination port source port numbered 80 respectively.

Analysis of experimental results is made on validity of existing firewall policy, where method searches 1 to n-1 dimensional shapes in n-dimensional space. Figure 1-(a) has a results which drew for the firewall policy where destination port and destination IP address are of local network of class C using firewalk method. This consists of sent packet which combines all IP address and well-known port. Therefore, this is same as the actual firewall policy. In this example, the policy was applied to blocked policy only at specific destination port. Figure 1-(b) is results which drew firewall policy for C class. In this example, this policy is applied to blocked policy at specific destination port and specific destination IP address. Figure 1-(c) and figure 1-(d) is results which are drew for firewall policy for firewall of actual network. The figure 1-(c) and figure 1-(d) is a combination of figure 1-(a) and figure 1-(b). We investigated the firewall policy through all sent probing packets in a local network of C class on twenty. As shown in Figure 1, the firewall policies were presented to four



(a) A policy blocked only at specific destination port

(b) A policy blocked at specific destination IP and destination port

(c) A policy combined figure 1-(a) and figure 1-(b) from network A

(d) A policy combined figure 1-(a) and figure 1-(b) from network B

Figure 1. Example of the existing firewall policy analysis

types.

To infer a firewall policy which is suitable for method of searching shapes in *n-1* in *n*-dimensional space through analysis of firewall policy. We should be describing a method that perform search effectively. We describe method for searching 1-dimensional straight line in 2-dimensional plane. We search a straight line which is not diagonal line, but it should have characteristic of vertical or horizontal. The method for search of straight line uses a line sweep algorithm. The idea behind algorithms is to imagine that a line (often a vertical line) is swept or moved across the plane, stopping at some points in Euclidean space. The algorithm calculates inferring a firewall policy at specific points. Once the line search is completed results are calculated Typical example of Sweep algorithm application is on 2-dimensional space, search the crossing points where line exits. We apply sweep algorithm to search start and end points of crossing line. The sweep line is the inclined line at an angle of 45 degree because lines to be searched are horizontal or vertical lines. Also this algorithm can search both horizontal and vertical lines simultaneously. Searching intervals of the sweep line are modified by changing the crossing points. If the line before sweeping and the other after sweeping are the same line, then searching interval is increased. If it is not the same line, then searching interval is decreased.

The line sweep algorithm executes as follows:

1) Generate packets covering sweep line which is an angle of 45 degree between start point and end point

2) Random probe for the generated packets

3) Searching the cross point for correspondent permission rules

4) Move the sweep line as searching interval and generate packets covering sweep line

5) Compare before sweeping and after sweeping

6) Repeat until the total space to search

Figure 2 shows an example of the line sweep algorithm on a 2-dimensional space. The sweep line is indicated by gray dotted line, and the sweep line of previous step is indicated by black line. According to step 2 to step 4 in figure 2, the searching interval is increased by double because the sweep line and the cross line are same. If the sweep line and cross line are differ, the interval would be half. We search the start point and the end point of the cross line and apply the searched cross line to the firewall rule. The sweep line is implemented by sending probing packets. Line sweep algorithm has an advantage of discovering the points crossed horizontal and vertical lines easily. Also, it overcomes the weak point that the probing packets are classified as port scan packets by the firewall.

If the n value defines the length of the x-axis and m value

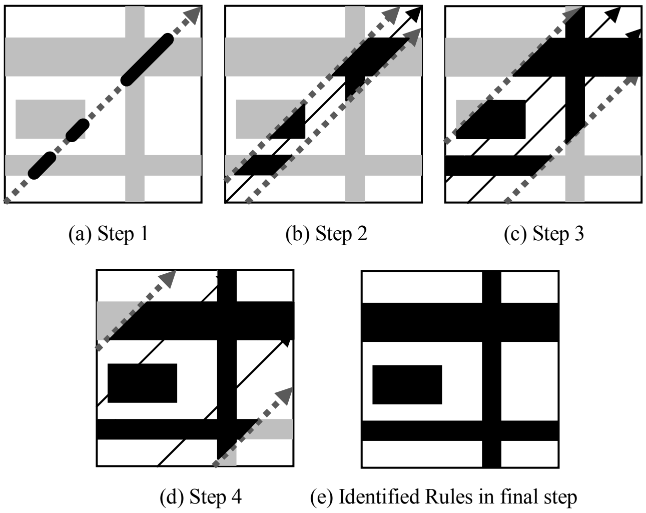


Figure 2. A simple example of line sweep algorithm

was defines the length of the y-axis from the proposed method, time complexity is O(n*m) in the worst case. Average time complexity of Region Growing method in the previous work was O(k*log n*log m), when the number of search shapes was defined as k. We propose method having the advantage regardless of the number of search shape. Region Growing method has been reduced time through expand by exponential increase about search space. If search shape is close to the line, time complexity is O(k*n*m). Therefore, we propose method that is better than previous work in the worst case.

IV. Inferring Method

We infer both the policies of the Internet firewall and host firewall, and then integrate those policies. To infer the policy, we need to find out the packets which are not filtered through several firewalls from the source host to the destination host. These packets are used to construct the integrated firewall policy.

To discover the permitted rules from all the firewalls on routing path, we generate probing packets, send them and infer the policy by analyzing response packets. If probing packets are generated using all the fields such as protocol, source address, source port, destination address and destination port, we can infer the firewall policy more accurately. But it takes many hours and consumes system resources. So we need to generate probing packets intelligently.

We generate probing packets using only destination address and port. In other words, we infer the policy related to two fields. There are three reasons which choose destination address and port. First, the proposed method in this paper is extended easily in case of adding other fields. Searching 1-dimensional lines in 2-dimension space can extend to searching 2-dimensional shapes such as rectangles in 3-dimensional space. Second, the policies of most of firewalls are consisted of destination address and port to control inbound packets from the Internet. Third, using two fields is proper to utilize the inferred policy because the inferred policies use for trying to the Internet attack or connecting to target network from distributed processing applications such as P2P. These policies are enough for connecting from external to internal network.

A. *Inferring the Internet firewall*

The Internet firewall is deployed at external point of network. Inferring the firewall policy should need to separate the policies of the Internet firewall and host firewall. Host firewall policy is possible to be inferred if the host is live. One of several reasons to make inferring the policy cannot be difficult ,suppose that the host is always live. Therefore we should infer each of the firewall policies. To infer the Internet firewall policy, we use the method of Firewalk. Probing packets are generated with TTL values assigned the length of path to firewall plus one, and are sent them to destination address. If probing packets are not filtered by the firewall, we can receive the ICMP Time Exceed message. If filtered, we cannot receive any response packets. But some of firewalls filter out ICMP packets to the Internet. In this case, it is impossible to infer the separate policy for Internet firewall and host firewall.

We can understand whether we can infer the policy using the method of Fire walk or not by sending DNS queries twice to DNS server inside target network. One query is sent with large TTL values that DNS server can receive the query certainly, and another query is sent with the TTL value which is assigned the length of path to firewall plus one. If the response of first query is received and the response of second query is not received, we do not infer the Internet firewall policy because ICMP packets are filtered by the Internet firewall. We researched to find DNS server per each AS and utilized it in this paper. Our proposed method is valid because 19,290 of AS(Autonomous System)es operate DNS server in 33,756 of total ASes. And it is proper since most of DNS server is operated inside target network and permits inbound query packets. If the Internet firewall policy can be inferred, we infer the policy using Line sweep algorithm discussed in Chapter 3.

B. *Inferring a host firewall*

Host firewall policy installed in PCs, server systems and network devices. Host firewall has various policies. Software firewall of PCs is enabled or disabled. Server systems apply only rules for operated services to the firewall policy. Special devices such as network devices or printers cannot support firewall functions or can permit limited packets for own services. Hosts which have Specific IP addresses are not existed or may be off.

Most of the time is waiting time; we didn't receive a response for sent probing packet. If infer an Internet firewall, which is able to receive ICMP packets and does not take much time. But, host firewall is without the response should be prepared to be intelligently, because most of host firewall does not send a response. Means intelligently prepare should have reduced the number of probing packets with no response to identify quickly for no response.

Host of specific IP does not exist or power of host system should be identified quickly. To do this, we should take advantage of the well-known port. We send two packets once

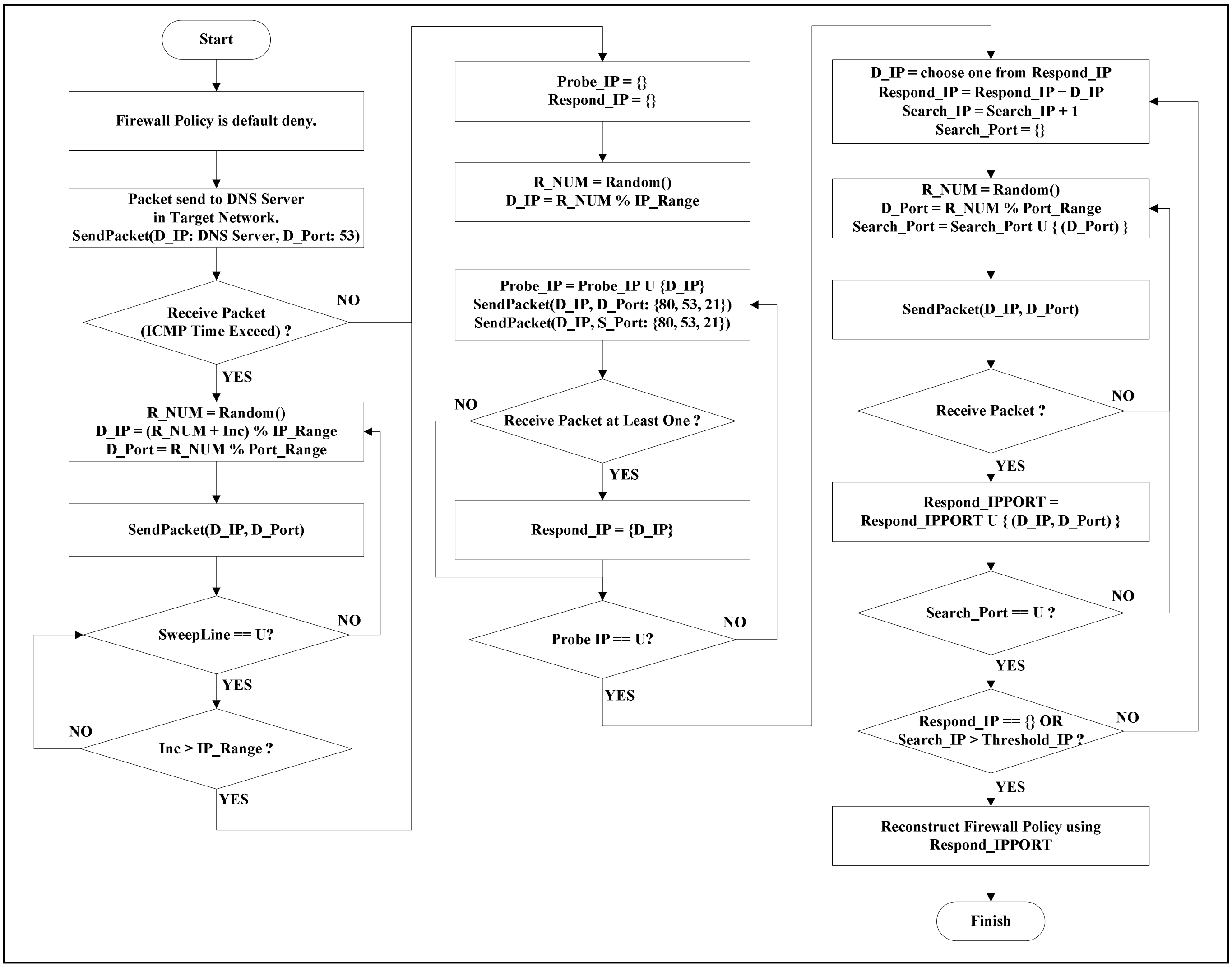

Start
Firewall Policy is default deny.
Packet send to DNS Server
in Target Network.
SendPacket(D_IP: DNS Server, D_Port: 53)
Receive Packet
(ICMP Time Exceed) ?
NO
YES
R_NUM = Random()
D_IP = (R_NUM + Inc) % IP_Range
D_Port = R_NUM % Port_Range
SendPacket(D_IP, D_Port)
SweepLine == U?
NO
YES
Inc > IP_Range ?
NO
YES
Probe_IP = {}
Respond_IP = {}
R_NUM = Random()
D_IP = R_NUM % IP_Range
Probe_IP = Probe_IP U {D_IP}
SendPacket(D_IP, D_Port: {80, 53, 21})
SendPacket(D_IP, S_Port: {80, 53, 21})
Receive Packet at Least One ?
NO
YES
Respond_IP = {D_IP}
Probe IP == U?
NO
YES
D_IP = choose one from Respond_IP
Respond_IP = Respond_IP − D_IP
Search_IP = Search_IP + 1
Search_Port = {}
R_NUM = Random()
D_Port = R_NUM % Port_Range
Search_Port = Search_Port U { (D_Port) }
SendPacket(D_IP, D_Port)
Receive Packet ?
NO
YES
Respond_IPPORT =
Respond_IPPORT U { (D_IP, D_Port) }
Search_Port == U ?
NO
YES
Respond_IP == {} OR
Search_IP > Threshold_IP ?
NO
YES
Reconstruct Firewall Policy using
Respond_IPPORT
Finish

Figure 3. Flow chart for inferring a firewall policy

to the source port and once to the destination port about TCP/IP of target host; set to the port 80(HTTP), 21(FTP) and 53(DNS). If we receive the TCP ACK/RST or ICMP Port Unreachable responses, we consider that the host can respond some of requests. We infer the firewall policy which has only these hosts.

To infer the firewall policies of these hosts, we send probing packets to many source ports and destination ports. Ports are the logical numbers to distinguish the communication in the applications and consist of well-known ports from 0 to 1023, registered ports from 1024 to 49151 and dynamic ports from 49152 to 65535. Specially, well-known ports were defined as specific services by IANA. We probe only well-known ports instead of total ports.

Inferring a firewall policy based on port numbers can be classified into three categories. First, the policy responds to all ports. Second, the policy is responds to specific ports. Third, the policy is not responds to all ports. If the policy of first example, where firewall is not installed, like network switch device or network printer, or the firewall of host is disabled. Second example, the firewall of host is enabled. This firewall is used to allow Internet connection for only specific application. The host of third policy in the example has not used IP address, or shut off the power on host allocated of IP address, and the firewall is blocked to all packets. Third policy is already searched well-known port, searched first firewall policy and second firewall policy. The two firewall policies perform for

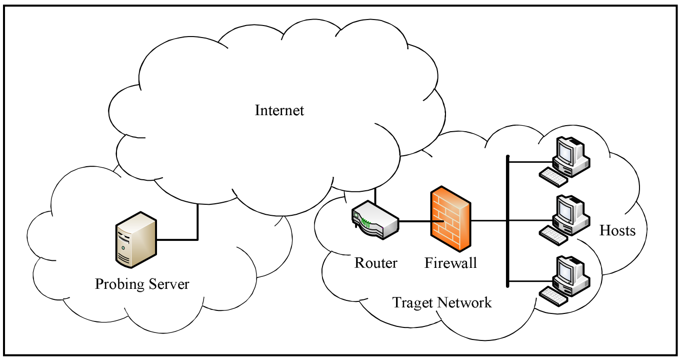


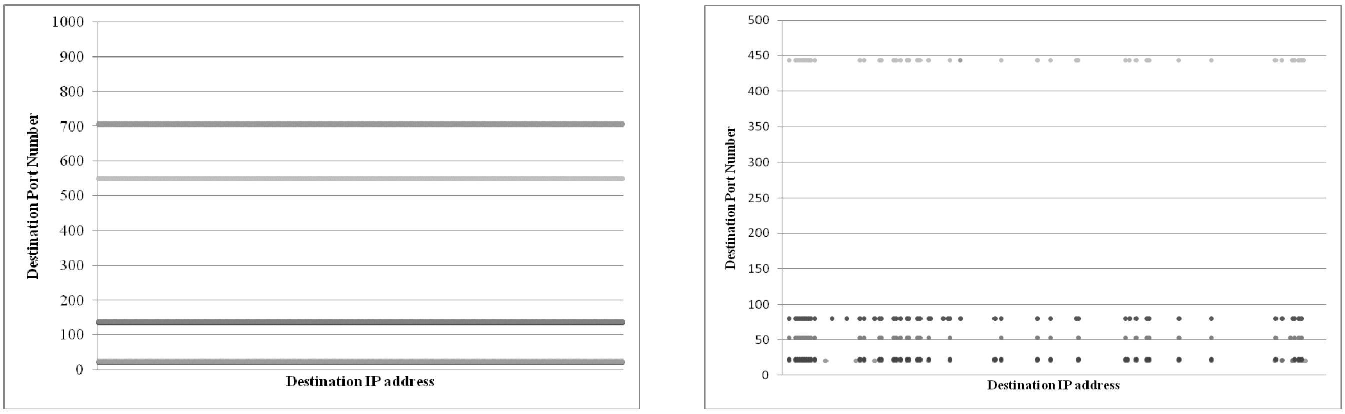
Figure 4. Configuration of network for inferring a firewall policy

TABLE II. THE INTERNET FIREWALL POLICY OF NETWORK A

Rule	Protocol	Src IP	Src Port	Dst IP	Dst Port	Action
R1	ICMP	Any	Any	192.168.10.*	-	Permit
R2	TCP	Any	Any	192.168.10.*	22:23	Deny
R3	TCP	Any	Any	192.168.10.*	135	Deny
R4	TCP	Any	Any	192.168.10.*	137	Deny
R5	TCP	Any	Any	192.168.10.*	139	Deny
R6	TCP	Any	Any	192.168.10.*	550	Deny
R7	TCP	Any	Any	192.168.10.*	707	Deny
R8	TCP	Any	Any	192.168.10.*	1:1023	Permit
Default	Any	Any	Any	Any	Any	Deny

TABLE III. THE INTERNET FIREWALL POLICY OF NETWORK B

Rule	Protocol	Src IP	Src Port	Dst IP	Dst Port	Action
R1	ICMP	Any	Any	192.168.10.*	-	Deny
R2	TCP	Any	Any	192.168.10.*	21:23	Permit
R3	TCP	Any	Any	192.168.10.*	53	Permit
R4	TCP	Any	Any	192.168.10.*	80	Permit
R5	TCP	Any	Any	192.168.10.*	443	Permit
Default	Any	Any	Any	Any	Any	Deny



(a) Distribution of packets denied from network A of C Class

(b) Distribution of packets permitted from network B of C Class

Figure 5. Distribution of response packet for inferring a firewall policy

possible host response. All hosts do not need to be used for inferring the Internet firewall policy. We discover the threshold of the number of IP addresses by experiment. We probe the IP addresses to infer the host firewall policy using this threshold value, too.

Inferred host firewall policy can infer to the Internet firewall against target network. Of course, if inference of the Internet firewall policy is impossible according to blocked ICMP packet, then only apply. The Internet firewall is inferred by combining the common policy of each host and the firewall policy is determined as the first firewall policy example.

V. EXPERIMENTAL RESULT

To infer policies of the Internet firewall, policies should be applied on actual network presented in the previous chapter. To infer policies of the Internet firewall for configuration of network in Figure 4. Shows the Firewall infer system to detection packet of send and receive by used Internet connection, where security system such as firewall or IDS does not exists. Tool packet of detection used for sending and receiving is Hping. Hping is used as a tool for the detection packet for sending and receiving, also offer variety of options and provide information to enough for analyzed packet of receiving.

We performed a detection of firewall policy on two networks A and B. The Network A is the network of any university in a variety server, personal computer, printer and network device is a mixed network. The Network B consisting of server farm and maintain a high level security. A is Network such as Firewalk of existing network, length value of the TTL is router path plus 1 as a response when TCP/IP packet is sent. Firewalk is simple tool to control the TTL value. But, our proposed method effectively controls the TTL value of the sent a packet by applying the Line sweep algorithm. Figure 5-(a) shows result of response of sent packet for a infer policy of the Internet firewall about network A. In this case, detection of firewall policy in run time was 19 minutes and total number of sent packets where detected and were 2,286 in number. Table 2 shows firewall policy result. Existing methods were compared with FireCracker. When detection was made by FireCracker run time was found was found to be 25 minutes and total number of sent packets were 3,056. Verify the accuracy of inference policy in all cases, sent a packet has to completely draw the firewall policy. At this point of time, firewall classified as port scan to detect the packets which are been blocked or not. And proper time interval to detect the sent packets The accuracy of the inferred firewall policy can be calculated by number of packets in the denominator and matching the two policy number of packets in the numerator. In our work, the accuracy of the proposed detection method 98%, and the accuracy of FireCracker is 98%. These results are primarily firewall policy of the network A because to block is based on port, difference between accuracy and number of sent packet are interpreted.

For Network B a strong firewall policy is been applied, control of the TTL value used to infer policy of the Internet firewall is impossible. Therefore, to detect host firewall and it should be to infer Internet firewall. Figure 5-(b) shows method proposed in this work to detect of packet for response result. In these cases Firecracker could not detect a firewall policy and cannot be compared. Figure 5-(b) draw firewall policy is shown Table 3. In this case, sent packet draw every field of the complete firewall policy and result of accuracy of derive firewall policy is 99%.

VI. Conclusion

Internet connected to each corporate or organization prevents internal hosts by firewall so as to filter out inbound/outbound bad traffic to/from the network. In this paper, we propose a method to infer policy of the structure of firewall for exist source host to destination host in the whole path of network for filtering the traffic in the network. Proposed method can deny the permission of the packet from source host to the destination host. Thus, when we want to send a random packet to destination host, this packet denied or restricted by the firewall and can determine previous deny and permission by firewall rule. We don't have to use the packets denied and can replace them with packets permitted.

However, if firewall regard probing as malicious and drop the packet, it should be impossible to make consecutive probing. That is a really important but practically insoluble problem. Therefore, to avoid the problem, we sent only 2 probing packets in one transmission interval and confirmed the response. Then, we performed the next behavior after enough transmission interval. The following research will reflect the problem, and distinction between inbound and outbound in the firewall policy also will be a future work. In addition, it should be expanded to protocol, source Port, destination IP and destination Port. In this paper, though we only infer the stateless based policy, the state-based policy also ought to be inferred in the future.

References

[1] Taghrid Samak, Adel El-Atawy, Ehab Al-Shaer, and Hong Li. "Firewall policy reconstruction by active probing: An attacker's view." In The Second Workshop on Secure Network Protocols (NPSec 2006), 2006.

[2] Taghrid Samak, Adel El-Atawy, and Ehab Al-Shaer. FireCracker: A Framework for Inferring Firewall Policy using Smart Probing. Icnp, pp.294-303, 2007 IEEE International Conference on Network Protocols, 2007

[3] David Goldsmith and Michael Schiffman. Firewalking: A traceroute-like analysis of ip packet responses to determine gateway access control lists, http://www.packetfactory.net/firewalk/firewalk-final.html, October 1998.

[4] Alain Mayer, Avishai Wool, and Elisha Ziskind. Fang: A firewall analysis engine. SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy, 00:0177, 2000.

[5] Avishai Wool. Architecting the Lumeta Firewall Analyzer. In Proceedings of the Tenth USENIX Security Symposium, August 13–17, 2001, Washington, DC, USA, 2001.

[6] Ehab S. Al-Shaer and Hazem H. Hamed. Firewall policy advisor for anomaly discovery and rule editing. In IFIP/IEEE Eighth International Symposium on Integrated Network Management (IM 2003), pages 17–30, 2003.

[7] Ehab S. Al-Shaer and Hazem H. Hamed. Discovery of policy anomalies in distributed firewalls. In The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), 2004.

[8] Lihua Yuan, Jianning Mai, Zhendong Su, Hao Chen, Chen-Nee Chuah, and Prasant Mohapatra. Fireman: A toolkit for firewall modeling and analysis. In 2006 IEEE Symposium on Security and Privacy (S&P 2006), pages 199–213, 2006.

[9] Ken Cutler John Wack and Jamie Pole. Guidelines on firewalls and firewall policy. NIST(National Institute of Standards and Technology), January 2002. Special Publication 800-41.

[10] Mohamed G. Gouda and Alex X. Liu. Firewall design: Consistency, completeness, and compactness. In The 24th International Conference on Distributed Computing Systems (ICDCS 2004), pages 320–327, 2004.

[11] James W. Mickens, John R. Douceur, William J. Bolosky and Brain D. Noble. "StrobeLight: Lightweight availability Mapping and Anomaly Detection."USENIX'09 Proceedings of the 2009 conference on USENIX Annual technical conference USENIX Association Berkeley, CA, USA ©2009

[12] Michael Shamos, Dan Hoey, "Geometric intersection problems", Proc. 17th IEEE Symp. Foundations of Computer Science, p. 208–215, 1976.