

Firewall Penetration Testing

Reto E. Haeni
r.haeni@cpi.seas.gwu.edu

**The George Washington University
Cyberspace Policy Institute
2033 K Str. Suite 340 N
Washington DC 20006**

Washington DC, January 1997

Index

1. Abstract	3
2. Introduction	4
3. What is a Firewall?	4
3.1 Limitations of Firewalls	6
4. Evaluating a Firewall's Security	7
5. Problems of Firewall Penetration Testing	8
6. Basic Rules at the beginning	9
7. Questions to answer with Firewall Testing	10
8. Methodology	11
8.1 Indirect information collection	11
8.2 Direct information collection	12
8.3 Attack from the outside	14
8.3.1 Packet filtering Firewalls	14
8.3.2 Proxy Firewalls	19
8.4 Attack from the inside of the network	20
9. Testing techniques/tools	21
10. Conclusions	23
11. References	24
12. Appendix	25

1. Abstract

Firewalls are often regarded as the only line of defense needed to secure our information systems. A firewall is a device that controls what gets in and comes out of our network. Unfortunately, a firewall has also its weaknesses if not installed properly and if we don't implement an appropriate security policy.

In this paper, I describe a methodology to perform firewall penetration testing. Before we perform the actual testing, I also discuss how to decide who will perform it. The firewall vendor and hackers are in my point of view not a solution. We are looking for an independent group that we trust for integrity, experience, writing skill and technical capabilities. If we have these resources in our company then we can perform the test by ourselves; otherwise we can outsource it.

The firewall testing is divided into four steps:

- ◆ **Indirect information collection**
- ◆ **Direct information collection**
- ◆ **Attack from the outside**
- ◆ **Attack from the inside**

We have principally two types of firewall and I list here the most basic attack approaches. These attacks are tailored to the type of firewall we are testing.

Packet filtering firewall

- * Blind IP-Spoofing
- * Non blind IP-Spoofing
- * Source porting and source routing

Application level firewalls (proxies)

- * Bad security policy
- * Policy poorly implemented
- * SOCKs incorrectly configured
- * Brute force attacks
- * Enabled services/ports

Security scanners can be of help in conducting firewall testing but cannot replace manual tests. To just run a security scanner against a firewall should not be accepted by the client as a penetration test.

If a penetration test is done properly by experienced people it can provide valuable feedback on the effectiveness of a firewall. It can also be misleading. It does not mean that "we" are secure now! Passing a firewall test simply means that the firewall defeated all of our attack approaches. Maybe a hacker can think of something else and break into our systems exploiting a weakness we did not test for. However, firewall testing gives us a basic understanding that our firewall is working properly.

2. Introduction

Presently, it is almost impossible to open a computer oriented newspaper or magazine without seeing ads for new security tools or reports on system/network intrusions or denial of service attacks. Despite these reports, corporations want to have access to the resources on the Internet and to be able to provide information and services to their customers using this way of communication. While connection to the Internet has its big advantages (mainly in relation to research areas) the dangers can be substantial. Not only might employees surf the Internet (and therefore maybe less work gets done) but the risk of data disclosure (by intent or accident) increases with and internet connection.

While the above described dangers exist if the corporation's network is not directly connected to the Internet, Internet connections introduce an additional threat. While in the past (and it is still the case at the moment) system penetrations were mainly attempted from the inside of a network, suddenly with the connection to the Internet, we are vulnerable to attacks from all over the world.

The biggest threat is probably the initial time. When we are accessible from the Internet but our defenses may be not yet in place and/or tested.

*When they have not secured good terrain they can be attacked.
When their battle array is in disorder they can be attacked.*

Sun Pin [12]

To counter the threat of network/system intrusion and to be able to define what services are provided to the inside users, people are building and installing firewalls. Often, a firewall is regarded as the solution to all the network and system security problems for the whole organization. Unfortunately, firewalls (even correctly installed and set up) are not the magic bullet against all security threats. In addition, firewalls are often installed but either are not properly configured or are poorly managed and updated. A firewall needs a substantial amount of maintenance to prevent "erosion". Operating System updates and security patches have to be applied as well as applying updates and patches provided by the firewall vendor. In addition, to install a firewall leads to a false sense of security. Although this security device is in place, we are still vulnerable to a lot of threats that either take advantage of holes in the security policy or in the implementation of the firewall. Also if a firewall is guarding our network, we must not forget about the individual host security.

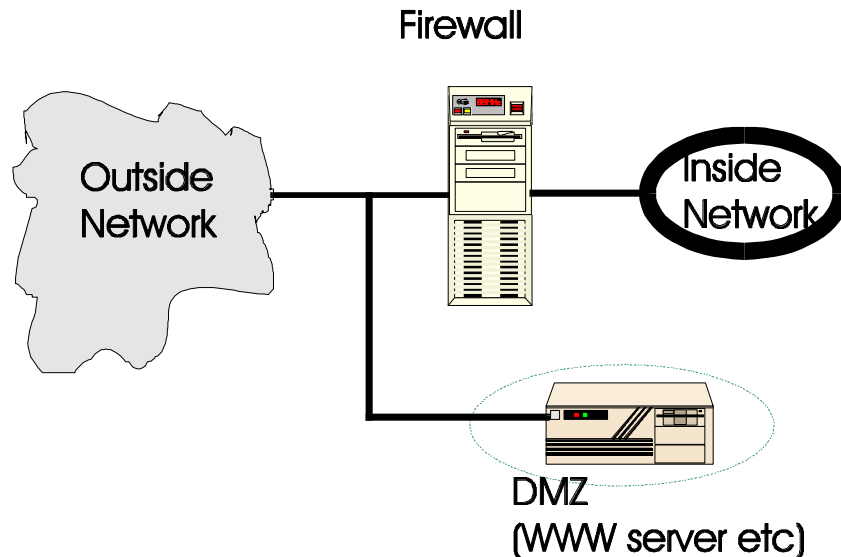
One way to provide a better understanding of both how well a firewall is installed and how well the security policy is implemented is to test a firewall. This topic will be discussed in this paper.

3. What is a Firewall?

To provide improved security, we have to have a way of controlling access to our systems and networks. This is principally what a firewall provides.

Firewalls control what gets in and comes out of our Network

To determine if data or access requests are allowed to pass the firewall (from or to the inside Network) the firewall has to be able to authenticate messages and the traffic has to be directed through the firewall. This leads us to the first question - where to situate the firewall.



Typically, the firewall is placed between an organization network and the outside world (although we also discussed the idea to place a firewall [application] on each local machine with its advantages and disadvantages). The outside world can be the Internet or another (untrusted) corporate network. We may want to protect information inside our network from being accessed by building internal firewalls.

If we want to provide access to the general population to a part of our services (like our WWW server or an anonymous FTP server) then we have to establish a demilitarized zone DMZ. These machines are connected to the outside world before the firewall performs its security functions and therefore are not protected by any means by the firewall. The description often used for these servers, “sacrificial lambs”, is colorful enough to describe that they must not contain any non public or sensitive information as the threat is very real for them to be overtaken by hackers.

How do firewalls fulfill their task?

Principally, there are two types of firewalls [2]

- Packet filtering gateways
- Application level gateways

Packet filtering gateways use the source or destination host to determine if the packet is allowed to pass the gateway. In general, no context is needed or kept and decisions are made based by the information in the header of the current packet. You can define which addresses are trusted to pass the gateway. This sort of “firewall” is normally quite cheap to implement, as today most

routers have filtering capabilities. However, logging and alarming is normally not supported by routers and FTP, X11 and DNS services are not easy to implement properly [2]. Another problem that could exist with using a packet filter as our firewall is the handling of IP fragments are handled. Normally, fragments are passed through the gateway as they are no threat to the inside system (either they can be reassembled and therefore the address/port of the first fragment was valid or they are dropped by the destination host) but if information leakage is a concern of yours, a packet filter may not be the best solution. A packet filtering gateway also has to have the capability to detect from which side (on which interface) a packet is arriving or it is vulnerable to an IP-spoofing attack (see 8.3.1.2)

Application level gateways (Proxies) do not rely on general purpose mechanisms to allow traffic to pass but use special purpose code for each desired service. We don't rely on the settings of filter rules and therefore we are not vulnerable to their possible interactions.

One of the big advantages of application level gateways is their capability to log all incoming and outgoing traffic. You can establish rules that allow only certain persons/departments to use an outgoing service (like WWW or FTP). This can help to prevent disclosure or theft of company property. In addition to logging, proxy type firewalls have the capability to examine these logs and produce an alarm (e-mail message, pager notification) to the responsible administrator. These messages are created when the connection attempts that are made seem to be hostile or non authorized. With these capabilities, the security personnel can either react to an attack or, after the incident, reconstruct what happened.

How does a proxy work? A proxy (also referred to as circuit level gateway) relays TCP connections. The caller connects to a TCP port on the firewall which then connects to the other side of the gateway and serves as a relay during the call. The proxy copies bytes back and forth. In this way, we do not have any direct connection from the inside to the outside world. Proxy type firewalls also provide us with better authentication capabilities. They let pass inside requests without further authentication if the authorization rules we established are fulfilled. We also can define outside hosts that can talk directly to inside machines without further authentication (for example to use a printer port on an internal machine). These firewalls also provide access control for incoming traffic. One implementation example in this case would be the *challenge response* one-time password authentication. If we want to establish a telnet session (just for example) to an inside host from home, then we are connected to the firewalls telnet port first. As usually, we are asked for an ID (login name) which we provide. Then, we get a random *challenge* back, something like 29503. This challenge is a integer number which we are entering in our token (looks like a small calculator). From this token, we get back the response *q4f2k81p* and this will authenticate us to the system. Now we are passed to our system and we log in there normally. An advantage of this system is that an attacker not only has to know a password but would also have to possess the hardware device (token). The *challenge response* authentication is not the only one-time password system available, but it provides in my point of view the best security of the ones with which I am familiar.

3.1 Limitations of Firewalls

While firewalls have their undeniable advantages in providing network security, they have also their limitations. To begin with, it is another single point of failure. We are connected to the

outside world by a router, which is our typical single point of failure. While one can argue that by installing a firewall in addition to the router we don't have an additional limitation I think that having two single points of failure installed successively is indeed an additional limitation. If the reliability of our connection to the outside world is vital, we might have to consider (hot) standby systems to counter this limitation.

Another draw back to firewalls is that they can be quite costly. While free firewall kits (TIS firewall toolkit, for example) exist and in some cases can be a sufficient solution to a problem, professional firewalls can cost substantial money for an enterprise. This cost factor can sometimes be the reason that a company decides not to connect their networks or decides to run with a low level of security.

While traffic flow on the Internet is high today and transmission rates are falling, the performance of a firewall can be the limiting factor. This is probably not a major concern if you want to connect your networks to the Internet, as the bandwidth is mostly small, but it can be a problem if you want to connect high speed intranets. I personally think that this is not too great concern as computing power is doubled roughly every 12 months and therefore solves the problem in time, as you will probably be able to exchange a firewall box more often and more easily for a faster one than to replace a costly high speed network.

Although firewalls are installed to provide network security, they are still vulnerable to attacks. They can be defeated by direct attacks; or the easier way to "break through" a firewall is by never touching it but simply bypassing it.

4. Evaluating a Firewall's Security

Before we buy a firewall (this would be the preferred option) or at least after we install one, we want typically to evaluate the security this device provides. To do so, we have principally the following options, as we do not have a firewall certification authority (and I am even not sure if it would be a good idea trying to standardize something so flexible, see also [7]).

- Vendor information
- Design analysis
- Examine Logs
- Firewall testing

The **firewall vendor** can provide us information on the level of trust that can be expected from a firewall. However, I don't believe that we can rely solely on this information as there are two potential problems that are involved when vendors are "certifying" one of their own products. It is true, that they know their creation (hopefully) best but they also will never be objective. At one hand, they will not be too thrilled to point you to a potential (maybe still undiscovered by third parties) weakness in a highly competitive market. On the other hand, there is a reason that resulted in a particular design and they will maybe not flexible enough to objectively rate the dangers posed to this design.

The second possibility would be a **design analysis**. This is an all theoretical approach. What do I mean with this exactly? You get as much design information on a particular product and preferably the source code and you begin to dig for security problems. These problems can be located in all

kind of places and therefore the search will be extremely time consuming. Just as an example, to have a look how the firewall is handling buffers and buffer overflows would be surely one of the many points to check. The problem with this approach is that it is very time consuming and requires considerable knowledge on firewall implementations. The additional and crucial part of it is that you rarely will get enough information and/or source code to do this test properly and even after testing the firewall toolkit, we would have to consider all the possible interactions with the operating system and the hardware on which we plan to run the firewall on.

The third approach would be to **examine logs**. For this, we install the firewall and set up the filter rules according to our policy. Then we let the firewall running for a certain time (week(s)) with the most detailed logging available. After a certain amount of time, we examine the logs in detail. This may take quite some time but we will get a sense of whether or not the firewall is working properly. In addition, logs will also be created if a service is used which we did not know was enabled or being used on the system. By examine the logs, we can detect and correct it. There are principally two problems with this approach. For one, when we find a security problem by examining the logs it can be already too late and an intruder can already have broken into our systems. Second, logging will never discover all attempts that have been made, and if we had a “smart” intruder, logs may have been altered.

The last item on my list is **firewall testing**, which leads us (finally) to the topic of this paper. This is in my opinion the most valuable technique of those discussed and should be performed on a regular basis. With this technique, we perform tests to attempt to penetrate the firewall as well as to bypass it. Technical problems such as known vulnerabilities as well as misconfiguration and badly implemented security policies are exploited, if possible. Principally, we challenge the firewall from a hackers standpoint.

5. Problems of Firewall Penetration Testing

As we have seen in the previous section, each of the evaluation approaches had its advantages but also its disadvantages. This is not different when using firewall penetration testing to evaluate the security of your firewall. The problems begin when you want to find information on this topic. There is not really much publicly available. In addition to the fact that firewall testing is a fairly new approach, nobody is really interested in providing too much information, as the people who could provide insights want to sell this service to their customers and not make the knowledge available for free. In addition to the limited amount of information, there are not too many good tools that you can use for this task. Most security scanners are based on operating system (OS) security which is different from network security. At this moment, ISS is the only vendor that provides a firewall testing tool (commercial version). This situation should get better in the near future as multiple other software products are in development state at the moment.

Another problem that we face with firewall testing is, that it is not as advanced as the design of firewalls. Vendors have been building firewalls for quite a while now and have developed a working knowledge of what is needed and what threats exist. Thus, firewall design approaches are far mor advanced than testing approaches, as this is a fairly new field.

One of the critical questions in relation to the topic discussed in this section is who performs the penetration testing. We have the following options:

- Testing by the vendor
- Hiring hackers
- Do it ourselves
- Third party

I personally would not recommend letting the vendor perform the test. Typically, we don't install firewalls solely by ourselves; the vendor is involved with it as well. Therefore during installation, they should already consider all the security problems. I also ask myself the question if they can come up with new threats during the test-phase that they did not think of during the design or installation of their product. Another thought is that the vendor of a product is maybe less creative when testing it as they know too well what is working on their firewall (or is supposed to work!).

The second possibility is to hire hackers. I strongly feel that nobody should do this. Although hackers may have the technical knowledge to test a firewall, I believe that they cannot have the integrity to ensure that your company's secrets are safe with them. They are criminals, as they probably obtained their knowledge with attempts to break into foreign systems (maybe yours). You will never be sure that hackers won't make any security holes they find available to other hackers and/or download access or alter your proprietary information and data. I also do not really like the term hackers. It has (besides its origin that describes good programmers) a too glorious sound. Vandals and jerks would be more my choice of names, and you probably wouldn't like to hire someone that falls in this category.

Third, firewall testing can be done by the company's employees themselves. Here, we have to make sure that the people that are involved with installing or managing/updating the firewall are not doing the testing. Tests have to be performed by an independent group. Principally, what we are looking for is integrity, independence, and experience. Another important rule is that the technical capability exists to perform the testing and last but not least that the testers have the ability to formulate the test results in a report in an understandable (also by non technicians) way. If all of the above points can be covered with a company's own resources, then we can do the test by ourselves. If not, and this will be quite often the case, we have to outsource this task. We are still looking for the same criteria here as before, but it is probably not easy to make the choice which firm to hire. Cost can be an important factor as a firewall test can be quite costly. It helps to get a quote form different firms; these should include a prior test report and references.

6. Basic Rules at the beginning

Now, we start discussing how to perform a firewall penetration test. Probably the most important point overall is planning. Included in the planning phase is the first step in the procedure: to get the approval of the management! This should be written down and must be obtained in advance. Don't do a test on a firewall or network just because someone (or a friend) is calling you and says

“Hey Joe, I just finished installing a firewall - Could you knock a bit on its doors to look if it is secure?”. You can get in serious trouble with this.

A part of this management approval will be an authorization list. This list establishes what we are allowed to do during testing and what we are not. This can mean for example to scan ports on a firewall but not perform *syn flood* attacks to disable the firewall. Then it is important that we (should go without saying) don't do more than is authorized by this list. Generally this means that we look for (potential) weaknesses but we do not exploit them. In the same spirit we are not disrupting or changing any systems except for what is approved by the client.

Finally, after doing most of the paperwork, we are getting more practical.

7. Questions to answer with Firewall Testing

As with every professional task, we have defined goals to achieve. In the case of firewall testing, we want to know first if the firewall policy (part of the information technology (IT) policy) is correctly implemented.

An internet firewall policy should contain the following points:[1]

- Security Requirements
 - * access control
 - * assurance
 - * logging
 - * alarming
 - * availability
- Required Functionality
 - * outgoing services
 - * incoming services
 - * services provided to the internet

I do not discuss policy issues here as this is not the scope of this paper but you will find a lot of publications discussing this topic if you search for it on the internet.

Another important point that we want to verify is that no leakages in the perimeter exist. We want our firewall to be the only way to communicate with the outside world. Unfortunately, the topology of a network is often not entirely known and there exist other connections to (mostly) the internet. These can be dial-up connections, ISDN connections or even T1 lines that are existent without knowledge of the network administrators. This is particularly the case in R&D environments where traditionally the groups/individuals have considerable autonomy. An additional danger exists with dial-in lines where the user can access the computer at work from home. The appearance of additional connections can increase when a firewall is set up as suddenly the access to services like WWW or FTP are restricted. Here, we need to have a clear policy that we can enforce that punishes such behavior or we will never get this problem under control. The strongest and best maintained firewall is of no use if we simply can bypass it.

We also want to determine if the logging of the events is adequate. We want not only to be able to control who is getting access to our network but also to track down an intruder and to find security holes when/after an incident occurs. In the same spirit, we want to test if the alarming is adequate. When a probe or attack is launched against our network or gateway, the firewall has to detect this and alarm the network and/or security administrators that they can take adequate actions. This alarming is typically an e-mail message and a message to the administrator's pager.

As the last item in this section, I want to address a more sensitive point. We can also determine if the reaction from the administrators is according to the intrusion detection policy. First of all, we must have such a policy. Then, the management has to decide if we really want to test this policy, as you have to launch the probing without announcing it to the involved network and security administrators. You can probably provoke a negative "corporate feeling" if you wake them up in the middle of the night when the alarming of the firewall is going off, they race to the office and spend the next 10 hours trying to keep the (so believed) hacker out of the system and trying to reconstruct what happened. They will probably not appreciate it if it was only a test. On the other hand if you announce a firewall test 24 hours before it happens, it is quite probable that they run around, applying patches and stuffing holes that they know to exist. The decision which way to go (or finding something in between) has to be made by the management.

8. Methodology

The methodology that I will discuss in the next section on how to perform firewall penetration testing is not THE methodology in this field. As this is a quite new field, a lot of approaches are valid and I leave it to the reader to come up with something different. For me, this methodology worked best from the few I found existing. For a different approach see for example [7].

The firewall testing is structured in the following four steps:

- ◆ Indirect information collection
- ◆ Direct information collection
- ◆ Attack from the outside
- ◆ Attack from the inside

In the following sections, I will describe each of these steps more thoroughly.

8.1 Indirect information collection

First, we want to find out as much as possible about our target. To do so, we collect information first in a way that can not be detected by any logging or alarming system. For this step, we use publicly available information from sources outside the network. These are services like *nslookup* or *whois* to get an idea about the structure of the targeted network. This should principally not reveal inside information on the network but sometimes topologies are exported although this should not be the case. Another source of information is to search on the internet. We can also access the targets anonymous FTP and WWW servers if they are available and hope we find something there (try to list directories on WWW servers if this is not disabled). We also search newsgroups for postings made by employees of the target. If mail headers are not rewritten, we

may find more specific sender addresses than just the general e-mail address. This would be for example *j.doe@mars.planet.solar.com* instead of *j.doe@solar.com*. We also search for peoples e-mail address in white page databases for the same reason.

The above approaches are quite simple but can give us a first idea of the target and its security implementation. By far the best resources for this type of data gathering are mailing lists. Here, I would like to point out specially to the firewall mailing list [4].

For example, we could follow the discussion how to solve the following problem.

```
-----  
Hello everybody,  
  
I was recently testing a XXXXXX firewall 1 (release-2) and discovered  
the following:  
  
With setting up the firewall's filter rules, I block all inbound  
traffic. This includes ICMP packets. However, when I am pinging the  
firewall with ping -l XXXXX, the firewall -1's GUI seems somehow to  
crash and so do all the filter rules.  
  
During the time that I did the pinging, I was able to telnet to the  
firewall and with this to bypass the filter rules that seem to have  
crashed with the GUI. The filter rules and the GUI came back to life  
several minutes later but too late.  
  
As I did more tests, it seems that there is also a logging problem.  
The firewall was so busy logging the ping, it forgot to do anything  
else.  
  
-----
```

*Comment: While the text describes an actual posting, I changed
the structure of it so don't waste your time using a search
engine to find out who did the posting and crack this firewall.
The problem has been fixed in the mean time.*

If the guys from *evil.com* are also reading this newsgroup, this site is easy pray if the administrator did not take the right precautions before making this posting. Mails like this help also in our testing task as we could have the same configuration to test or maybe the origin of this mail is even our targeted network.

8.2 Direct information collection

This step can sometimes not easily be separated from the last. The main difference would be that we are now approaching the targeted network and that most of our steps should be detected. We would typically begin with looking for additional information that the company's name server could have been stored on the network topology. Valuable information can also be discovered in

bounced mail headers. If we want to get more information on vital systems in our target network, we can send an e-mail message to a non-existent user. The bounced mail header could contain valuable topographic information as you can see in the following example:

SMTP <jdoe@solar.com>

Please reply to **Postmaster@mars.planet.solar.com**

if you feel this message to be in error.

Received: from **pluto.planet.solar.com** ([128.104.9.2]) by

mars.planet.solar.com

(Netscape Mail Server v2.01) with ESMTTP id AAA165

for < **jdoe@solar.com** >; Sat, 16 Nov 1996 13:57:40 -0500

Received: from **planet.solar.com** (**root@earth.planet.solar.com**

[128.164.9.3]) by **mars.planet.solar.com** (8.7.1/8.7.1) with ESMTTP id

NAA29812 for < **jdoe@solar.com** >; Sat, 16 Nov 1996 13:56:44 -0500 (EST)

Received: from **evil.satan.com** (**lucifer@evil** [128.104.9.3]) by

planet.solar.com (8.7.1/8.7.1) with SMTP id NAA29804 for <

jdoe@solar.com >; Sat, 16 Nov 1996 13:56:40 -0500 (EST)

By sending just one e-mail message (from lucifer@evil.satan.com to jdoe@solar.com), we got in this case information on three hosts, of which one will be their main e-mail gateway.

To get other topological information, we will launch in addition a scan of the entire address space of the targeted network. To do so, we use automated tools like SATAN that perform this task for us. We can there define network classes or even just the network name which would be in this case *solar.com*. Principally, we should only be able to see the firewall or even nothing when using this approach. When additional hosts are detected by this scan, they are not protected by the firewall and therefore we can probably enter through these systems.

While we used network scanning to determine if the security parameter is intact, we use stealth scanning to determine which ports of the firewall are open and are therefore a potential point of entry. The difference between stealth scanning and the way security scanners do scanning (what we used before) is that the scanners are invoking a complete TCP session which can be detected by the targeted systems. At this point, we are still interested that as little as possible of our actions get detected and therefore we are interested in an alternative way to do scanning.

Due to a bug in most existing TCP implementations [13], we can use the following approach to see if a port is open or not

A → T	system A(ttacker) sends a FIN packet to T(arget)
A ← T	if T returns a RST message, the port is not listening
A ! T	if no message is returned to A, the port is probably listening and an attack can be launched.

The following stealth scanning approach takes advantages of a different bug in the Linux (TTL-bug) or BSD (non-zero window) UNIX implementations.

A → T system A sends an ACK packet to T
A ← T T sends a RST packet back as no connection is pending

if now the TTL (Time To Life) is low or the window is not 0, the port will be probably listening.

Most of the activities that I described in this section should be detected from the targeted systems. However, the information collections should still be on such a level that the alarming of the firewall was not set off.

8.3 Attack from the outside

This section now discusses direct attacks from the outside. Our targets are on one hand the firewall but also other designated systems like WWW server or hosts that seems to have a connection to the outside world and are not protected with a firewall.

Although if the main target is the firewall, we want to try overtaking the WWW server (or FTP server, outside mail-server) first. This can give us additional advantages if the firewall (which should principally not be the case) trust these hosts or there were also examples that the filesystem of the firewall was mounted to the WWW server for easier maintenance. To have one of these systems (lets take the WWW server for example) under our control has also its advantages when we are later (following section) trying to launch an IP spoofing attack.

Another general idea is, that we are launching our attacks from different subnets as the firewall could trust on of these addresses. Quite a small chance but worth a shot.

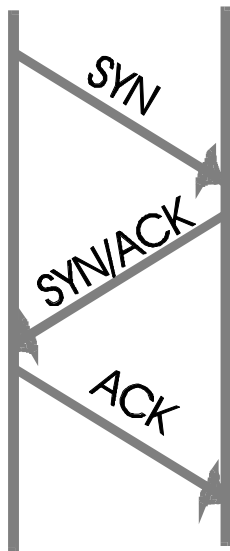
When we now begin with the testing of the firewall, we have to ask ourselves what the best approach is. This is now influenced by the design of the firewall. We have principally two different types of firewalls. The **packet filtering firewall** and the **application layer firewall (proxy)**. Both demand a different approach.

8.3.1 Packet filtering Firewalls

This firewall type uses the IP addresses to do authentication. That is the point where we base our attack. A lot of software based firewalls cannot tell from which network interface a packet is coming. Therefore they have no means to tell if a packet is really coming from our side of the network or is just pretending to be by having an IP address that belongs to this network

This lack of verification can be exploited by IP spoofing attacks. I will describe a blind and a non blind IP spoofing attack more in detail but before this, we have to have deeper knowledge on how a TCP connection is established.

8.3.1.1 TCP Connection Establishment



A TCP connection is established by a connection request from Host A to Host B. This connection request bears its own Initial Sequence Number (ISN) from Host A.

Host B will then acknowledge the ISN from Host A with the ACK flag and sends its own ISN to Host B with the SYN flag enabled.

As for the last handshake in this connection establishment phase, Host A acknowledges back the ISN that Host B sent in the previous message.

We want to know in addition how this Initial Sequence Number is created. Principally, it is a 32 bit counter. This counter is then advanced every second by 128,000 units. In addition, every TCP connection advances this counter by an additional 64,000 units.

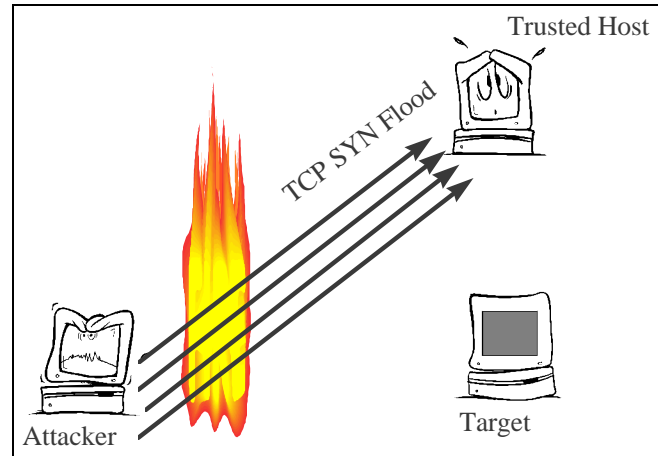
This predictability of the initial sequence number makes it possible to launch IP spoofing attacks.

8.3.1.2 blind IP Spoofing attack

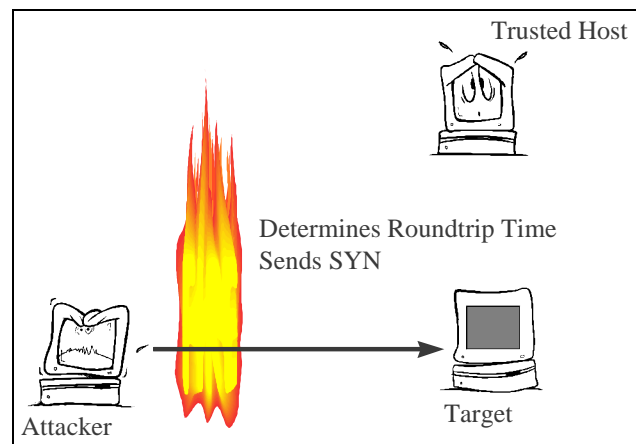
For this attack, the attacker is located outside of the firewall and takes advantage of the before discussed circumstance that the firewall cannot determine from what side an IP packet is coming.

The Attacker wants to establish a TCP connection with the target host. This TCP connection is typically a *r** service (mostly a *rlogin*) that uses the IP address as authentication and no password is needed. Therefore, we need a pattern of trust. The host that the attacker pretends to be must be trusted by the target. To determine this, we can use tools as *rpcinfo* or *showmount -e* (the latter one shows mounted filesystems and indicates therefore a trusted relationship). Another method would be brute force guessing of neighbors of the target host.

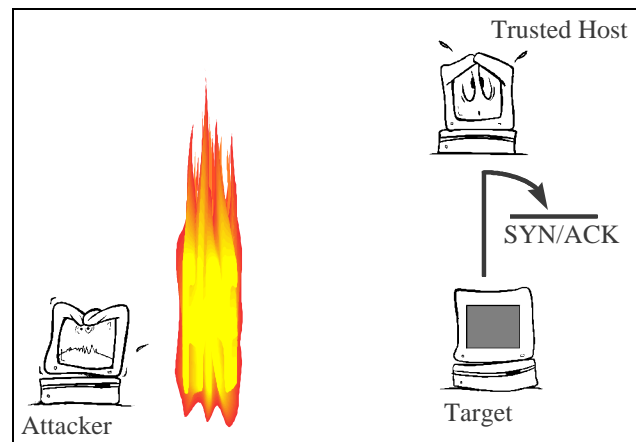
The first step in the IP Spoofing attack is to disable the trusted host. For this, we can typically use a TCP Syn Flood attack [3]. The trusted host has to be disabled so that the targeted host thinks that the traffic that the attacker will generate emanates from there. If the trusted host is still reachable, it will interfere in our attack and send RST packets to indicate that it did not start a connection and the target will ignore further packets with our ISN.



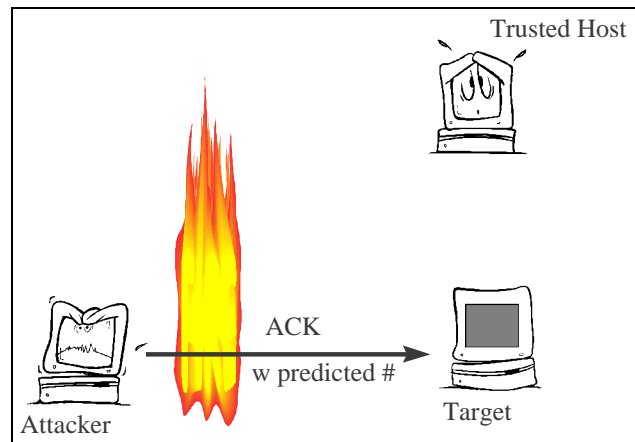
Next, we have to prepare for guessing the target's ISN. For this, we take advantage that the SMTP port (25) is often open so that mail can be passed through the firewall. Through this port we open a TCP connection to the target host and get back his initial sequence number. We are repeating this several times to get an idea on the roundtrip time so that we can guess the target's ISN more precisely. After doing so, we are sending a connection request (SYN) to the target where we are using the IP address of the disabled trusted host.



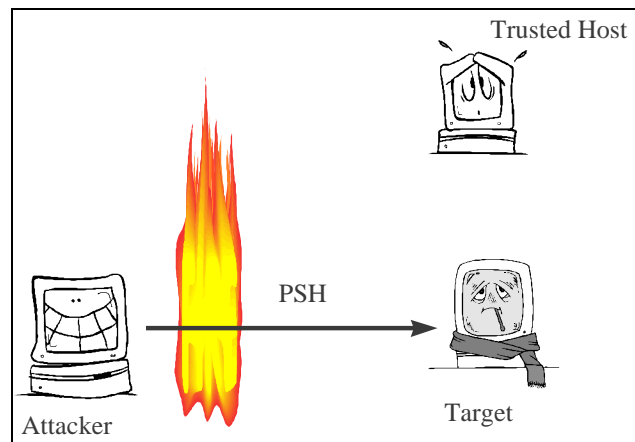
The target sends now its response with its ISN to the trusted host as this appears to be the sender. This host is still busy waiting that the connections that we opened with our SYN flood are continued and drops all incoming TCP traffic. Therefore, the target does not get a response back from the trusted host.



The attacker sends now the acknowledgment to the dropped response and acknowledges the predicted sequence number (+1) of the target.

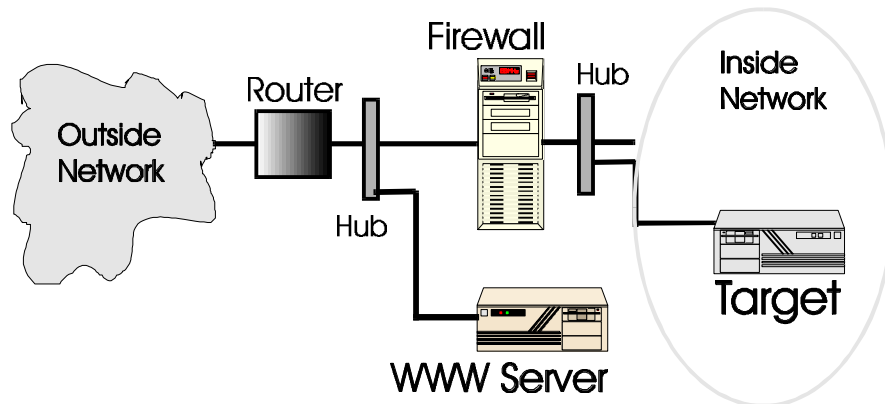


When we were correct with our guessed initial sequence number of the target then we have now a session established. If our guessed ISN is too low in relation to what the target was expecting, then the packet will be discarded. If our guessed ISN is too high, the packet will be regarded as a future one and held in the queue. We established this way a login to the target where we need no further authentication with the rights of a standard user. We will typically try if the target accepts *root* logins over the network as this would give us complete power over this system. Still, we never see an output from the target and have to guess what our commands are provoking.



8.3.1.3 non-blind IP Spoofing attack

One of the main disadvantages with the previous described IP Spoofing attack is, that we never



see the actual output of our target. We can guess in a lot of cases what this output could be but we are restricted in the use of commands. This problem can be solved in a few cases where we can actually launch a non blind IP Spoofing attack. To do so, we need a similar topology as that

described in the above picture. The firewall has to be unable to detect on which interface a packet is arriving and therefore can not tell if a packet comes from the inside network or the outside world. This may sound as a quite basic feature but a lot of today's firewall don't have this capability. In fact, a software based firewall is mostly not capable of distinguishing between the inside and the outside interface and makes the decision based upon the IP address. Our attack approach takes advantage of this. Another restriction for this attack is, that our target and the WWW server are not separated with a router (or an intelligent bridge) that filters the traffic. The attack is quite straight forward. We are attacking and overtaking the WWW server (normally enough security holes there). In addition, we have to gain root access on this machine. As the web server is freely accessible from the internet we don't have problems connecting to it. After we have gained root access, we change the servers IP address to an unused IP address from the same C class network that our target belongs to and that our target potentially trusts. To make the change active, we now reboot the server after we set up an account for us to come back. After rebooting, the server will now have the new IP address. As this address belongs to the inside network, all traffic to and from this host can now freely pass the firewall. We are still able to connect for example with *telnet* to the server and launch an attack from there to the target. As the traffic will pass the firewall, we have now established a non-blind IP spoofing attack.

8.3.1.4 Source porting and source routing

Filter rules on packet filtering firewalls are typically based on source and destination port addresses. On a TCP/IP enabled host, we have 65,535 possible virtual ports. Some of these ports are assigned to specific services and a lot are for general use. While we could have the policy that we allow *FTP* services but not *telnet* services to pass our firewall, when source porting is enabled, an attacker can modify telnet to make the connection come from source port 20 (port 20 is assigned to the FTP service).

Source routing is an option in the IP protocol that allows defining how packets are routed between source and destination. If source routing is allowed, we are often able to bypass the filter

rules of the firewall. We could also probably turn a blind-IP-spoofing attack into a non-blind attack if we can define how the packets are routed.

Both options, source porting and routing, pose a threat to our network security and should not be allowed.

8.3.2 Proxy Firewalls

Proxy type firewalls require a different approach as the authentication is not based on IP addresses. Proxies are demons and don't have much code that could have bugs to exploit in them. To my knowledge, no proxy exploitation due to a programming error is known to date. Authentication is normally quite strong by using hardware tokens and one-time passwords so that also brute force attacks are not very dangerous. However, to check for default accounts and passwords or simply password guessing is still a valid tool.

However, I came up with a possible attack if a firewall is running the NTP (network time protocol) service in combination with the one-time password SecurID. SecurID uses hardware tokens in combination with passwords to authenticate users. The one-time password is time based. The user enters an 8 digit code before the assigned password. This 8 digit code changes every 60 second. Normally, this prevents from sniffer attacks where passwords are collected. If the attacker could install a sniffer, he could store the login sequence with a timestamp. If the hacker can launch his attack from a high level (penetrated the targets service provider first), he can try to shut down the firewall (e.g. ping with oversized packets, flooding...). When the firewall comes up again, it will ask the current time through NTP. The attacker can now forge the NTP responses and set the time of the firewall back. When the attacker now replays the SecurID login sequence at the right time, he should be able to gain access our network. I never heard that this attack was described or used but it is still a valid threat and shows us that critical hosts should not rely on any information (in this case the time) from outside our trusted network.

Most of the tests that we run when testing a proxy firewall will be tests for wrong configurations or poor implementation of the security policy. SOCKs can serve here as an example. SOCKs is a library for proxy application firewalls that was designed to allow certain services to pass the firewall. Unfortunately, SOCKs is often misconfigured in a way that the administrator established the rules to allow certain services through the firewall but forgot to define the rules necessary for denying access to intruders. The firewall will seem to work fine until a hacker finds this misconfiguration or it is discovered in a firewall test.

Another test will be if we can connect to the network without going through the firewall (additional connections).

The following list covers the most dangerous threats that a firewall should block. It is not complete at all and a longer list can be accessed at <http://www.iss.net/vd/>, the vulnerability database of ISS (Internet Security Systems)[15].

- SOCKs misconfigured
- bad policy (or poorly implemented)
- brute force attacks
- anonymous FTP allowed
- tftp allowed
- r-commands allowed
- X-Windows / Open Windows allowed
- Portmapper
- NFS world mountable
- Win95/NT file sharing allowed
- Open ports (hint, check Real Audio port 7070)

A method that can possibly lead to success is the combination of attacks or trying to keep the firewall busy (e.g. ping it with large packets) during an attack. The firewall can be so busy handling the ping requests that it “forgets” its filtering rules.

However, while being creative is a good thing - we shall never forget that we are probably not allowed to shut down the firewall. Therefore our probing should not lead to denial of service attacks. If possible and during a very specified time window of the testing, we can also try if denial of service attacks can force the firewall into a failure condition. After a denial of service attack, we are interested in the state of the firewall. There are four potential states:

- ◆ Survived the attack and continues to operate normally
- ◆ Shut down and restart - reset to normal operation
- ◆ Shut down and stopping all traffic
- ◆ Shut down and allowing all traffic through

While we would like to see the firewall in the first state after a denial of service attack, the next two can be still acceptable, while we definitely want to avoid the latest one.

8.4 Attack from the inside of the network

While our previous approaches were concentrated on attacks from the outside, we are trying to attack the firewall from the inside in our last part of the firewall penetration testing.

Why should anyone try to attack a firewall from the inside- you might ask. There may be different reasons. With additional security (what a firewall provides) access has been made more difficult. Before the firewall was installed, an employee could telnet into its computer during non office hours and check the mail or send his newest papers to a friend using FTP. Even worse, access to the internet (surfing) can be disabled (or audited) for a group of users. There are a lot of reasons

that someone is not too happy with this firewall and he (or she) can try to disable it to gain back the old capabilities.

In this case, the target is the firewall or probably more often the operating system on which the firewall is running. This belongs now not any more to network security but more to host security as this specific host (the firewall) is attacked. There are a few quite good tools that you (and anyone else!) can use to test for OS vulnerabilities (SATAN, ISS.....).

9. Testing techniques/tools

Until now, I have described the methodology on firewall testing and how we are doing it in theory but I have not discussed how to perform an actual test.

There are four approaches:

- ◆ Manual probing
- ◆ Interactive testing
- ◆ Security Scanners
- ◆ Hacking tools

While manual probing and interactive testing are related to each other, the latter two approaches are more or less automated. With manual tests, you have the big advantage that you really know what is going on at the target system. You see the output/response that your probing and testing creates and you can react accordingly. This is much more flexible than using automated tools or hacking scripts. The danger of disrupting services is also not as great as with the other approaches. Unfortunately, this takes a lot of time and needs a considerable degree of knowledge. You are sitting behind your terminal equipped with a checklist of vulnerabilities that you want to test and you certainly spend a lot of time typing and searching for possible entry points. This method requires in addition also good knowledge of the security problems of your target system.

A possibility to get results faster is the use of security scanners. Until now, there is not much available that target firewalls and scans for problems. Although a lot of scanners are declared as “network security tools”, they target more host security and scan mostly for operating system vulnerabilities and dangerous enabled services. While this can help when testing a firewall, its not enough. At this moment (Dec 96), the only available package that has an option for firewall testing is the firewall scanner of ISS [14]. Even if this option is available, it means that it adds a number of firewall security checks (we discussed all of these in this paper) to the Internet Security Scanner.

Hacking tools can be easily found on the internet. Just go to Alta Vista (or your favorite search engine) and do a search for “UNIX hacking” and you will get links to around 250 so called “underground” pages where self defined “hackers” make their knowledge (sometimes knowledge and a lot of noise) openly available. With a little time, you will find tools for exploiting a lot of vulnerabilities. However, the use of these tools is not without problems. Before using them, analyze the source code thoroughly until you know what the program is really doing and then test

it in a closed environment on one of your own machines before letting it free. You never will be really sure what these tools are doing so be extremely careful.

In my opinion, automated tools can make a firewall test more efficient and take care of lengthy tasks but cannot replace them. To run just ISS against a firewall and declare it as a firewall penetration test is not an option in my point of view and should not be accepted by the client.

10. Conclusions

While firewall design is quite advanced, firewall penetration testing is a quite new field and still not very widely used. In addition, it's quite hard to get valid information on how to do the testing.

However, if a penetration test is done properly by experienced people it can give us a valuable feedback on the effectiveness of a firewall. These penetration tests have to be conducted regularly as firewalls are "eroding" with time. Right after the installation of a firewall, penetration testing should not discover big security problems if the firewall is installed properly and the policy is implemented by setting up the correct filter rules. Repeated firewall testing gives us a feedback if the firewall is maintained properly and no holes are punched through that could be exploited by hackers.

While passing a firewall test gives us an indication about our network security, it can also be misleading. It does not mean that "we" are secure now! Passing a firewall test means just that the firewall defeated all of our attack approaches. Maybe a hackers can think of something else or we forgot to test for a particular bug. It is important that we are still concerned about the individual host security when we have a firewall.

One of the most important steps in firewall testing is to choose who is doing the test and the planning of the test. It is important that the test is conducted by someone who is not involved in the maintenance or setup of the firewall. Therefore, testing should be done by an independent group that we trust for integrity, experience, writing skill and technical capabilities.

Before the testing starts, we have to define our goals and get the management approval in writing. We want typically to test if the information policy is correctly implemented and no leakage exist in the perimeter as well to test for the correct working of the firewall.

Security scanners can be of help in conducting firewall testing but cannot replace manual tests. To just run a security scanner against a firewall should not be accepted by the client as a penetration test. In addition, the functionality of these security scanners is still quite limited and they are mostly host security testing tools and not really designed for testing network security.

11. References

- [1] Boran Sean, IT Security Cookbook, Draft V0.84 1996
- [2] Cheswick / Bellovin, Firewalls and Internet Security, Addison-Wesley, 1994
- [3] daemon9 / route/ infinity, IP-spoofing demystified, Phrack Magazine 48,
<http://www.fc.net/phrack/files/p48/p48-14.html>
- [4] Firewall mailing list, subscribe Firewalls@GreatCircle.COM
- [5] Garfinkel / Spafford, Practical Unix & Internet Security, O'Reilly 1996
- [6] Kaufmann / Perlman / Speciner, Network Security, Prentice Hall 1995
- [7] Marcus J. Ranum, On the topic on Firewall Testing,
<http://www.v-one.com/newpages/fwtest.html>
- [8] Marcus J. Ranum, Thinking about Firewalls,
<http://www.tis.com/docs/products/gauntlet/ThinkingFirewalls.html>
- [9] Micahel Surkan, Daemons defy hackers, PCWeek Feb 5 1996
- [10] Network Security Conference proceedings, SANS 96
- [11] Peter Stephenson (interview), Cracking an Internet Firewall,
<http://www.venida.com/file/white/infosec1.htm>
- [12] Ralph D. Sawyer, SUN PIN - Military Methods, Westview Press, 1995
- [13] Uriel Maimon, Port Scanning without the SYN flag, Phrack Magazine 49,
<http://www.fc.net/phrack/files/p49/p49-15.html>
- [14] White paper Firewall Scanner, Internet Security Systems 1996
- [15] White paper Internet Security Scanner, Internet Security Systems 1996

12. Appendix

◆ Presentation slides