Building Your Firewall Rulebase

Lance Spitzner
Last Modified: January 26, 2000

Building a solid rulebase is a critical, if not the most critical, step in implementing a successful and secure firewall. Security admins and experts all over the Internet argue what platforms and applications make the best firewalls. We compare stateful inspection tables, application based filtering, fragmentation and reassembly, etc. However, all of this is meaningless if your firewall rulebase is misconfigured. Far too often in my security audits I see $50,000 firewalls exposing organizations to great risk, all because of a misconfigured rule. That is the purpose of this paper, to help you plan, build, and maintain a solid and secure firewall rulebase. The information covered here applies to most firewalls, but I will be using Check Point FireWall-1 as an example. Regardless of what type of firewall you use, the basic concepts of rulebase design remain the same.

The Key to Success

Before we jump in, I would like to share with you the key to success, *simplicity*. The key to a secure firewall is a simple rulebase. Your organization's number one enemy is a misconfiguration. Why should the badguys try to sneak spoofed, fragmented packets past your firewall when you have accidentally left imap wide open? To keep your rulebase simple, keep it short. The more rules you have, the more likely you or someone else will make a mistake. The fewer rules your rulebase has, the easier it is to understand and maintain. By keeping your rulebase short and simple, you have won half the battle. A good rule of thumb is to have no more then 30 rules. With 30 rules, its relatively easy to understand what is going on. Between 30 and 50 rules, things become confusing, the odds grow exponentially that something will be misconfigured. Anything over 50 rules and you end up fighting a losing battle. I personally have witnessed organizations with over 300 rules in their rulebase, nobody (including myself) had a clue as to what was going on. When you start having this many rules, you need to take a serious look at your overall security architecture, and not just your firewalls. The moral of the story is, the fewer rules you have, the simpler your rulebase and the less likely you are to have misconfigurations (and a more secure firewall).

Fewer rules have the added bonus of improving performance. By parsing only a small number of rules, your firewall saves CPU cycles. Most firewalls are extremely efficient, so you will see little difference. But it can never hurt.

Building Your Rulebase

So, how do we build a secure rulebase? Well, we will do just that in this paper. We will go through step by step and build a firewall rulebase. We will start with a fictional organization's security policy. Based on that policy, we will then develop a firewall rulebase. Along the way, I will cover some of the Do's and Don'ts of rulebase configuration. Also, we will cover rules that every firewall should have. Hopefully by the end of the paper you will have a better understanding of how to build a secure rulebase for your organization.

The Security Policy

Remember, your firewall (and your firewall rulebase) are nothing more then a technical implementation of your security policy. Management builds the security policy, which defines *what* is to be enforced. The firewall is a technical tool, which is *how* the policy gets enforced. So, before we start building our rulebase, we have to understand our security policy. Since this paper focuses on rulebase design, we will keep our security policy relatively simple. Fortunately for us, our organization has a simple security policy, which management has outlined as follows.

1. Internal employees are allowed unrestricted access to the Internet.
2. Provide the Internet access to the company's web server and Internet email
3. Any traffic coming into the corporate internal network must be securely authenticated and encrypted

Obviously most organization's security policies are far more complex then this. However, for the purposes of this paper, this will do. Do not be deceived, you will soon see how even this simple security policy can quickly grow in complexity.

The Security Architecture

As a security administrator, our first step is converting the security policy to security architecture. Lets now go through and convert each security policy bullet into technical implementation.

1. The first one is easy. Anything from the internal network is allowed outbound to the Internet.

2. The second security policy bullet gets tricky. We are required to set up both a web and mail server for the company. We will do this by putting them in a DMZ. A DMZ (Demilitarized Zone) is a separate network where you put systems you do not trust. Since anyone on the Internet will be accessing both our web and mail server, we cannot trust them. Also, systems in the DMZ can never initiate connections to the internal network, since they are not trusted. There are two kinds of DMZs, protected and unprotected. Protected DMZ's are a separate segment off the firewall. Unprotected DMZs are the network segment between the router and the firewall. I prefer to use protected DMZs, so that is where we will place both our mail and web server.
3. The only traffic that we have coming from the Internet to our internal network is remote administration. We have to give our system admins remote access to their systems. We will do this by allowing only the encrypted service ssh into our internal network.
4. There is one additional goody we have to add, DNS. Though not stated in our security policy, we have to provide this service. Being the secure admins we are, we will implement Split DNS. Split DNS means to split the functionality of DNS on two different servers. We will do this by using an External DNS server that the Internet will use to resolve our company's domain information, and an Internal DNS server that our internal users will use. The External DNS server will be placed on the protected DMZ with the mail and web server. The Internal DNS server will be placed on the Internal network. This protects our Internal DNS server from being compromised by the Internet. The Internal DNS server has information which could be used to map our internal network.

This is what our firewall rulebase has to enforce. Looks simple? It isn't. We will end up using 11 rules for this security policy. Lets see how.

Rule Order.

Before we begin building our rulebase, there is one thing I want to cover, rule order. As you will soon learn, which rules go in what order are critical. Having the same rules, but placing them in a different order, can radically alter how your firewall works. Many firewalls (such as SunScreen EFS, Cisco IOS, and FW-1) work by inspecting packets in a sequential manner. When your firewall receives a packet, it compares it against the first rule, then the second, then the third, etc. When it finds a rule that matches, it stops checking and applies that rule. If the packet goes through each rule without finding a match, then that packet is denied. It is critical to understand that the first rule that matches is applied to the packet, not the rule that best matches. Based on this, I like to keep the more specific rules first, the more general rules last. This prevents a general rule being matched before hitting a more specific rule. This helps protect your firewall from misconfigurations. To learn more about how a firewall state table works, check out Understanding the FW-1 State Table.

The Rulebase

Time to get to the good stuff, building the rulebase! Below I have briefly outlined each rule, why I selected that rule, and the importance it has. To go into more detail on the rule (including a picture of the rulebase), follow the links.

1. **Default Properties**: The first step is to eliminate anything that is allowed. We want to be sure we are starting with a clean slate and no packets are getting through. Unfortunately, CheckPoint FireWall-1 comes with a variety of services wide open, by default. Our first step is to turn off these default properties.
2. **Internal Outbound:** Our first rule is to allow anyone on the internal network outbound. As stated in the security policy, all services are allowed.
3. **Lockdown:** Now we add our lockdown rule, blocking any access to the firewall. This is a standard rule that every rulebase should have. No one should have access to the firewall but the firewall admins.
4. **Admin Access:** No one can connect to the firewall, including the admins. We have to create a rule allowing administrative access to the firewalls.
5. **Drop All:** By default, FW-1 drops all packets that do not match any rules. However, these packets are not logged. We change this by creating a Drop All AND Log rule and add it to the end of the rulebase. This is a standard rule that every rulebase should have.
6. **No Logging:** Often there is a great deal of broadcast traffic on the network that the firewall drops and logs, which can quickly fill up the logs. We create a rule that drops/rejects this traffic, but does NOT log it. This is a standard rule base that you may want to use.
7. **DNS Access:** We want to give Internet users DNS access to our DNS server..
8. **Mail Access:** We want to give Internet and internal users smtp access to our mail server.
9. **Web access:** We want to give Internet and internal users http access to our web server.
10. **Block DMZ:** Our internal users have wide open access to our DMZ, which we need to stop.
11. **Internal POP Access:** Give our internal users POP access to the mail server.
12. **Sneaky Rule:** Your DMZ should never initiate traffic to your internal network. If your DMZ does, this may mean your DMZ was compromised. I like to add a rule that denies, logs, and alerts me whenever there is any traffic from the DMZ to my internal network.
13. **Admin Access:** We give our admins (limited to specific source IPs) encrypted access to the internal network.

14. **Performance**: Last, we review our rulebase for performance. When possible, move your most commonly used rules towards the top of the rulebase. This improves performance since the firewall parses fewer rules.
15. **IDS**: For those of you who would like some basic scan detection, this will help.
16. **Additional Rules** There are additional rules you can add. Examples include:
    -> Block any connections from doubleclick.com, one of the many Internet advertisers. This should help stop the on-line ads you have to wade through on the net. Saves your users time and improves performance
    -> To block AOL ICQ connections, don't block the ports, but the destination AOL servers.

Change Control

Once you have all your rules in place, I highly recommend you update them with comments, and keep them updated. Comments help you keep track which rules do what By having a better understanding of the rules, there is less chance for misconfiguration. For larger organizations with multiple firewall admins, I recommend the following information be put into comments whenever a rule is modified. This will help you track who changed which rules, and why.

- Name of person modifying rule
- Date/time of rule change
- Reason for rule change.

Audit

Once you have finished your firewall rulebase, it is critical that you test it. We all make mistakes its the good admins who follow up and find them. To learn more about testing your firewall rulebase, check out Auditing Your Firewall Setup.

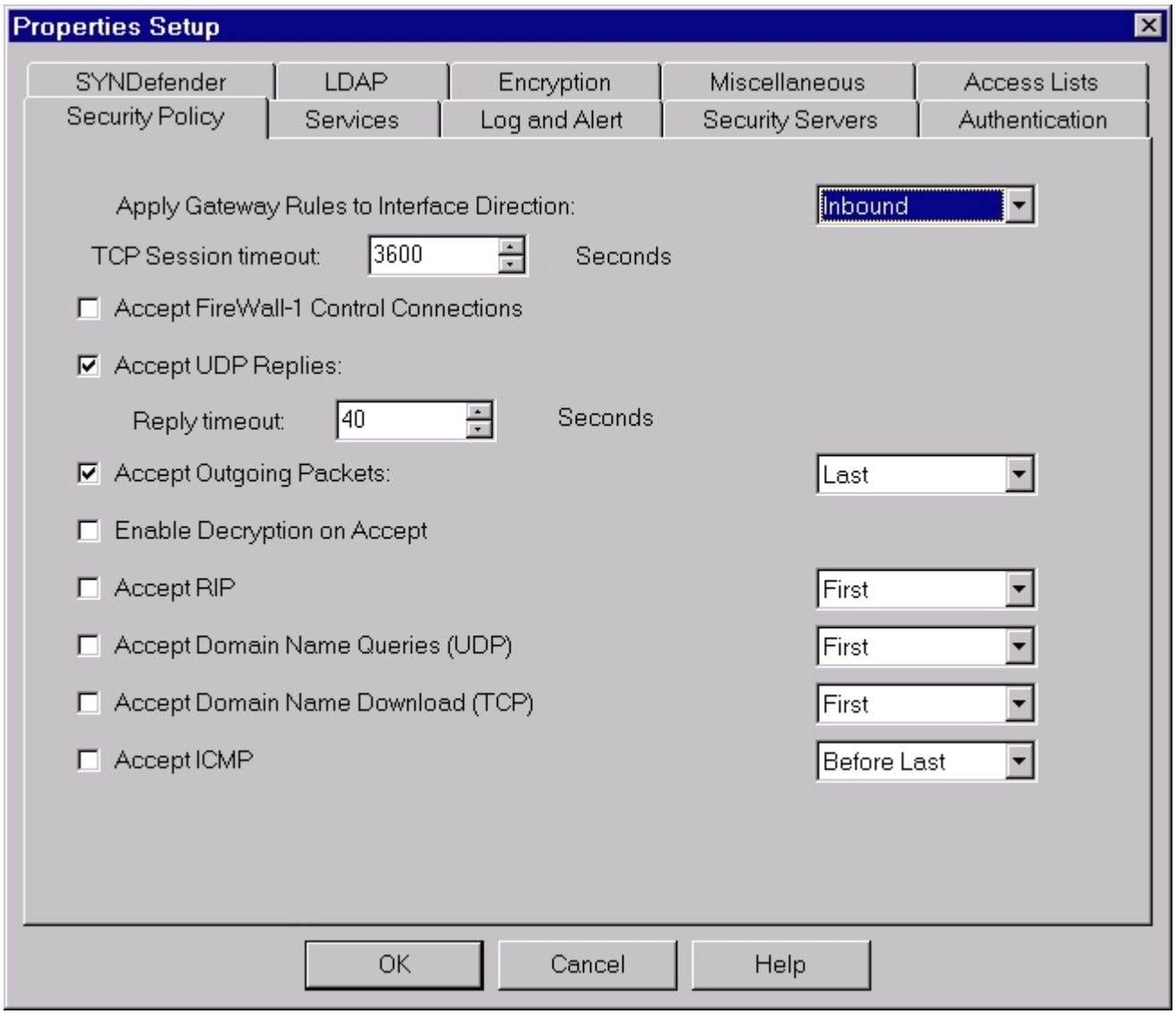
Conclusion

A firewall is only as good as it's implementation. In today's dynamic world of Internet access, it is easy to make mistakes during the implementation process. By building a solid and simple rulebase, you create a more secure firewall.

Author's bio
Lance Spitzner enjoys learning by blowing up his Unix systems at home. Before this, he was an *Officer in the Rapid Deployment Force,* where he blew up things of a different nature. You can reach him at *lance@spitzner.net*

Default Properties

Notice below that all default services are disabled. By default, FW-1 leaves a variety of vulnerable services open to the world. Be sure to turn them off. If you need any of these services, turn them on later in the firewall rulebase. This gives you greater control AND logging of those services (default properties do not log). Keep in mind, if you are managing more then one rulebase, any changes you make to default properties apply to all rulebases. This is another advantage to clearing default properties, you get more granular control by using the rulebases.



Internal Outbound

Okay, this is where we let everyone on the internal network unrestricted access to the Internet. This setup is not secure. You want to limit only what is absolutely required for outbound access (such as limiting outbound traffic to http and DNS queries only). If possible, also proxy all outbound traffic. I used this unrestricted rule on purpose to demonstrate a point. Some sites use this unrestricted "Internal Outbound" rule, without realizing the problems they are causing. With a rule like this, rule base ordering becomes absolutely critical. As you follow along and see how the additional rules are added, you will better understand the importance of rulebase ordering. If management allows you to limit what is allowed outbound, do it, it is far more secure. If you can't, see how a rule like this causes problems, requiring critical rulebase ordering.

Lockdown

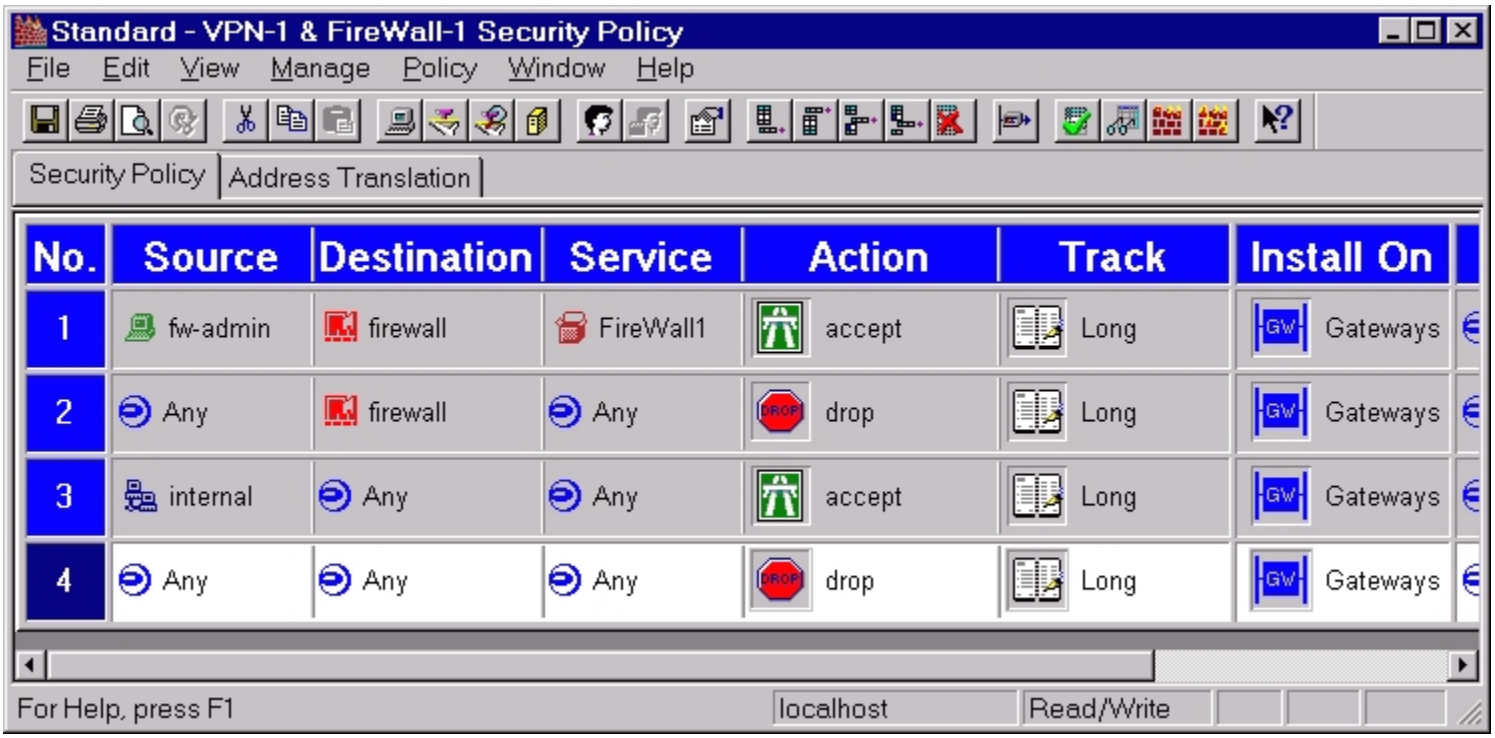
The lockdown rule protect your firewall, denying any traffic to it. This rule is critical, as this is one of the primary resources you need to protects. Some people mistakenly call this the "ghosting" or "stealth" rule, thinking it hides the firewall. There is no true way to hide your firewall, sooner or later that will be discovered. There are too many tools and techniques out there designed specifically for identifying firewalls. Notice how this rule is placed ***before*** the Internal Outbound rule. Positioning is critical. If this rule was after, then anyone on the internal network would have access to the firewall, because the Internal Outbound rule would match first. This is the first of many rules where positioning is critical.

Admin Access

Everyone is locked out of the firewall, including the admins. We will give them access, but only to the specific services they need. Fortunately, CheckPoint has these services predefined. Also, we limit what sources can access these services by specific systems. Once again, notice the rule positioning, this rule goes before the Lockdown rule. For larger organizations with several distributed firewalls, you may have to make several rules similar to this one. For example, you may have to create one rule to connect to the Management Station, and another for connecting to the Firewall modules. For this example, we are assuming that both the Management and Firewall module are on a single system. Also, many of you may want to give your admins remote access to the operating systems. In that case, you would add the service "ssh" (or some other secure remote access) to this rule.

Log Denied

By default, if any packet does not match any rule, then that packet is dropped. If the firewall does not explicitly allow the service, then it is not allowed. However, these packets are not logged by default. You definitely want to log this traffic, much of your unauthorized traffic happens here. To do that, we create a drop all and log rule, which gets placed at the end of the rulebase. This is another of those rules that all firewalls should have, if not already by default.

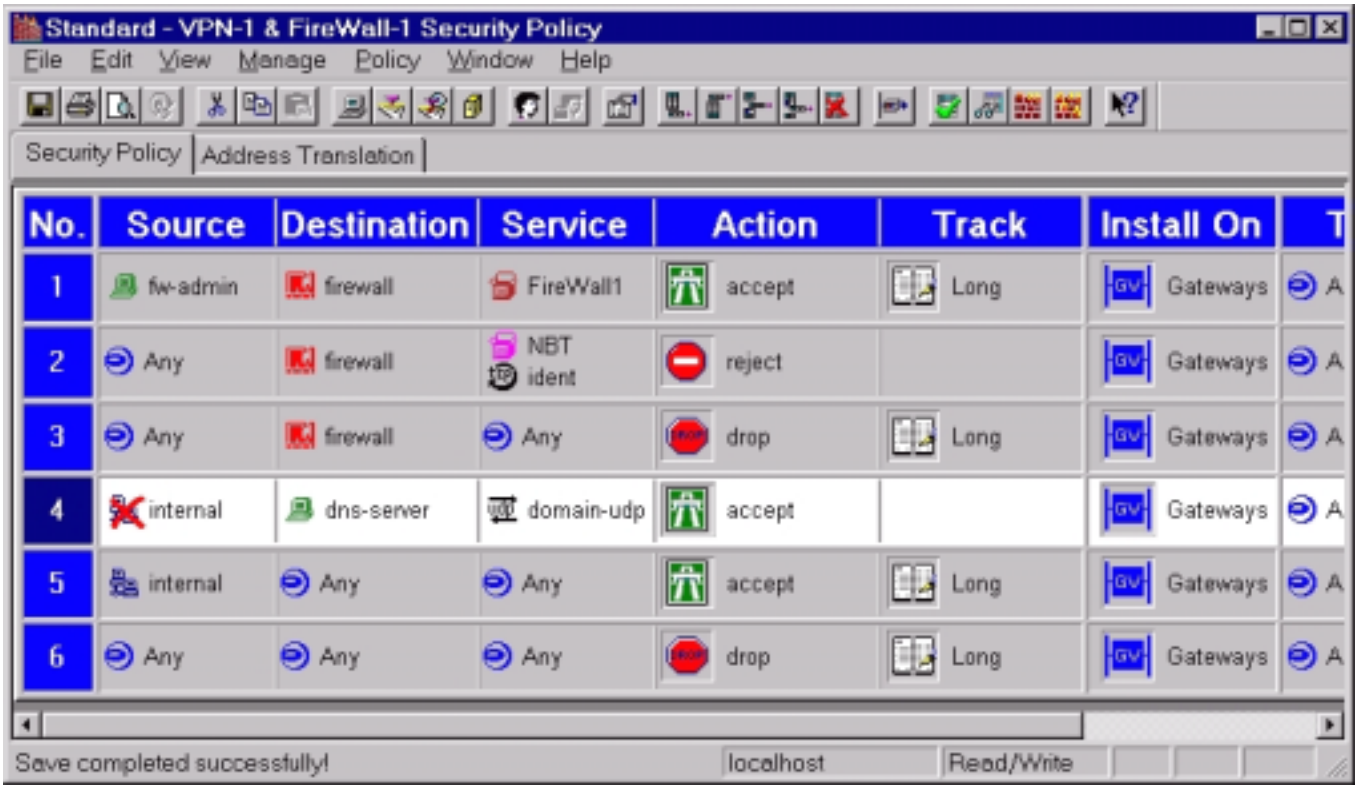


No Logging

Often your network will see a lot of broadcast traffic that is filling up your logs, especially chatty protocols such as NetBIOS. You may not want to log this traffic. Remember our last rule that drops and logs everything that is not explicitly allowed? Well, we create a rule before the Drop All rule that drops the chatty traffic, but does NOT log it. We will also add ident, an unreliable protocol used by mail servers to identify the user sending mail. Notice how we use "Reject" instead of "Drop". Reject quickly closes the connection by sending RST packets. This helps increase the response time for mail, since the ident protocol gets a "RST" instead of timing out. For NetBIOS, it does not matter. Keep in mind, if you are not logging the traffic, this will make it more difficult to troubleshoot if you have problems in the future. You may have to temporarily disable the rule to troubleshoot specific NetBIOS or Ident issues.

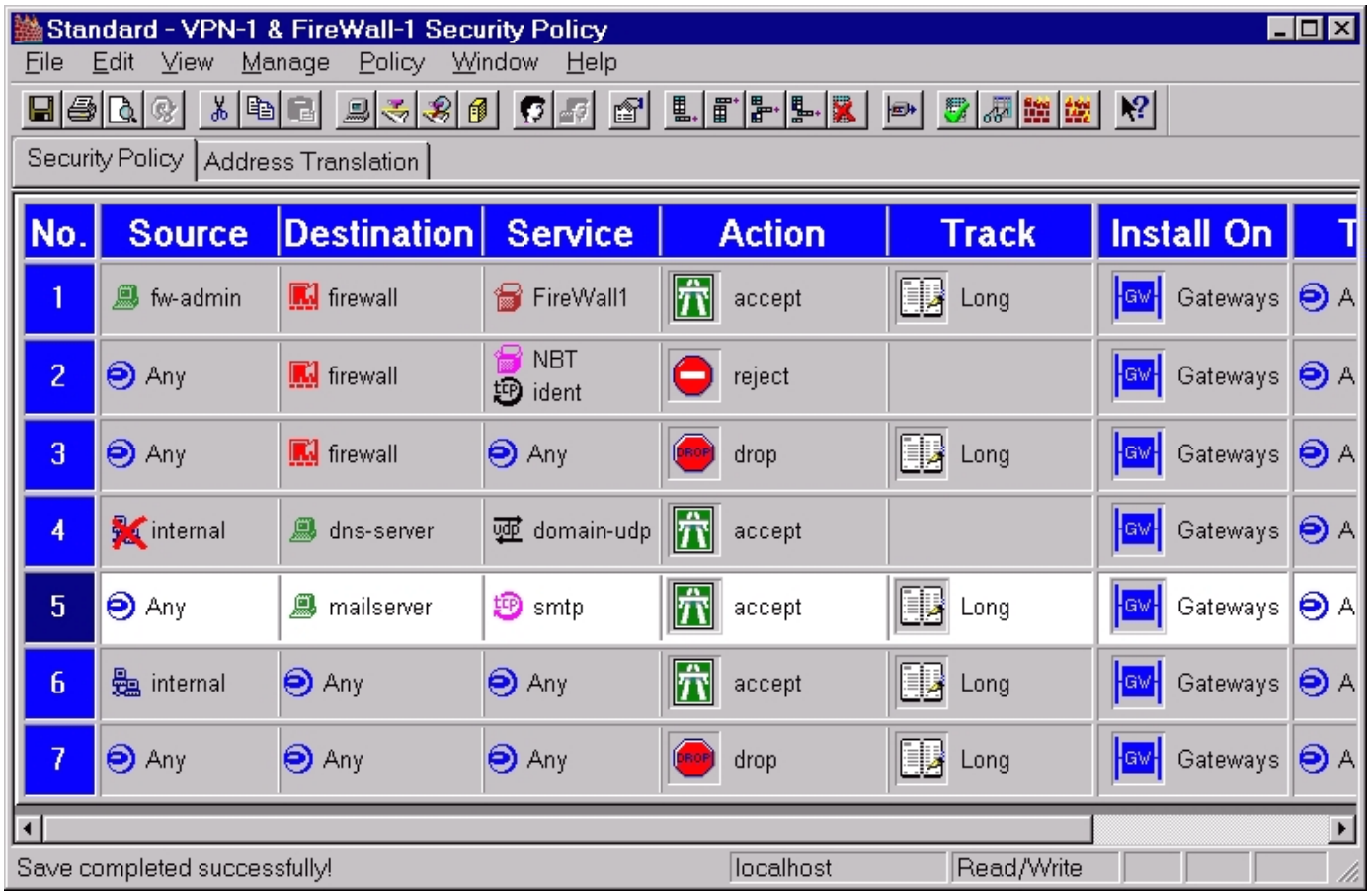
DNS Access

Here we give the Internet DNS access (strictly 53/UDP) to our Name server. Notice how the source is everything ***but*** the internal network (use of negation). We do not want our internal network using this DNS server, as they will be using the internal DNS server. Notice how this rule goes ***before*** the Internal Outbound rule. Also, notice how we chose not to log this traffic. Logging these sessions will quickly fill up your firewall logs, while providing little information (in my opinion). You may or may not want to log these sessions, that is up to you.



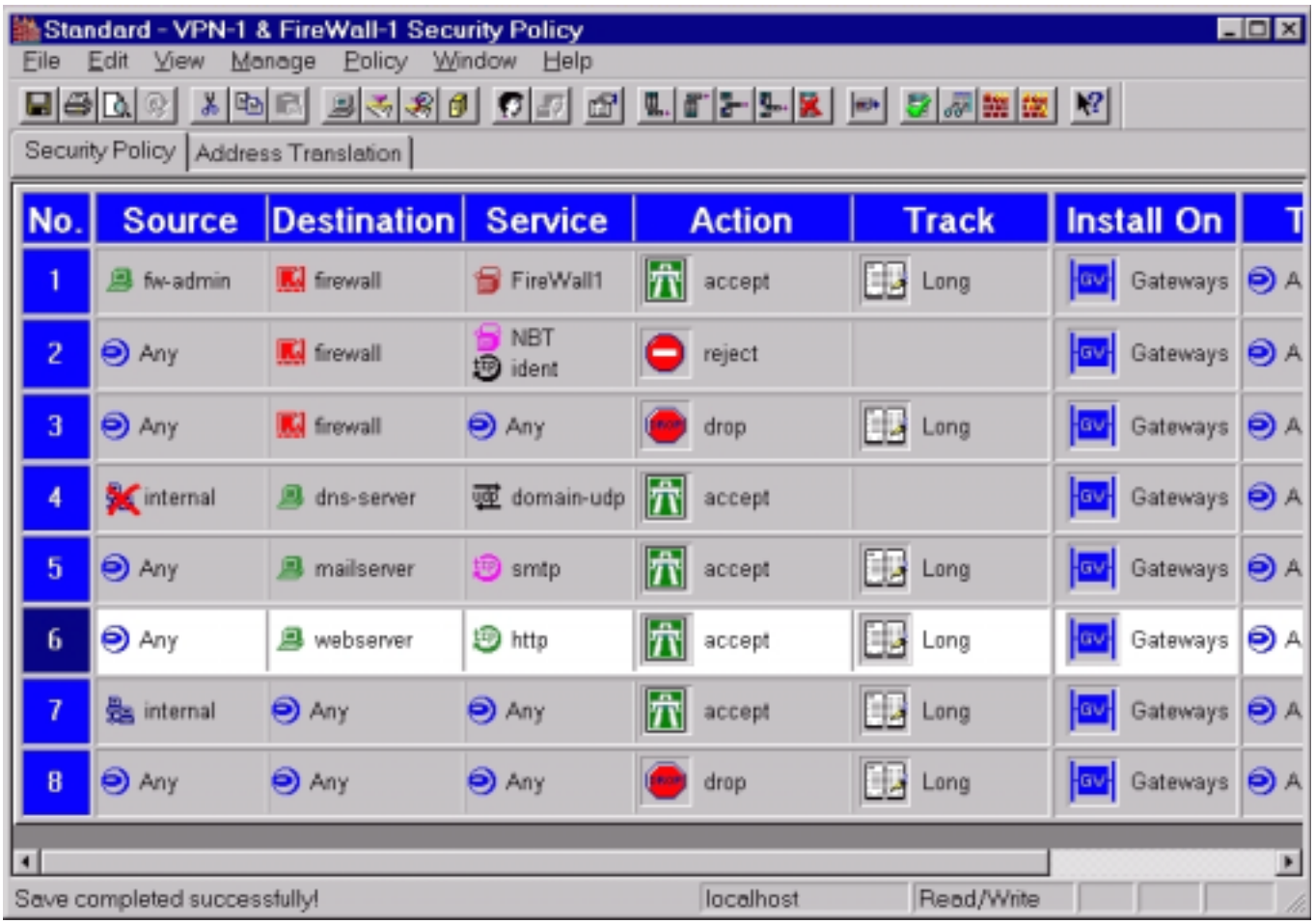
Mail Access

Here we give everyone SMTP access to the mail server. This is required for our internal users to be able to send mail, and Internet to send us mail. Since we are good security admins, we have turned of mail-relay on the mail server.



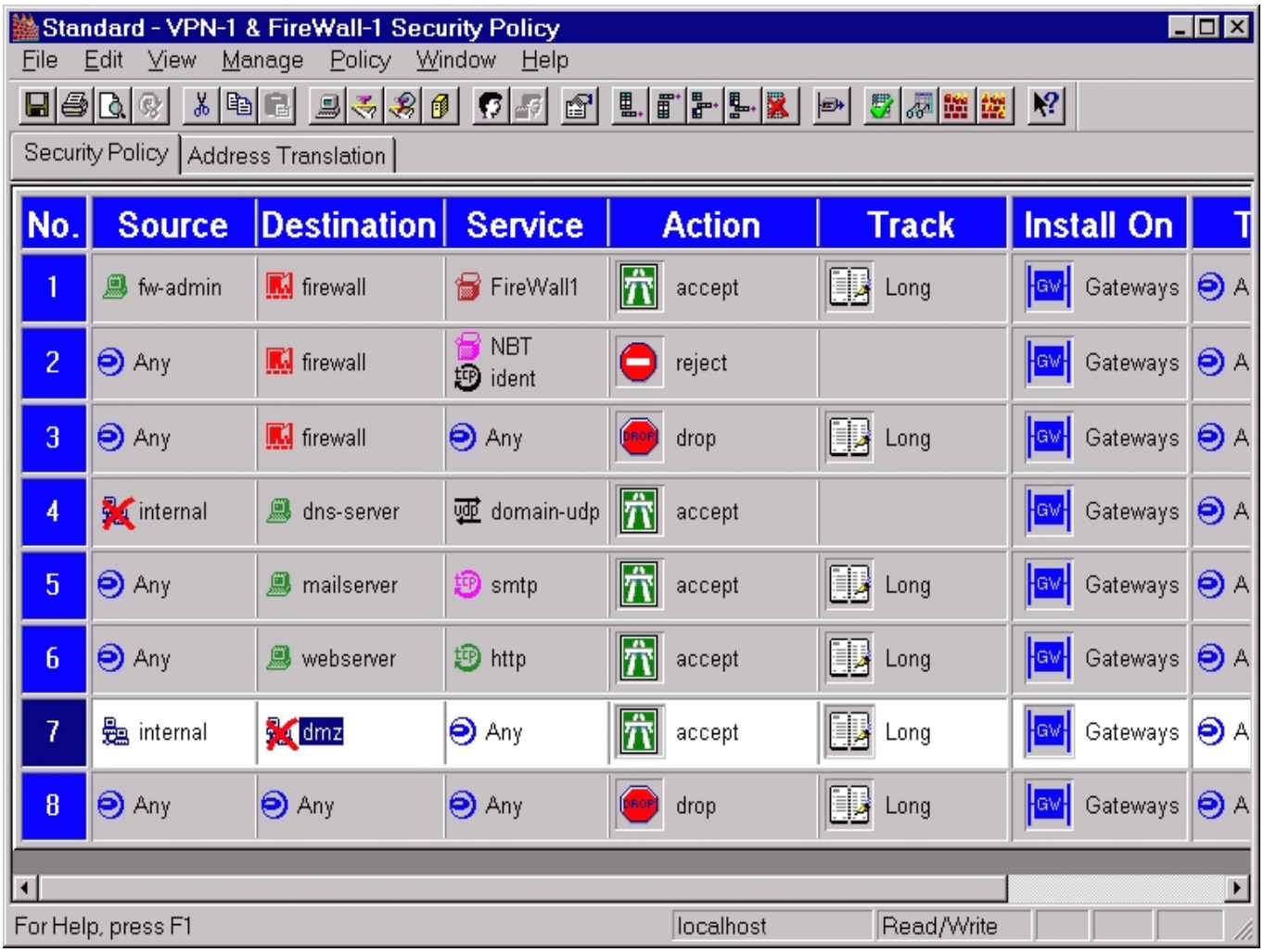
Web Access

Just like SMTP, we want to give everyone (both Internet and internal) http access to our web server. If we were to block this, marketing and sales would have our heads.



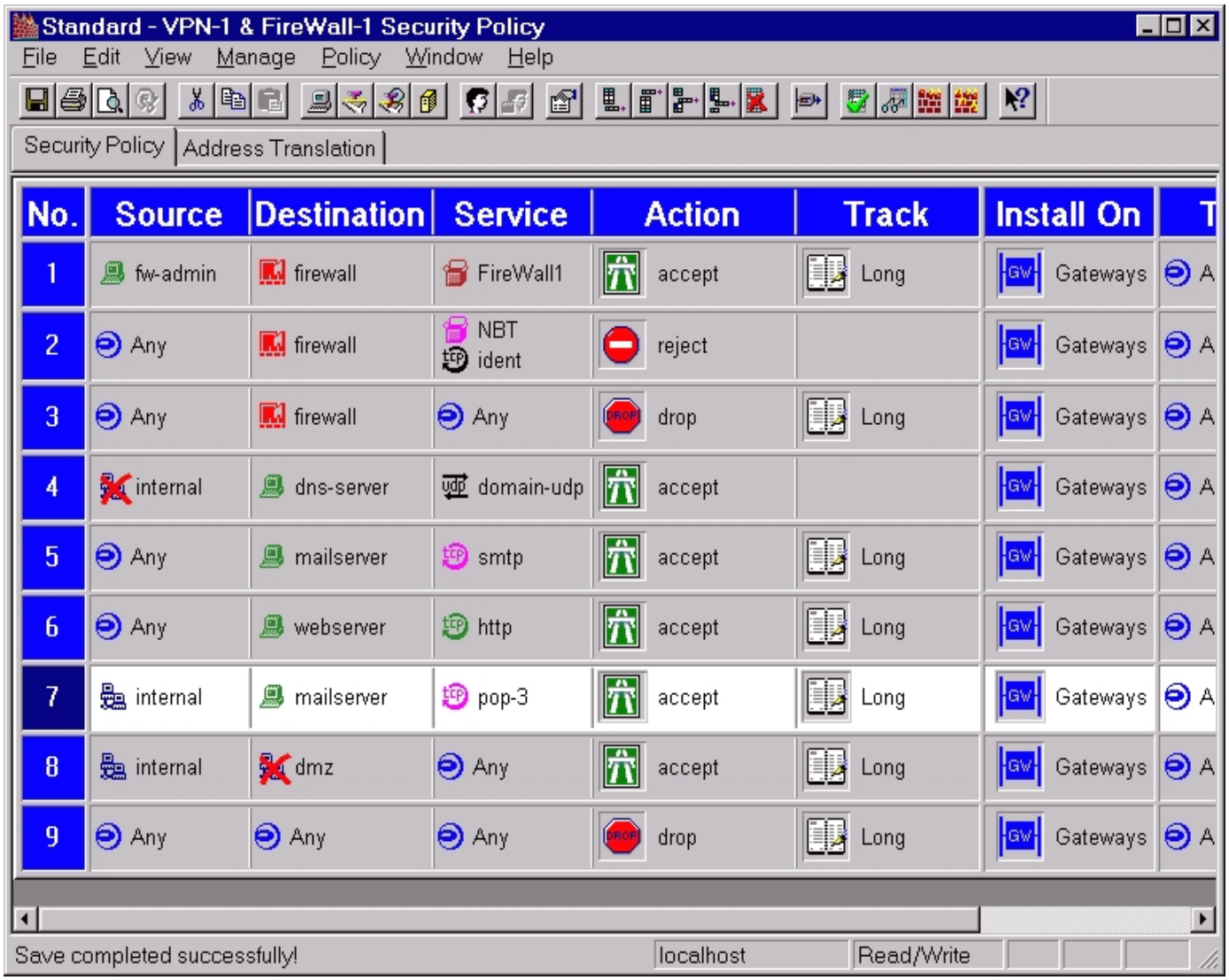
Block DMZ

Right now all internal users have wide open access to the DMZ. This is not good, as the entire idea behind the DMZ is it is an isolated, untrusted network. We don't want our internal users accidentally bringing in something from the DMZ. So, we block our internal users having any other access to the DMZ. Now, instead of creating another rule, we just change the Internal Outbound rule to say Internal can go anywhere but the DMZ. This saves us from creating another rule. Remember, simplicity is good. If the use of negation confuses you, then create an additional rule that denies the Internal network access to the DMZ and place it before the Internal Outbound rule.



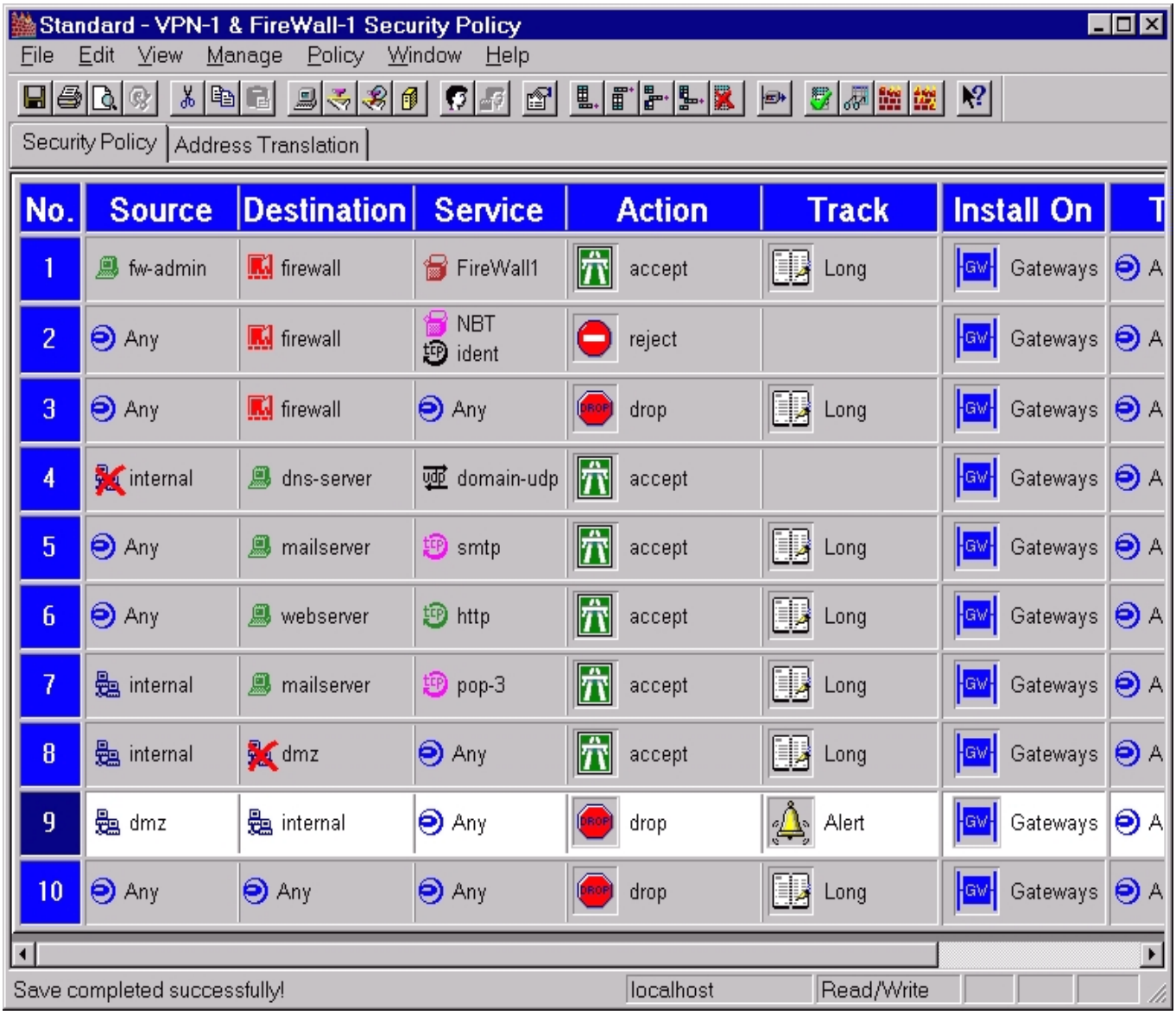
Internal POP Access

We have to give our internal users POP access to the mail server. This rule has to go before the Block DMZ rule, or our internal users would never be able to access the mail server. NOTE: A more advance setup would be to have a secondary mail server on the internal network pull the mail from the external server at specified intervals. Internal users would then get their mail from the internal mail server. However, detailing such a setup is beyond the scope of this paper (however, I may have to change the scope of this paper if I keep getting hate mail about the mail server setup pictured here :)



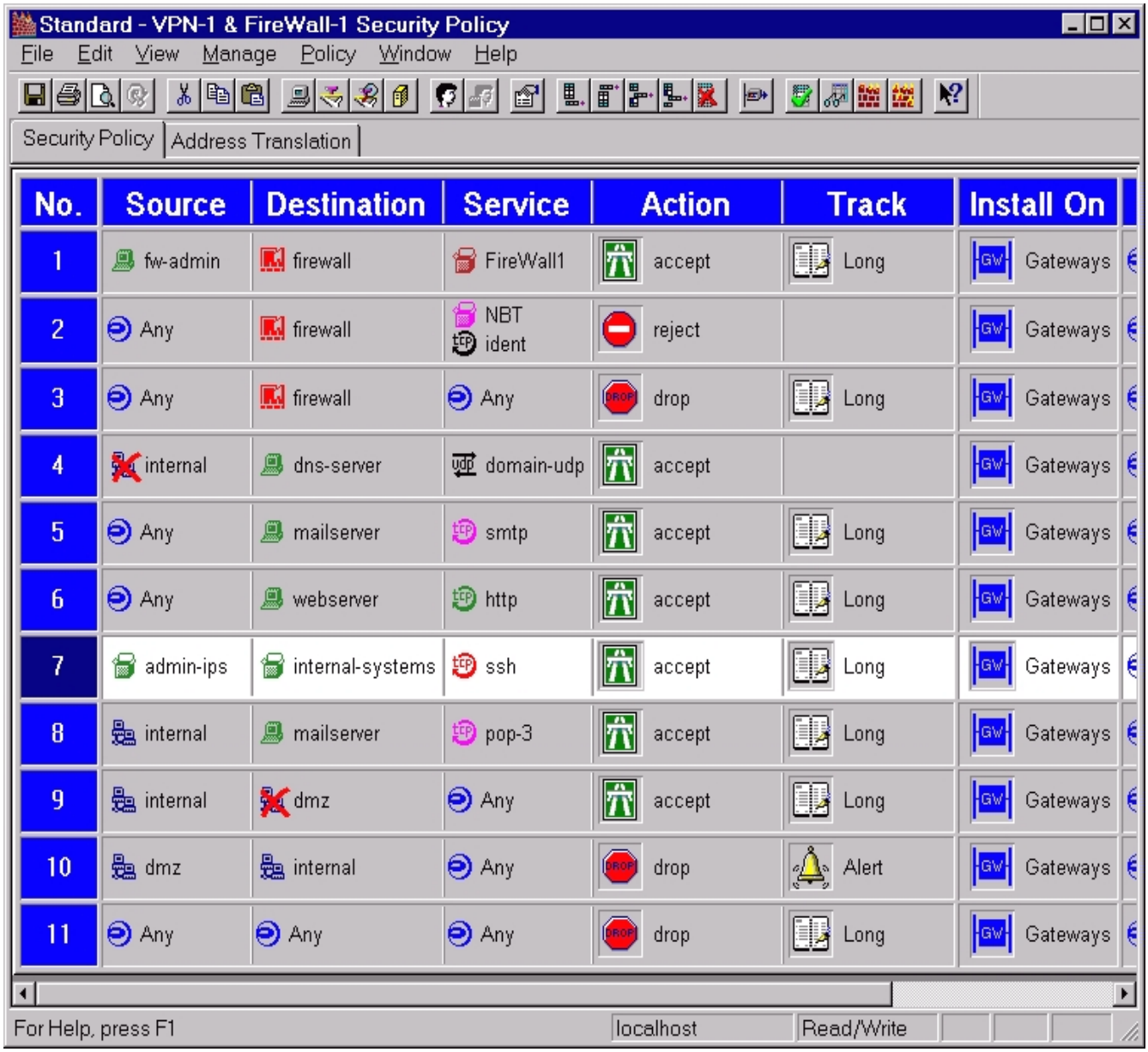
Sneaky Rule

I like this rule. The problem with logging is there is so much of it. What is important, what isn't? This rule helps simplify that. This rule looks specifically for any traffic initiated from the DMZ going to the Internal network. This should never happen, as the DMZ is an untrusted network. By creating this rule and giving it an alert, we can quickly be notified when this occurs. Something like this is one of the first indications that your DMZ may have been compromised.



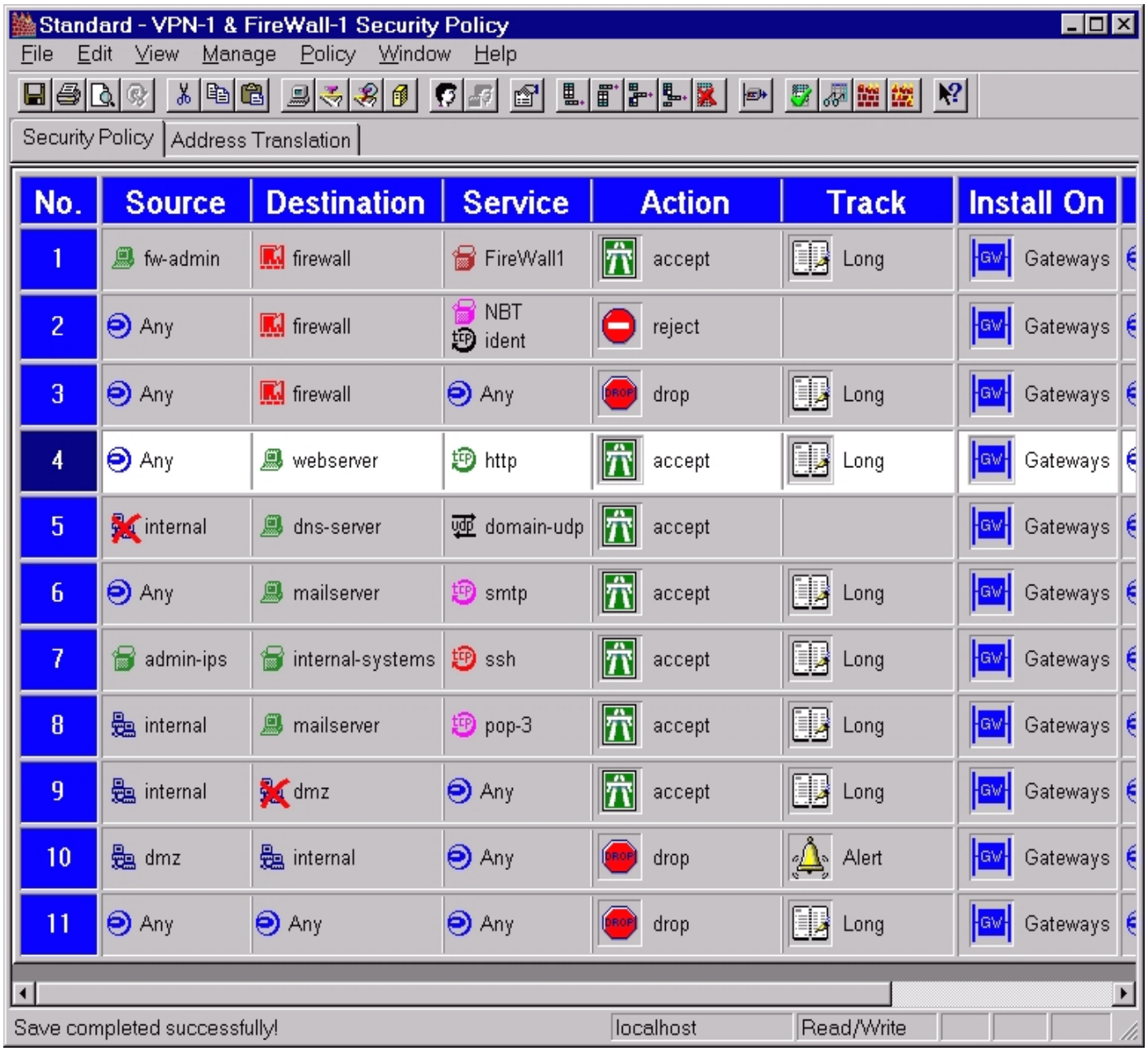
Admin Access

Don't forget our security policy. Management stated that any access from the Internet to the internal network had to be encrypted. However, the admins need remote access for those emergencies. For this, we will allow only specific IPs ssh access to specific systems in our internal network. Notice the placement of this rule, specific rules first, general rules last. A form of single sign-on authentication is being used on the hosts, as per the security policy.



Performance Tweaks

Once you have your rulebase complete, review the rule base to see if you can improve performance. Security is priority number one, but if you can improve performance while maintaining security and simplicity, go for it. The idea is to place the most commonly used rules first. This way the firewall has fewer rules to parse to get to the most commonly used rules. For many organizations, this will have little impact. However, for organizations with large rulebases, or if the majority of their traffic is a single service (such as a web server farm) then this can help. For our organization, we will say that our web server receives the most traffic. So, we move the rule as high as possible. Notice how I did NOT place this before the Firewall Lockdown rule, this is on purpose. Remember, security before performance. I never place anything before the Firewall Lockdown unless absolutely necessary. Also, notice how the rules are grouped together logically. First comes the "Firewall" rules, followed by the "DMZ" rules, followed by the "Internal" rules. Logically grouping rules helps you keep track of what is going on. Remember, keep it simple.



No.	Source	Destination	Service	Action	Track	Install On
1	fw-admin	firewall	FireWall1	accept	Long	Gateways
2	Any	firewall	NBT ident	reject		Gateways
3	Any	firewall	Any	drop	Long	Gateways
4	Any	webserver	http	accept	Long	Gateways
5	internal	dns-server	domain-udp	accept		Gateways
6	Any	mailserver	smtp	accept	Long	Gateways
7	admin-ips	internal-systems	ssh	accept	Long	Gateways
8	internal	mailserver	pop-3	accept	Long	Gateways
9	internal	dmz	Any	accept	Long	Gateways
10	dmz	internal	Any	drop	Alert	Gateways
11	Any	Any	Any	drop	Long	Gateways