A quantitative study of firewall configuration errors

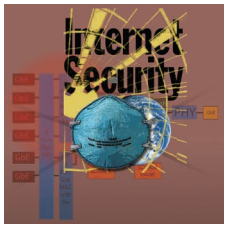
ARTICLE *in* COMPUTER · JULY 2004

Impact Factor: 1.44 · DOI: 10.1109/MC.2004.2 · Source: IEEE Xplore

1 AUTHOR:

Avishai Wool
Tel Aviv University
106 PUBLICATIONS **3,595** CITATIONS

A Quantitative Study of Firewall Configuration Errors



The protection that firewalls provide is only as good as the policy they are configured to implement. Analysis of real configuration data shows that corporate firewalls are often enforcing rule sets that violate well-established security guidelines.

Avishai Wool
Tel Aviv University

Firewalls are the cornerstone of corporate intranet security. Once a company acquires a firewall, a systems administrator must configure and manage it according to a security policy that meets the company's needs. Configuration is a crucial task, probably the most important factor in the security a firewall provides.[1]

Network security experts generally consider corporate firewalls to be poorly configured, as witnessed in professionally oriented mailing lists such as Firewall Wizards (http://honor.icsalabs.com/mailman/listinfo/firewall-wizards). This assessment is indirectly affirmed by the success of recent worms and viruses like Blaster[2] and Sapphire,[3] which a well-configured firewall could easily have blocked. However, no quantified studies directly confirm the extent of the problem because corporate firewall policy configuration files, or *rule sets*, are highly sensitive and therefore closely guarded.

For the past four years, I have been leading the development of the Firewall Analyzer software (www.algosec.com), which evolved from a Bell Labs project[4] into a commercial product.[5] During this time, I have had the opportunity to analyze rule sets obtained from a variety of corporations. In this article, I focus on rule sets for Check Point's FireWall-1 product (www.checkpoint.com) and, specifically, on 12 possible misconfigurations that would allow access beyond a typical corporation's network security policy. By documenting the frequency of misconfigurations in actual firewall data, I was able to check whether the configuration quality is correlated with other factors—specifically, the operating system on which the firewall runs, the firewall's software version, and a new measure of rule-set complexity.

DATA COLLECTION

Between 2000 and 2001, a total of 37 Check Point FireWall-1 rule sets were collected from organizations in the telecommunications, financial, energy, media, automotive, and healthcare market segments as well as from research labs, academic institutions, and network security consulting firms.

Table 1 lists some basic statistics of these rule sets: number of rules in the set, number of network objects defined in the database that supports the rules, and number of network interface cards on the firewall.

Table 2 lists the distribution for the three operating systems running the firewalls—specifically, Sun Solaris, Nokia IPSO, and Microsoft Windows NT, and Table 3 shows the distribution for the software versions that Check Point's product went through during the time the rule sets were collected. The version is relevant to this discussion because Check Point introduced several changes to default configuration settings in version 4.1 that should have helped eliminate a few common configuration errors.

Before we draw any conclusions based on this data, we need to bear in mind some caveats that affect the significance of the findings. First, 37 rule sets form a very small sample—the number of installed Check Point firewalls is estimated to be hundreds of thousands.

Published by the IEEE Computer Society

Furthermore, these rule sets are not a random sample. They came from organizations willing to pay for an audit of their firewall rule set by an external company. This could have biased the sample toward badly configured firewalls.

On the other hand, obtaining any number of real firewall rule sets from operational firewalls is rare. In fact, I am not aware of any previously published quantitative study of this type, so the data itself constitutes a contribution to research in this area.

RULE-SET COMPLEXITY

Firewall administrators can intuitively classify a rule set as "complicated" or "simple." I wanted to quantify this intuition into a concrete measure of complexity. The raw number of rules is an obvious parameter to consider in defining a measure of rule-set complexity. However, this number by itself is insufficient for two reasons.

First, a single Check Point FireWall-1 rule can list multiple source, destination, and service objects. Thus, evaluating the "real" number of rules would require counting the cross products of all these object types—a tedious calculation, which would lead to a somewhat unintuitive measure. Instead, I chose a simpler method: Add the total number of database objects that support the rule set to the number of rules.

Second, a Check Point FireWall-1 rule applies simultaneously to all traffic crossing the firewall from any interface to any other interface. The number of possible interface-to-interface paths through the firewall increases quadratically with the number of interfaces, complicating the administrator's task. Precisely, if the firewall has i interfaces, the number of different interface-to-interface paths through the firewall is $i(i-1)/2$. To take this additional complexity into account, I chose to add this last quantity to the measure.

Thus we obtain the following simple, intuitive measure of rule-set complexity:

$$RC = \text{Rules} + \text{Objects} + \text{Interfaces}(\text{Interfaces} - 1)/2$$

where RC denotes rule complexity, Rules denotes the raw number of rules in the rule set, Objects denotes the number of network objects, and Interfaces denotes the number of interfaces on the firewall.

CONFIGURATION ERRORS

To quantify a firewall's configuration quality, we must define what constitutes a configuration error.

Table 1. Statistical properties of the collected rule sets.

Property description	Minimum	Maximum	Average
Number of rules[1]	5	2,671	144.0
Number of objects[2]	24	5,847	968.0
Number of interfaces[3]	2	13	4.1

[1] Total rules in the rule set (including rules for network address translators).
[2] Network objects (hosts, subnets, and groups of these) defined in the database supporting the rules.
[3] Network interface cards on the firewall.

Table 2. Distribution of rule sets by operating system.

Operating system	Distribution (percent)
Sun Solaris	48.7
Nokia IPSO	35.1
Microsoft Windows	16.2

Table 3. Distribution of rule sets by software version.

FireWall-1	Distribution (percent)
Version 3.0	2.7
Version 4.0	18.9
Version 4.1	73.0
Version NG	5.4

In general, the definition is subjective, since an acceptable policy for one corporation could be completely unacceptable for another. Furthermore, the data for this study does not identify which of the corporation's machines are user desktops, which are Web servers, which are file servers, and so on, which means that we don't know the policy's semantics.

To be as objective as possible, I adopted the stance of an external auditor, counting as errors only those configurations that represented violations of well-established industry practices and guidelines.[6-8] Thus, these findings offer only a rough baseline, and the protection that the surveyed firewalls offer may well be worse than the results reported here suggest.

The following 12 items counted as configuration errors for this evaluation:

1. *No stealth rule*. To protect the firewall itself from unauthorized access, it is common to have a "stealth" rule of the form: "From anywhere, to the firewall, with any service, drop." The absence of such a rule to hide the firewall counted as a configuration error.

2-4. *Check Point implicit rules*. Besides the regular user-written rules, the Check Point FireWall-1 GUI has several checkboxes that produce implicit rules. These rules control both the Internet's Domain Name System (DNS), separately over TCP and UDP, and the Internet Control Message Protocol (ICMP). However,

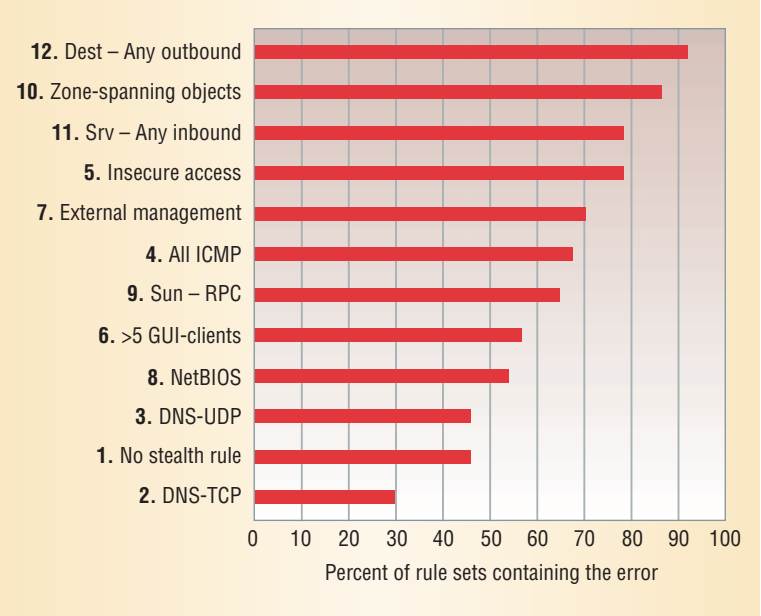


Figure 1. Distribution of configuration errors. Numbers on bar descriptions correspond to the configuration error descriptions in the text.

the implicit rules are very broad, basically allowing the service at hand from anywhere to anywhere. Since DNS is one of the most attacked services,[8] writing narrow, explicit rules for it is more secure. Likewise, with any-to-any ICMP, attackers can scan the internal net and propagate worms like Nachi/Welchia.[9] Each of the three possible implicit rules—DNS-TCP, DNS-UDP, and ICMP—counted as one error.

5. *Insecure firewall management*. Access to the firewall over insecure, unencrypted, and poorly authenticated protocols—such as telnet, ftp, or x11—counted as one error.
6. *Too many management machines*. Firewalls should be managed from a small number of machines. Allowing management sessions from more than five machines was counted as a configuration error. While this threshold is somewhat subjective, most experts agree that it is reasonable.
7. *External management machines*. An error was counted if machines outside the network's perimeter could manage the firewall. The preferred way for administrators to manage the firewall from home is from the "inside" through a virtual private network.
8. *NetBIOS service*. NetBIOS is a set of services that Microsoft Windows operating systems use to support network functions such as file and printer sharing. These frequently attacked services are very insecure.[8] Allowing any NetBIOS service to cross the firewall in any direction counted as an error.
9. *Portmapper/Remote Procedure Call service*. The portmapper daemon assigns TCP ports to implement RPC services, a Unix mechanism that has a long history of being insecure. Among other services, RPCs include the Network File System protocol, which potentially exposes all the organization's file system. Allowing traffic to the portmapper (TCP or UDP on port 111) counted as an error.
10. *Zone-spanning objects*. A Check Point network object is a named definition of a set of IP addresses. Zone-spanning objects include addresses that reside on more than one "side" of the firewall—for example, some IP addresses internal to the firewall and others external. Note that for a firewall with more than two interfaces, each interface defines another "side." Zone-spanning objects cause many unintended consequences when used in firewall rules. For example, when administrators write a rule, they usually assume that the object is either internal or external, and this assumption affects how they write the rule. Zone-spanning objects break this dichotomy—with disastrous results.[10,11] Any use of zone-spanning objects in the rule set counted as an error.
11. *"Any" service on inbound rules*. Allowing "Any" service to enter the network is a gross mistake, since "Any" includes numerous high-risk services, including NetBIOS and RPC. Allowing such access was counted as an error.
12. *"Any" destination on outbound rules*. Because internal users typically have unrestricted access to the Internet, outbound rules commonly allow a destination of "Any." Unfortunately, firewalls commonly have more than two network interfaces—more than 86 percent of the firewalls in this study did. Typical usage for a third interface is to attach a *demilitarized zone*—that is, a subnet dedicated to the corporation's externally visible servers. In such cases, free Internet access also gives internal users free access to the servers in the DMZ. Worse, it often allows the DMZ servers free access to the internal network, because the predefined "Any" network object is inherently zone-spanning.[10] Therefore, allowing such access counted as an error.

Item 12 is probably the most subjective error counted. It is possible to safely use a destination of "Any" by carefully adding other rules that restrict the unwanted access. Nevertheless, finding "destination = Any" outbound rules in a firewall audit should, at least, raise a warning flag.

RESULTS AND ANALYSIS

Figure 1 shows the raw distribution of configuration errors discovered in the data. The results can only be characterized as dismal. Most of the errors appeared in most of the firewalls studied; in fact, nine of the 12 errors appeared in more than half the firewalls.

Even if we discount the two most frequent errors (items 10 and 12), which may be somewhat controversial, the results show that almost 80 percent of firewalls allow both the "Any" service on inbound rules (item 11) and insecure access to the firewalls (item 5). These are gross mistakes by any account.

Only one of the firewalls exhibited just a single misconfiguration. All the others could have been easily penetrated by both unsophisticated attackers and mindless automatic worms.

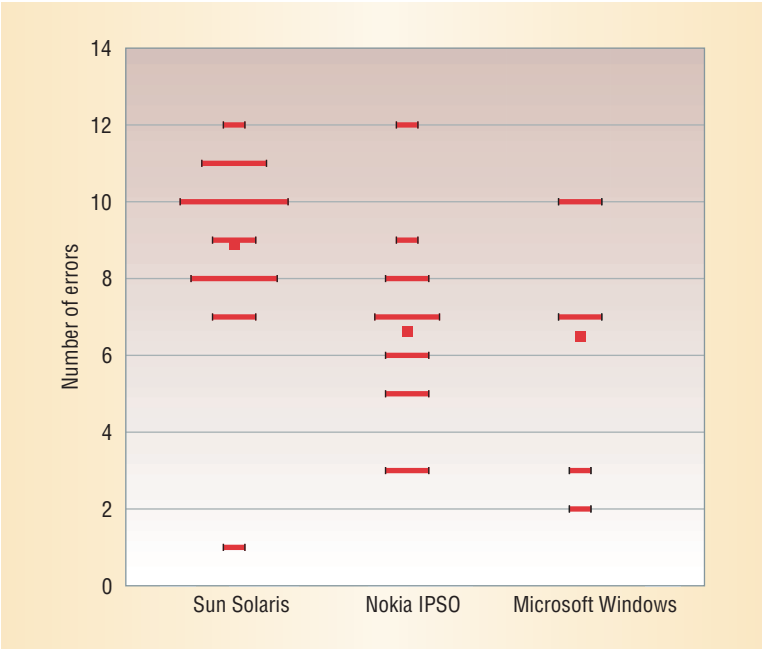


Figure 2. Number of errors as a function of the firewall's operating system. The bar widths are proportional to the number of configurations. The red squares mark the average number of errors per operating system.

Does the operating system matter?

For the period of data collection, Solaris was the oldest platform that Check Point supported, and Windows NT was the most recent. Nokia IPSO, a hardened version of BSD Unix, was developed specifically for security appliances like a firewall. Sun-based firewalls were, generally speaking, more typical of larger organizations, and smaller organizations used Nokia- and Microsoft-based systems.

Because of this distribution, I tested whether a correlation existed between the operating system and the number of errors. The Check Point FireWall-1 administration GUI and rule-set format are both operating-system agnostic, so it might seem that the operating system would be irrelevant. In fact, I think the operating system, per se, is irrelevant to configuration quality, and the results from this study do not in any way constitute purchasing recommendations.

However, the choice of operating system could reflect some other factor that could influence the configuration quality and manifest itself as a correlation between the number of errors and the operating system.

For example, the three platforms have distinct price ranges: During the data collection period, the Check Point FireWall-1 software and underlying hardware for a typical Sun-based firewall cost more than a Nokia system, which cost more than a Windows system. The systems' advertised performance had the same ranking order, with Sun-based systems marketed as being more appropriate for more demanding, high-traffic networks.

It seems reasonable to assume that an organization choosing to purchase the more expensive, higher-performance system might also have better firewall administrators and a higher awareness of network security. Thus, a priori, we can hypothesize that firewalls running over a Sun platform would be better configured than those running Nokia, which would be better configured than those running Microsoft Windows.

Figure 2 shows that, if anything, the opposite is true. The trend seems to be that Sun-based systems are more poorly configured than the two other platforms—although there is significantly less data for Windows-based systems.

It seems unlikely that the firewall administrators of Sun-based systems are less knowledgeable, so another factor is likely at work. The real issue here may lie in the tendency of Sun-based systems to have longer histories, to have multiple administrators managing them, and—in general—to be more complex.

Does the firewall version matter?

During the time the rule sets were collected, Check Point's product went through three major software versions (4.0, 4.1, and NG). In early 2000, the older 3.0 version was still being used in the market, even though the vendor no longer supported it, so our data also contains one rule set from version 3.0, as Table 3 shows.

The firewall version is relevant to our discussion, since Check Point introduced several changes to the default configuration settings with version 4.1. The changes should have helped eliminate some of the configuration errors.

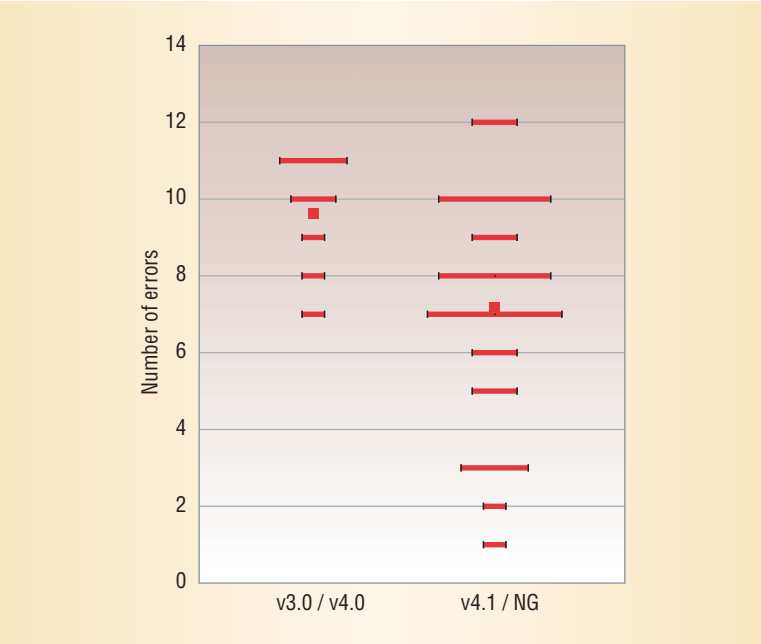


Figure 3. Number of errors as a function of firewall version. The bar widths are proportional to the number of configurations. The red squares mark the average number of errors in each version.

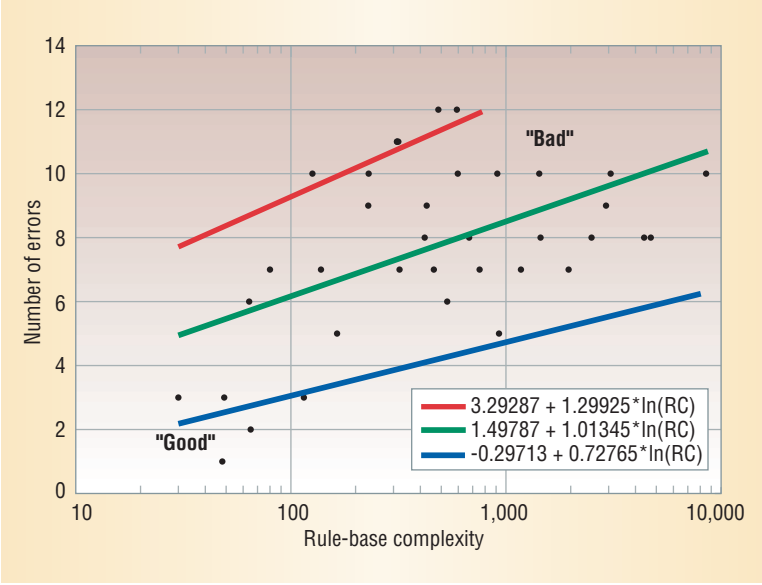


Figure 4. Number of errors as a function of rule-set complexity. The green line represents the least-squares fit; the red and blue lines represent one standard deviation above and below the least-squares fit.

First, the default values for the GUI fields controlling the DNS-TCP, DNS-UDP, and ICMP implicit rules were set to "false," whereas in version 4.0 and earlier the default was "true." Thus, if the administrator accepted the default settings, a firewall running version 4.1 would avoid three of the identified errors (items 2-4).

Second, the version 4.1 GUI included a new policy wizard. If the administrator used the wizard, the configuration would include both a stealth rule and a rule to drop all NetBIOS traffic, thus avoiding two more errors (items 1 and 8, respectively).

Therefore, we can hypothesize that version 4.1 and later configurations would likely show a lower number of errors than version 4.0 and earlier. Optimistically, we could hope to see a total drop of five errors between versions. In fact, Figure 3 shows a decrease in the average number of errors from 9.63 for versions 3.0/4.0 to 7.17 for versions 4.1/NG.

The benefit is significant, though not a five-error drop. The most likely reason is that the upgraded product improvements apply only to users that install new firewalls or at least create a new policy from scratch. If a user merely upgraded a rule set from version 4.0 to 4.1, the converted rule set maintained all its previous, insecure semantics.

Complexity matters: Small is beautiful

The RC measure showed a wide range in complexity values. The average RC was 1,121, the lowest value was 30, and the highest was an astonishing 8,521.

Figure 4 shows a scatter plot of the number of errors versus RC. While the plot is fairly sparse, the empty lower-right quadrant indicates that there are no good high-complexity rule sets. The only reasonably well-configured firewalls—three errors or less—are very simple, with RC values under 100. However, a small and simple rule set is no guarantee of a good configuration. The figure shows that a small rule set can be configured quite badly: Two configurations with RC values under 100 include six or more errors.

In fact, the RC measure yields a crude but fairly accurate prediction of the number of configuration errors: A linear regression shows that a rule set of complexity RC is predicted to have about ln(RC) + 1.5 errors. This is the formula for the central green line in Figure 4.

The conclusion to draw here is obvious: Limiting a firewall's rule-set complexity as defined by RC is safer. Instead of connecting yet another subnet to the main firewall and adding more rules and more objects, it's preferable to install a new, dedicated firewall to protect only that new subnet. Complex rule sets are apparently too difficult for administrators to manage effectively.

This study clearly shows that corporate firewalls are often enforcing poorly written rule sets. However, it includes some useful observations for improving rule-set quality as well. For example,

later versions of the Check Point FireWall-1 software include features that can noticeably improve security—mostly in newly installed systems. Furthermore, low-complexity rule sets appear to be better configured. Thus we can conclude that for well-configured firewalls, "small is beautiful." ■

References

1. A. Rubin, D. Geer, and M. Ranum, *Web Security Sourcebook*, Wiley Computer Publishing, 1997.
2. CERT Coordination Center, "CERT Advisory CA-2003-20: W32/Blaster Worm," 11 Aug. 2003; www.cert.org/advisories/CA-2003-20.html.
3. D. Moore et al., "The Spread of the Sapphire/Slammer Worm," 2003; www.caida.org/outreach/papers/2003/sapphire/sapphire.html.
4. A. Mayer, A. Wool, and E. Ziskind, "Fang: A Firewall Analysis Engine," *Proc. IEEE Symp. Security and Privacy* (S&P 2000), IEEE Press, 2000, pp. 177-187.
5. A. Wool, "Architecting the Lumeta Firewall Analyzer," *Proc. 10th Usenix Security Symp.*, Usenix Assoc., 2001, pp. 85-97.
6. W.R. Cheswick and S.M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison Wesley, 1994.
7. D.B. Chapman and E.D. Zwicky, *Building Internet Firewalls*, O'Reilly & Assoc., 1995.
8. SANS Institute, "The Twenty Most Critical Internet Security Vulnerabilities," v. 4.0, 2003; www.sans.org/top20/.
9. Symantec Security Response, "W32.Welchia.Worm," Aug. 2003; http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html.
10. A. Wool, "How Not to Configure Your Firewall: A Field Guide to Common Firewall Misconfigurations," presentation slides (invited talk), *15th Large Installation Systems Administration Conf.* (LISA), Usenix Assoc., 2001.
11. A. Wool, "The Use and Usability of Direction-Based Filtering in Firewalls," *Computers & Security*, in press; available online 2 Apr. 2004; www.sciencedirect.com/science/journal/01674048.

Avishai Wool *is an assistant professor at the School of Electrical Engineering, Tel Aviv University, and chief technical officer at Algorithmic Security, a network security company that he cofounded. His research interests include firewall technology, network and wireless security, data communication networks, and distributed computing. Wool received a PhD in computer science from the Weizmann Institute of Science, Israel. He is a senior member of the IEEE and a member of the ACM and Usenix. Contact him at yash@acm.org.*