Firewall Configuration Errors Revisited

Avishai Wool
School of Electrical Engineering,
Tel Aviv University, Ramat Aviv 69978, ISRAEL
E-mail: yash@acm.org.

November 6, 2009

Abstract

Practically every corporation that is connected to the Internet uses firewalls as the first line of its cyber-defense. However, the protection that these firewalls provide is only as good as the policy they are configured to implement. The first quantitative evaluation of the quality of corporate firewall configurations appeared in 2004, based on Check Point FireWall-1 rule-sets. In general that survey indicated that corporate firewalls were often enforcing poorly written rule-sets, containing many errors. One important finding was that high rule-set complexity was positively correlated with the number of detected configuration errors. Another finding was an indication that rule-sets from later software versions had slightly fewer errors.

The goal of this work is to revisit the first survey, and to test whether its findings remain valid. The current study is much larger, and is based on newer data, collected from firewalls running later firewall versions. Furthermore, for the first time the study includes configurations from two major vendors: both Check Point firewalls and Cisco PIX firewalls. Finally, the study considers three times as many possible configuration errors, consisting of 36 vendor-neutral errors instead of the 12 used in the 2004 study.

In order to compare the complexity of configurations from different vendors, this work also introduces a novel uniform complexity measure, called the *firewall complexity* ($\mathcal{FC}$), that applies to both types of firewalls.

The findings of the current study indeed validate the 2004 study's main observations: (a) firewalls are (still) poorly configured, and (b) a rule-set's complexity, as measured by the new $\mathcal{FC}$ measure, is (still) positively correlated with the number of detected configuration errors. These findings hold for rule-sets from both vendors. Thus we can conclude that, for well-configured firewalls, "small is (still) beautiful". However, unlike the 2004 study, there is no significant indication that later software versions have fewer errors (for either vendor). This is apparently because the vendor-neutral errors that this study focuses on are all controlled by the firewall's basic filtering capability—which has not changed significantly between versions.

1 Introduction

1.1 Background

Firewalls are the cornerstones of corporate intranet security. Once a firewall is acquired, a firewall administrator has to configure and manage it to realize an appropriate security policy for the particular needs of the company. This is a crucial task; quoting [RGR97]: "The single most important factor of your firewall's security is how you configure it".

It is generally accepted among network security experts that corporate firewalls are poorly configured. Anecdotal evidence of this sentiment can be found in such mailing lists as the Firewall Wizards list [FWW08]. Furthermore, the poor state of firewall configuration can be observed, indirectly, by the success of worms and viruses like Blaster [CER03] and Saphire [MPS+03], which could have been easily blocked by a well configured firewall. This state of affairs was validated in a 2004 study [Woo04a], which for the first time provided a quantitative evaluation of the quality of corporate firewall configurations collected in 2001.

However, firewall vendors regularly release new software versions for their products. In addition, the attention that is paid to firewall rule-set quality has increased, due to regulations such as the Sarbanes-Oxley act [SOX02] and the CobiT framework [Cob08]; the Payment-Card Industry Data Security Standard (PCI DSS) [PCI06]; and the NIST standard 800-41 [NIS02]. All these relatively new regulations include specific sections dealing with firewall configuration, management, and audit. Thus, one may hypothesize, or hope, that the quality of corporate firewall configurations has improved over time. The goal of this work is to test this hypothesis, and to check whether the findings of [Woo04a] are still valid.

1.2 Contributions

The data for the [Woo04a] study was collected in 2001, and clearly much has happened in the network security arena in later years. Therefore, one may challenge the validity of [Woo04a] claiming (or hoping) that the situation has improved over time. Moreover, [Woo04a] was fairly small in scope, covering only 37 rule-sets, all from Check Point firewalls, and considering only 12 possible errors, 8 of which were specific to Check Point FireWall-1. One could argue that the sample in [Woo04a] was not indicative, and that the detected problems were specific only to that vendor.

To address these possible critiques, the current study has the following features: (1) It is based on newer configuration data, collected from firewalls running later firewall versions; (2) It is significantly larger, covering more than twice as many rule-sets as [Woo04a]; (3) For the first time, this study includes rule-sets from *two* leading firewall vendors: Check Point FireWall-1 (cf. [Wel02]) and Cisco PIX (cf. [CF01]); and (4) this study considers three times as many configuration errors, consisting of 36 vendor-neutral items instead of the 12 used in [Woo04a].

The rule-sets were obtained from a variety of corporations that used the AlgoSec Firewall Analyzer between 2003–2005 [MWZ05, Alg09]. Note that corporate firewall rule-sets are considered to be highly sensitive, and were provided under non-disclosure agreements that limit or delay our ability to discuss them publicly. Hence, publishing quantified statistics about these rule-sets may be viewed a contribution in itself.

An important contribution of this work is the identification and selection of the 36 vendor-neutral firewall configuration errors to search for. In principle, errors should be defined in terms of each corporation's network security policy. However, our stance is that of an external auditor, and we do not really know these policies. Therefore, the selected errors are all clear violations of well established practices and guidelines (cf. [SAN07], [PCI06], [NIS02]).

The study's first finding is that serious errors are still alarmingly frequent: For instance, the infamous Microsoft services, that are a vector to numerous Internet worms (cf. [CER03]), are allowed to *enter* organiztions' networks from the outside in 42% of the surveyed firewalls. The most frequent error (allowing outbound SMTP from over 256 IP addresses[1]) occured in over 80% of the surveyed firewalls. Moreover,

[1]This setting may be appropriate for small corporations, but is a significant error in larger corporations that run their own mail

since we only consider violations of "best practices" and do not attempt to understand each network's semantics, the findings should be viewed as a rough baseline: the protection offered by the firewalls we surveyed may well be worse than suggested by this work.

One of the findings of [Woo04a] was that Check Point firewalls with versions 4.1 or later had slightly fewer configuration errors than those with earlier software versions. In contrast, the current study does *not* show this effect—neither for Check Point firewalls nor for Cisco PIX firewalls. The data shows that the distribution of the number of configuration errors is essentially independent of the firewall software version.

Probably the most important finding of [Woo04a] was that rule-set complexity was positively correlated with the number of detected errors—and for this purpose [Woo04a] defined a rule-set complexity measure denoted by $\mathcal{RC}$. Unfortunately, the $\mathcal{RC}$ measure is not suitable for Cisco PIX firewalls. Therefore, in order to compare the complexity of configurations by two different vendors, this work introduces a novel complexity measure, called the *firewall complexity* ($\mathcal{FC}$), that applies to both types of firewalls, and should be suitable to other firewall vendors' configuration languages as well. A key feature of the $\mathcal{FC}$ measure is that it produces comparable values between Check Point FireWall-1 and Cisco PIX. Furthermore, the $\mathcal{FC}$ values are comparable to the older $\mathcal{RC}$ values on Check Point FireWall-1.

What we shall see is that a rule-set's complexity, as measured by the new $\mathcal{FC}$ measure, is (still) positively correlated with the number of detected configuration errors. These findings hold for rule-sets from both vendors. Thus we validate and strengthen the finding of [Woo04a], that, for well-configured firewalls, "small is (still) beautiful".

The study also allows a preliminary investigation on whether the choice of vendor had an impact on the quality of the configuration. One finding is that Check Point FireWall-1 configurations tend to be more complex than Cisco PIX firewalls. Another finding is that in general Cisco PIX firewalls have somewhat fewer errors than Check Point firewalls of comparable $\mathcal{FC}$ complexity. However, the larger Cisco PIX firewalls are essentially as poorly configured as the worst Check Point firewalls.

Organization: The next section describes the data collection, provides some basic statistics of the data, indicates some caveats in the methodology, and describes the new $\mathcal{FC}$ measure. Section 3 describes the selection criteria for the configuration errors used in this study. Section 4 analyzes the data, describes the detected error frequencies, discusses the effects of the firewall's software version and rule-set complexity, and also compares the relative complexity and number of errors between the two vendors. We conclude in Section 5. Appendix A includes the full list of possible configuration errors that were used.

2 Methodology

2.1 Check Point firewalls

The data for this work includes 54 Check Point FireWall-1 rule-sets that were collected between 2003–2005. The rule-sets came from a variety of organizations, from the telecommunications, financial, energy, media, automotive, and healthcare market segments. Some basic statistics of the rule-sets are shown in Table 1.

The collected rule-sets came from three Check Point major software releases: 4.0, 4.1, and NG, with the NG rule-sets spanning several minor releases. For this study I grouped all NG rule-sets up to field-

servers; Letting so many machines send e-mail directly, bypassing the corporate mail server, indicates that the corporation is not properly controlling its outbound e-mail.

	Minimum	Median	Maximum
#Rules	2	79	617
#Objects	19	572	5443
#Interfaces	2	4	18

Table 1: Statistical properties of the collected Check Point FireWall-1 rule-sets. #Rules counts the total number of rules in the rule-set (including NAT rules). #Objects counts the number of network objects (hosts, subnets, and groups of these) defined in the database supporting the rules. #Interfaces counts the number of network interface cards on the firewall.

CHECK POINT FIREWALL-1 VERSION			
4.0	4.1	NG/NG-FP3	NG R55
4 (7.4%)	30 (55.6%)	17 (31.5%)	3 (5.5%)

Table 2: Distribution of Check Point FireWall-1 rule-sets by software version. For each software version we show both the absolute number of rule-sets and their percentage (in parenthesis)

pack 3 in one category called "NG/NG-FP3", and all later rule-sets in a separate category called "NG R55". See Table 2 for a distribution of versions. The firewall version is relevant to our discussion since [Woo04a] reported that rule-sets collected from later Check Point FireWall-1 versions tended to have fewer configuration errors—see Section 4.3 for a discussion.

2.2 Cisco PIX firewalls

The data for this work also includes 30 Cisco PIX rule-sets that were collected between 2003–2005. Some basic statistics of the rule-sets are shown in Table 3. The collected rule-sets include files from Cisco PIX versions 4.4–7.0. Since there were very few rule-sets from version 7.0 I grouped the rule-sets into four categories: 4.4, 5.0–5.2, 6.0–6.2, and 6.3–7.0. Table 4 shows the distribution of rule-sets into versions. See Section 4.3 for a discussion.

2.3 Caveats

When we consider the significance of the findings, we have to bear in mind several caveats.

The current study includes a total of 84 rule-sets. This is more than double the sample of [Woo04a], but still a very small sample—The number of operational firewalls in the world is estimated to be hundreds of thousands. Furthermore, these rule-sets were not selected randomly: they were provided by organizations that wanted to perform an audit of their firewall rule-set, and were willing to purchase software for this purpose. Ancdotal evidence suggests that, in some cases, these organizations felt their firewall rule-sets were too complex to manage effectively without better software systems. Therefore, our sample may be biased toward complex, badly configured, firewalls.

In defense, I claim that the arguments of [Woo04a] are still valid: The findings agree with the experience of many colleagues in the network security industry, and the poor state of firewall configuration can be observed, indirectly, by the huge proliferation of network worms like Blaster [CER03] and Saphire [MPS+03] that could have been easily blocked by a well configured firewall.

	Minimum	Median	Maximum
#Lines	71	365	3259
#Interfaces	2	4	8

Table 3: Statistical properties of the collected Cisco PIX rule-sets. #Lines counts the total number of lines in the configuration file. #Interfaces counts the number of network interface cards on the firewall.

CISCO PIX VERSION			
4.4	5.0–5.2	6.0–6.2	6.3–7.0
3 (10%)	7 (23.3%)	11 (36.7%)	9 (30%)

Table 4: Distribution of Cisco PIX rule-sets by software version. For each software version we show both the absolute number of rule-sets and their percentage (in parenthesis)

Finally, obtaining and reporting on *any* number of real rule-sets from operational firewalls is extremely rare: These are considered to be highly sensitive files. So there is significant value in analyzing the data that *can* be obtained.

2.4 A New Measure of Firewall Complexity

One of the discoveries of [Woo04a] was that more complex rule-sets have more errors in them. To make such statements precise, in [Woo04a] I introduced a measure of complexity defined as

$$\mathcal{RC} = \#\text{Rules} + \#\text{Objects} + \binom{\#\text{Interfaces}}{2},$$

where #Rules is the raw number of rules in the rule-set, #Objects is the number of network objects, and #Interfaces is the number of interfaces on the firewall. I believe that the $\mathcal{RC}$ measure was quite successful in capturing the essence of rule-set complexity. However, for this study I realized that I needed to improve upon it, for two reasons. First, I found that $\mathcal{RC}$ was not suitable for Cisco PIX firewalls, and second, I found that $\mathcal{RC}$ didn't discriminate between more and less complex rule-sets as well as I initially hoped.

2.4.1 A comparable measure

In order to compare the complexities of Cisco PIX firewalls to those of Check Point firewalls I needed a measure that produced comparable values for the two vendors. However, I realized that $\mathcal{RC}$ cannot be applied to Cisco PIX, since Cisco PIX and Check Point FireWall-1 are configured differently (see [Woo04b] for details). For the purpose of measuring complexity, the most important differences are:

1. A Cisco PIX configuration includes a separate set of rules (called an `access-list`) for each *interface*, whereas a Check Point FireWall-1 has a single set of rules that applies to all the interfaces—making #Rules incomparable.

2. Cisco PIX configurations do not have a separate object database, and up to version 6.0, Cisco PIX configurations did not even allow the definition of non-trivial objects, such as objects containing anything other than a single subnet. Thus #Objects is ill-defined for Cisco PIX firewalls.

2.4.2 More interfaces imply higher complexity

Beyond the immediate need for a measure that is suitable for Cisco PIX firewalls, I realized that $\mathcal{RC}$ could be improved upon for Check Point firewalls as well. Specifically, I felt that $\mathcal{RC}$ didn't give enough weight to the number of interfaces: as Table 1 shows, none of the surveyed Check Point firewalls has more than 18 interfaces (the median number was 4), yet it is not uncommon to find Check Point firewalls with hundreds of rules and thousands of objects. Since #Interfaces is only *added* to the $\mathcal{RC}$, albeit quadratically, its contribution is often dwarfed by the two other terms. For instance, I felt that a firewall with 12 interfaces is *much* more complex than one with 3 interfaces with the same numbers of rules and objects—and this intuition is not well captured by the old $\mathcal{RC}$ measure since the growth in the number of interfaces only contributes an extra $(66 - 3) = 63$ points to the $\mathcal{RC}$ of the more complex firewall.

2.4.3 The new $\mathcal{FC}$ measure

For all these reasons I decided to design a new measure of complexity. To differentiate the new measure from the old I call the new measure the *firewall complexity*, denoted $\mathcal{FC}$. Of course, this new measure still had to be objective, intuitive, and simple to compute.

For Cisco PIX firewalls, the simplest measure of complexity, and one which is often used in discussions with Cisco PIX administrators, is the number of lines in the configuration file. This was my first attempt to pick a new measure of complexity. However, I found the raw number of lines to be slightly misleading, especially for very small configurations. This is because even the smallest Cisco PIX configuration file includes a few tens of "boilerplate" lines that have little to do with traffic filtering. To compensate for this, I chose to use the following definition.

Definition 2.1 *Let #Lines denote the number of lines in the ASCII file containing the complete Cisco PIX configuration file. Then the* firewall complexity *of a Cisco PIX firewall is:*

$$\mathcal{FC}_p = \textit{\#Lines} - 50.$$

To define a comparable measure for Check Point firewalls, I tried to capture the results of an imaginary over-simplified "Check Point-to-PIX converter". Such a convertor would need to replicate the single Check Point FireWall-1 rule-set and place a copy on each Cisco PIX interface. Therefore, for the $\mathcal{FC}$ measure I chose to *multiply* the number of rules by the number of interfaces. However, object definitions in (newer versions of) Cisco PIX are global (not per-interface), so $\mathcal{FC}$ only needs to *add* add the number of Check Point FireWall-1 objects, once. These choices have the nice side-effect of giving much more weight to the number of interfaces than the old $\mathcal{RC}$ did, thereby addressing a shortcoming of the $\mathcal{RC}$ measure. Thus, we arrive at the following definition.

Definition 2.2 *Let #Rules denote the raw number of rules in the Check Point FireWall-1 rule-set, let #Objects denote the number of network objects, and let #Interfaces denote the number of interfaces on the firewall. Then the* firewall complexity *of a Check Point FireWall-1 firewall is:*

$$\mathcal{FC}_c = (\textit{\#Rules} \times \textit{\#Interfaces}) + \textit{\#Objects}$$

Figure 1 shows the distribution of the rule-set complexity as measured by $\mathcal{FC}$ over the surveyed Check Point FireWall-1 and Cisco PIX firewalls. While the precise value assigned by the new $\mathcal{FC}$ to a Check Point FireWall-1 rule-set differs from the value computed by the old $\mathcal{RC}$ for the same rule-set, the figure shows

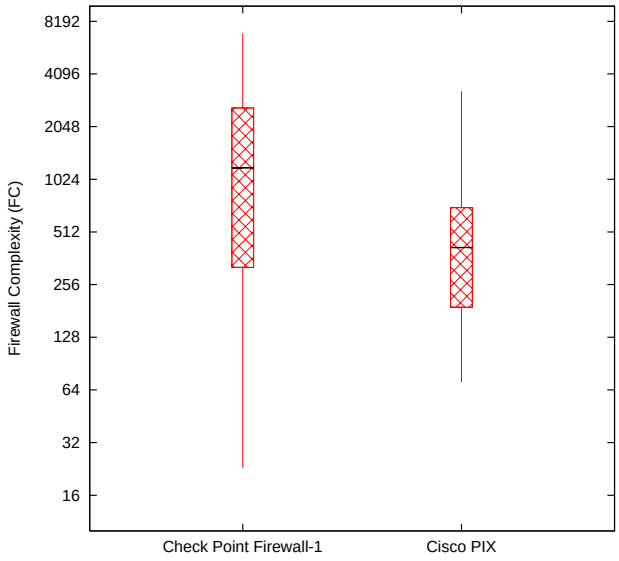

Figure 1: Firewall complexity distribution (log scale) separately for Check Point FireWall-1 and Cisco PIX. For each vendor we see a "bar and whiskers" column for the distribution of errors. The bottom and top of the whiskers mark the minimum and maximum values, the bottom and top of the bar mark the first and 3rd quartiles, and the black line within the bar marks the median.

that the genaral range of values remains the same: 10–10,000. But more importatly, the $\mathcal{FC}$ values assigned to Cisco PIX rule-sets fall into the same range as well.

The figure also shows that Check Point FireWall-1 firewalls typically have a higher complexity: the median $\mathcal{FC}$ value for Check Point firewalls is 1117 versus 315 for Cisco PIX (note that the Y-axis is log-scaled). My interpretation is that this is a real finding rather than an artifact of the $\mathcal{FC}$ metric: From inspecting the configuration data directly, it seems to me that indeed Check Point FireWall-1 configurations tend to be more complex.

3 The Selection of Configuration Errors

3.1 Selection Goals and Criteria

In [Woo04a] I considered twelve configuration errors. However, eight of those errors were specific to Check Point firewalls, which makes them unsuitable in a multi-vendor study. Therefore, in this study I used a different list of errors. The current list is much larger list, consisting of 36 errors. All the errors are vendor-neutral and create a risk to the network behind the firewall. The full list of errors appears in Appendix A.

As in [Woo04a], I took the stance of an external auditor. Thus, the errors I counted are all violations of well established practices and guidelines (cf. [SAN07], [PCI06], [NIS02]), that are independent of each organization's specific requirements and policies. As such, the findings should be viewed as a rough baseline: The protection offered by the surveyed firewalls may well be worse than suggested by this work.

Note that a single badly written rule may trigger multiple counted errors, since some errors contain other,

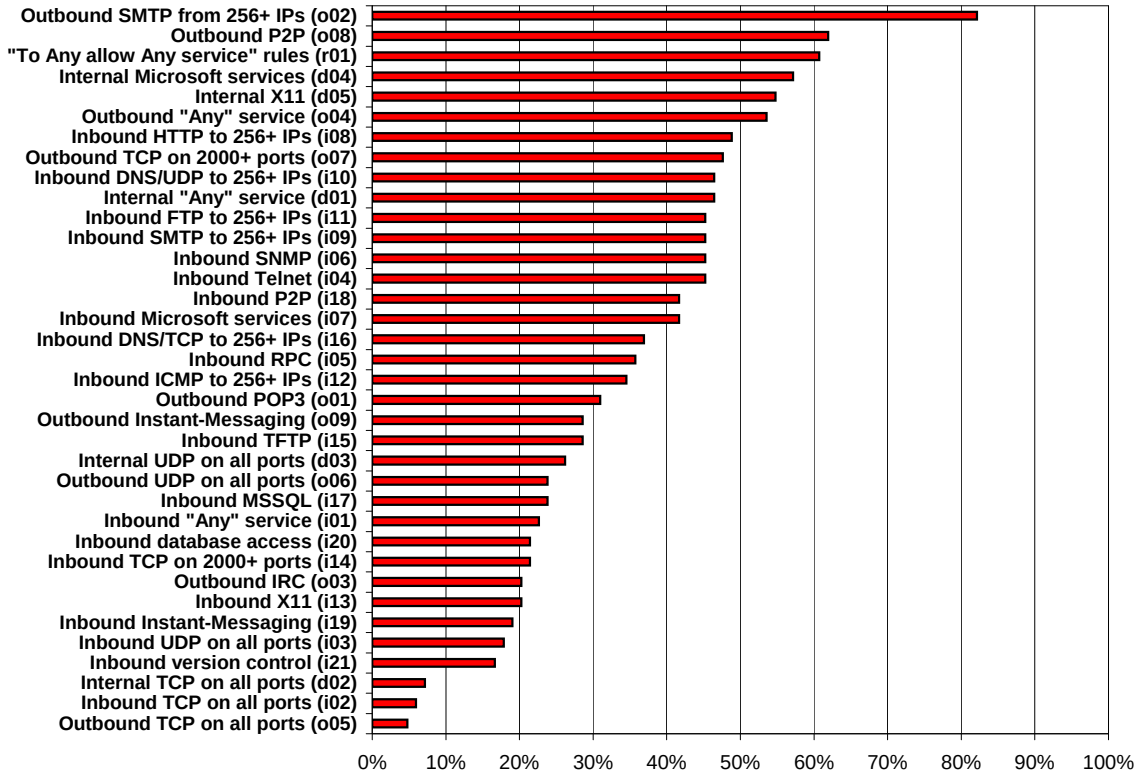

Figure 2: Distribution of configuration errors.

more specific, errors. For instance, allowing Telnet (TCP on port 23) is a special case of allowing “All TCP ports”, which in turn is a special case of allowing “Any service”. To avoid inflating the error counts because of this containment phenomenon, a more specific error was only counted if it was triggered by some rule that does not trigger a more general error. Continuing the previous example, a rule-set *can* trigger both the “Any service” and the “All TCP” errors — but only if the configuration includes two separate rules, one allowing Any service, and the other allowing all TCP ports.

One may argue that not all the configuration errors in the list are equally severe: For instance, some services have a poorer security history than others; also having all 65536 TCP ports open is probably more risky than having just the Telnet port open. However, in order to arrive at a single, easy to understand, number, I did not take such fine-grain considerations into account. The simplicity of counting each error type once allowed me to avoid introducing severity levels, and let me side-step the question “How many medium risk errors is one high risk error worth?”.

Furthermore, one may argue that a configuration with ten rules allowing Any service is riskier than a configuration with just one such rule. My opinion is that the number of erroneous rules is a good measure for the level of effort that the firewall administator would need to spend *fixing* the configuration, but less indicative of the amount of *risk*: If an attacker can enter the network, then the network is at risk—and it is less important if the attacker has one or ten firewall rules that allow him in. Therefore, again in the interest of simplicity, I ignored the number of rules contributing to each error, and opted for Boolean indicators for each error.

3.2 Error Categories

The counted configuration errors (refer to Appendix A) are organized into four categories: inbound traffic, outbound traffic, internal traffic, and inherently risky rules. The errors in each category have an identifying code with a leading letter indicating the category: "i" errors for inbound traffic, "o" errors for outbound traffic, "d" errors for internal, and "r" errors for risky rules.

The majority of configuration errors (21 of the 36) are for inbound traffic ("i" errors). This is in line with the firewall's perceived primary job, which is to protect the internal resources from outsider attach. These errors cover things such as allowing Any traffic inbound, or allowing services such as NetBIOS, Telnet, RPC, SNMP, and several other known-risky services, through the firewall in the inbound direction.

I also wanted to count errors that are related to services such as HTTP, DNS, and FTP. These are services that are often necessary to the running of business, so allowing then inbound should not be counted as an error *per se*. For instance, HTTP must be allowed to reach the organization's web servers. However, allowing HTTP to reach machines that are not hardened web servers is risky—many sensitive machines (like e-mail servers, phone switches, routers, and firewalls) provide a web interface.

The challenge is that we do not necessarily know which IP addresses belong to the relevant servers and which do not. To get around this challenge, I chose to use *thresholds*. The key idea is that it is very rare to find hundreds of different web servers on one network. Thus, if HTTP is allowed to reach over 256 IP addresses—more than a full class-C subnet—this is almost always the result of badly written rules. Several of the inbound traffic configuration errors include such thresholds.

Configuration errors for outbound traffic ("o" errors) contribute 9 of the 36 items. Most of these deal with services that are considered to be risky in all directions. In addition I counted errors for indiscriminate outbound e-mail traffic (again using the notion of a thresholds) and for IRC, which is notorious for carrying the command-and-control channel of "bot-nets".

Internal traffic through the firewall (between separate internal network segments) contributes 5 of the 36 items ("d" errors): these all deal with services that are considered to be risky in all directions. Finally, I also counted a special "r" error for the presence of rules that are inheretly problematic: An error is counted if the rule-set has "To Any allow Any service" rules.

4 Analysis and Discussion

4.1 Firewalls are still badly configured

Figure 2 shows the raw distribution of configuration errors that were discovered in the data. The results are perhaps unsurprising in view of [Woo04a]: Generally speaking, the surveyed firewalls are poorly configured.

In the inbound direction, over 45% of the firewalls allowed DNS, or FTP, or SMTP, to reach over 256 addresses (items i10, i11, and i09 respectively). Possibly more worrisome is that 42% of firewalls allowed inbound NetBIOS traffic (item i07).

In the outbound direction the situation is apparently worse. Over 80% of firewalls allow broad outbound SMTP access (item o02)—and over 60% of firewalls allow outbound P2P (o08), services which hardly ever have any business use. Finally, over 60% of firewalls have rules of the form "from somwhere to Any allow Any service" (r01): very lax rules, which constitute gross mistakes by any account.

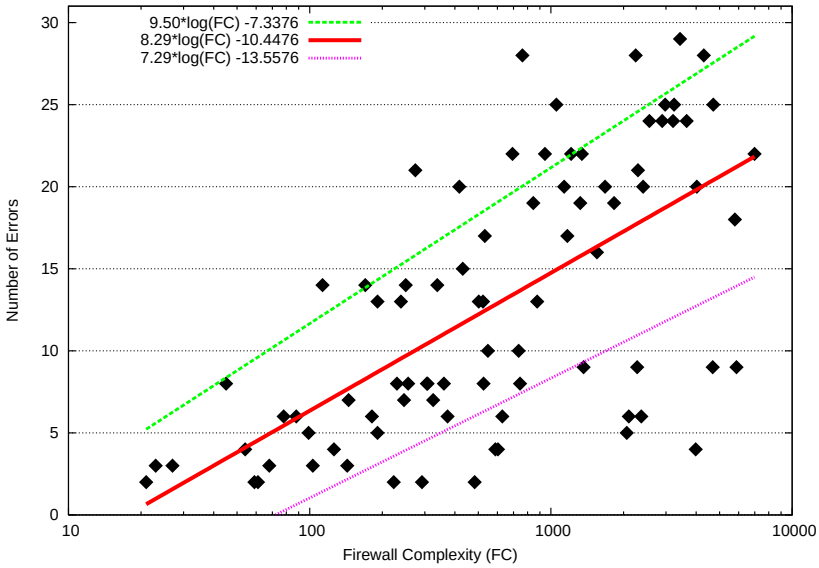

Figure 3: Number of errors as a function of the rule-set's complexity $\mathcal{FC}$ (log scale). The central red line represents the least-squares linear regression fit, and the green and purple lines represent one standard deviation above and below the least-squares fit.

4.2 Complexity Matters: Small is Still Beautiful

One of the main findings of [Woo04a] was that high rule-set complexity was positively correlated with the number of detected configuration errors. Figure 3 shows a scatter plot of the number of errors versus $\mathcal{FC}$, and demonstates the same phenomenon in the current survey. We can observe that there are very few high-complexity rule-sets that are well-configured (the lower-right quadrant of the figure is very sparse). Furthermore, the figure clearly shows a correlation between the rule-set complexity, as measured by $\mathcal{FC}$, and the number of detected errors.

In fact, the $\mathcal{FC}$ measure gives us a crude but fairly accurate prediction of the number of configuration errors: a linear regression (the central line on Figure 3) shows that the number of errors in a rule-set of complexity $\mathcal{FC}$ is roughly captured by the following formula:

$$\#\text{Errors} \approx 8 \log_{10}(\mathcal{FC}) - 10.$$

We can see that the current survey validates the finding of [Woo04a]: it's safer to limit the complexity of a firewall rule-set. Instead of connecting yet another subnet to the main firewall, and adding more rules and more objects, it seems preferable to install a new, dedicated, firewall to protect only that new subnet. Complex firewall rule-sets are apparently too complex for their administrators to manage effectively.

As an aside, note that both Check Point and Cisco (and indeed other major firewall vendors) are now offering virtualized solutions, in which a single piece of hardware can be set up to function as multiple separate firewalls. Besides offering various operational and commercial advantages, our results suggest that using multiple small (virtual) firewalls is likely to be more secure than a single large firewall.

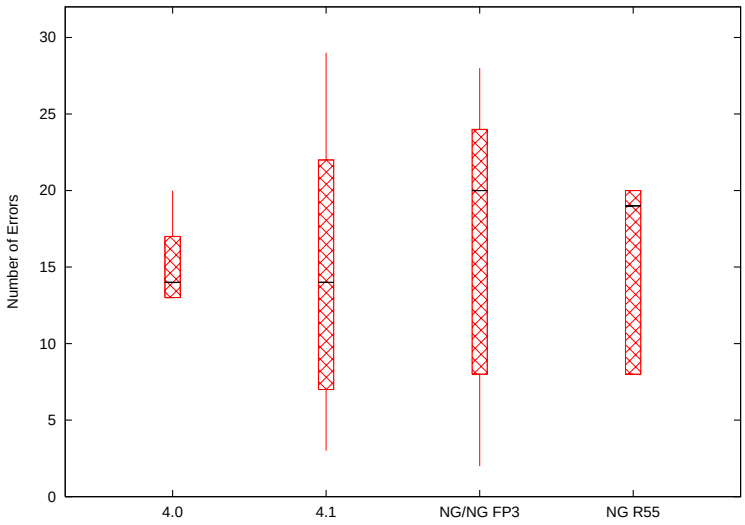

Figure 4: Errors versus version — comparing the distribution of errors between Check Point FireWall-1 versions 4.0, 4.1, NG up to FP3, and NG R55. For each software version we see a "bar and whiskers" column for the distribution of errors. The bottom and top of the whiskers mark the minimum and maximum values, the bottom and top of the bar mark the first and 3rd quartiles, and the black line within the bar marks the median.

4.3 Does the Firewall's Version Matter?

One of the findings of [Woo04a] was that Check Point firewalls with versions 4.1 or later had slightly fewer configuration errors than those with earlier software versions. I wanted to test whether the same trend can be observed in this study as well.

Figures 4 and 5 show the distribution of errors across the different versions, separately for Check Point FireWall-1 and Cisco PIX. It is clear from both figures that, contrary to the findings of [Woo04a], the effect of the software version on the number of configuration errors is insignificant: the distribution of the number of errors is essentially independent of the firewall software version. The data does *not* support the hypothesis that later software versions are correlated with fewer errors (for both vendors).

I believe that [Woo04a] detected such a trend because 8 of the configuration errors it considered were controlled by global options in the Check Point FireWall-1 user interface (rather than by explicit user-defined rules). The default settings for those gloal options were improved in v4.1, producing the detected effect. However, all the configuration errors considered in this study are vendor-neutral and are controlled by explicit user-defined rules: i.e., by the basic filtering functionality of the firewalls. This functionality has not changed syntactically or semantically, in neither vendor's products, during the period that data was collected. Therefore we see that later software versions do not help users write better filtering rules.

4.4 Does the Vendor Matter?

Given the fact that the data includes rule-sets from two vendors, with rather different configuration languages and tools, I wanted to investigate whether the choice of vendor had an impact on the quality of the

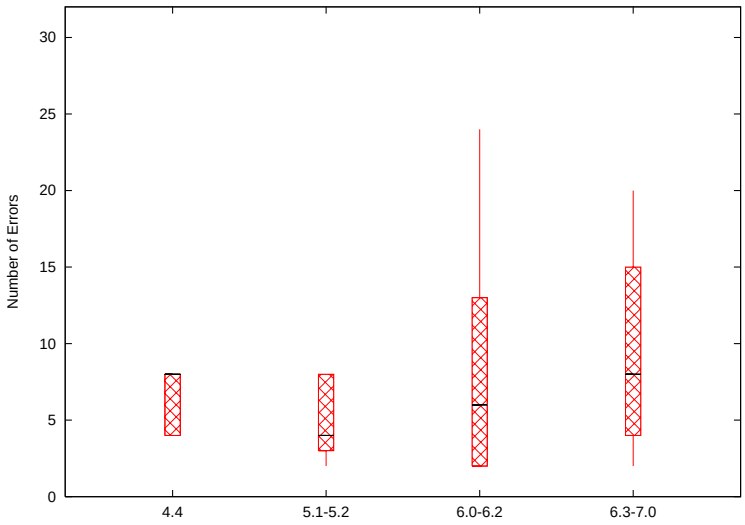

Figure 5: Errors versus version — comparing the distribution of errors between Cisco PIX versions 4.4, 5.x, 6.0-6.2, and 6.3-7.0. For each software version we see a "bar and whiskers" column for the distribution of errors. The bottom and top of the whiskers mark the minimum and maximum values, the bottom and top of the bar mark the first and 3rd quartiles, and the black line within the bar marks the median.

configuration.

We already saw in Figure 1 that Check Point FireWall-1 configurations tended to be more complex than Cisco PIX firewalls—and from Figure 3 we learned that more complex configurations typically have more errors. Hence, we can expect Check Point firewalls to have more errors than Cisco PIX firewalls just because of their greater complexity. To test the extent of this phenomenon, Figure 6 repeats the data of Figure 3 but with separate markers and regression lines for the two vendors.

As expected, the figure shows that in general Cisco PIX firewalls seem to exhibit fewer errors than Check Point firewalls: the median number of errors for the Cisco PIX firewalls is 6 versus 14 for Check Point firewalls. Furthermore, the vertical distance between the least-squares regression lines indicates that Cisco PIX firewalls have approximately 5 errors less than Check Point firewalls of comparable $\mathcal{FC}$ complexity. But a closer look reveals that small Cisco PIX firewalls, with $\mathcal{FC} < 1000$, indeed have significantly fewer errors than small Check Point firewalls. However, the larger Cisco PIX firewalls, with $\mathcal{FC} \geq 1000$, are essentially as poorly configured as the worst Check Point firewalls. Note that this latter observation needs to be validated with more data: the surveyed firewalls only include 5 Cisco PIX firewalls with $\mathcal{FC} \geq 1000$.

As background for a possible explanation, other sources of information indicate that the Cisco PIX configuration language and operational environment is more difficult to learn and use than Check Point FireWall-1: E.g., Cisco PIX involves a complex notion of "direction" (cf. [Woo04b]); Cisco PIX mandates the use of network address translation[2] (NAT) statements—even in the case that no translation is needed; Finally, many Cisco PIX firewalls are managed from the command line whereas Check Point firewalls are managed using a graphical user interface. As a result, one often sees questions on how to confiure the basic functionality of a Cisco PIX on mailing lists such as Firewall-Wizards [FWW08].

[2]The strict requirement of writing NAT statements was relaxed with Cisco PIX version 7.x.

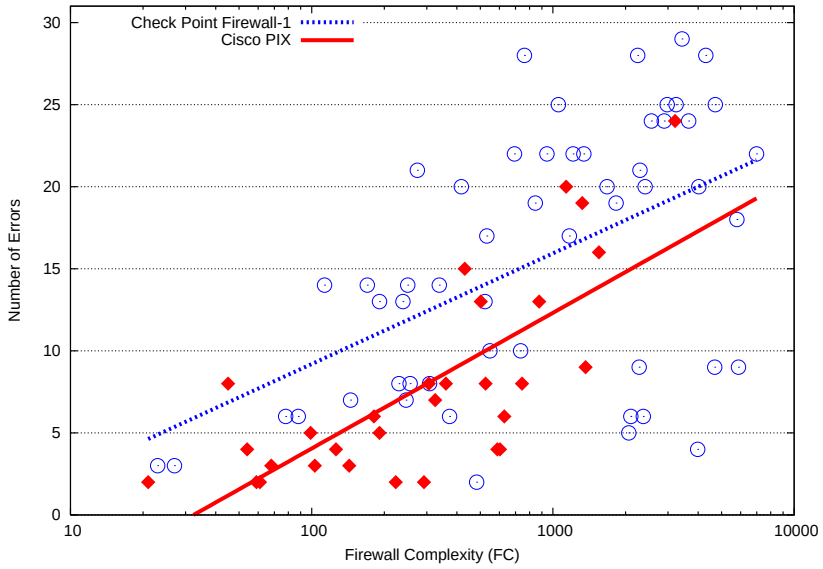

Figure 6: Number of errors as a function of the rule-set's complexity $\mathcal{FC}$ (log scale), separately for Check Point FireWall-1 (blue circles) and Cisco PIX (red diamonds). The dotted blue line and solid red line represents the least-squares regression fit of the Check Point firewalls and Cisco PIX firewalls, respectively.

With this background, my interpretation of Figure 6 is that, perhaps paradoxically, the steeper learning curve and harsher work environment of a Cisco PIX encourage administrators to produce smaller configurations—that are less error-prone than comparable Check Point firewalls. However, once the Cisco PIX configuration complexity becomes large—the number of errors seems similar to the numbers in comparable (complex) Check Point firewalls.

5 Conclusions

As we've seen, this work follows the methodology of [Woo04a]. The current study is much larger, and is based on newer data, collected from firewalls made by two major vendors, running later firewall versions. Finally, the study considers three times as many possible vendor-neutral configuration errors. In order to compare the complexity of configurations from different vendors, this work also introduced a new uniform *firewall complexity* ($\mathcal{FC}$) measure, that applies to both firewalls brands and probably to others as well.

Unlike [Woo04a], the current study shows no significant indication that later software versions have fewer errors (for either vendor)—it seems that later software versions do not assist the administrators to make fewer errors.

However, the findings do validate the main observations of [Woo04a]: (a) firewalls are (still) poorly configured, and (b) a rule-set's complexity is (still) positively correlated with the number of detected configuration errors. These findings hold for rule-sets from both vendors. Thus we can conclude that, for well-configured firewalls, "small is (still) beautiful".

References

[Alg09] Algorithmic Security's Firewall Analyzer, 2009. http://www.algosec.com.

[CER03] CERT advisory CA-2003-20: W32/Blaster worm, 11 August 2003. http://www.cert.org/advisories/CA-2003-20.html.

[CF01] D. W. Chapman and A. Fox. *Cisco Secure PIX Firewalls*. Cisco Press, 2001.

[Cob08] Control objectives for information and related technology (CobiT) v4.1. IT Governance Institute (ITGI), 1992–2008. http://www.isaca.org/cobit/.

[FWW08] Firewall Wizards. Electronic mailing list, 1997–2008. Archived at https://listserv.icsalabs.com/pipermail/firewall-wizards/.

[MPS+03] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The spread of the Sapphire/Slammer worm, 2003. http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html.

[MWZ05] A. Mayer, A. Wool, and E. Ziskind. Offline firewall analysis. *International Journal of Information Security*, 2005.

[NIS02] Guidelines on firewalls and firewall policy. NIST Special Publication 800-41, 2002. http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf.

[PCI06] PCI data security standard (PCI DSS) v1.1. Payment Card Industry - Security Standards Council, 2006. https://www.pcisecuritystandards.org/.

[RGR97] A. Rubin, D. Geer, and M. Ranum. *Web Security Sourcebook*. Wiley Computer Publishing, 1997.

[SAN07] SANS Institute. The twenty most critical Internet security vulnerabilities, version 4.0. http://www.sans.org/top20/, 2007.

[SOX02] Public company accounting reform and investor protection act of 2002 (Sarbanes-Oxley Act), 2002. Wikipedia article at http://en.wikipedia.org/wiki/Sarbanes-Oxley_Act.

[Wel02] D. D. Welch-Abernathy. *Essential Checkpoint Firewall-1: An Installation, Configuration, and Troubleshooting Guide*. Addison-Wesley, 2002.

[Woo04a] A. Wool. A quantitative study of firewall configuration errors. *IEEE Computer*, 37(6):62–67, 2004.

[Woo04b] A. Wool. The use and usability of direction-based filtering in firewalls. *Computers & Security*, 23(6):459–468, 2004.

A Appendix: The Selected Configuration Errors

A.1 Errors Related to Inbound Traffic

Twenty one of the errors I counted are related to traffic crossing the firewall from the outside to the inside. These include:

i01: Inbound "Any" service Allowing the "Any" service to enter the network is extremely risky since the "Any" service includes many vulnerable services.

i02: Inbound TCP on all ports Allowing TCP on all ports inbound is almost as risky as allowing Any service, since this includes many vulnerable TCP services. On PIX firewalls, this risk is often the result of neglecting to specify a port number on an access-list or conduit statement: omitting the port number is interpreted as "all ports".

i03: Inbound UDP on all ports Similar to allowing all TCP ports but for UDP.

i04: Inbound Telnet Telnet is a remote-login service that is not encrypted. It is only authenticated by simple passwords, that are transmitted in the clear. Anyone snooping the traffic can read all the transmitted information.

i05: Inbound RPC The RPC service is a mechanism for remote procedure calls in Unix environments. It is implemented using the `portmapper` daemon, which assigns TCP ports to the services. RPC services have a long history of being insecure, and among others they include NFS (which potentially exposes all the organization's file system).

i06: Inbound SNMP SNMP is the Simple Network Management Protocol, which allows scanning and identifying the network infrastructure. It is only authenticated by a simple password, called the "community string". The community string is often left at its default setting of "public", and transmitted in the clear. Several serious vulnerabilities have been reported in SNMP.

i07: Inbound Microsoft services NetBIOS is the name of a set of services that are used by the Microsoft Windows family of operating systems to support network functions such as file and printer sharing. These services are very insecure, and include some of the most attacked services [SAN07]. I counted an error if the firewall allowed any of these services to cross it.

i08: Inbound HTTP to 256+ IPs HTTP is the main protocol used for web browsing, so allowing it inbound is not counted as an error *per se*. However, allowing HTTP reach machines that are not hardened web servers is risky: Many sensitive machines provide a web interface: These include e-mail servers, printers, phone switches, routers, and firewalls. Therefore allowing HTTP to reach over 256 IP addresses is counted as an error because it is very rare to find so many web servers on one network—such access is almost always the result of badly written rules.

i09: Inbound SMTP to 256+ IPs SMTP is the main E-mail protocol—and E-mail is a vector for many viruses and worms. Nevertheless, allowing SMTP inbound is a requirement for many organizations, so inbound SMTP is not counted as an error. However, it is very rare to find more than 256 mail servers on one network, so allowing SMTP to reach over 256 IP addresses is counted as an error.

i10: Inbound DNS/UDP to 256+ IPs DNS is the Domain Name Service on UDP port 53 or on TCP port 53. DNS is one of the most attacked services in use [SAN07]. As before, inbound DNS is often required—so it's not counted as a risk—but allowing DNS to reach over 256 DNS servers on one network *is*.

i11: Inbound FTP to 256+ IPs FTP is the File Transfer Protocol, which is necessary for many organizations. Serious vulnerabilities have been found in many versions of FTP server software. As before, an error is counted if FTP is allowed to reach over 256 servers on one network.

i12: Inbound ICMP to 256+ IPs ICMP is the Internet Control Message Protocol, which is necessary for many network management tools (like `ping`). Full ICMP access allows outsiders to scan the network,

it lets worm-generated traffic cross the perimeter, and it can be used to mount denial-of-service attacks. An error is counted if ICMP can reach over 256 internal machines.

i13: Inbound X11 X11 is a popular graphical window system developed at MIT and implemented on UNIX systems to allow UNIX-based applications to be run from multiple types of terminals, PCs, and workstations. X11 is not encrypted and uses weak authentication. An error is counted if X11 is allowed inbound.

i14: Inbound TCP on 2000+ ports Allowing traffic over 2000 TCP ports inbound is almost as risky as allowing all TCP ports (item i02), since this includes many vulnerable TCP services. Such access is usually the result of an "permit all TCP ports" rule that is slightly mitigated by a few earlier "deny TCP on port X" rules.

i15: Inbound TFTP TFTP is the Trivial File Transfer Protocol. It is not encrypted and not authenticated in any way. Anyone who can use TFTP inbound can upload any file onto any device that responds to TFTP, such as routers, firewalls, and other communication equipment.

i16: Inbound DNS/TCP to 256+ IPs This is the same as item i10 except over TCP.

i17: Inbound MSSQL The Microsoft SQL Server (MSSQL) contains several serious vulnerabilities that allow remote attackers to obtain sensitive information, alter database content, compromise SQL servers, and, in some configurations, compromise server hosts. It's in the SANS Top 20 list [SAN07].

i18: Inbound P2P Peer-to-peer services (napster, edonkey, gnutella, kazaa) are used to download, and distribute many types of data (e.g., music, video, graphics, text, source code, and proprietary information to name a few). These services are in the SANS Top 20 list [SAN07].

i19: Inbound Instant-Messaging Instant Messaging services (MSN, yahoo, aol) let users use text-based chat, check remote web based email, do voice chat, perform video communication, and send and share data files. These services are in the SANS Top 20 list [SAN07].

i20: Inbound database access Database protocols (SQLnet, mysql, postgresql) provide remote access to relational databases. Allowing direct inbound access to your corporate databases is risky. These services are in the SANS Top 20 list [SAN07].

i21: Inbound version control Version control systems (cvs, subversion) provide tools to manage different versions of documents or source code. Both systems have serious known vulnerabilities. These services are in the SANS Top 20 list [SAN07].

A.2 Errors Related to Outbound Traffic

For outbound traffic I counted three specific services that are particularly problematic in the outbound direction (o01, o02, and o03), plus six errors that are similar to inbound errors, as follows:

o01: Outbound POP3 POP3 is used to download E-mail from mail servers to desktop computers. Since E-mail is a vector for viruses and worms, many organization only allow POP3 to reach their internal E-mail servers, and forbid any access to external E-mail servers. An error is counted if POP3 is allowed outbound.

o02: Outbound SMTP from 256+ IPs Outbound SMTP (E-mail) should only originate from mail servers, not from individual desktops. An error is counted if over 256 IP addresses can send outbound SMTP through the firewall.

o03: Outbound IRC Internet Relay Chat (IRC) is an interactive chat service. Many worms and Trojans use outbound IRC as a communications vector.

o04: Outbound "Any" service Similar to the i01 item.

o05: Outbound TCP on all ports Similar to the i02 item.

o06: Outbound UDP on all ports Similar to the i03 item.

o07: Outbound TCP on 2000+ ports Similar to the i14 item.

o08: Outbound P2P Similar to the i14 item.

o09: Outbound Instant-Messaging Similar to the i19 item.

A.3 Errors Related to Internal Traffic

For traffic between different internal network segments that are separated by the firewall the errors are a small subset of the inbound errors, as follows:

d01: Internal "Any" service Similar as the i01 item.

d02: Internal TCP on all ports Similar as the i02 item.

d03: Internal UDP on all ports Similar as the i03 item.

d04: Internal Microsoft services Similar as the i07 item.

d05: Internal X11 Similar as the i13 item.

A.4 Inherently Risky Rules

In addition to all the traffic-based errors described above, I also counted a special error for rules that are inheretly problematic, even if there are some other rules that mitigate some of their bad effects. An error is counted (with code **r01**) if the rule-set has "To Any allow Any service" rules.