Fast, Cheap, and In Control: A Step Towards Pain Free Security!

Sandeep Bhatt, Cat Okita, and Prasad Rao – Hewlett-Packard

ABSTRACT

We hypothesize that it is possible to obtain significant gains in operational efficiency through the application of simple analysis techniques to firewall rule sets. This paper describes our experiences with a firewall analysis tool and metrics that we have designed and used to help manage large production rule sets. Firewall rule sets typically become increasingly unwieldy over time. It is common for firewalls to have hundreds, or even thousands, of rules. Not surprisingly, administrators have a hard time keeping track of how the rules interact with each other, resulting in many partially effective or completely ineffective rules, and unpredictable behavior. Our tool can be used to identify these problematic rules. Further, given two rule sets, our tool produces a comprehensive list of the traffic that is only permitted or denied by one rule set, rather than both. As such, we can compare the existing rule set with a second rule set containing the proposed changes. The administrator can then visually check if the difference in traffic patterns corresponds to what he or she intended in proposing the changes. Additionally our tool collects various metrics that help the administrator to gauge the 'health' of the firewall. The tool is designed to be extensible to multiple vendor products.

Introduction

Securing the increasingly complex and highly networked environments of today is a challenging and frustrating task. Between compliance demands, such as Sarbanes-Oxley and PCI, new applications and services, and increased end-user awareness of security issues, it is a challenge to maintain security in any environment. As the complexity of an environment increases, the complexity of the configurations required to secure access to the environment increases, as does the amount of time required to maintain, update and validate the configurations. Changes to the security configurations start to produce unexpected side effects, and often result in a reluctance to make any changes at all!

Similarly, the amount of time required to debug obvious access issues increases dramatically with the size and complexity of the configurations involved – and skyrockets when multiple groups are involved in debugging any single issue. Configuration issues which do not result in immediately obvious issues such as permitting additional access that either fails to be noticed (or is so useful that nobody wants to mention it!) are often missed, and are hard to find without extensive, repeated effort on the part of an experienced system administrator (nowadays affectionately referred to as a 'resource').

Migration between vendors, or versions of security devices is also challenging – there is no standard configuration language for security devices like firewalls, and most vendors do not provide migration tools between their own OS releases, let alone from the OSes of other vendors.

Compounding the above issues, since the security of an environment is often a case of proving a negative, there is a definite need for straightforward, easily understood metrics that can be used to describe the state of a given configuration. Alas, while there are many tools that can be used to secure an environment – firewalls, intrusion detection/prevention devices, virus scanners, et al. – there is a dearth of simple, multi-platform tools that can be used to analyze and measure existing installations.

Problem

In this work, we have elected to focus on ways to improve firewall rule set management, which many security managers have identified as an ongoing challenge. Typical operational issues include how to determine the effect of adding or removing a firewall rule, clean up messy firewall rule sets and debug firewall rule related issues; other challenges include reporting the ongoing status of firewall operations in an easy to evaluate format.

The increasing commoditization of firewall management through outsourced and managed services, as well as decreasing amount of time or skill (or both) available to manage firewalls in house has also led to an increase in "cargo cult" style firewall operations, and a "once in, never out" rule set management.

Methodology

Our goal was to produce a high value, fast, lightweight tool that can be used to improve firewall rule set management and acts as an easy to implement supplement to existing systems and processes, rather than adding overhead and cost.

The methods used must be vendor and product independent, and be easily extensible to additional products.

Approach

To meet our goal of producing a minimally intrusive and maximally effective tool, we elected to restrict ourselves to analysis of security configurations (e.g., firewall configurations and router ACLs), and more specifically to the structural properties of the rule sets.

We do not evaluate whether a given configuration enforces the 'correct' policy, or adheres to some specific set of best practices which may or may not be applicable.

These decisions have multiple advantages – analysis takes place offline, with no requirement for agents or special access to the security infrastructure. Analysis is repeatable, and can be used to show improvement (or lack of improvement) in the configuration(s) over time. Further, as the analysis is based on the structural properties of the rule sets, it is vendor agnostic, and allows for rule set comparison between different products.

We first identify the minimal set of information required to describe the action of the rule sets from various configurations, which allows us to translate the configurations to a ''standard'' grammar. As a result, it becomes extremely straightforward to handle and compare multiple types of configurations.

We next identify a set of universal issues that typically cause hard to predict, difficult to diagnose (or inexplicable) effects on firewall management, and then develop a set of common ideal conditions for firewall rule sets to avoid those issues, and define metrics which can be used to measure the degree to which a given firewall rule set differs from the ideal.

Finally we describe the implementation and use of the prototype tool to improve the management of production firewall rule sets, and lessons learned.

Sidebar: Terminology

Location: Source or destination in a rule
Service: Port or protocol
Action: What to do when a rule matches
Rule: A source, destination, service & action combination
Object: Location or Service
Block: A subset of a Rule or Object

For the purposes of this paper, a firewall rule set is a collection of declarations consisting of a source location, destination location, service and action declaration. We do not address transformations such as NAT or PAT in this paper, although the work is extensible. Since each location and service can be a group of locations and services, and there are no constraints preventing overlapping locations or services, it is unfortunately simple to define (and hard to discover) rules which are partially or completely similar to other rules.

Realization

There are policy and vendor independent characteristics that can be generalized as being universally common to a well managed firewall rule set, and which can be used as a basis for metrics to evaluate and improve firewall rule sets.

We posit that the ideal structural properties of a firewall rule set are:

- Non-Interference
    - Rules should not interfere with other rules. Rules should not partially or completely overlap other rules, or be partially or completely overlapped by other rules except in the case where the more specific rule is acted on first, and is completely overlapped by the less specific rule, and the action of the less specific rule differs. Overlapping rules are described as 'eclipsing' or 'eclipsed' rules, and are broken down into 'interfering blocks.'
    - Objects are unique. Objects do not define the same location or service.
- Simplicity
    - There are no unused non-default objects defined
    - Only rules which can be triggered are defined in the rule set
    - Rules permit only what is required by policy
- Consistency
    - Rules actions are consistent. If rules interfere, except in the case noted above, they should have the same action.
    - Object naming style is consistent if named objects are used.

Unfortunately, since firewalls are highly idiosyncratic, it is not possible to discuss firewall rule sets without noting the existence of configuration and device specific corner cases.

- Object Template Re-Use: Objects in multiple firewall policies used by a common management interface must remain identical across all firewalls using them.
- Rule Set Commonality: Rules which can not be triggered may only be permitted if a single rule set common to multiple firewalls is in use.

Interpretation

Given the ideal characteristics shown above, we describe a set of metrics that we can use to measure how well a firewall rule set is managed. In order to be meaningful across multiple firewalls, and multiple types of firewalls, the metrics selected must also be policy independent and vendor independent.

We use Non-Interference and Consistency to provide an understanding of the complexity of a given firewall rule set, while Simplicity and Effectiveness measure the functionality of the rule set.

- Non-Interference (Rules): The fraction of rules in a firewall rule set which do not interfere with other rules.
- Non-Interference (Objects): The fraction of objects in a firewall rule set which do not define the same location or service.
- Simplicity (Rules): The fraction of rules in a firewall rule set which can be triggered.
- Simplicity (Objects): The fraction of objects in a firewall rule set which are defined and used.

Since we have explicitly stated that policy analysis is out of scope for the purposes of our analysis, we will avoid the question of how to measure 'permit only what is required by policy' at this time.

- Consistency (Rules): The fraction of interfering rules in a firewall rule set which have the same action.

Measuring inconsistent object naming style and object template re-use is challenging, and is left to future work.

Based on the above properties, we propose a new metric – *effectiveness* – that can be used to evaluate the complexity of a rule set. This metric essentially captures the degree to which different rules are independent of one another; the intuition is that the greater the overlap, the more complex the rule set and hence more costly to manage. This will also allow us to track the effectiveness of firewalls over time as their rule sets evolve.

- Effectiveness: A measure of the fraction of a given Rule or Object that is not interfered with.

We also investigate other metrics such as frequency of log hits and interference counts, simplicity measures, and consistency measures over rules and objects.

Implementation

Our prototype tool currently handles Checkpoint configurations; the Checkpoint configuration file formats are notably different from the single file, single line style configurations of most other firewalls. We have done proof of concept checks against Cisco PIX/ASA, pf and ipfilter to confirm our ideal state hypothesis, but have not yet implemented parsers for those configurations.

Given two rule sets, the tool produces a comprehensive list of the traffic that one rule set will let through but not the other one. As such, we can use it to compare the existing rule set with a second rule set containing the proposed changes. The administrator can visually check if the difference in patterns of allowed packets corresponds to what he or she intended in proposing the changes.

The tool is implemented in Java SDK 1.6. It can be invoked either from a command line or from a Web interface. This Web interface is implemented using Jetty (Version 6.1). The tool requires the configuration files (object files and rule files) to be transferred to a directory accessible to the firewall analyzer (read only permission is enough). The user can use the web front end to explore the results of the analysis, compare the results of two analyses and run queries via a form interface.

The raw configuration files are parsed using a recursive descent parser that converts the raw configuration files to an intermediate format. The tool interprets rules and objects (represented in this intermediate format) geometrically and computes overlaps between them using computational geometry algorithms. These results are stored in a data structure with multiple indices (rule ids, object ids etc.) so that they are efficiently retrievable by the servlet and query algorithms. (Details in http://www.hpl.hp.com/techreports/2007/HPL-2007-154R1.html .)

Case Studies

The challenges described by firewall managers can be split into three groups – comparison, remediation and reporting.

The anonymized examples below are taken from live production environments, and showcase the use of the tool and metrics which we have described for firewall rule set comparison, remediation and reporting.

Case Study 1: Rule Set Comparison

In this case study, we describe the use of our tool to compare, identify and resolve the differences between two ostensibly identical rule sets.

An end-of-life Checkpoint Firewall-1 NG FP3 firewall was scheduled for replacement with a pair of redundant firewalls running Checkpoint NGX R60, centrally managed by Checkpoint Provider-1.

As the Checkpoint configurations were not compatible between versions, and Checkpoint did not provide a migration tool, a manual rule and object transfer between the old and new environments was required. Since this migration was a bug-for-bug firewall rule set migration, the task of manually re-entering the firewall rules from the old environment to the new environment was delegated to front line support staff, with validation of the copied rules being performed by a senior engineer.

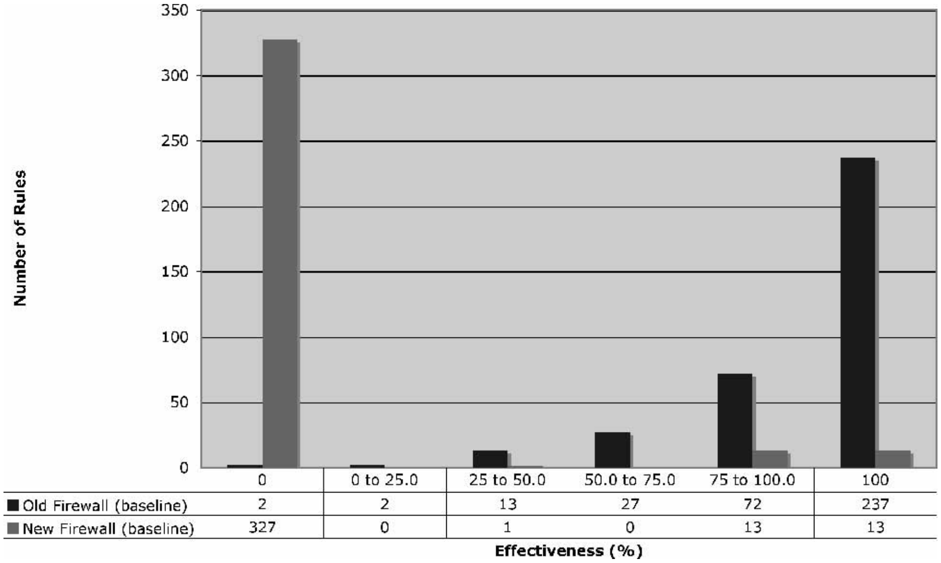
An initial comparison of the old and new rule set rule effectiveness made it immediately obvious that the two configurations were significantly different.

The gross difference in rule sets was swiftly explained by noting that although the total number of active rules was correct (the new firewall has an additional rule for state synchronization), the old configuration had six deny rules, while the new configuration had 12.

A quick visual comparison of the two rule sets also revealed a number of rules that were missing, out of order, with incorrect actions, or missing objects.

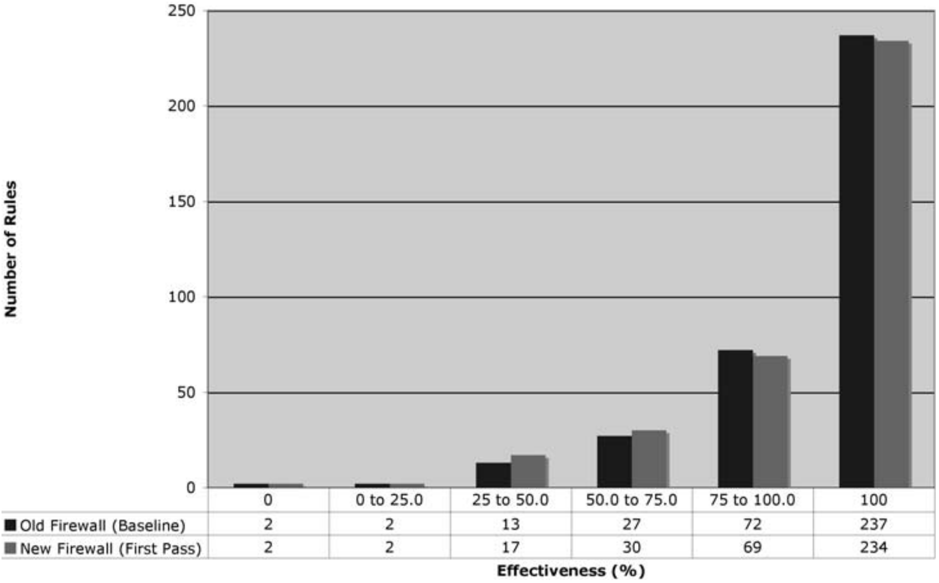
Source	Destination Firewall	New Firewall Action	New Firewall Rule ID	Old Firewall Action	Old Firewall Rule ID
10.0.6.230	192.168.10.21	322	accept	352	drop
10.0.6.231	192.168.10.21	322	accept	352	drop
10.0.6.232	192.168.10.21	322	accept	352	drop
172.16.0.0-172.16.32.20	192.168.11.150	311	accept	352	drop
172.16.0.0-172.16.32.20	192.168.10.150	353	drop	310	accept
172.16.32.22-172.16.35.255	192.168.11.150	311	accept	352	drop
172.16.32.22-172.16.35.255	192.168.10.150	353	drop	310	accept
172.16.64.0-172.16.72.255	192.168.11.150	311	accept	352	drop
172.16.64.0-172.16.72.255	192.168.10.150	353	drop	310	accept
172.16.128.0-172.16.144.255	192.168.11.150	311	accept	352	drop
172.16.128.0-172.16.144.255	192.168.10.150	353	drop	310	accept

Table 1: Case Study 1 – Rule set comparison – Object TCP-3389 interference.



	0	0 to 25.0	25 to 50.0	50.0 to 75.0	75 to 100.0	100
■ Old Firewall (baseline)	2	2	13	27	72	237
■ New Firewall (baseline)	327	0	1	0	13	13

Figure 1: Case Study 1 – Rule set comparison baseline.



	0	0 to 25.0	25 to 50.0	50.0 to 75.0	75 to 100.0	100
■ Old Firewall (Baseline)	2	2	13	27	72	237
■ New Firewall (First Pass)	2	2	17	30	69	234

Figure 2: Case Study 1 – Rule set comparison first pass.

These issues were straightforward to identify, and correct, and would certainly have been identified through manual examination. Once the first pass for gross errors was complete, the effectiveness of both the old and new rule sets was nearly identical.

	Old Firewall (Baseline)	New Firewall (Baseline)
Drop Rules	6	12
Accept Rules	347	342
Total Rules	353	354

Although the configurations appear to be nearly identical from a rule effectiveness standpoint, comparing used objects showed a high number of objects in use in only one of the two configurations – 65 objects unique to the old configuration, and 77 objects unique to the new configuration. 100 objects in the two configurations interfered.

Further, as demonstrated by the object interference example in Table 1, each object could interfere with multiple rules, and either partially or completely.

More specific examination of the interfering objects revealed that the object definitions also varied between the old and new firewalls, and object names had not been consistently defined between the old and new firewalls. Object names had been entered with varying case (TCP vs tcp vs Tcp), different separating characters (TCP-22 vs TCP_22 vs tcp22) and with completely different names (TCP-22 vs SSH).

Resolving the interfering objects then became an iterative process of resolving one set of interference and using the tool to compare the configurations again, as resolving interference in one rule frequently affected interference in other rules.

Case Study 2: Rule Set Remediation

In this case study, we describe the use of our tool in combination with log file analysis to identify partially and completely ineffective rules, and then examine the effect of removing the identified rules from the rule set.

We were asked to identify what rules in a given set of approximately 350 rules were not being used, based on six months of firewall logs, and identify the impact(s) of removing those rules.

The method we selected was a combination of configuration analysis and log analysis. Log analysis was used to identify those rules which had few or no hits during the monitored period, and could be removed, while configuration analysis was used to identify eclipsed rules, and the impact of removing rules.

As shown in Figure 3, log analysis revealed that a sizeable number of rules had received no hits at all during the six month analysis period.

Based on input from the customer and review of the rules, the decision was made to remove all rules which had fewer than 1,000 hits over six months, with the exception of two specified IP ranges.

Based on that specification, a new version of the rule set was created, and the old and new rule sets compared to determine the impact of the proposed changes.

As shown in the figures below, removing the seldom and never used rules resulted in a dramatic reduction in both the number of rules and the number of objects in use.

While the number of rules and objects in the new configurations had decreased significantly, there were still a number of interfering objects and eclipsing rules

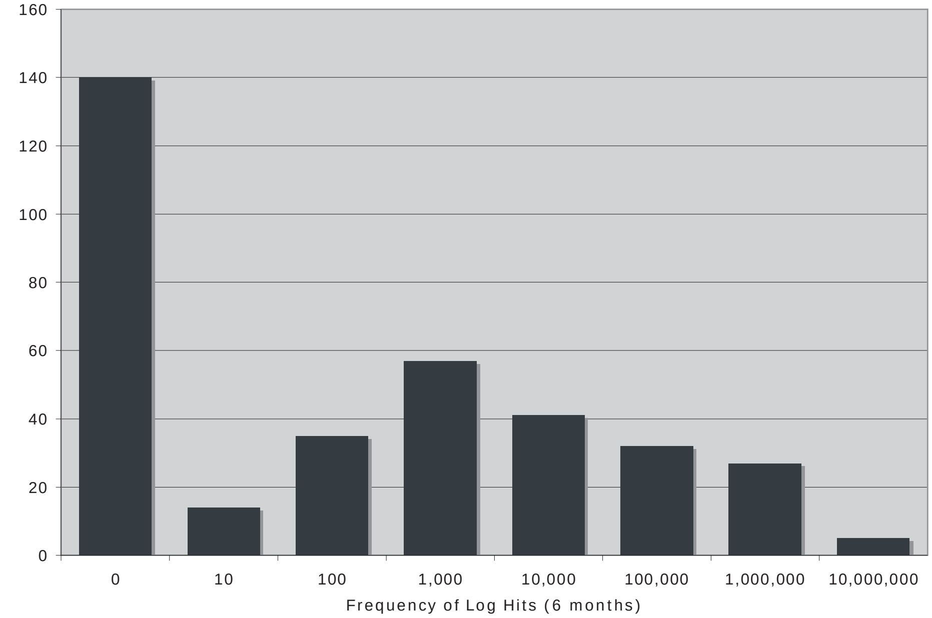


Figure 3: Case Study 2 – Rule set remediation – Rule hit frequency figure.

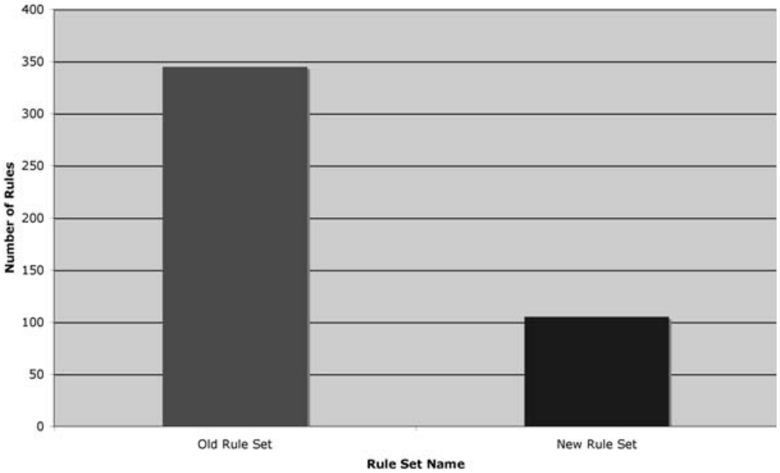


Figure 4a: Case Study 2 – Rule remediation – Number of rules.

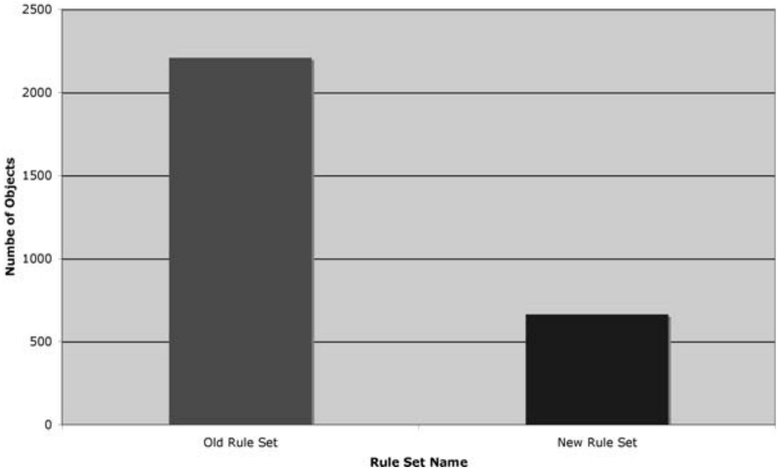


Figure 4b: Case Study 2 – Rule remediation – Number of objects.

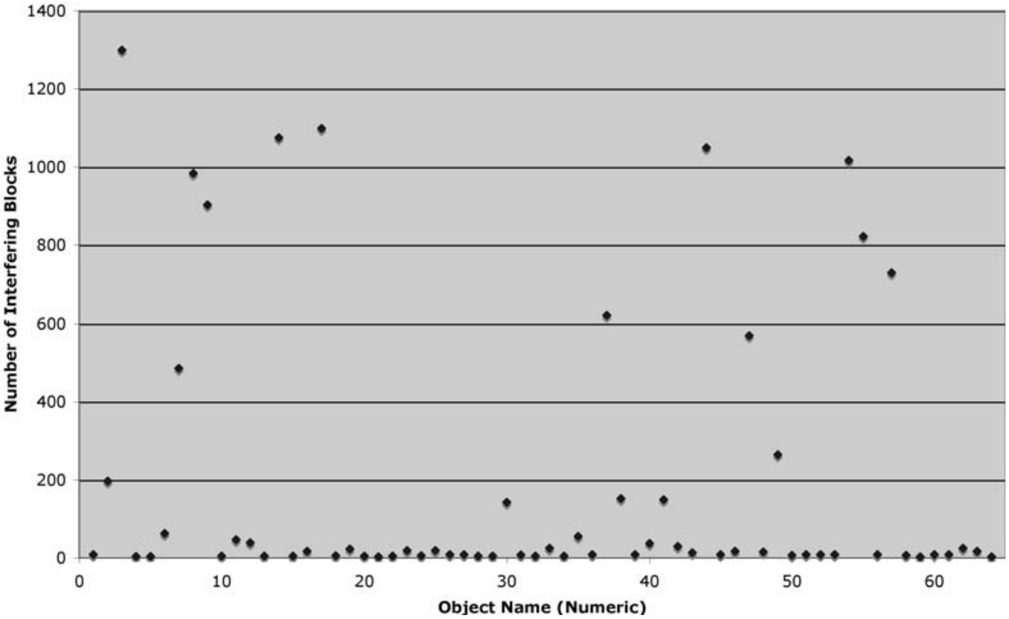


Figure 5: Case Study 2 – Rule set remediation – Number of interfering blocks in rule set.

in the new configuration, pointing towards a number of additional rule set improvements which we did not investigate at that time.

Rule Set Reporting

We previously described six metrics which we believe are a good measure of how well a firewall rule set is managed. In the first four sections we apply these metrics to a collection of 50+ Checkpoint Firewall-1 configurations spanning 2+ years, 34 different firewalls and multiple business types, and discuss the results; the last section looks at a single firewall over time, and through a migration to new hardware.

- Non-Interference (Rules): The fraction of rules in a firewall rule set which do not interfere with other rules.
- Non-Interference (Objects): The fraction of objects in a firewall rule set which do not define the same location or service.
- Simplicity (Rules): The fraction of rules in a firewall rule set which can be triggered.
- Simplicity (Objects): The fraction of objects in a firewall rule set which are defined and used.
- Consistency (Rules): The fraction of interfering rules in a firewall rule set which have the same action.
- Effectiveness: A measure of the fraction of a given Rule or Object that is not interfered with.

Non-Interference

As one might hope, in general, as the number of rules in a firewall rule set increases, the number of

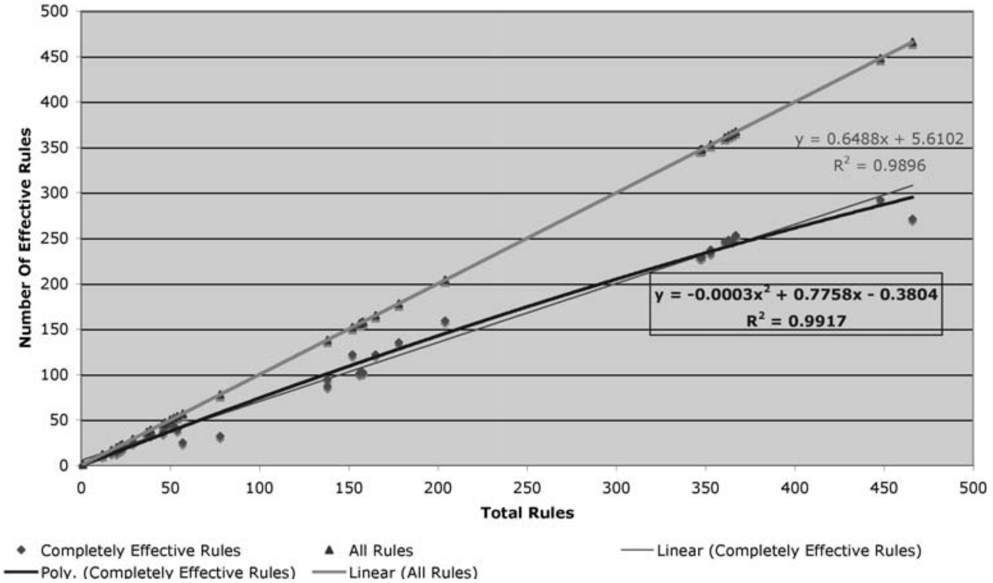


Figure 6a: Case Study 3 – Rule set reporting – Non-interference (rules).

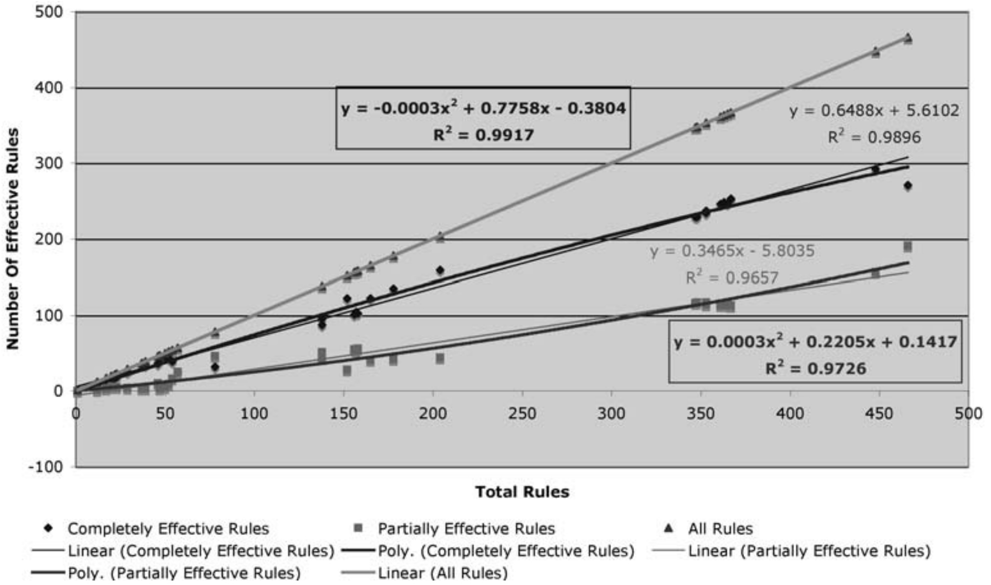


Figure 6b: Case Study 3 – Rule set reporting – Non-interference (rules).

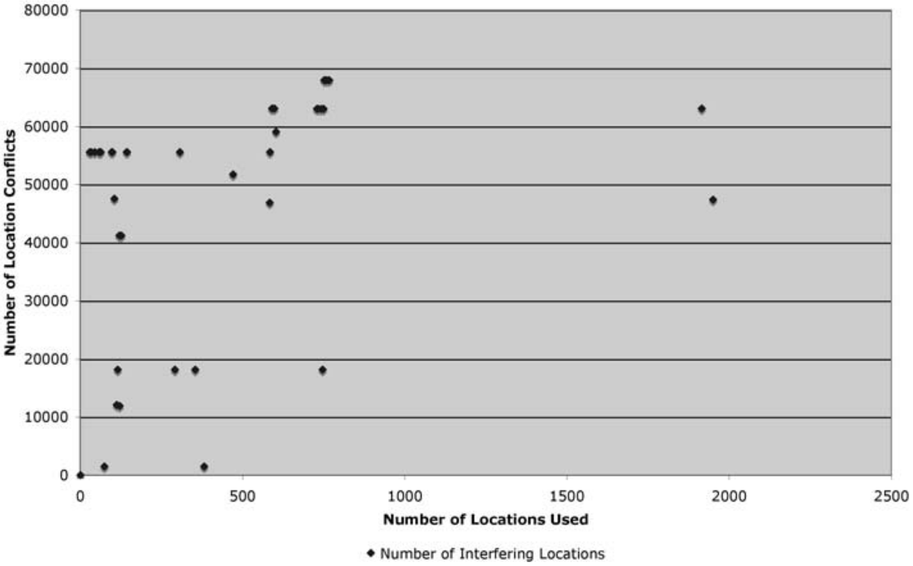


Figure 7: Case Study 3 – Rule set reporting – Non-interference (objects).

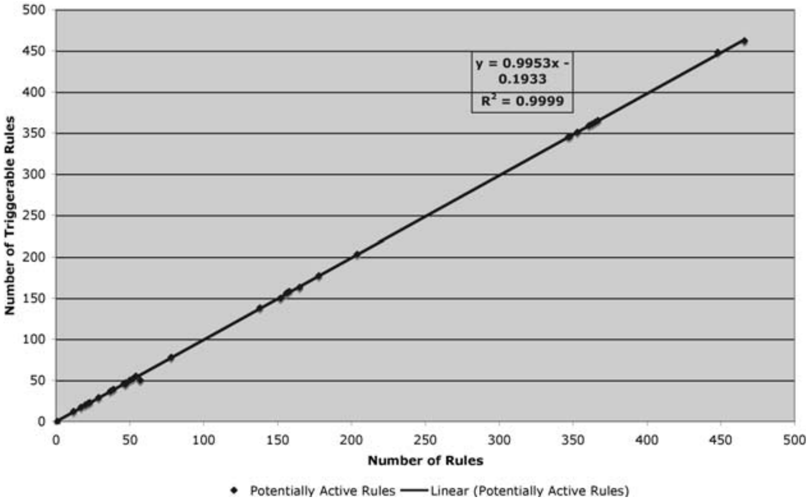


Figure 8a: Case study 3 – Rule set reporting – Simplicity (rules).

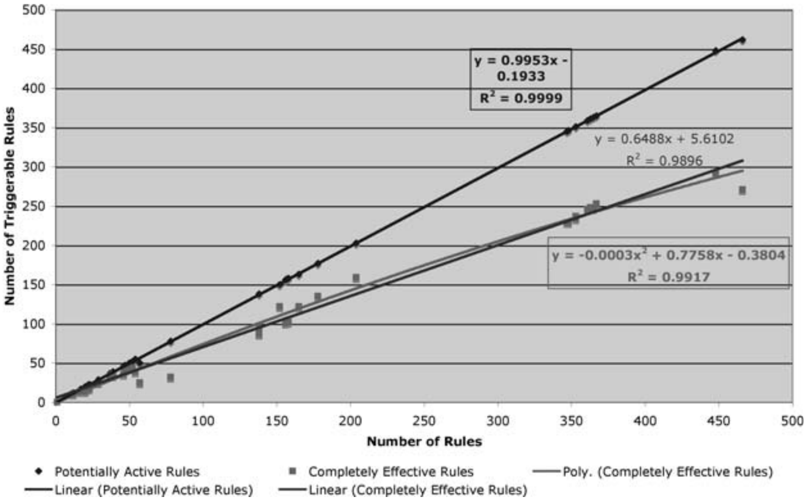


Figure 8b: Case study 3 – Rule set reporting – Simplicity (rules).

non-interfering rules also increases. However, as shown in Figure 6a, as the number of rules increases, the number of non-interfering rules drops away from the total number of rules.

More clearly, Figure 6b suggests that the number of interfering rules is proportionate to, and increases with, the total number of rules. There is also an interesting hint that the number of non-interfering and interfering rules may intersect, and that the number of interfering rules will exceed, and eventually overwhelm the number of non- interfering rules, for a sufficiently large number of rules.

As Checkpoint uses shared object definitions for all firewalls being managed by the same management station, the number of non-interfering objects is heavily dependent on the number of objects defined in that management instance, as the loose plot below demonstrates. It is clear, however, that interference between objects is to be expected in all rule sets, and increases with the number of objects defined on a given common management instance.

Simplicity

If we consider Simplicity of rules strictly from the standpoint of rules which could be triggered – that is to say, any rule which is not completely eclipsed, Figure 8a shows that the relationship between number of rules, and number of potentially active rules is linear, and almost exact.

This is misleading, as Figure 8b shows; as the number of rules increases, the number of completely effective rules immediately drops away from the number of potentially active rules, showing that rules can be expected to interact in unexpected ways. The data obtained for object simplicity is inconclusive, and suggests that our approach for examining objects in Checkpoint configurations managed by a shared management instance needs to be revisited.

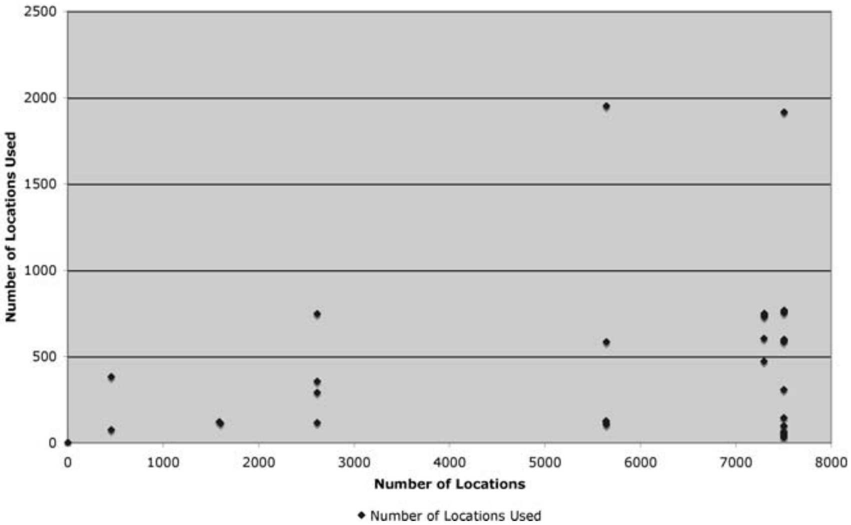


Figure 9: Case Study 3 – Rule set reporting – Simplicity (Objects).

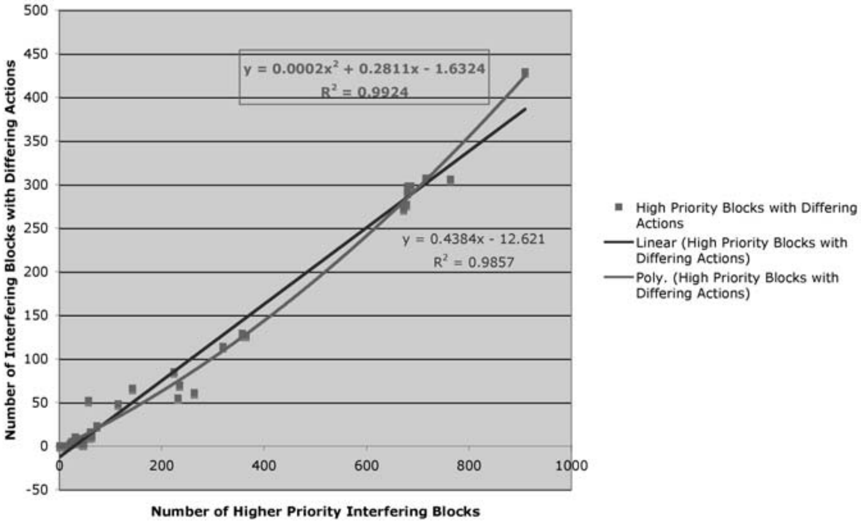


Figure 10: Case Study 3 – Rule set reporting – Consistency.

Consistency

Here we look specifically at the case of interfering rules, where the action of the rule taking priority differs from the action of the rule(s) being eclipsed.

Unsurprisingly, we see that the number of interfering rules with inconsistent actions increases as the size of a rule set increases. We have not calculated the case where a more specific rule is acted on first, and is completely overlapped by a less specific rule separately, but other configuration analysis work suggests that this case does not account for the majority of rule interference.

Rule Effectiveness (%)	R^2 Value (poly)
Ineffective Rules	0.6684
0-25	0.6367
25-50	0.9195
50-75	0.9429
75-100	0.9449
Partially Effective	0.9726
Completely Effective	0.9917

Table 2: Case Study 3 – Rule set reporting – Effectiveness.

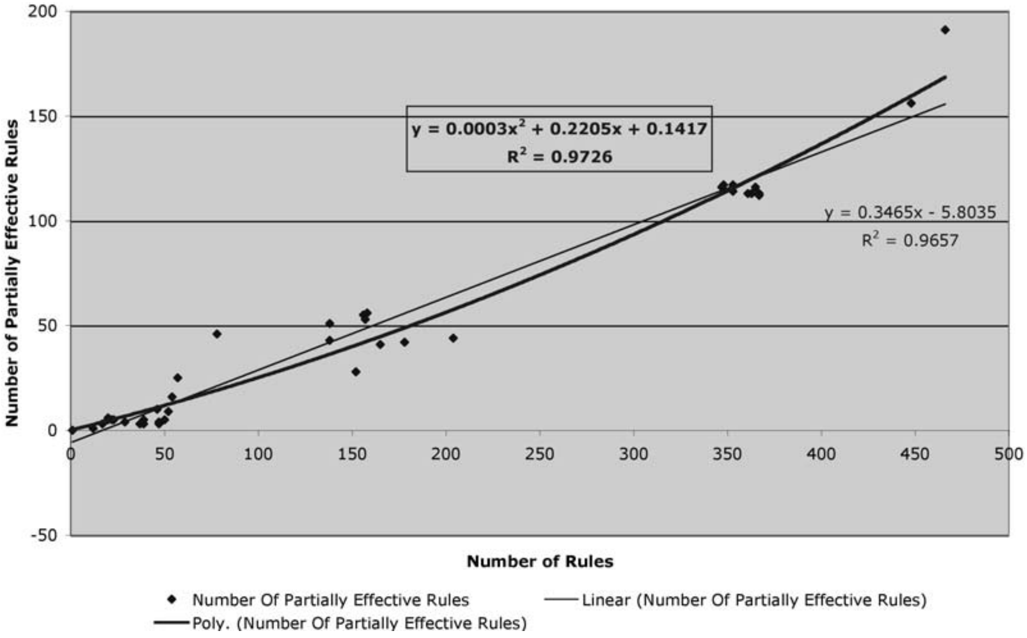


Figure 11a: Case Study 3: Rule set reporting – Effectiveness.

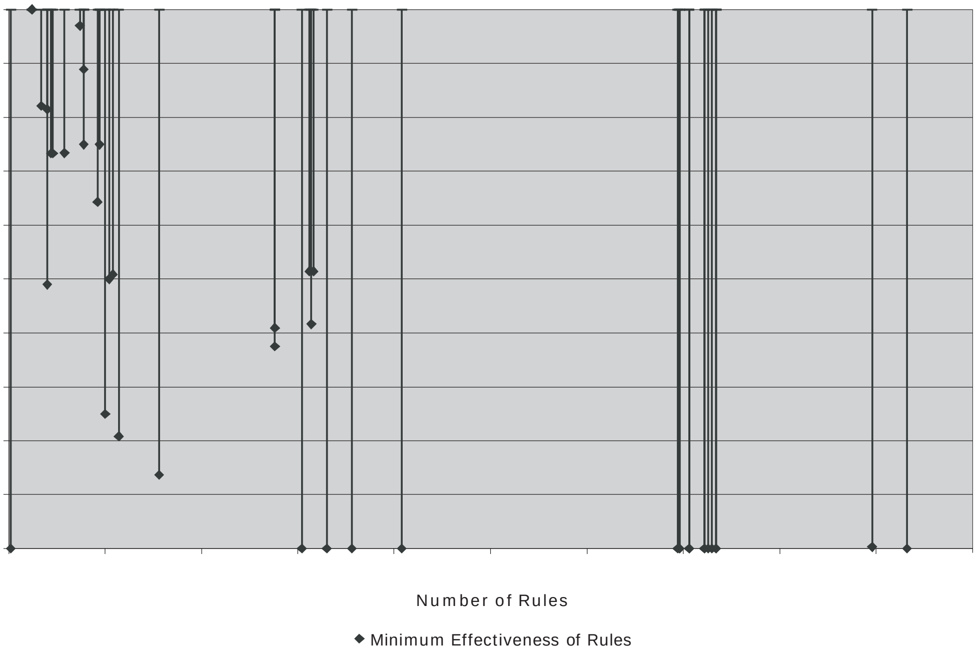


Figure 11b: Case Study 3: Rule set reporting – Effectiveness.

Effectiveness

As shown above, there is a direct correlation between the number of rules, and the number of partially effective rules. Less obviously, the variance in rule effectiveness also increases as the rule set size increases. While the overall R^2 value for partially effective rules is excellent, breaking the rule effectiveness into ranges is suggestive. The lowest correlation values are associated with the most ineffective rules, suggesting that these rules may be easier to detect and resolve.

Trends Over Time

The following data is drawn from an ongoing firewall migration project which has been in progress for nearly two years, and continues onward with an extended period of overlapping operation for the old and new firewalls.

When the configuration was initially migrated from the old firewall to the new firewall, a manual cleanup of the rule set took place; currently most rule set changes are expected to be implemented on both the old and new firewalls.

Unsurprisingly, we see that both the old and new rule sets show a steady increase in the number of rules and locations defined over time

Similarly, we also see the number of partially effective rules increasing over time, as the number of rules in the rule set increases.

The number of partially effective rules in the new firewall rule set is initially lower than the number of partially effective rules in the old firewall rule set, thanks to a manual clean up of the rule set, prior to migration. It is abundantly clear, however, that the effect of the rule set clean up was only temporary.

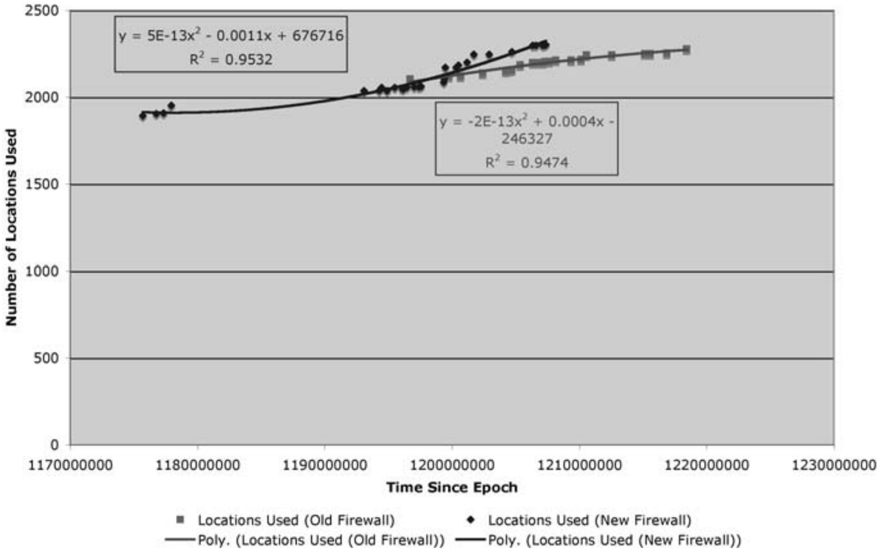


Figure 12a: Number of locations used over time.

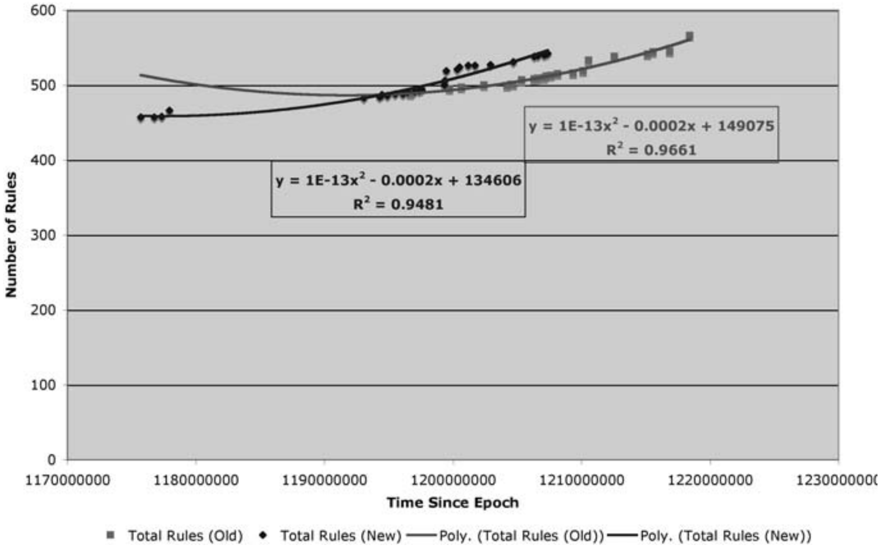


Figure 12b: Number of rules over time.

Further, if we examine Figure 13b, it is clear that the majority of rule changes which result in partially effective rules are for rules in the 75-100% effectiveness range.

Discoveries

Our findings in this work range from confirming relatively obvious intuitions (such as simpler configurations are better) to some more surprising results. Our discoveries here summarize the case studies, as well as additional experiences not described in this paper.

1. The number of rules in a rule set is highly correlated to the number of partially and completely effective rules. This suggests that our intuition that larger rule sets are more complex is likely to be correct.
2. There is a high correlation between the number of interfering rules, and the number of interfering rules with conflicting actions. While it is likely that some of these interfering rules have been knowingly configured (e.g., a permit rule for a host in a larger subnet which is denied), in practice it appears that most interfering rules are unintended.
3. The majority of partially effective rules are 25-99% effective. Further, when a rule set undergoes a manual clean up process, the number of completely ineffective rules is typically reduced dramatically, while the number of partially effective rules remains relatively unchanged. This suggests that complex interference patterns are more difficult for people to detect and resolve.
4. The effectiveness of a rule set decays visibly over time. If a rule set is cleaned up, the effectiveness immediately begins to decay again.
5. Most firewall rule sets have some amount of interference, but the amount of interference

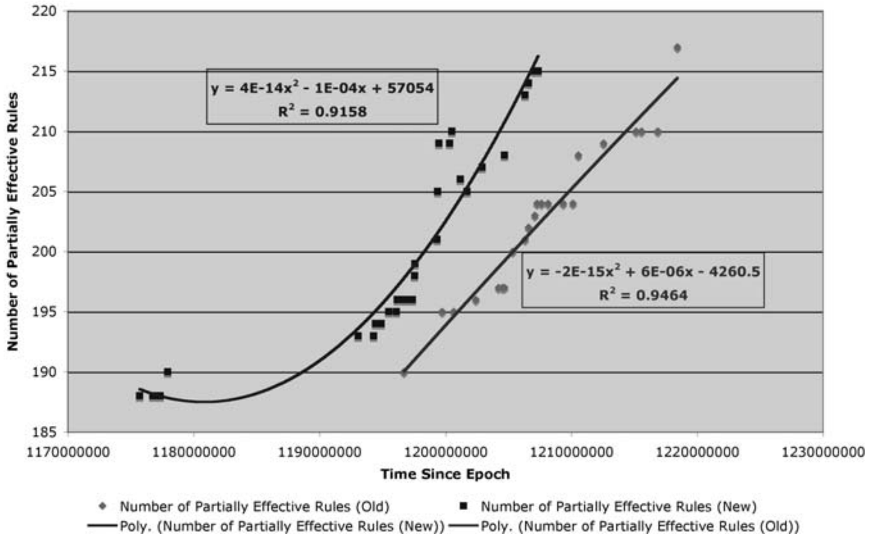


Figure 13a: Trends over time – Rule interference.

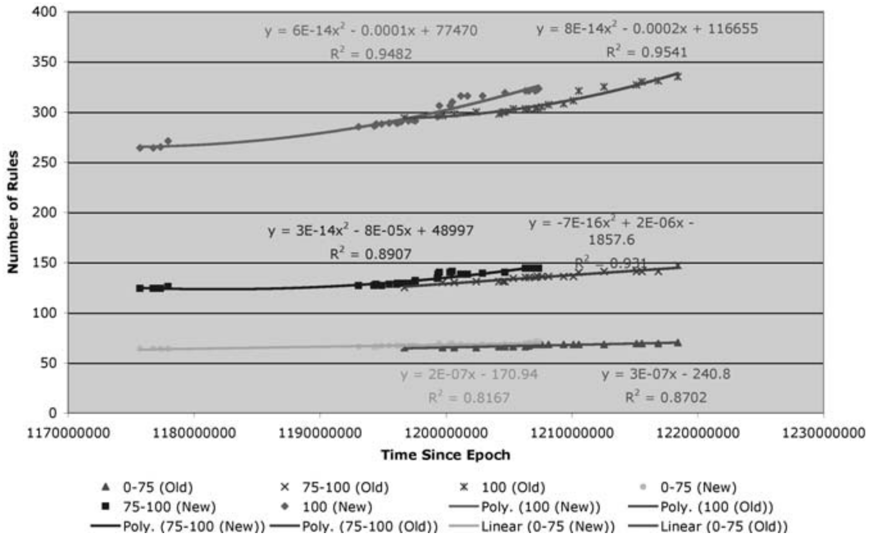


Figure 13b: Trends over time – Distribution of paritally effective rules.

varies dramatically, and increases as number of rules increases.

6. While effectiveness varies dramatically, in general, more effective rule sets have less variance in effectiveness, suggesting that less confusing rule sets are easier to manage.
7. In general, in the absence of a major clean up effort, the number of rules and objects used in a given rule set always increases. Since the number of partially and completely effective rules also increases as the number of rules increase, this points to the management of firewall rule sets as write-once, remove-never devices.
8. Even when a rule set is cleaned up, the number of rules decreases, but the number of objects remains constant or increases, suggesting that object management is an ongoing challenge.
9. There is no clear relationship between the number of rules and number of locations.

Unfortunately, it appears that objects are a weak source for information about the state of rule sets, at least as we are currently measuring and examining objects. It is likely that the use of shared objects by all firewalls managed by a given Checkpoint management instance is the cause of this issue, and we hope to better address the role of objects as a means of measuring firewall rule sets in the future.

Ultimately, our holy grail is to be able to correlate the above metrics with quantities dear to higher level management such as cost, 'ease of management' etc. We need to measure these 'manager-friendly' metrics to see what configuration and network metrics correlate with them – this is a part of our ongoing efforts.

Usage

The tool and metrics we describe are extremely useful for rule set comparison and clean up, as they automate the process of finding conflicts, vastly reducing the amount of time and effort required.

As well as being used for massive rule set remediation projects, we suggest that an ideal usage would be to include rule set analysis as a part of routine change management, to identify unanticipated effects from rule and objects additions and removals.

We also hope that our metrics work will serve as a measure of health for firewalls – and can drive decisions like "when should I clean up my rule set?" or "Is my outsourced firewall administrator doing their job properly?"

Future Work

Going forward, we intend to improve our understanding of firewall management. Specific areas of interest include:

- Analyzing configurations from a wider variety of firewalls to discover commonalities and consider additional metrics.
- Tracking the effect of using the metrics we have described to improve firewall management over time.
- Correlating the metrics we have described to the rate of change, number of incidents, and time to resolve incidents in the environment.
- Correlating the metrics we have described to cost.

Related Work

There has been a flurry of work on firewall analysis in recent years. The most relevant is the Firewall Analyzer (FA) product from Algorithmic Security Inc [ALG] that builds on previous research reported in [MWZ] and [WOO]. FA analyzes a number of different types of firewalls, reports common vulnerabilities found in the rule set, and detects rules that are redundant (i.e., are eclipsed by higher-priority rules). FA does not check equivalence of rule sets.

The Solsoft Firewall Manager [SOL] converts high-level specifications of allowed and prohibited traffic into firewall rule sets. However, it does not analyze existing rule sets on a firewall to see if they comply with the high-level specifications.

There are also a number of academic papers on firewall analysis. The paper [AH1] recognizes different types of conflicts that can occur between a pair of rules, but does not address the more general problem of one rule being eclipsed by a set of rules. The authors extend their study to rules on multiple firewalls in a tree network [AHB], but the results are again limited to interactions between pairs of rules. The paper [GLI] proposes a simplistic way to design rule sets such that every packet is associated with exactly one rule; the problem with this approach is that one can design much more compact rule sets if multiple rules (resolved by a priority mechanism) can apply to any packet. The paper [EMU] investigates efficient data structures to process basic firewall queries.

The paper [VPR] addresses problems of generating and analyzing rule sets for networks with multiple firewalls and addresses the problem of checking equivalence of rule sets. However, the paper relies on exhaustive analysis of all possible packets, and it is unclear whether the methods scale to large networks. The paper [BRR] describes a tool to configure and analyze rule sets for networks with multiple firewalls. It employs efficient algorithms that scale for large networks, and while the techniques are relevant for checking rule set equivalence, the paper does not explicitly address the equivalence problem.

The paper [MKE] proposes the use of binary decision diagrams to analyze rule sets. These diagrams allow you to query the firewall rule set but do not easily support the comparison of two rule sets.

Conclusions

Offline configuration analysis is a fast, lightweight, and reliable way to identify the effectiveness of existing firewall configurations, improve maintainability, hasten debugging, and assist with policy, audit and compliance.

Author Biographies

Sandeep Bhatt is a researcher in the Systems Security Laboratory at HP Labs. Since joining HP Labs in 2004, he has focused on the management of distributed access control in enterprise applications, on algorithms for firewall analysis, and on end-to-end access control in enterprise networks. Before joining HP Labs, Sandeep led the Systems Performance team at Akamai Technologies, and was Director of Network Algorithms at Telcordia Technologies where he led the development of fault diagnosis systems for large-scale telecommunication networks. Sandeep was also on the faculty of Computer Science at Yale University, with appointments at Caltech and Rutgers University. His research interests have included algorithms for parallel computation, network communication, and VLSI layout. He received Ph.D, S.M and S.B degrees in Computer Science from MIT.

Cat Okita has more than 10 years of experience as a senior systems, security and network professional in the Financial, Internet, Manufacturing and Telecom sectors. She has designed, managed and contributed to a variety of geographically diverse projects and installations. Her assignments have included proposing and seeing to completion a wide variety of security, Internet, enterprise and monitoring projects, including highly available, redundant fault tolerant systems for security, web services, logging, monitoring, statistics gathering and analysis. Her research interests include privacy, reputation and practical security.

Cat has spoken at LISA and Defcon about identity and reputation, co-chairs the 'Managing Sysadmins' workshop at LISA, and programs for fun, in her spare time.

Prasad Rao is a member of the Systems Security Laboratories within HP labs, where he works in security analytics and metrics. At HP he has been active in the Vantage project that analyzes end-to-end access control across various layers in the enterprise. As a part of this project, he has built technologies to analyze firewalls with very large rule sets (more than 7500 rules) and objects (more than 12000 objects) for presence of operator errors and vulnerabilities – this technology is currently being tested within HP. In addition he has contributed to ongoing work in other areas such as role discovery, securing printing software and privacy compliance.

At Telcordia Technologies he was the technical lead and architect of the Smart Firewalls project (a part of the Dynamic Coalitions Program). Additionally, at Telcordia technologies he built a deductive rule-based system in large scale worker scheduling programs, which had to optimize worker utilitization and cost under arbitrarily stated union rules and practical constraints.

For his Ph.D. thesis in logic programming and deductive databases at the State University of New York at Stony Brook, Dr Rao built designed and implemented the tabling engine of the XSB logic programming language – which set speed records for standard benchmarks in the field of deductive databases compared to the state of the art in the early '90s.

Bibliography

[ALG] Algorithmic Security Inc., *Firewall Analyzer: Make Your Firewall Really Safe*, white paper, 2006, http://www.algosec.com .

[AH1] Al-Shaer, E. and H. Hamed, "Firewall Policy Advisor for Anomaly Discovery and Rule Editing," *Proceedings IEEE/IFIP Integrated Management Conference*, 2003.

[AHB] Al-Shaer, E., H. Hamed, R. Boutaba, and M. Hasan, "Conflict Classification and Analysis of Distributed Firewall Policies," *JSAC*, 2005.

[BRR] Bhatt, S., S. Rajagopalan, P. Rao, "Automatic Management of Network Security Policy," *Proceedings MILCOM*, 2003.

[EMU] Eppstein, D. and S. Muthukrishnan, "Internet Packet Filter Management and Rectangle Geometry," *Proceedings ACM SODA*, 2001.

[GLI] Gouda, M. and X-Y. Liu, "Firewall Design: Consistency, Completeness and Compactness," *Proceedings IEEE International Conference on Distributed Computing Systems*, 2004.

[MKE] Marmorstein, R. and P. Kearns, "An Open Source Solution for Testing NAT'd and Nested iptables Firewalls," *Proceedings LISA '05*, 2005.

[MWZ] Mayer, A., A. Wool and E. Ziskind, "Fang: A Firewall Analysis Engine," *Proceedings IEEE Symposium on Security and Privacy*, 2000.

[SOL] Solsoft Inc., http://www.solsoft.com .

[VPR] Verma, P. and A. Prakash, "FACE: A Firewall Analysis and Configuration Engine," *IEEE Symposium on Applications and the Internet*, 2005.

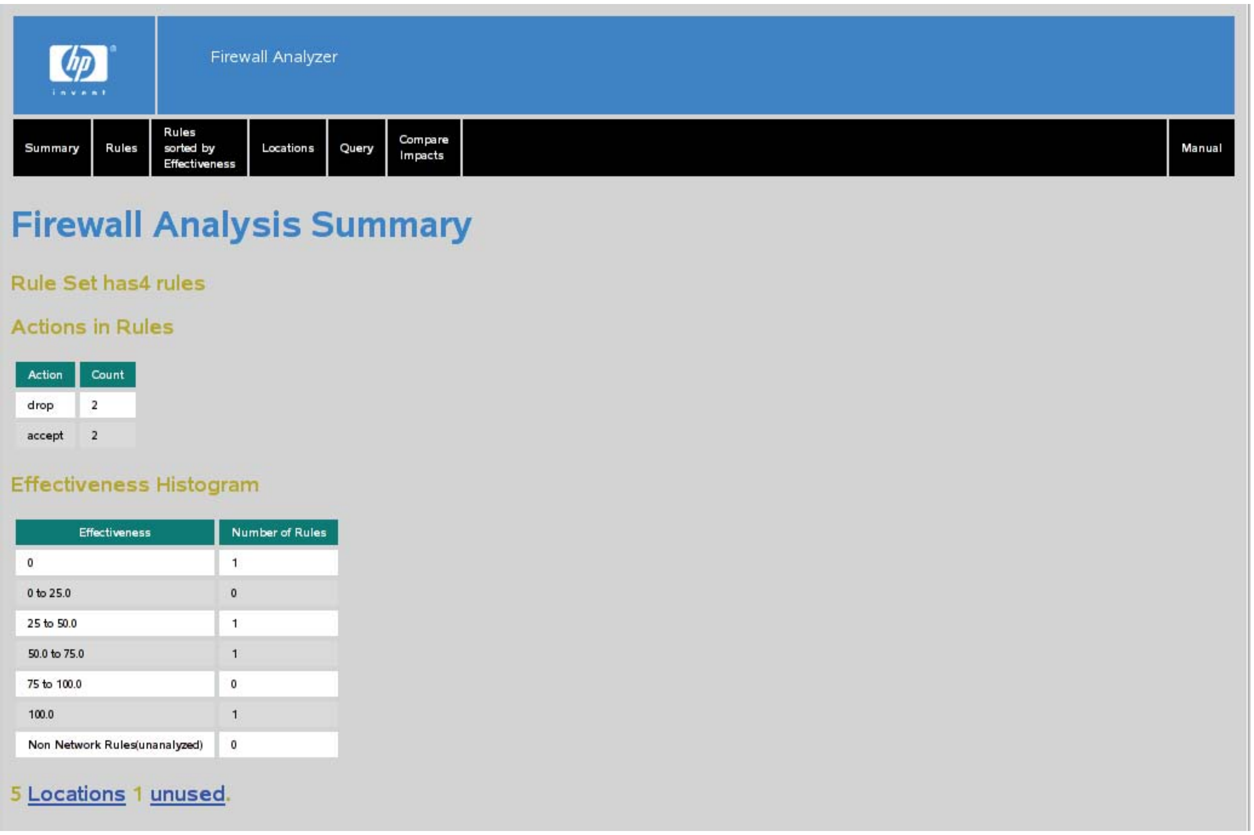
[WOO] Wool, A., "Architecting the Lumeta Firewall Analyzer," *Proceedings 10th USENIX Security Symposium*, 2001.

Appendix 1

To briefly demonstrate the tool, consider the following first match firewall rule set, which contains several errors:

Action	**Service**	**Source**	**Destination**
Permit	SSH	10.0.0.0/8	10.0.0.0/8
Deny	SSH	10.0.0.0/8	10.3.1.0/24
Permit	SSH,https	10.0.0.0/8	10.3.1.61/32
Deny	*ANY*	*ANY*	*ANY*

The tool produces the following overall summary:



Firewall Analysis Summary

Rule Set has4 rules

Actions in Rules

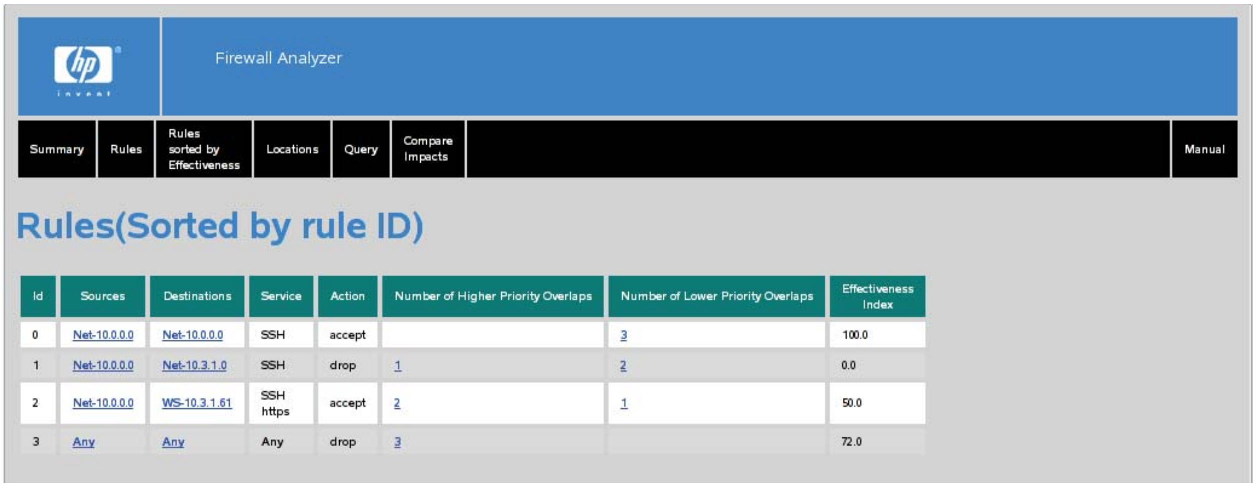
Action	Count
drop	2
accept	2

Effectiveness Histogram

Effectiveness	Number of Rules
0	1
0 to 25.0	0
25 to 50.0	1
50.0 to 75.0	1
75 to 100.0	0
100.0	1
Non Network Rules(unanalyzed)	0

5 Locations 1 unused.

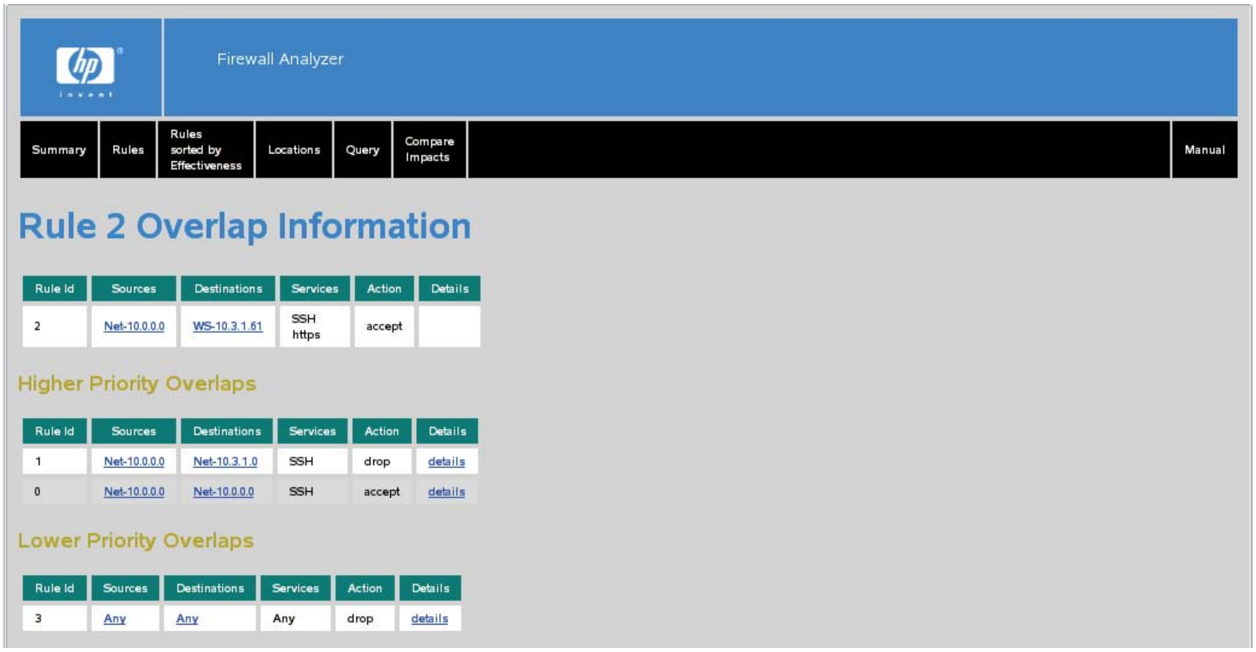
From the Summary, we can then obtain more specific information about the rule set...



Rules(Sorted by rule ID)

Id	Sources	Destinations	Service	Action	Number of Higher Priority Overlaps	Number of Lower Priority Overlaps	Effectiveness Index
0	Net-10.0.0.0	Net-10.0.0.0	SSH	accept		3	100.0
1	Net-10.0.0.0	Net-10.3.1.0	SSH	drop	1	2	0.0
2	Net-10.0.0.0	WS-10.3.1.61	SSH https	accept	2	1	50.0
3	Any	Any	Any	drop	3		72.0

... and we can drill down for detail about the overlapping rules:



Rule 2 Overlap Information

Rule Id	Sources	Destinations	Services	Action	Details
2	Net-10.0.0.0	WS-10.3.1.61	SSH https	accept	

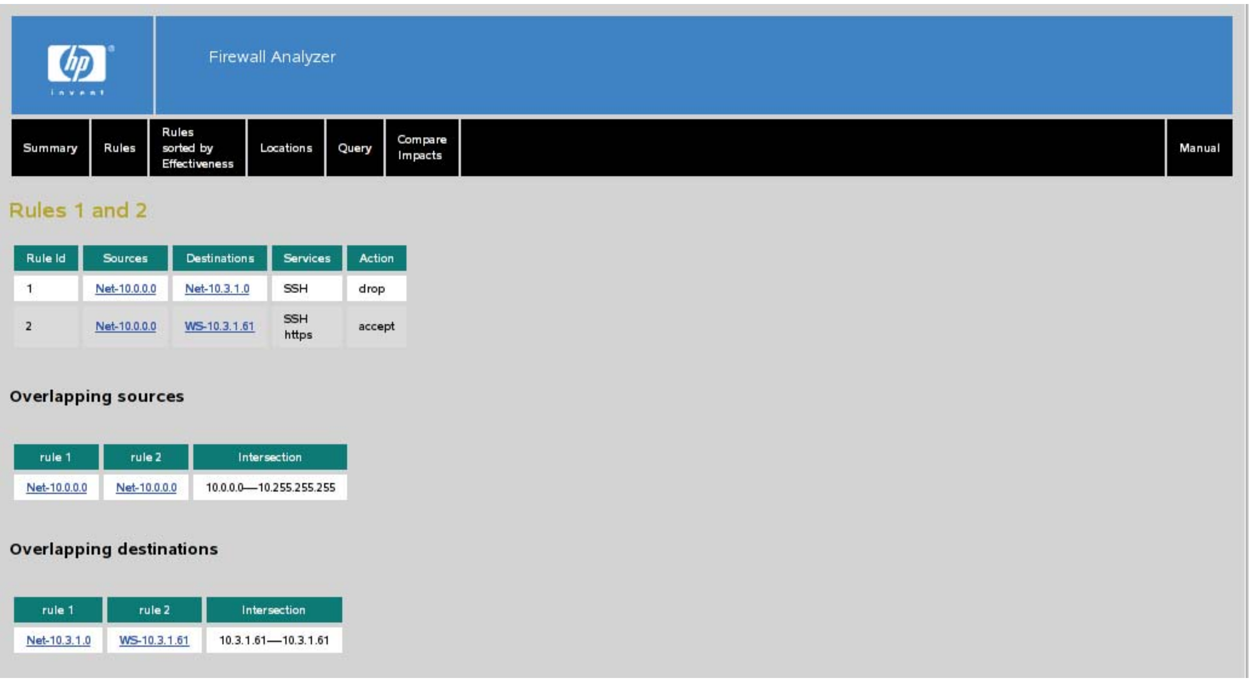
Higher Priority Overlaps

Rule Id	Sources	Destinations	Services	Action	Details
1	Net-10.0.0.0	Net-10.3.1.0	SSH	drop	details
0	Net-10.0.0.0	Net-10.0.0.0	SSH	accept	details

Lower Priority Overlaps

Rule Id	Sources	Destinations	Services	Action	Details
3	Any	Any	Any	drop	details

... and finally procure further details about the specific overlaps:



Rules 1 and 2

Rule Id	Sources	Destinations	Services	Action
1	Net-10.0.0.0	Net-10.3.1.0	SSH	drop
2	Net-10.0.0.0	WS-10.3.1.61	SSH https	accept

Overlapping sources

rule 1	rule 2	Intersection
Net-10.0.0.0	Net-10.0.0.0	10.0.0.0—10.255.255.255

Overlapping destinations

rule 1	rule 2	Intersection
Net-10.3.1.0	WS-10.3.1.61	10.3.1.61—10.3.1.61