Internet Firewall Vulnerability Analysis Method

Cho Hong, LING
Department of Computer Science, University of Auckland
clin123@ec.auckland.ac.nz

Abstract

Firewall is the main defence of our network and is no guarantee the present of hidden vulnerabilities. Under there is no well-defined general methodology for testing the firewalls, this is always a big headache for the administrators. Two firewall testing approaches are being compared and the correctness of the testing result is reported as well.

1. Introduction

As the Internet is growing in an incredible speed, the number of attacks on the network is also raising rapidly according to the Incidents report from CERT [6]. Firewall is the main and first layer of defence of our network. However, it does not guarantee a perfect protection for the network. Number of vulnerabilities is found in the firewall regularly [6]. Therefore, a thorough test for the firewall should be performed regularly. Unfortunately, there is no well-defined and effective general methodology for testing the firewall vulnerabilities. This paper is going to compare two methodologies on firewall vulnerability analysis and define a new method to compare the results from different analysis approach. It first introduces two firewall testing approaches and followed by a comparison. Finally, a new method is defined to show the correctness of the results.

2. Methodology

Different approaches use to analyse firewall produce results in different prospective and limitation. Since there are different types of firewall product, some approaches may only be able to analyse certain type of firewall. A general mechanism for firewall vulnerability analysis is needed. This section is going to compare two firewall testing approaches.

2.1 Counting Configuration Errors

In [1], a set of Check Point FireWall-1 [4] sample rule sets are collected from the real operating firewalls at different organizations. The rule sets are being analysis to identify possible errors according to the twelve categories of configuration errors, which is conducted by the author. A plot is produced from the processed data, which display the order of different configuration errors, as show in Figure 1 [1]. (Please refer to Appendix A for the detail of the twelve configuration errors.)

Figure 1 displays how often a certain configuration error is found in the sample rule sets [1]. Obviously, error 12: Any destination on outbound rules, is the most often configuration error made by the administrator while error 2: DNS-TCP implicit rules, is the least often error to be made. This figure is useful to get an idea on which kind of errors is usually present in the firewall configuration. However, it does not show the errors in the other categories of firewall vulnerability, like design and implementation errors, and only applies to the Check Point FireWall-1 [4] firewall product.

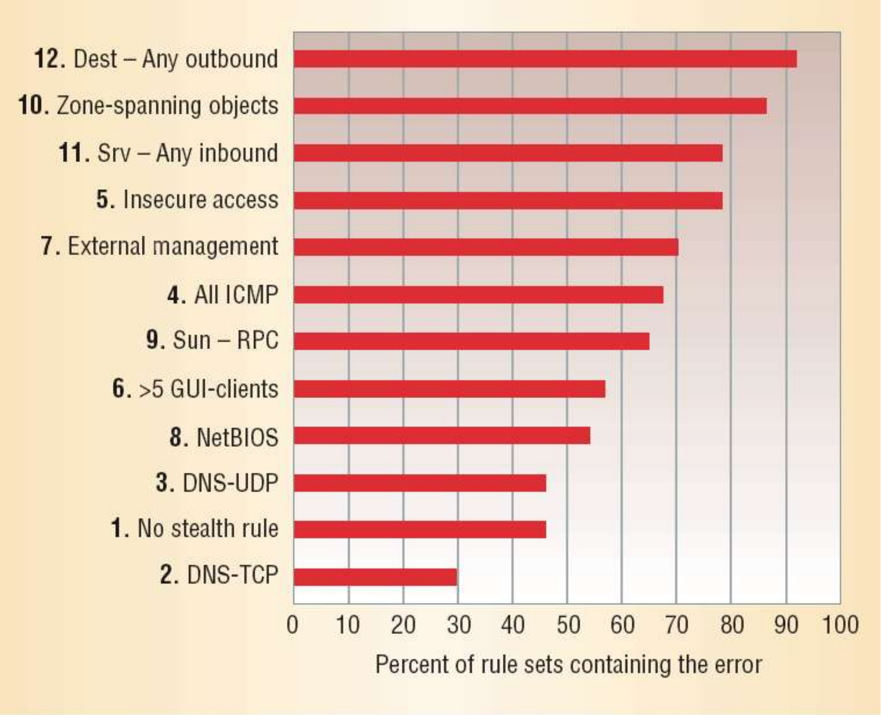

Figure 1. Distribution of configuration errors. [1]

2.2 The Matrix

Compare with [1], [2] uses a very different approach to analyse the vulnerabilities in the firewall. A set of two-dimensional related matrices is being constructed to analyse the vulnerabilities in the firewall, and the resulting matrices provide various perspectives on firewall vulnerabilities.

The matrices are constructed by twenty known firewall vulnerabilities from the vulnerability databases and reports including different type of firewall products. This is fundamentally different from the approach used in [1], which make an analysis on the collected raw data from a single firewall product. Therefore, the matrices can be used to analyse different type of firewall products, instead of the product specific approach used in [1].

The row and column of the three matrices in [2] are the combination of three main attributes:

- Firewall operations: It includes different type of firewall operations, which is described in the firewall data flow model in [3].
- Vulnerability causes: It consist of a set of taxonomy used in [2], which is a summarization of [5] taxonomy with minor changes. It categorizes different type of errors in the firewall that cause vulnerability.
- Vulnerability effects: It is a set of taxonomy used in [2], which is presented in [5]. It categorizes different type of effects or attacks to the firewall that is caused by the vulnerability in the firewall.

(Please refer to Appendix A for the detail of each attribute.)

These three attributes generate three matrices with different combination, as illustrate in Figure 2. The first matrix is cross reference of firewall operations with vulnerability causes. Each cell is filled with the classified vulnerabilities according to the analysis to the twenty collected firewall vulnerabilities from the database.

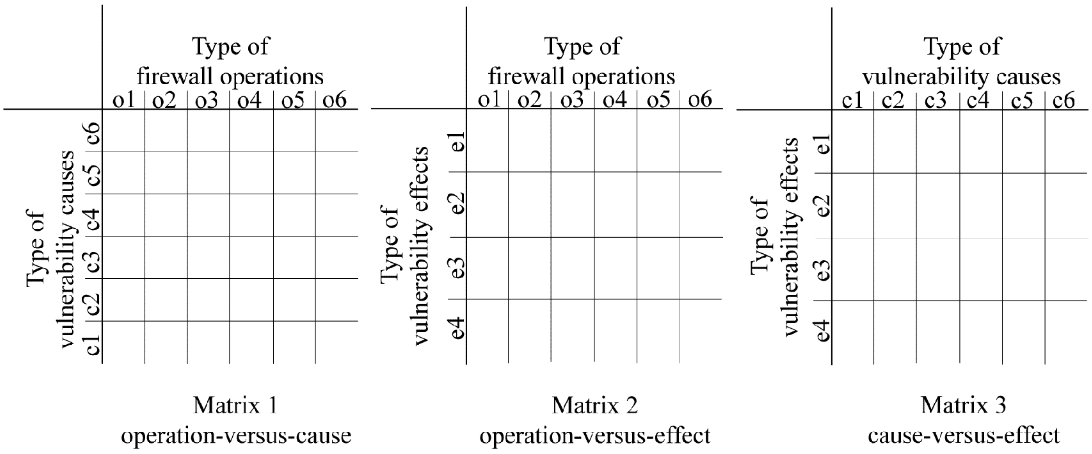

Figure 2. The Three matrices

The first matrix displays the type of firewall operation that is most vulnerable to, as well as which kind of errors is most often found in certain type of firewall operation

[2]. It provides the ability to make a two-way analysis between the attributes. Moreover, three related matrices are being used to do the analysis. The second matrix is cross references of firewall operation with vulnerability effects while the third matrix is cross references of vulnerability causes with vulnerability effects, as illustrate in Figure 2.

2.3 Check Point Firewall-1 Analysis

The testing methodology in [2] comprises two phases, the modeling phase and the ranking phase. In the modeling phase, the testing firewall is mapped into an operation model according to the firewall data flow model [MEFS 01]. This divides the firewall into several layers, each with different operations, as show in Figure 3. It provides an in sign of how the packet is being processed by the firewall and yield a better chance to identify the potential weaknesses of the firewall.

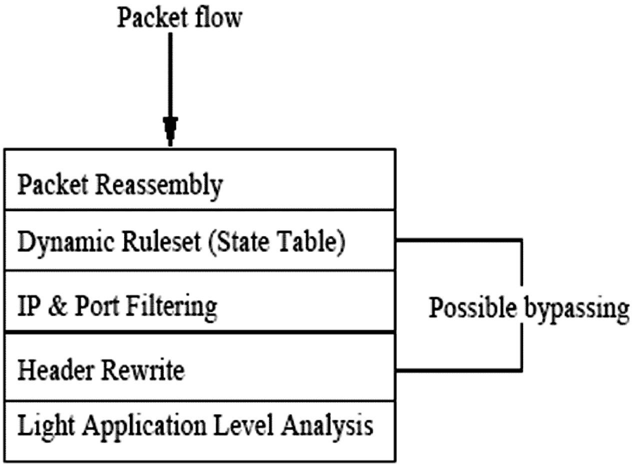

Figure 3. Firewall-1 operation [2]

The ranking phase produces the operation and cause ranking to display the importance of different category firewall operations and vulnerability causes according to the ranking. First, each vulnerability effects will assign a weight base on the effect of the attack to the testing firewall. For Firewall-1, here is the weight for each vulnerability effects:

- “Execution of code: 3
- Change of target: 2
- Access to target: 1
- DoS: 4” [2]

The operation ranking is using the operation-versus-effect matrix to calculate the importance of each firewall operation. The score of each firewall operation is calculated by:

Score of operation = Sum of the cells in the column of particular operation (Number of vulnerabilities in the cell x Weight of the vulnerability effect)

The resulting operation ranking in decreasing order of importance:

1. “IP and port filtering
2. Legality checks
3. Application level
4. Packet reassembly
5. Dynamic rule-set
6. NAT/PAT” [2]

The cause-versus-effect matrix is used to rank the vulnerability causes. The score of each vulnerability cause is calculated by:

Score of vulnerability cause = Sum of the cells in the column of particular vulnerability cause (Number of vulnerabilities in the cell x Weight of the vulnerability effect)

Result of the vulnerability cause ranking, in decreasing order of importance:

1. “Validation
2. Design
3. Boundary checking
4. Authorization, serialization, aliasing, and domain” [2]

By combining the results of these two ranking, an order for testing errors can be established. It shows us the greatest possible place of having errors is in the operation of IP and port filtering with vulnerability cause of Validation, followed by Design, Boundary checking, Authorization, serialization, aliasing, and domain. The second greatest possible place is in the operation of Legality checks with various vulnerability causes according to the ranking of it, followed by the Application level, Packet reassembly, Dynamic rule-set and NAT/PAT operations [2].

2.4 Comparison

The methodology for testing the firewall is very different between [1] and [2]. [2] cover wider category of firewall vulnerabilities with less detail while [1] only cover one category with more detail. The result of [2] provides a direction to identify the possible vulnerabilities in the firewall as well as the types of attacks that the administrator should be aware. The matrices show the vulnerability trends and help administrator to predict where the new possible vulnerabilities is [2]. However, [1] display the possible errors in more detail by stating the exact configuration errors, which let administrator easier to figure out what the error is. The main advantage of the methodology in [2] is a proactive approach is used rather then the reactive approach used in [1].

3. Compare the results

This section attempts to compare the result from these two papers [1], and [2] by defining a new method. Although the comparison can only apply to a part of the results, due to the vast differences in two approaches, it still can display some similarities of the results. The first step is mapping the twelve configuration errors in [1] into the vulnerability causes taxonomy in [2], then compute the weight for each Vulnerability causes by the sum of percentage of each configuration errors show in Figure 1. The following table display this approach.

Vulnerability causes	According Configuration errors	Weight
Validation error	Error 12: Any destination on inbound rules Error 11: Any service on inbound rules	92 (Error 12) + 88 (Error 11) = 180
Authorization error	Error 5: Insecure firewall management Error 1: No stealth rule	88 (Error 5) + 46 (Error 1) = 134
Serialization/Aliasing error	Error 6: Too many management machines	56 (Error 6)
Boundary checking error	Error 10: Zone-spanning objects Error 4: All ICMP	87 (Error 10) + 67 (Error 4) = 154
Domain error	Error 7: External management machines Error 9: Portmapper/Remote Procedure Call service	70 (Error 7) + 64 (Error 9) = 134
Weak/Incorrect design error		0

Other error	Error 8: NetBIOS service Error 3: DNS over UDP Error 2: DNS over TCP	53 (Error 8) + 46 (Error 3) + 30 (Error 2) = 129

A vulnerability causes ranking is produced according to the computed weight, in decreasing order of importance:

1. Validation
2. Boundary checking
3. Authorization, and domain
4. Others
5. Serialization/Aliasing

The following table display a side-by-side comparison of the vulnerability causes ranking between [1], and [2].

Vulnerability causes ranking in [2]	Vulnerability causes ranking for [1]
1. Validation 2. Design 3. Boundary checking 4. Authorization, serialization/aliasing, and domain	1. Validation 2. Boundary checking 3. Authorization, and domain 4. Others 5. Serialization/Aliasing

The above table indicate the similarity and differences of the results and approaches in these two papers. The approach used by [1] cannot address the design error in the firewall, due to it only analyses the configuration rule-set of the firewall. There are three configuration error rules cannot be classified into any of the vulnerability causes taxonomy. It is because those three error rules are related to particular protocol service that is vulnerable to be attacked. It seems to be outside the scope of the vulnerability of the firewall itself. Therefore, they fall into the Others category.

By removing the Design and Others category in the vulnerability causes ranking of [2] and [1] respectively, a similarity between the two causes ranking can be observed. They are both having the Validation category with highest ranking, Boundary checking with second rank, and the Authorization and domain at third. The only difference is the Serialization/Aliasing category in the vulnerability causes ranking for [1] is not having the same ranking as Authorization and domain. Overall, these two vulnerability causes ranking are very similar to each other. It seems that the analysed result by [2] is having a close match with the finding by [1], although their approaches for analysing the firewall are very different. Hence, the result from these two papers seems to be very convincing, as they confirm the results with each other.

4. Conclusion

This paper displays and compares the methodology of firewall analysis used in [1] and [2]. It also confirms the uniformity of the results from [1] and [2] by defining a method to compare the results. This provides a forceful argument for the accuracy of the results in two approaches [1], [2]. While the approach used in [1] is product specific, the approach used in [2] is too complex. Although [2] can apply on any kinds of firewall products, it still needs to be refined to reduce the ambiguity in the process. It seems to be more work need to be done in order to gain a comprehensive and general mechanism for firewall vulnerability analysis.

References

[1] A. Wool, "A Quantitative Study of Firewall Configuration Errors", *Computer* 37:6, 62-67, June 2004.

[2] Seny Kamara, Sonia Fahmy, Eugene Schultz, Florian Kerschbaum, Michael Frantzen, "Analysis of Vulnerabilities in Internet Firewalls," Computers and Security, volume 22, issue 3, pp. 214-232, April 2003.

[3] Michael Frantzen, Florian Kerschbaum, Eugene Schultz, Sonia Fahmy, "A Framework for Understanding Vulnerabilities in Firewalls Using a Dataflow Model of Firewall Internals," Computers and Security, volume 20, issue 3, pp. 263-270, May 2001.

[4] “Check Point Products”, http://www.checkpoint.com, 2004

[5] Wenliang Du and Aditya P. Mathur, “Categorization of software errors that led to security breaches,” in Proceedings of the 21st National Information System Security Conference (NISSC’98), 1998, http://www.cerias.purdue.edu/homes/duw/research/paper/nissc98.ps, 2004

[6] “CERT Coordination Center”,http://www.cert.org, 2004

Appendix A

Part 1

The twelve configuration errors:

1. No stealth rule.
2. Check Point implicit DNS over TCP rule.
3. Check Point implicit DNS over UDP rule.
4. Check Point implicit all ICMP rule.
5. Insecure firewall management.
6. Too many management machines.
7. External management machines.
8. NetBIOS service.
9. Portmapper/Remote Procedure Call service.
10. Zone-spanning objects.
11. Any service on inbound rules.
12. Any destination on outbound rules. [1]

Part 2

"Vulnerability causes include:

Validation error: A validation error occurs when the program interacts with the environment without ensuring the correctness of environmental data. There are three types of environmental data that need validation: input, origin, and target. Input validation ensures that the input is as expected. This includes the number, type and format of each input field. Origin validation ensures that the origin of data is actually what it is claimed to be, e.g., checking the identity of the IP source. Target validation ensures that the information goes to the place it is supposed to. This includes ensuring that protected information does not go to an untrusted target. _ Authorization error: An authorization error (called authentication error in the origin Du and Mathur taxonomy) permits a protected operation to be invoked without sufficient checking of the authority of the invoking agent.

Serialization/Aliasing error: A serialization error permits the asynchronous behaviour of different system operations to be exploited to cause a security violation. Many time-of-check-to-time-of-use flaws fall into this category. An aliasing flaw occurs when two names for the same object can cause its contents to change unexpectedly, and, consequently, invalidate checks already applied to it.

Boundary checking error: A boundary checking error is caused by failure to check boundaries and ensure constraints. Not checking against excessive values associated with table size, file allocation, or other resource consumption, leads to boundary checking errors. Buffer overflow is a result of a boundary checking error.

Domain error: A domain error occurs when the intended boundaries between protection environments have "holes." This causes information to implicitly leak out.

Weak/Incorrect design error: A design error occurs can be traced to the system design phase. For example, a weak encryption algorithm falls into this category.

Other error: Any error that does not fall into any of the above categories. Note that this is not a comprehensive or complete list– we only include the most common causes that lead to firewall vulnerabilities.

Vulnerability effects include:

Execution of code: This occurs when a vulnerability can lead to code being illegitimately executed. This includes, but is not limited to, code written by an attacker.

Change of target resource: This occurs when a vulnerability allows the state of a resource to be illegitimately changed by an attacker. A resource could be a host, a firewall rule table, or any entity that should be protected by the firewall.

Access to target resource: This occurs when a vulnerability allows an attacker illegitimate access to a target resource. Again, a resource may be any entity that is protected by the firewall. Examples of this vulnerability effect include allowing an attacker to read the firewall rule tables, or to find out which services are available on a protected host.

Denial of service (DoS): This occurs when a vulnerability is exploited to disrupt a service provided to legitimate users. Services in the context may range from packet forwarding or network address translation to administration.” [2]