Passive Traffic Inspection for Automated Firewall Rule Set Generation

Georg-Christian Pranschke[1], Barry Irwin[2] and Richard J Barnett[3]
Security and Networks Research Group
Department of Computer Science
Rhodes University
Grahamstown, South Africa
E-Mail: [1]g05p3292@campus.ru.ac.za [2]b.irwin@ru.ac.za [3]barnettrj@acm.org

***Abstract*—The introduction of network filters and chokes such as firewalls in existing operational network is often problematic, due to considerations that need to be made to minimise the interruption of existent legitimate traffic. This often necessitates the time consuming manual analysis of network traffic over a period of time in order to generate and vet the rule bases to minimise disruption of legitimate flows. To improve upon this, a system facilitating network traffic analysis and firewall rule set generation is proposed. The system shall be capable to deal with the ever increasing traffic volumes and help to provide and maintain high uptimes. A high level overview of the design of the components is presented. Additions to the system are scoring metrics which may assist the administrator to optimise the rule sets for the most efficient matching of flows, based on traffic volume, frequency or packet count. A third party package - *Firewall Builder* - is used to target the resultant rule sets to a number of different firewall and network Filtering platforms.**

***Index Terms*—firewall, c automated configuration, network traffic analyser, pcap, netflow**

I. Introduction

IN order for firewalls to serve their intended purpose, it is imperative that they are correctly configured for the requirements and environment in which they are operated. Combining the various technologies involved into a well configured firewalling solution is often a non trivial task in itself [6]A misconfigured firewall will, almost certainly, only provide the illusion of network security [10], and may well adversely affect legitimate traffic. While configuring firewalling solutions protecting small networks and correctly documented networks may be a relatively straight forward task for an experienced network administrator, it does become a very much harder task when dealing with poorly documented legacy and organically developed networks. The process of configuring and deploying a firewalling solution is further complicated when a firewall is to be introduced into a network segment that previously did not have any choke point controls.

The authors would like to acknowledge the financial support of Telkom SA, Business Connexion, Comverse SA, Stortech, Tellabs, Amatole, Mars Technologies and THRIP through the Telkom Centre of Excellence in the Department of Computer Science at Rhodes University.

This research is focused on the feasibility of automatically generating the an appropriate configuration input for *Firewall Builder* [2], based on an automated analysis of traffic collected at the intended point of insertion. The intention of this is to provide a means to analyse network traffic in setups in which to manual analysis is prohibitive due to traffic volumes or time constraints.

The remainder of this paper is structured as follows. After a brief problem statement in section II, in which we describe in what situations and setups the system is to be employed, we turn to a brief review of related research in section II and to the high level design overview of the proposed system in section III. Section IV describes possible future extensions to the system and concludes the paper.

II. Related Research

The pros and cons of netflow in analysing network traffic have been researched extensively [4], [9], [7], [5] - with the common conclusion that it is adequate if sampling rate and sampling interval (so called bin sizes) are chosen carefully as their values directly correspond to memory utilisation and cpu-time in netflow enabled Cisco routers. The shortcommings of netflow, pointed out in the related research are almost of no concern to the proposed project because the flow creation routine treats the packet-level traces as one continuous bin - and therefore all packets are analysed and no sampling artifacts created[5]. Because a flow is only created when a SYN-ACK is observed there is also no vulnerability to SYN flooding. There is a great deal of literature about rule collisions and redundancies such as described in [8].

III. Proposed Solution

In order to minimize downtime when deploying a new firewalling solution, the authors propose a system, which aims to automate as much of the firewall configuration process as possible with the goals of increasing the accuracy of the generated rule base and reducing the time required in comparison to traditional manual analysis. . The system consists

of two distinct components, one for analysing the traffic at the proposed choke point and one for generating a rule set to match the observed traffic. The traffic analyser should be capable of analysing either live traffic at the node or trace files recorded at the node, in *pcap* or *Cisco NetFlow* format.

The output of the automated traffic analysis is in a format similar to that of *NetFlow*. The resulting flows are stored in a database upon which the GUI-based rule generator can act. The system proposes an initial a set of rules based on the observed flows and allows the administrator to review and refine these rules from within the GUI. The refined rules are then be exported to an intermediate format for processing by *Firewall Builder*. The use of this third-party tool set allows for the targeting of a number of different filtering platforms. It is expected that this solution will speed up the process of configuring and deploying firewalls considerably because the administrator does not need to concern himself with a tedious manual traffic analysis, or the intrinsic details of writing firewall policies for a particular firewalling solution.

The traffic analyser uses *libpcap* [1](*WinPcap [3]* on Windows) to handle both, live traffic and *pcap* dump files. The processing of these two types of input is nearly identical. The strategy to obtain the same custom flows that are extracted from *NetFlow*, is to screen the packet data for TCP 3-way handshakes and TCP FIN and RST packets.

The ACK packets involved in the three way handshake can be determined through the packet's sequence numbers . This is used to establishes the sources and destinations and hence the direction of the traffic flows. The difference in the timestamps between the setup and tear-down packets allows for an estimation of the duration of any given connection. The packets that have neither SYN nor FIN flags set are matched to one of the existing flows and their payloads added to the total volume of traffic, and the packet counters increased for a specific flow Because IP datagrams may arrive out of order, care must be taken when reconstructing the flows so as to not disregard valuable information. Care is taken in that non SYN or FIN packets without a corresponding flow are used to create a new flow

The flow information is then stored in a relational database for later analysis by the rule generator. The database table that records the traffic flows should feature fields for a flow identifier number, the flows type of service, the timestamp of the SYN - ACK packet, the timestamp of the FIN - ACK packet, the total packet count in the flow, the total volume of traffic transferred in the flow so far, the flows source address and port and the flows destination address and port.

The basic strategy to automatically generate a rule set is to divide the network into an 'inside' and an 'outside', with both sides initially starting off with the least possible privileges (deny all). All incoming flows targeted at commonly known services are permitted. Flows targeting high port numbers are only allowed as a response to outgoing flows. The presence of services that are commonly considered outdated such as TELNET is flagged and a suggestion for their replacement made. This basic configuration can then be refined by the administrator by either individually allowing or denying flows or by specifying wildcards on IP, protocol or port level.

IV. Conclusion

Since the proposed system is still at the proof of concept phase, most future extensions considered at this time are related to adding features that will make it a stable production release. Two particular focus areas are the addition of IPv6 support, and the generation of additional scoring metrics based on flow size, duration, frequency and packet counts that can be used to further optimize the generated rule sets. Although this research is still at a very early stage, it is anticipated that the approach of automatic firewall rule set generation by means of passive traffic inspection will prove feasible and that a working prototype can be developed within the given timeframe. It is hoped that the proposed system will not only be quicker, more convenient and accurate than traditional manual configuration allowing for faster turnaround in the deployment of new firewalling solutions. This should result in decreased risk and cost for organisations deploying such solutions.

References

[1] Tcpdump/libpcap public repository. Online: http://www.tcpdump.org.

[2] What is firewall builder. Online: http://fwbuilder.org/about.html.

[3] Winpcap: The windows packet capture library. Online: http://winpcap.org.

[4] Choi, B.-Y., and Bhattacharyya, S. Observations on cisco sampled netflow. *SIGMETRICS Perform. Eval. Rev. 33* (2005), 18 – 23.

[5] Estan, C., Keys, K., Moore, D., and Varghese, G. Building a better netflow. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2004), ACM, pp. 245–256.

[6] Ogletree, T. *practical firewalls*. Que, 2000.

[7] Sommer, R., and Feldmann, A. Netflow: information loss or win? In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment* (New York, NY, USA, 2002), ACM, pp. 173–174.

[8] Venanzio Capretta, Bernard Stepien, A. F. S. M. Formal correctness of conflict detection for firewalls. *FMSE '07: Proceedings of the 2007 ACM workshop on Formal methods in security engineering* (2007), 22 – 30.

[9] Wallerich, J., Dreger, H., Feldmann, A., Krishnamurthy, B., and Willinger, W. A methodology for studying persistency aspects of internet flows. *SIGCOMM Comput. Commun. Rev. 35*, 2 (2005), 23–36.

[10] Zwicky, E. D., Cooper, S., and Chapman, D. B. *Building Internet Firewalls*. O'Reilly, 2000.

Georg-Christian Pranschke is currently reading for his Honours Degree at Rhodes University. He has research interests in Information Security, particularly Shellcoding, and Networking Technologies