Project Proposal

Georg-Christian Pranschke

Supervisor: Mr. Barry Irwin

Computer Science Department, Rhodes University

27th February 2009

1 Principle Investigator

Georg-Christian Pranschke
3 Milner Street
Grahamstown 6139
+27 - 83 - 58 30 193
g05p3292@campus.ru.ac.za
Supervised by: Barry Irwin, M.Sc. (CISSP)

2 Project Title

The proposed area of research for this project is in network security and firewall rule generation. The proposed title is: Automated firewall rule set generation through passive traffic inspection.

3 Background

ACM Classification System (1998) C.2.0 Security and protection

Introducing firewalls and other choke point controls in existing networks is often problematic, because in the majority of cases there is already production traffic in place that cannot be interupted. This often necessitates the time consuming manual analysis of network traffic in order to ensure that when a new system is installed, there is no disruption to legitimate flows. To improve upon this situation it is proposed that a system facilitating network traffic analysis and firewall rule set generation is developed.

4 Aim of project

To facilitate network traffic analysis and firewall rule generation by developing a system that automatically analyses the existing network traffic over a period of time and proposes a set of firewall rules to match the observed traffic. The

proposed rules can then consequnetly be further refined by the network administrator, thus essentially eliminating the need for tedious manual analysis, allowing for faster and possibly more accurate turnaround in the deployment of new firewalling solutions. This should also result in decreased risk to organisations deploying such solutions.

5 Intended outcome

A working system or series of tools that facilitate(s) the analysis of either live traffic flow or a recorded pcap trace file. The output of the traffic analyser should be useable as input to a tool such as fwbuilder, which would allow for cross platform deployment. Ideally the resultant tool chain should be cross platform and be able to run on Unix-like and windows systems. Main target platforms and their corresponding firewall solutions are FreeBSD and ipfw and pf, Linux and iptables and ip chains, Windows and its firewall policies and Cisco IOS and its ACLs.

Rules should be scored in terms of their percentage contribution to the traffic composition. A user should be able to select from matching at the following sorts of granularity:

IP	IP communications
Protocol	Level TCP, UDP, ESP etc
Port	level
ICMP	including subtypes should also be able to be matched.

6 Intended approach

The system shall be implemented as a classic client - server process, with the traffic analyser as the server and the GUI / rule generator as the client part. Because most machines to be chosen as future choke points are not likely to be desktop machines, this architecture does allow the the traffic analyzer to run on machines without desktop capabilities, such as rack mounted servers without X. The traffic analyser shall be implemented using libpcap, the de facto standard for packet capture and analysis across platforms (WinPcap on Windows). The GUI / Rule generator shall be implemented using the Qt widget set. The use of pcap and Qt and the fact that most work is to be done in a Unix environment infers the use of C/C++.

Due to the potentially very large data sets processed by the analyser, it is necessary to temporarily store its output in a database and have the rule generator work on data obtained from it. A particularily interesting database for this job is PostgreSQL, because of its native support for an IP data type.

7 Project progression time line

Duration	Activity
4 weeks	research period
4 weeks	traffic analyzer (rapid prototype)
2 weeks	traffic analyzer (code cleanup / rewrite)
1 week	traffic analyzer (porting)
3 weeks	GUI - Front end development
1 week	GUI - Front end (porting)
1 week	package creation
	writeup

8 Expected results

A fully functional cross platform system, running stable on at least the two mainly targeted platforms, FreeBSD and Windows that integrates fluently with Firewall Builder.

9 Possible extensions

The package could be further refined to produce optimized rule sets, that for example pass the highest volume traffic through the fewest rules. This could be done by analyzing the meta data the system will provide. Another possible extension is the inclusion of an intrusion detection and prevention system such as snort, and consequently its automatic configuration. Finally a web based or other alternative front ends to the analyzer could be developed.

References

[1] Firewall builder cookbook. Online: http://www.fwbuilder.org/guides/.

[2] Qt - a cross-platform application and ui framework. Online: http://www.qtsoftware.com.

[3] Sqlite. Online: http://www.sqlite.org.

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

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

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

[7] Cisco ios ipsec accounting with cisco ios netflow. Tech. rep., Cisco Systems, 2004.

[8] Cisco cns netflow collection engine user guide, 5.0.3. Tech. rep., Cisco Systems, 2005.

[9] Introduction to cisco ios netflow - a technical overview. Tech. rep., Cisco Systems, 2007.

[10] Blanchette, J., and Summerfield, M. *C++ GUI Programming with Qt 4*. Prentice Hall, 2006.

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

[12] Cohen, E., Duffield, N., Kaplan, H., Lund, C., and Thorup, M. Algorithms and estimators for accurate summarization of internet traffic. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (New York, NY, USA, 2007), ACM, pp. 265–278.

[13] 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.

[14] Garcia, L. M. Programming with libpcap - sniffing the network from our own application. *hackin9 3* (2008), 39.

[15] Garfinkel, S., Schwartz, A., and Spafford, G. *Practical Unix & Internet Security*. O'Reilly, 2003.

[16] Hamed, H., and Al-Shaer, E. Dynamic rule-ordering optimization for high-speed firewall filtering. In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security* (New York, NY, USA, 2006), ACM, pp. 332–342.

[17] Noonan, W., and Dubrawsky, I. *Firewall Fundamentals*. Cisco Press, 2006.

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

[19] Owens, M. *The Definitive Guide to SQLite*. Apress, 2006.

[20] Siyan, K. S., and Parker, T. *TCP Unleashed*. SAMS Publishing, 2002.

[21] 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.

[22] Uribe, T. E., and Cheung, S. Automatic analysis of firewall and network intrusion detection system configurations. *J. Comput. Secur. 15*, 6 (2007), 691–715.

[23] 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.

[24] 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.

[25] Yin, X., Yurcik, W., Treaster, M., Li, Y., and Lakkaraju, K. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (New York, NY, USA, 2004), ACM, pp. 26–34.

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