Flow analysis in the security context

Tobias Limmer and Falko Dressler

University of Erlangen-Nuremberg, Department of Computer Science 7
{limmer,dressler}@informatik.uni-erlangen.de

Recent statistics in the Internet show high growth rates for computers infected by malicious applications, like so-called bots. These are installed on computers owned by unsuspecting users at home or within poorly maintained networks. A few years ago, malicious applications, particularly worms, propagated by using network-wide scans to detect vulnerable hosts, and infected them afterwards (popular examples are Code Red and Nimda). These scans have been easily detected by network monitoring equipment as often this type of malware exploited known vulnerabilities using the same attack pattern. So payload-based intrusion detection systems like Snort are well suited for the detection of these worms as they used signatures for different exploits to detect traffic belonging to these malicious applications.

Later generations of malware did not depend on network-wide scans to detect and infect vulnerable hosts [1]. Instead, they relied on social engineering techniques to persuade users to execute malware, or zero-day client-side vulnerabilities like those found in Internet browsers. Especially these kinds of exploits are not easily distinguishable from normal network traffic. This explains a current trend in security-related monitoring systems: not the exploit of vulnerable systems is being detected but aftereffects of a successful exploit are detected, e.g. installed and active versions of malware generate typical traffic patterns that may be recognized by intrusion detection systems.

Another challenge of current network monitoring systems are high data rates in networks that do not allow deep packet inspection to detect traffic patterns. So, available information is often reduced to network flows that offer an aggregated view to the monitored traffic. This type of data offers reduction rates of usually more than 20:1 in the standard setting and still allows easy detection of traffic anomalies. For best performance, flow aggregation is directly performed on hardware routers, which often support the export of flows in the IPFIX format, or its predecessors Netflow v9/v5.

Current malware often causes a high number of failed connections. The limited view offered by flow-based data only allows the use of heuristic methods to determine the state of a connection, e.g. if it was successfully established, abnormally terminated or blocked by a firewall. We performed several experiments in which we analyzed results gained by analyzing flow data with a packet-based TCP defragmentation module and determined which properties describe different connection states best.

For the experiments, we used our network monitoring toolkit Vermont that offers flow-based analysis of network streams. It is based on a modular structure that allows import and export of flow data in the standardized IPFIX protocol, as well as included packet payload using the PSAMP protocol. Main goal during the development of Vermont was to maintain full reconfigurability during execution of the program. This enables adaptive reaction to new demands in dynamic distributed environments.

References

[1] M. Allman, V. Paxson, J. Terrell, *A brief history of scanning.* Proceedings of 7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007), San Diego, CA, USA, pp. 77-82, October 2007.