Flow-based Worm Detection using Correlated Honeypot Logs

Falko Dressler, Wolfgang Jaegers, and Reinhard German

Computer Networks and Communication Systems, University of Erlangen,
Martensstr. 3, 91058 Erlangen, Germany
{dressler,german}@informatik.uni-erlangen.de

Abstract. Attack detection in high-speed networks is a hot research topic. While the performance of packet oriented signature-based approaches is questionable, flow-based anomaly detection shows high false positive rates. We tried to combine both techniques. In this paper, we study the applicability of flow-based attack detection. We installed a lab environment consisting of a monitoring infrastructure and a well-controlled honeypot. Using correlated honeypot logs and flow signatures, we created a first set of attack pattern. The evaluation of the approach was done within our university network. On the positive side, we were able to prove the successful detection of worm attacks. Problems can occur if incomplete monitoring data is used.

1 Introduction

Besides denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks, worm and virus based assaults dominate the security threats in the Internet. The research on network security can be distinguished into attack detection and appropriate counteracting. Basically, attack detection is fundament for securing computer networks. In the past decades, many so-called intrusion detection systems (IDS) have been developed. Several taxonomies have been created that deal with attack and detection techniques [1–3]. Following these discussions, we use the term attack detection instead of intrusion detection as a more general denotation referring to a wide range of attacks including intrusion, system break-down, and resource exhaustion. A common classification criterion distinguishes host-based detection, i.e. attack detection executed on the protected host providing access to log file entries or the state of running processes, and network-based detection, i.e. operation on monitored network traffic within a particular network. Network-based detection can deal with a huge number of connected hosts. On the other hand, the captured network traffic is the only source of information that can be used. Additionally, we distinguish between signature based detection, also known as knowledge-based detection, and anomaly detection. The former method comprises techniques that dispose information about known attacks by pattern-based searching similar occurrences. Anomaly detection uses an opposite approach: based on information about normal network or system behavior, a significant derivation from this reference model is

considered as indicator of a potential attack. However, such derivations can have other reasons than attacks, resulting in false positives. From the performance perspective, a third criterion must be considered. Usually, the detection part is decoupled from the monitoring part. Therefore, the question arises, what data and to which cost must be captured in order to provide sufficient information for attack detection. In real networks, usually flow information is available, i.e. statistical information about specific data flows that are, for example, distinguished by the IP-5-tuple. Obviously, flow data is not applicable for packet-oriented signature detection. Inspired by works on fingerprinting attacks [4] and on using honeypots for signature creation [5], we searched for another approach. In this work, we discuss the possibilities and advantages of performing signature based techniques on flow data as well. We developed flow data based signatures using a well-defined environment: a lab environment containing a honeypot. Based on lab measures with our well-controlled honeypot, we correlated flow information associated to particular attacks. For testing purposes, we evaluated this approach on the lab environment as well as on our productive university network. We were able to find the analyzed worm attacks and finally determined open issues such as the need for adequate handling of incomplete monitoring data.

2 Related Work

Signature-based detection was the first kind of attack detection deployed in the Internet. While statistical conclusions are not possible, well-known attacks can be efficiently detected using this methodology. Open-source tools such as snort [6] and Bro [7] are widely used in the Internet. *Anomaly detection* allows to detect new kinds of attacks or slightly modified variants that cannot be detected by knowledge-based systems and it copes with the high performance demands in current backbone networks. A typical example is D-WARD [8], a network-based DDoS detection and defense system. The detection method uses predefined models of normal traffic and estimates deviations from this model. *Statistical flow-based detection* is a new mechanism that evolved together with high-performance monitoring techniques. Especially, flow monitoring has several performance advantages in high-speed networks [9]. Most recent developments include aggregation techniques that provide further reduction of monitoring data [10]. In the context of attack detection, three approaches should be mentioned. Mahoney et al. developed a self-learning system for traffic classification [11] and Conti et al. analyzed fingerprints of attack tools [4] These studies open a new research field for signature-based traffic analysis using flow data.

3 Lab Environment

Honeypot lab – For our tests, we deployed a monitoring infrastructure in a well-controlled lab environment consisting of a honeypot, a flow monitor, and a database system that collects all flow data for later analysis. The installation is depicted in figure 1. We used three PCs (2.80GHz, 512MB RAM, Suse Linux

10.1). The honeypot is directly connected to the Internet. It uses the *honeyd*[1] software package that already contains a number of scripts simulating well-known security vulnerabilities. A switch is used to duplicate the traffic from and to the honeypot. Connected to this switch is the monitoring PC using *vermont*[2], a high-speed flow monitor that is exporting flow information using the IPFIX (IP flow information export) protocol. The collector is the third PC using the *nasty*[2] software to store received flow information in a MySQL database. All PCs are also connected to our internal network to simplify operation and maintenance.

Honeypot configuration – We installed several virtual machines on the honeypot that simulate well-known vulnerabilities. This list includes ssh, telnet, finger, and apache flaws as well as mydoom, kuang2, and cmd.exe worms. These scripts either simulate native OS vulnerabilities or flaws that have been created by a particular worms. For later analysis, all scripts write log information (see listing 1.1). Additionally, all connections are logged by the honeyd system. *University network* – For evaluation purposes, we used a monitoring installation operated by the computing center of our university to test and evaluate our developed rules. Basically, this environment is very simular to the lab installation. We used vermont (Dual-Xeon 2.00GHz, 1GB RAM, Debian Linux) and nasty (Dual-Pentium4 2.40GHz, 1GB RAM, FreeBSD 6.1) to gather and collect flow information and to store them in a MySQL database. Based on these data, we were able to test and to improve developed flow signatures.

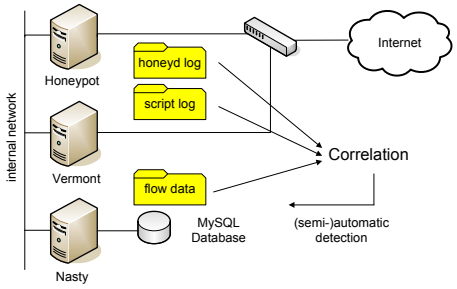

Fig. 1. Lab environment with database and log files for event correlation

Listing 1.1. Log file of mydoom.pl

```
2006-08-05 20:26:58 +0200: mydoom.pl[14383]: connection from 193.138.232.84:50894 to
    192.44.88.40:1080
2006-08-05 20:26:58 +0200: mydoom.pl[14383]: unknown command: 0x05 0x01 0x00
2006-08-06 04:00:09 +0200: mydoom.pl[17135]: connection from 82.127.224.169:2783 to
    192.44.88.23:3127
2006-08-06 04:00:09 +0200: mydoom.pl[17135]: file upload attempt from
    82.127.224.169:2783
2006-08-06 04:01:03 +0200: mydoom.pl[17135]: file uploaded to /usr/local/share/honeyd
    /scripts/mydoom/82/127/224/169/2783/FILE.17135, 104448 byte(s) written
```

4 Searching Worm Signatures

4.1 Flow analysis

Data sources – Figure 1 shows the available log data. The honeypot generates log information on detected attacks, suspicious activities, and network statistics.

[1] `http://www.honeyd.org/`

[2] `http://vermont.berlios.de/` (vermont and nasty)

Simultaneously, we collect flow information in a MySQL database. These information build the basis for developing meaningful flow signatures. We evaluated the signatures by comparing the detected attacks (only considering the flow information) with the honeypot data. Additionally, we tested the applicability in high-speed networks, i.e. in our university network. Since the deployed monitor collects all flow information from and to the university network, we should be able to detect the same attacks here as well.

Manual flow analysis – Starting with the first attacks detected by the honeypot, we searched for corresponding entries in the flow database. We identified pattern for three attacks: a mydoom backdoor on port 3127 and a Dabber and an unknown attack on the Sasser hole. For the pattern, we used the following information: destination port number, time interval of an active attack, number of associated connections, and the number of transmitted bytes. To give an example, the Dabber attack first targets the port 5554 and each connection transmits between 2000 and 6000 byte. Within 5 seconds, connections on ports 445, 8967, 1023, and 9898 are opened. Similar pattern are used for tools such as Snort as well. Nevertheless, these tools are able to search full packet data only while we try to work on flow information.

Detection scripts – We developed three scripts that scan for the mentioned attacks. Figure 2 depicts the simplest example, the scan for Dabber worms. Due to space restrictions, the diagrams and algorithms for the other worms are not shown here.

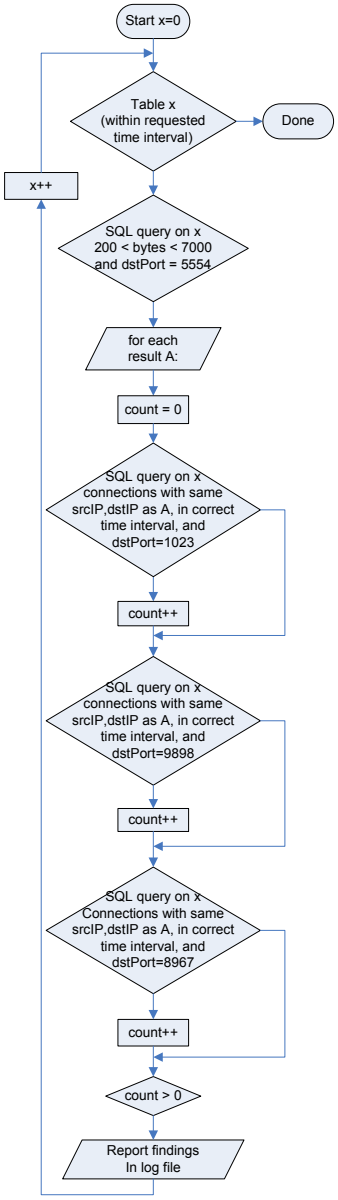

Fig. 2. Dabber detection

Observed data – The results presented in this work-in-progress paper refer to two measurements of 48h each starting on Sept. 12 and Sept. 27, respectively. In total, we found 7 MyDoom and 16 Sasser attacks (three of them Dabber). In the same time intervals, we collected flow data in the lab and in the university network. To give an impression of the amount of data to be analyzed: on Sept. 27, we observed 988k packets or 251k flows in the lab while we received about 5.165M packets or 90M flow records in the

university networks. The used monitoring tools reported some packet loss in the latter measurement. Obviously, the number of flow records to search is quite large in high-speed environments.

4.2 Evaluation

We used the three developed scripts to search both databases (lab and university) for our investigated attack patterns. Finally, we were able to detect several events. As an example, we show the results from observations from Sept. 27 in table 1. We selected this measurement because the mydoom script produced a number of false positives after the first run. Modifications to the script, i.e. to the pattern that we were searching for, reduced this rate dramatically. For example, we explicitly excluded a planetlab testbed client.

Date	Time	Attacker	Destination	Worm	Honeyd	Lab	Uni
27.09.2006	00:07:29	131.188.x.y	131.94.x.y	-			x
27.09.2006	19:47:12	125.51.251.56	192.44.88.58	mydoom	x	x	
27.09.2006	23:23:31	24.92.254.246	192.44.88.46	mydoom	x	x	
27.09.2006	23:29:27	189.141.119.151	192.44.88.99	mydoom	x	x	x
28.09.2006	16:32:23	81.222.45.162	192.44.88.72	mydoom	x	x	x
28.09.2006	20:19:33	82.16.82.224	192.44.88.87	mydoom	x	x	
28.09.2006	02:56:55	61.181.216.245	192.44.88.202	sasser	x	x	x
28.09.2006	03:27:36	61.181.216.245	192.44.88.240	sasser	x	x	
28.09.2006	04:07:18	218.90.153.34	192.44.88.249	sasser	x	x	
28.09.2006	15:12:38	121.227.18.48	192.44.88.234	sasser	x	x	

Table 1. Attacks as detected by the honeypot and by our flow pattern (snapshot)

Obviously, the detection quality in the lab network is very good (actually around 100%). This gives us the evidence that the approach works in principle. Looking at the results as provided by the same scripts working on the much larger database for the university network, we first recognize very few false positives (73 log entries). For example, one connection (the first one in the table) that matched our pattern, turned out to be legitimate traffic – while we were not able to figure out what really was going on, the "detected" worm was not found on that machine. Nonetheless, only 73 false positives in such a large number of analyzed flow records is good news. Secondly, it – mysterically – seems to be the case that some attacks were not detected looking at the data gathered in the university network while of course, the connection did use this path. We believe that some rules in our pattern do not match the corresponding flows due to changed statistical characteristics. The monitoring environment seemed to loose a small percentage of all transmitted packets. Therefore, the flow statistics change.

5 Conclusion

In this work-in-progress paper, we discussed the feasibility of flow-based attack detection. Using a well-controlled lab environment, we developed pattern that represent flow statistics using correlated honeypot data. In conclusion, it can be said that we were able to prove the applicability of this approach in our lab environment. We were able to find flow signatures of mydoom and sasser worms. Thus, the first results encourage further work on this topic. Trying the same algorithms in our university network, we identified some problems: packet loss in the monitoring setup leads to reduced detection ratio. In future work, we will elaborate this issue in two directions. First, we will try to improve the monitoring environment to reduce loss rates and secondly, we will work on statistical measures to counteract incomplete information. Obviously, more scripts and better honeypot technology is needed to observe the behavior of other worms and to develop correlated attack pattern.

References

1. Bace, R., Mell, P.: Intrusion Detection Systems. Nist computer security special publication sp 800-31, National Institute of Standards and Technology (2001)
2. Mirkovic, J., Reiher, P.: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. ACM SIGCOMM Computer Communication Review **34** (2004) 39–53
3. Dressler, F., Münz, G., Carle, G.: CATS - Cooperating Autonomous Detection Systems. In: 1st IFIP International Workshop on Autonomic Communication (WAC 2004), Poster Session, Berlin, Germany (2004)
4. Conti, G., Abdullah, K.: Passive visual fingerprinting of network attack tools. In: ACM workshop on Visualization and data mining for computer security, Washington, DC, USA (2004) 45–54
5. Kreibich, C., Crowcroft, J.: Honeycomb: Creating Intrusion Detection Signatures Using Honeypots. ACM SIGCOMM Computer Communication Review **34** (2004) 51–56
6. Beale, J., Caswell, B.: Snort 2.1 Intrusion Detection. 2nd edn. Syngress (2004)
7. Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks **31** (1999) 2435–2463
8. Mirkovic, J., Prier, G., Reiher, P.: Attacking DDoS at the Source. In: 10th IEEE International Conference on Network Protocols (ICNP 2002), Paris, France (2002) 312–321
9. Molina, M.: A scalable and efficient methodology for flow monitoring in the Internet. In Charzinski, J., Lehnert, R., Tran-Gia, P., eds.: 18th International Teletraffic Congress (ITC18). Volume 5a of Providing Quality of Service in Heterogeneous Environments., Berlin, Germany, Elsevier (2003) 271–280
10. Dressler, F., Münz, G.: Flexible Flow Aggregation for Adaptive Network Monitoring. In: 31st IEEE Conference on Local Computer Networks (LCN): 1st IEEE LCN Workshop on Network Measurements (WNM 2006), Tampa, Florida (2006)
11. Mahoney, M.V., Chan, P.K.: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In: 8th ACM International Conference on Knowledge Discovery and Data Mining. (2002) 376–385