Nemea: Searching for Botnet Footprints

Tomas Cejka[1], Radoslav Bodó[1], Hana Kubatova[2]
[1]CESNET, a.l.e. Zikova 4, 160 00 Prague 6 Czech Republic
[2]FIT, CTU in Prague Thakurova 9, 160 00 Prague 6 Czech Republic
{cejkat,bodik}@cesnet.cz, kubatova@fit.cvut.cz

Abstract. Malicious network traffic originated by malware means a serious threat. Current malware is designed to hide itself from the eyes of victim users as well as network administrators. It is very difficult or impossible to discover such traffic using traditional ways of flow-based monitoring. This paper describes a network traffic analysis of a backbone network as an attempt to discover infected devices. Cooperation with forensic laboratory and analysis of samples of malware allow to gain information that can lead to find unwanted traffic. Special tailored Nemea framework with high speed monitoring pipeline was used to discover infected devices on the network.

Keywords. Nemea, traffic analysis, malware, infected devices discovery, application layer analysis, DNS, HTTP, forensic laboratory.

1 Introduction

This paper is focused mainly on network traffic monitoring and analysis on a backbone networks. Backbone networks are very specific because of the volume of traffic that flows through the network infrastructure. The growth of network speed and bandwidth makes monitoring and analysis complicated. There are several issues related to this area [10].

Standard monitoring mechanisms (well described e.g. in [8]) exploits various mechanism of traffic aggregation or sampling that allow storage of information about traffic. Many security threats are still detectable even though the used aggregation looses precise information about transferred packets. Information about communication over network is represented as so called flow record, i.e. tuple of network addresses, ports and protocol that uniquely identify "one connection" between two hosts.

Basic flow records contain volume statistics such as number of packets and number of bytes. Analysis of the flow records is used to detect Denial of Service attacks (DoS), scanning or sweeping of addresses or ports, brute-force attacks etc. However, there are malicious applications that generate traffic very similar to normal benign traffic. The malicious applications (usually called malware), forged by attackers, usually exploit standard protocols and services. Additionally, this traffic is composed of a few packets in long time periods. Basic volume information about malware traffic is almost useless for detection of infected devices.

As we will explain in following sections, malware sometimes generate specific patterns that appear in packets. The patterns are e.g. URL in the HTTP protocol or DNS lookups containing suspicious domain names. Detection of such traffic could be based on application layer analysis of transferred packets, sometimes called as deep packet inspection [14]. The examples of detection of suspicious traffic using application layer information can be found e.g. in [2, 3].

Application layer analysis is very difficult or impossible for large networks without any hardware acceleration. The main obstacle is the volume of data that must be processed. On current high-speed networks, it is needed to analyze traffic at speed over 10 Gbps. Such traffic is very difficult to be transferred into software for processing by standard tools. For this task, a special tailored hardware card [7] can be used for hardware accelerated preprocessing in monitoring probes. Using the monitoring probes it is possible to process headers of application layers export information for additional analysis.

The rest of this paper describes real use case of cooperation of monitoring system, network traffic analysis tools and forensic laboratory that is able to analyze malware and provide valuable information to discover infected devices on the network.

2 Malware Sample Analysis

Currently, most of devices that are connected into internet can be infected by a malicious software. When malware gets into a device, it usually works as a downloader and starts to download additional applications such as backdoor [13].

Malicious codes spread through the internet via many means, but so far most used channel is a simple e-mail. The most typical scenario includes the obfuscated executable sent as an e-mail attachment, latter upon delivery the user itself is forced (perhaps by an urgently composed text) to open an attachment and thus executing the malicious program on one of his devices. Our analyzed samples were acquired as an suspicious incoming e-mail and were securely sent to forensic laboratory [4] for extended analysis.

This section briefly describes a process of malware analysis performed in the forensic laboratory.

At first, malware samples are passed to several simple tools for extracting basic data and metadata from the samples. This phase includes `file`, `strings`, antivirus software, PE Explorer, ... Finaly samples are disassembled to gain basic overview of its functionality. The data, metadata and assembler code of the executable file supplies the first insight and sometimes it can provide e.g. domain names or IP addresses or other interesting keywords in readable text format.

The second step is execution of the malicious code in specially prepared virtualized environment. Isolated virtual machine contains tools that monitors system calls (e.g. using procmon [12], `strace`, ...) and state of the system. It is needed to be able to take snapshots of disk and memory in a different phases of malware execution (starting with the state before execution). Execution of malware can bring valuable information about the infiltration into the operating system as well as the activity of malicious process. The activity of the process include disk operations (writing data) or network communication (downloading other binaries, contacting command and control servers).

Traces of process monitor and network traffic dump (e.g. using Wireshark [11]) can be visualized by ProcDOT [15]. This application can generate call graph that represents behaviour of the analyzed sample. Network dump is useful source of addresses and domain names that should be traced and monitored in real network traffic. Communication with gained addresses can expose infected hosts on the network.

Forensic analysis discovered several characteristics of malware behavior. The interesting characteristics for our purposes are related to the network traffic generated by malware. Fig. 1 shows the example of captured traffic from testing virtual environment.

Fig. 2 shows visualisation of malware activity. It summarizes information of process monitoring and network monitoring. The figure contains part of the graph that was created by ProcDOT [15].

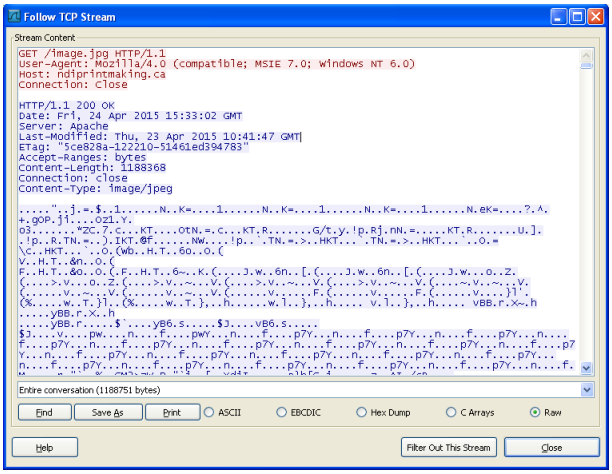

Figure 1: Network communication of infected computer with attacker's server.

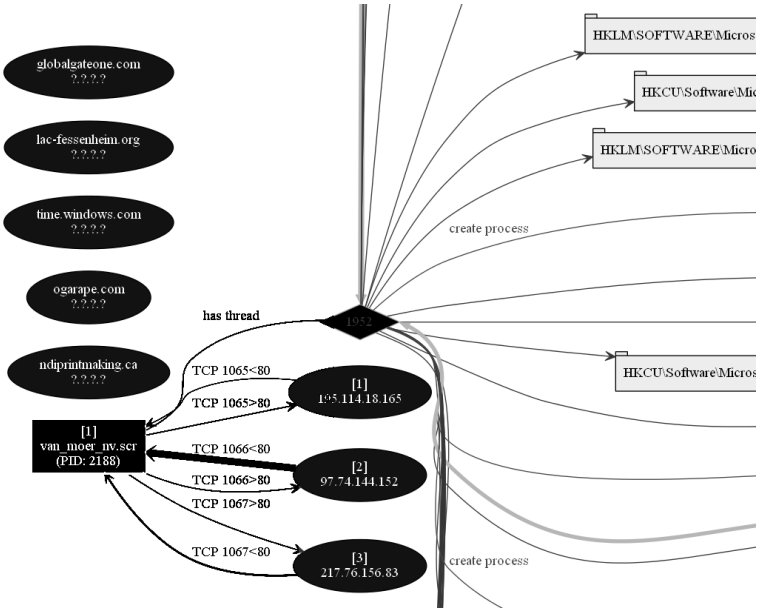

Figure 2: Part of the diagram of malware activity gained using ProcDOT in forensic analysis.

The laboratory analysis found the list of suspicious domain names and IP addresses that is shown in the following list and those pieces of information can be used to locate other infected nodes in monitored network:

- globalgateone.com
- globalgatetwo.com
- lac-fessenheim.org
- ogarape.com
- ndiprintmaking.ca
- 195.114.18.x

- 97.74.144.x
- 217.76.156.x

3 Real Network Monitoring – Infrastructure

Our monitoring infrastructure consists of monitoring probes [9] with the COMBO cards [7] for hardware acceleration, the open-source IPFIXcol collector [5] for collecting information about network traffic and finally the Nemea system [6] for stream-wise data analysis. The monitoring infrastructure is shown in Fig. 3.

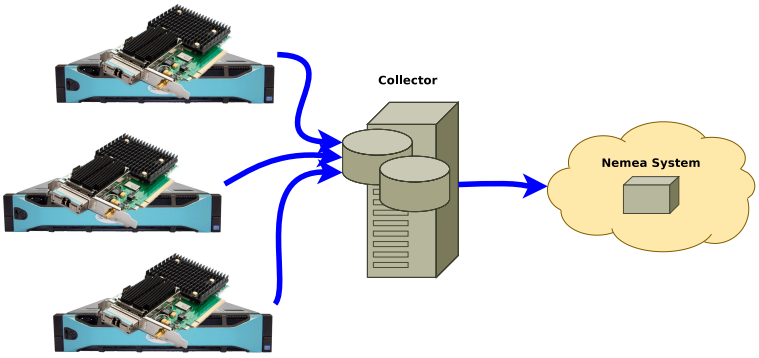

Figure 3: Monitoring infrastructure consisting of monitoring probes, collector and Nemea system.

The Nemea system a modular system for network traffic analysis and anomaly detection. It is composed by independent modules developed using the Nemea framework [1]. The framework tries to make development of Nemea modules easier and faster due to implementing common tasks in form of shared libraries.

Nemea contains several modules that work as a source of data for the Nemea system. An important example of such modules is a plug-in for IPFIXcol. This plug-in can export all flow records that collector receives from monitoring probe and pass them in format that Nemea modules understand. Using IPFIXcol plug-in, it is possible to get flow records extended by application layer information at near real-time into detection modules in the Nemea system. The example of the Nemea system configuration is shown in Fig. 4.

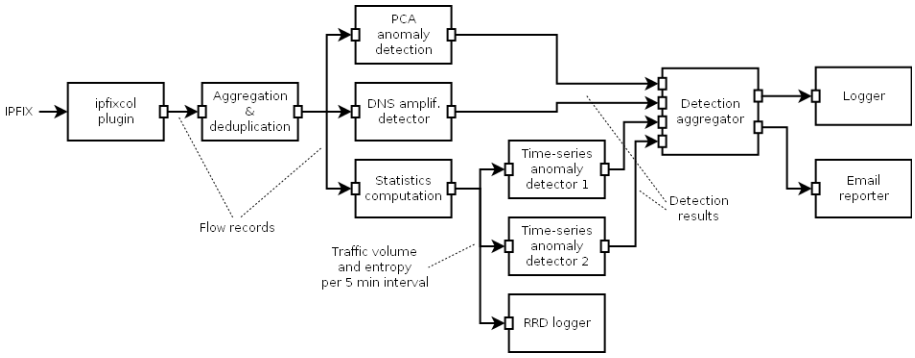

Figure 4: Example of interconnected modules of the Nemea system.

For the purposes of tracking of malicious traffic, a special filter module was used. The filter is able to take user-defined condition and apply it on incoming messages with information about

flow. Messages that do not satisfy the condition are dropped, the rest of messages is passed for next processed.

The module supports many operators that can be used in condition. We have used mainly matching regular expressions with values of fields extracted from application layer. The following listing shows the example of condition that filter messages of DNS traffic:

```
:DNS_NAME=~".*globalgate.*\.com$"||DNS_NAME=~".*ogarape\.com$"||
DNS_NAME=~".*lac-fessenheim\.org$"||
DNS_NAME=~".*ndiprintmaking\.ca$"
```

It is possible to filter traffic of discovered addresses that are probably command and control servers as follows (addresses are anonymized):

```
:SRC_IP==195.114.18.x||DST_IP==195.114.18.x
```

This condition was applied on HTTP data and we can discovered other potentially infected computers on our network infrastructure that communicate with the given server.

4 Results

After few days of monitoring, we have discovered 11 devices connected into the monitored network infrastructure that tried to resolve domain names used by malware samples or to communicate with discovered servers via the HTTP protocol.

Except domain names listed in previous section, we found that infected devices tried to query another suspicious domain name `dqwdwqwqdqwddqw.cn`. In addition, we discovered URLs that appeared in malicious HTTP queries that probably work as keep-alive ping: `/so/` and `/hr/`.

Most frequent source and destination IP address that appeared in our observation is from France. The second most frequent addresses belong to subnets of the Czech Republic, to the academic networks.

5 Conclusion

Discovering of infected computers on computer network infrastructure is a non-trivial but critically important task for every network operators. Attackers and authors of malicious code try hard to hide not only the running malware in the victim's system but also the traffic that malware produces. It is very difficult maybe impossible to find malicious traffic or traffic originated from infected devices using basic flow records.

This paper described the case study of cooperation of department of tools for monitoring and configuration with the forensic laboratory to discover infected devices on the monitored network infrastructure. As a result, 11 infected computers were discovered communicating with suspicious servers. In addition, it could be observed how infected devices communicated with command and control servers to let them know that they are still alive (infected).

The whole monitoring was allowed by monitoring infrastructure based on hardware accelerated monitoring probes that can operate at link speed (10 Gbps). Exported information about network traffic was extended by some headers of application protocols such as DNS and HTTP. The resulting flow records were processed by modular Nemea system that is designed for streamwise traffic analysis at near real-time.

After 10 days of monitoring, the most active command and control server that communicated from domain name `globalgateone.com` stopped responding, however, infected computers still continued to send their TCP SYN packets. The infected addresses discovered by our work were reported to members of CSIRT team.

Acknowledgments

This work was supported by the "CESNET Large Infrastructure" (LM2010005) funded by the Ministry of Education, Youth and Sports of the Czech Republic, MOBILITY 7AMB14SK177 Verification and dependability of digital systems design and by CTU in Prague that funded the grant No. SGS15/122/OHK3/1T/18.

References

[1] Bartos, V., Zadnik, M., Cejka, T.: Nemea: Framework for stream-wise analysis of network traffic. Tech. rep., CESNET (2013)

[2] Cejka, T., Bartos, V., Truxa, L., Kubatova, H.: Using Application-Aware Flow Monitoring for SIP Fraud Detection. In: Proc. of 9th International Conference on Autonomous Infrastructure, Management and Security (AIMS15) (2015)

[3] Cejka, T., Rosa, Z., Kubatova, H.: Stream-wise detection of surreptitious traffic over dns. In: Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on. pp. 300–304. IEEE (2014)

[4] CESNET, a. l. e.: FLAB forensic laboratory, `https://flab.cesnet.cz/`

[5] CESNET, a. l. e.: IPFIXcol, `https://github.com/CESNET/ipfixcol/`

[6] CESNET, a. l. e.: Nemea, `https://www.liberouter.org/nemea/`

[7] CESNET, a. l. e.: Programmable Hardware, `http://www.liberouter.org/technologies/cards/`

[8] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A.: Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. IEEE Communications Surveys Tutorials 16(4), 2037–2064 (2014)

[9] INVEA-TECH a.s.: FlowMon Probe – High-performance NetFlow Probe up to 10 Gbps, `http://www.invea-tech.com/products-and-services/flowmon/flowmon-probes`

[10] Kekely, L., Pus, V., Korenek, J.: Software defined monitoring of application protocols. In: INFOCOM, 2014 Proceedings IEEE. pp. 1725–1733. IEEE (2014)

[11] Orebaugh, A., Ramirez, G., Beale, J.: Wireshark & Ethereal network protocol analyzer toolkit. Syngress (2006)

[12] Russinovich, M.: Process Monitor v3.1, `https://technet.microsoft.com/en-us/library/bb896645.aspx`

[13] Skoudis, E.: Malware: Fighting malicious code. Prentice Hall Professional (2004)

[14] Velan, P., Celeda, P.: Next generation application-aware flow monitoring. In: Monitoring and Securing Virtualized Networks and Services, LNCS, vol. 8508, pp. 173–178. Springer (2014)

[15] Wojner, C.: ProcDOT — Visual Malware Analysis, `http://www.procdot.com/`