CANINE: A NetFlows Converter/Anonymizer Tool for Format Interoperability and Secure Sharing

Katherine Luo, Yifan Li, Adam Slagell, William Yurcik
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign
605 E. Springfield Avenue
Champaign, IL 61820
{*xluo1, yifan, slagell, byurcik*}*@ncsa.uiuc.edu*

Abstract

We created a tool to address two problems with using NetFlows logs for security analysis: (1) NetFlows come in multiple, incompatible formats, and (2) the sensitivity of NetFlow logs can hinder the sharing of these logs. We call the NetFlow converter and anonymizer that we created to address these problems CANINE: Converter and ANonymizer for Investigating Netflow Events). This paper demonstrates the use of CANINE in detail.

1 Introduction

A *network flow* is defined as a sequence of IP packets that are transferred between two endpoints within a certain time interval, and the most commonly used NetFlows formats are Cisco [1] and Argus [3]. With the increased use of NetFlows for network security monitoring [5], more and more tools based on NetFlows are being built and deployed. However, the different NetFlow sources, as well as collectors deployed, generate different incompatible versions of NetFlows. The different NetFlow formats impede the progress of network security monitoring since most tools that are based on NetFlows support only one format, but organizations often have hardware generating multiple formats. We were motivated to develop the CANINE to augment our existing flow tools [6, 7] by enabling them use NetFlows from the multiple sources here at the NCSA.

In addition to issues with format conversion, people often have concerns about information disclosure when sharing NetFlow logs—a source of sensitive network information. Consequently, we integrated anonymization capabilities with the converter. CANINE supports the anonymization of 8 fields common to all NetFlow formats: source IP address, destination IP address, starting timestamp, ending timestamp, source port, destination port, protocol and cumulative byte count. This combined converter and anonymizer has been vital to the development of visualization tools at the NCSA[6, 7] as it allows students to work with sensitive log data. We expect this work to likewise promote better insight into the use of NetFlows for security and network performance monitoring at other institutions.

The rest of the paper is organized as follows. Section 2 illustrates the system architecture of the CANINE. In section 3, we describe the the supported NetFlow conversion and anonymization methods. We conclude in Section 4.

2 System Architecture

The system architecture of CANINE is shown in Figure 1.

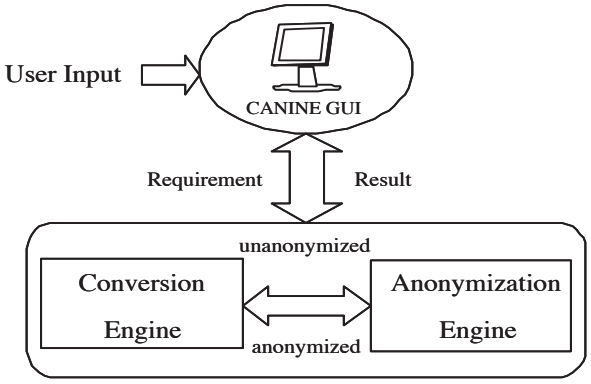

Figure 1: System Architecture of CANINE

CANINE consists of the two main modules: (1) the CANINE GUI and (2) the conversion/anonymization engines. The CANINE GUI accepts user input for NetFlow conversion and anonymization options, sends the request to the processing engine and summarizes the results of the performed actions in a pop-up window. First, the conversion engine reads the NetFlow data record from the source file and parses it into its component fields. Next it sends the unanonymized data to the anonymization engine. The anonymization engine houses a collection of anonymization algorithms, and it anonymizes the data according to the user's chosen options before it sends the data back to the

conversion engine. The conversion engine reassembles the anonymized data according to the conversion options and writes the records to the destination file. Statistics are collected and sent back to the GUI which displays them in a new window.

3 Demonstration of CANINE

The root window of CANINE is shown in Figure 2. In the source [destination] information fields, the user can designate the source [destination] NetFlow format and file. Below these fields, the user can choose the fields to anonymize and the specific anonymization algorithms to use—many fields have multiple anonymization options. Below that, the *task control* area is used to start [stop] anonymization and display the current progress. CANINE can be freely downloaded at: **http://security.ncsa.uiuc.edu/distribution/CANINEDownLoad.html**

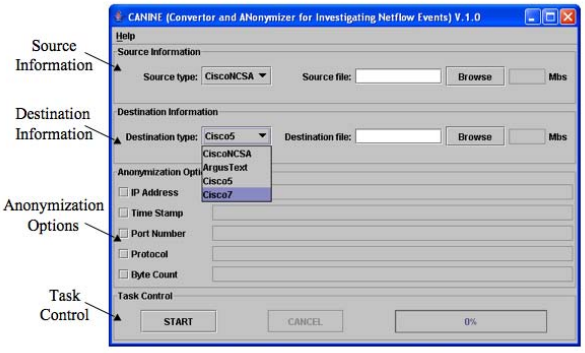

Figure 2: Main GUI of CANINE

3.1 NetFlow Conversion

CANINE's conversion engine currently supports conversion between Cisco V5, Cisco V7, Argus and NCSA unified formats. We briefly describe those formats below.

A. Cisco NetFlows

A *Cisco NetFlow* [2] record is a *unidirectional* flow identified by the following unique keys: source IP address, destination IP address, source port, destination port and protocol type. There are multiple versions of Cisco NetFlows (e.g., V1, V5, V7, V8 and V9). In all versions, the datagram consists of a header and one or more flow records. Most importantly, the header contains the version number and the number of records that follow in the datagram. For more details about the formats of each version, readers are referred to [1]. Currently, CANINE supports the most commonly used Cisco versions: V5 and V7. It will also support a mixture of V5 and V7 datagrams from an input file, though the output will all be in one format.

B. Argus NetFlows

Argus [3] views each network flow as a *bidirectional* sequence of packets that typically contains two sub-flows, one for each direction. Each flow record contains the attributes of source IP, source port, destination IP, destination port, protocol type, etc. There are two types of Argus records: the *Management Audit Record* and the *Flow Activity Record*, where the former provides information about Argus itself, and the latter provides information about specific network flows that Argus has tracked. For more details about the format, readers are referred to [4]. Note that unlike Cisco formats, Argus flows are ASCII text, rather than binary.

C. NCSA Unified Format

Since different versions of Cisco NetFlow Export datagrams are generated by the diverse routing equipment at the NCSA and because Cisco datagrams are of variable length, we have created the fixed length *NCSA Unified format* for use by our visualization tools ([6, 7]). This is important for efficiently supporting random access to records. The NCSA unified format contains the principle information about a network flow, as illustrated in Table 1 and serves as an internal format into which multiple versions of NetFlows can be transformed.

Table 1: NCSA unified record format

Data Field	**Length**(B)
version of Cisco NetFlow	1
padding (set to 0)	1
router's IP address	4
source IP address	4
destination IP address	4
source port number	2
destination port number	2
number of bytes	4
number of packets	4
protocol	1
TCP flags	1
start time (seconds since epoch)	4
milliseconds offset of start time	2
end time (seconds since epoch)	4
milliseconds offset of end time	2
padding (set to 0)	4

3.2 NetFlow Anonymization

CANINE's anonymization engine supports the anonymization of 8 data fields—only 5 unique types. Below we describe the anonymization options and their use within the latest version of CANINE.

3.2.1 IP Anonymization

We support three options to anonymize IP addresses. Note that either both the source and destination IP addresses are anonymized or both are unanonymized. You cannot

anonymize one without the other. The IP anonymization options GUI is shown in Figure 3.

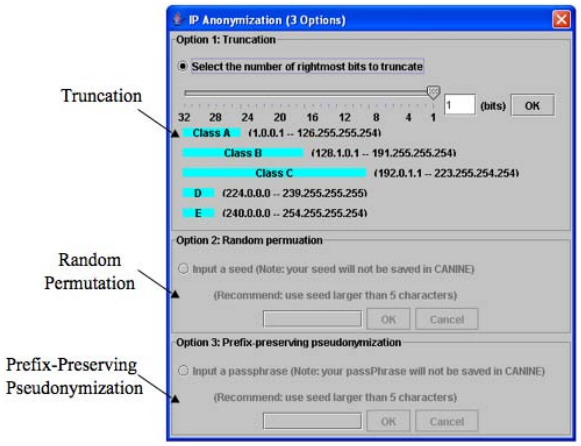

Figure 3: IP Anonymization Options

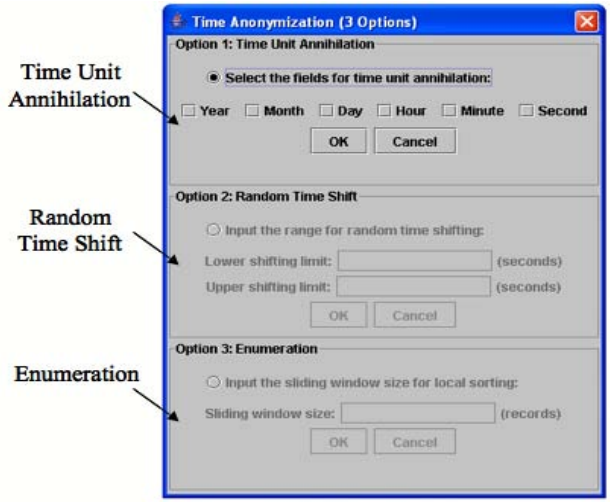

Figure 4: Timestamp Anonymization Options

A. Truncation

For IP address truncation, the user chooses the number of least significant bits to truncate. For example, truncating 8 bits would simply replace an IP address with the corresponding class C network address. Truncating all 32 bits would replace every IP with the constant *0.0.0.0*.

B. Random Permutation

We also support anonymization by creating a random permutation seeded by user input. We implement this algorithm through use of two hash tables for efficient lookup. Note that the use of tables means that the permutation will be different for two different logs anonymized at different times.

C. Prefix-preserving Pseudonymization

Prefix-preserving pseudonymization is a special class of permutations that have a unique structure preserving property. The property is that two anonymized IP addresses match on a prefix of n bits if and only if the unanonymized addresses match on n bits. We implemented the Crypto-PAn algorithm [8] for this type of anonymization, adding a key generator that takes a passphrase as input.

3.2.2 Timestamp Anonymization

Timestamps can be broken down into the units of *Year, Month, Day, Hour, Minute and Second*. We currently support three options to anonymize timestamps. The timestamp anonymization GUI is shown in Figure 4.

A. Time Unit Annihilation

We support the annihilation of any subset of the previously mentioned units. The user selects the time units to zero-out. For example, if someone wishes to obfuscate the date, they can remove the year, month and day information from the ending times. If they want to completely eliminate time information, they can select all of the time units for annihilation. Start times are adjusted so that the duration of the flow is kept the same.

B. Random Time Shift

In some cases it may be important to know how far apart two events are without knowing exactly when they occurred. For this reason, a log or set of logs can be anonymized at once such that all timestamps are shifted by the same random number. The user needs to designate the lower and upper shift limit, from which the random number of seconds is generated. If one uses this type of anonymization on two different log files at different times, then this random number will be different between the data sets. We warn users to be aware of the troubles with data mining (by indexing the timestamp) between sets anonymized at different times in this manner.

C. Enumeration

With this method, all time information is essentially removed except the order in which the events occurred. A random end time for first record is chosen, and all other records are equidistantly spaced from each other—temporally that is—while retaining the original order with respect to ending times. Start times are adjusted so that the duration of the flow is kept the same. Sorting cannot work perfectly on streamed data, and it would be extremely slow on large log files. So we make use of the fact that records come roughly sorted by ending times and sort locally. This has worked with great accuracy and efficiency.

3.2.3 Port Anonymization

We support two anonymization options for port numbers. The port number anonymization GUI is shown in Figure 5.

A. Bilateral Classification

Usually, the port number is useless unless one knows the exact value to correlate with a service. However, there is

Figure 5: Port Anonymization Options

one important piece of information that does not require one to know the actual port number: whether or not the port is ephemeral. In this way, we can classify ports as being below 1024 or greater than 1023. To keep the format the same for log analysis tools, port 0 replaces all ports less than 1024, while port 65535 replaces the rest of the port values.

B. Black Marker Anonymization

From an information theoretic point of view, this method is no different than printing out a log and blacking-out every port number. In a digital form, we just replace all ports with a 16 bit representation for 0.

3.2.4 Protocol Anonymization

Protocol information can be eliminated with CANINE. We do this by replacing the protocol number with the unused, but IANA reserved, number of 255. This is the maximal number for that 8 bit field.

3.2.5 Byte Count Anonymization

For user privacy, one may desire to eliminate byte counts. Thus we support black marker anonymization of this field. All byte counts are replaced with the constant of 0, an impossible byte count in reality because headers do account for some of those bytes.

3.3 Task Result Dialog

After the CANINE task finishes, a brief task summary will be shown to the user in a pop-up window (Figure 6).

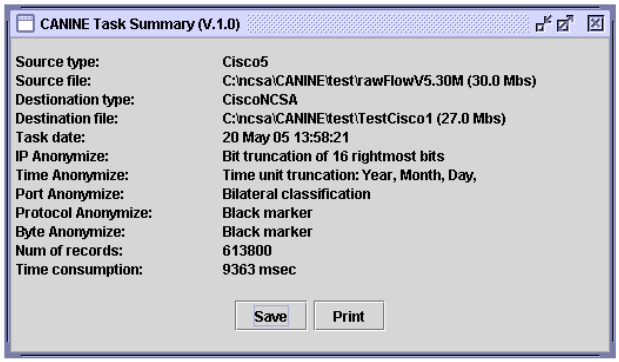

Figure 6: Task Summary Dialog of CANINE

The task summary includes the following information: source and destination formats/filenames, date of processing, anonymization methods used, number of records processed and the total processing time. The user can save and print the task summary for future reference.

4 Summary

In this paper, we put forth two important problems facing the developers of NetFlow based tools: (1) NetFlows come in different and incompatible formats, and (2) the sensitive nature of NetFlow logs make it difficult for developers to find good data sources. Our tool, CANINE, addresses both of these issues by giving users the ability to both convert and anonymize NetFlow logs.

While users have many options to anonymize NetFlows with CANINE, it can still be difficult to choose the correct options for a particular organization's needs. Thus, future work should focus on creating multiple, useful levels of anonymization that trade-off between the utility of the anonymized log and the security of the anonymization scheme. This work should also strive to help organizations map levels of trust shared with would-be receivers to these different levels of anonymization.

References

[1] "Cisco NetFlow Services and Applications White Paper", Jun 2005; <http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm>.

[2] K. Claffy, G. C. Polyzos and H. W. Braun. "Internet traffic flow profiling", UCSD TR-CS93-328, SDSC GA-A21526, 1993.

[3] C. Bullard, "Argus, the network Audit Record Generation and Utilization System", June 2005; <http://www.qosient.com/argus/>.

[4] C. Bullard, "Argus record format", June 2005; <http://www.qosient.com/argus/argus.5.htm/>.

[5] Yiming Gong, "Detecting Worms and Abnormal Activities with NetFlows", August 2004; <http://www.securityfocus.com/infocus/1796>

[6] K. Lakkaraju, W. Yurcik, A. Lee, R. Bearavolu, Y. Li and X. Yin, "NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness. VizSEC/DMSEC, held in conjunction with 11^{th} ACM Conference on Computer and Communications Security, Fairfax, VA, October 2004.

[7] X. Yin, W. Yurcik, M. Treaster, Y. Li and K. Lakkaraju, "VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness", VizSEC/DMSEC, held in conjunction with ACM Conference on Computer and Communications Security, Fairfax, VA, October 2004.

[8] A. Slagell, J. Wang and W. Yurcik, "Network Log Anonymization: Application of Crypto-PAn to Cisco NetFlows, Secure Knowledge Management Workshop, Buffalo, NY, 2004.