SCRUB NetFlows

A Software Tool for Multi-Field Multi-Level NetFlows Anonymization

<http://scrub-netflows.sourceforge.net/>

William Yurcik
Clay Woolam, Latifur Khan, Bhavani Thuraisingham

University of Texas at Dallas
Motivation: Anonymization?

Anonymization enables entities to share types of data that would otherwise not be shared

(1) Private Data
- User-identifiable information
  - user content (Email messages, URLs)
  - user behavior (access patterns, application usage)
- Machine/Interface addresses
  - IP and MAC addresses

(2) Secret Data
- System configurations (services, topology, routing)
- Traffic patterns (connections, mix, volume)
- Security defenses (firewalls, IDS, routers)
- Attack impacts
Motivation: Sharing?

- Chasing attackers away (to other organizations) does not improve security

- Security data is needed between organizations to correlate events across administrative domains (cumulative learning between organizations)
  - Detect attacks
  - Blacklist attackers and attacker techniques
  - Distinguishing between normal and suspicious network traffic patterns
**SCRUB** Infrastructure

- **SCRUB-tcpdump**
- **SCRUB-NetFlows**
- **SCRUB-PACCT**
- **SCRUB-Alerts**
- **CANINE** (format converter)

Organization Enabled for Distributed Sharing

- Command processes
- NetFlows (Cisco, Argus, IPFix)

Other Organizations
- MSSP
- CERT
- ISAC

IDS Firewall Virus

The University of Texas at Dallas
CANINE (Flocon’05)  
**a NetFlows Converter/Anonymizer**

- **CANINE**: Converter and ANonymizer for Investigating Netflow Events  
  [http://security.ncsa.uiuc.edu/distribution/CanineDownload.html](http://security.ncsa.uiuc.edu/distribution/CanineDownload.html)

- **Converter**  
  - Cisco V5 & V7, ArgusNCSA, CiscoNCSA, NFDump

- **Anonymizer**  
  - 5 NetFlow fields (multi-field)  
    (1) IP, (2) Timestamp, (3) Port, (4) Protocol, (5) Byte Count  
    - Multiple options for each field (multi-level anonymization)

- **Java GUI** – easy to use point-and-click
IP Address Anonymization in CANINE

The University of Texas at Dallas
New & Improved NetFlows Anonymizer

- ASCII-based PERL code
  - works on any NetFlows format converted to ascii
  - optimized code (multi-threaded parallelization)
- Anonymizes more NetFlow fields (10>5)
  - adding support for additional fields is minimal
  - (6) TimeStamp (first/last pkt) (7) TOS (8) TTL (9) TCP Flags (10) Packet Count
- Improved/More anonymization options per field
  - Fixes Crypto-PAn IP address anonymization flaw
  - Working on tailoring semantics to low/medium/high
- Command line operation
  - UNIX friendly, consistency with other SCRUB* tools
  - cascaded streaming operation available via piping
SCRUB-NetFlows
Multi-Level Anonymization Options

• Black Marker (filtering/deletion)
• Pure Randomization (replacement)
• Keyed Randomization (replacement)
• Annihilation/Truncation (accuracy reduction)
• Prefix-Preserving Pseudonymization (IP address)
• Grouping (accuracy reduction)
  – Bilateral Classification
• Enumeration (time, adding noise)
• Time Shift (time, adding noise)
Example: Timestamp Field (First/Last Pkt)

- **Black Marker**
  - replacement of field with a predefined constant (0)
- **Random Time Shift**
  - increments given time by a random value within a user defined window
- **Enumeration**
  - sorts entries by timestamp, applies black-marker
- **Distance-preserving pseudonymization**
  - preserve distance between two timestamps
- **More**
  - including pure/keyed randomization, truncation, unit annihilation
Addressing Crypto-PAn Flaw in SCRUB-NetFlows

• Crypto-PAn is widely used for prefix-preserving pseudonymization
  – flaw discovered – attacker can reverse-engineer the original prefix mapping in a given dataset

• Our use of Crypto-PAn
  – Begin with two separate instances of Crypto-PAn with two distinct keys: Crypt1 and Crypt2
  – Determine network and host portion of IP address
  – Run Crypt1 and Crypt2 on the IP address
  – Return the network of Crypt1 concatenated with the host given by Crypt2
Example usage

- Anonymizations done on one line of an Argus NetFlow
  - The program is told to black marker the source IP, randomize the destination IP, and black marker the first timestamp

```
$ ./scrub-netflow.pl -r ArgusData_146_78 -w AnonData -o "srcip bm dstip rand firsttimestamp bm"

Anonymizing ARGUS format

$ tail -n 1 AnonData
01 Jan 71 01:01:01 02 Oct 03 14:00:50 udp 10.10.10.1118 -> 39.7.114.87.55525 6 0
  4856   0       INT

$ tail -n 1 ArgusData_146_78
02 Oct 03 14:00:00 02 Oct 03 14:00:50 udp 132.156.189.139.1118 -> 228.154.76.120.55525 6
  9     4856   0       INT
```

The University of Texas at Dallas
Anonymization for Sharing: The Privacy vs. Analysis Tradeoff

while anonymization protects against information leakage it also destroys data needed for security analysis

– Zero-Sum? (more privacy <> less analysis & vice versa)

– We are now making measurements of the tradeoff
  • another story but we can talk off-line
Summary

- Critical need for security data sharing between organizations

- Anonymization can provide safe security data sharing
  - Multi-Field: prevent information leakage
  - Multi-Level: no one-size-fits-all anonymization solution

- **SCRUB-NetFlows** as part of a data sharing infrastructure (**SCRUB***) supporting multiple data sources
  - NetFlows is not the only data source of interest

- No “One-Size-Fits-All” anonymization policy
  - multi-level anonymization options can/should be tailored to requirements of sharing parties to optimize tradeoffs
  - privacy/analysis anonymization tradeoffs need to be characterized
SCRUB* References

Background on Using Anonymization to Safely Share Security Data


SCRUB* Tool (1) SCRUB-tcpdump <http://scrub-tcpdump.sourceforge.net/>


SCRUB* Tool (2) SCRUB-PACCT <http://security.ncsa.uiuc.edu/distribution/Scrub-PADownLoad.html>


SCRUB* Tool (3) SCRUB-NetFlows <http://scrub-netflows.sourceforge.net/>>


SCRUB-NetFlows
<http://scrub-netflows.sourceforge.net/>