A Method to Compress and Anonymize Packet Traces

Markus Peuhkuri

2001-11-02

Abstract

Data volume and privacy issues are one of problems related to large-scale packet capture. Utilizing flow nature of Internet traffic can reduce data volume. Removing sensitive information such as IP addresses enchanges privacy. Our method makes possible to have same replacement value for given IP address even if capture location or time is different.

Problems in packet capture?

- Data volume
  - pre-filtering and processing on capture card
  - persistent storage problem
- Data privacy
  - packets include sensitive data in
    - header
    - payload
  - TLS, SSH and IPSec helps for payload

Packet capture, why to do it?

- Measure sum effect of multiple
  - users
  - applications
  - operating systems
  - protocols
  - hardware
- with one (or a few) device(s)
- to provide data for
  - analysis
  - simulation
  - models

Other author information: Email: *Markus.Peuhkuri@hut.fi*; Telephone: +358-9-451 2467; Fax: +358-9-451 2474; Home page: `http://www.iki.fi/puhuri/`.
This work is supported by Academy of Finland contract for project MI^{2}TTA.

Compression by flows

- Better compression rate if you utilize structure of data than if data is “just bits”
  - RFC1144, RFC2507
- Data in flows (5-tuple)

  TCP sequence, ack numbers proceed, possibly same size

  UDP possibly same size

  1. keep track of every active flow (large id space)
  2. compare to previous packet
  3. short codes for common cases

What data not to include

- IP identification + fragment word
  - changes randomly
  - for most studies no-use
  - adds 32 or 0/24/40 bits for each packet
- checksums
  - no use afterwards, just check if ok (if possible)
- length and header length fields implicit
- TTL field and TOS/DS byte should be constant in a flow
  ⇒ record changes

What is sensitive

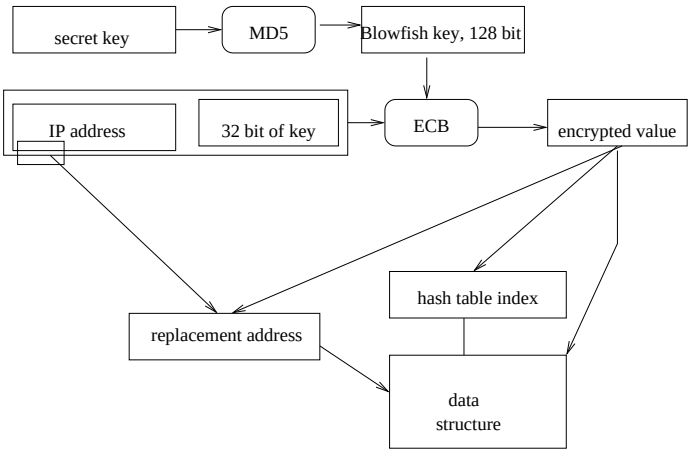
Person identification identifies communication parties to a single person (or one’s family)
⇒ IP address

Application identifier TCP/UDP port numbers – what applications are used

Payload

If you open a protected message, or learn the content of a message or learn that a message is sent or received.
⇒ *violation of communication secret*
⇒ a fine or maximum 1 years, 3 if special device is used

How to keep address secret



Steps

Initiate encryption (blowfish), then for each IP address

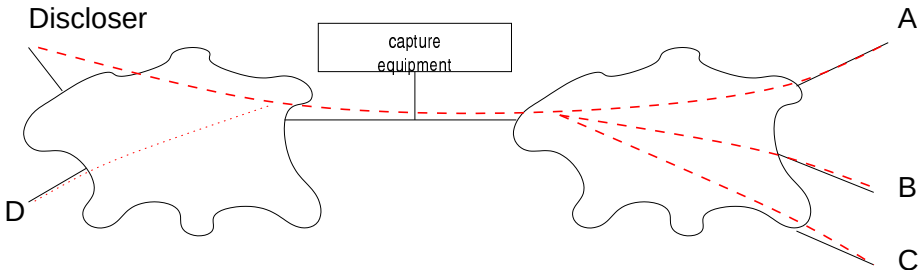
1. Encrypt
2. Check if in hash, if not then
   (a) insert into hash
   (b) write out record to stream
3. Replace real IP with 8 bits of clear and 24 bits from encrypted
   ⇒ codeIP

On decoding (off-line) each time encryption record is found

1. Check if known mapping secret ⇒ anonIP, if not then
   (a) pick random unused IP from that network
   (b) store (secret,anonIP) to (persistent) database
   (c) maintain hash table of codeIP ⇒ (secret,anonIP) mapping

Replace codeIP with anonIP in headers

Possible disclosure



Performance

Compression	Size [MiB]	Time [s]	Pkts/s
none	4,886	-	-
gzip	2,218	5,108	9,120
anon+flow	770	1,374	33,906
anon+flow+gzip	318	2,023	23,028
plain	1759	-	-

Conclusions

- Compression and desensitization needed
- Utilize flow nature of traffic in compression
- Desensitization works, but vulnerable to *choisen plain-text*
  ⇒ Some control to trace archives needed
- Same real IP maps always to the same anon IP if secret is the same
  ⇒ Possible to relate measurements in multiple locations
- Performance feasible, also memory requirements