Security Vulnerabilities and Solutions for Packet Sampling

Sharon Goldberg and Jennifer Rexford
Managing a network is all about measurement...

Load measurement at single node
Why? To characterize traffic mix, for billing, for intrusion detection, etc.
How? Uncoordinated sampling (each node selects packets independently)
Network Measurement via Packet Sampling

Managing a network is all about measurement…

Load measurement at single node
Why? To characterize traffic mix, for billing, for intrusion detection, etc.
How? Uncoordinated sampling (each node selects packets independently)

Load, loss, and delay measurement on a path
Why? Finding spatial paths of traffic thru network, path quality measurement
How? Coordinated sampling (packet selected by one node selected by all nodes)

IETF PSAMP: standardize packet sampling on linecards
Sampling should be passive (not modify traffic)
Network Measurement via Packet Sampling

Managing a network is all about measurement…

Load measurement at single node
Why? To characterize traffic mix, for billing, for intrusion detection, etc.
How? Uncoordinated sampling (each node selects packets independently)

Load, loss, and delay measurement on a path
Why? Finding spatial paths of traffic thru network, path quality measurement
How? Coordinated sampling (packet selected by one node selected by all nodes)

IETF PSAMP: standardize packet sampling on linecards
Sampling should be passive (not modify traffic)
Packet Sampling: The IETF PSAMP Framework

Each **Sampler** selects and stores a \( p \)-fraction of packets.

Sampling outcomes are exported from the **Samplers** to the **Collector**.

**Uncoordinated Sampling:**
~ Each Sampler select packets independently of other Samplers.

**Coordinated Sampling:**
~ A packet selected at one Sampler is selected at all Samplers.
~ Sampling outcomes are aggregated at the Collector.
Secure Packet Sampling

No adversarial host can craft a disproportionally selected packet stream

Who is the adversary?
- Botnet evading intrusion detection (uncoord)
- Botnet evading network traceback (coord)
- Greedy customer evading billing

Weak adversary: Crafts arbitrary packet streams

Strong adversary: Crafts arbitrary packet streams
Learns sampling outcomes

For example, by...
- Eavesdropping on export packets
- Observing billing information
Who is the adversary?
- Botnet evading intrusion detection (uncoord)
- Botnet evading network traceback (coord)
- Greedy customer evading billing

Weak adversary: Crafts arbitrary packet streams
Strong adversary: Crafts arbitrary packet streams
Learns sampling outcomes

Secure Packet Sampling

No adversarial host can craft a disproportionally selected packet stream

For example, by...
- Eavesdropping on export packets
- Observing billing information
Secure (Uncoordinated) Random Sampling

Samp(d) = 1 with probability \( p \)
0 with probability \( 1-p \)

Secure against weak adversary:

Each packet sampled randomly and independently

\( \Rightarrow \) adversary can’t predict if a packet will be sampled with probability better than \( p \)
Secure (Uncoordinated) Random Sampling

Secure against weak adversary:
- Each packet sampled randomly and independently
- Adversary can’t predict if a packet will be sampled with probability better than $p$

Secure against strong adversary:
- Each packet sampled randomly and independently
- Adversary can’t predict if a packet will be sampled with probability better than $p$ even if past sampling outcomes are known

Samp$(d) = 1$ with probability $p$
0 with probability $1-p$

Requires a cryptographically-strong random number generator
(e.g. RC4, AES in counter mode)
Hash-Based Coordinated Sampling

\[
\text{Samp}(d) = \begin{cases} 
1 & \text{if } f_k(d) \in [R_1, R_2] \\
0 & \text{else}
\end{cases}
\]

Suppose \(k, R_1, R_2\) are secret

With an unkeyed hash function, a weak adversary can break security:

Suppose \(k, R_1, R_2\) are secret

With high probability, packets evade selection

With an unkeyed hash function, a weak adversary can break security:

\[
\begin{array}{c}
S_1 \\
S_2 \\
R_1 \\
R_2
\end{array}
\]

Chooses arbitrary \([S_1, S_2]\) and send packets \(d\) is such that \(f(d) \in [S_1, S_2]\)

With high probability, packets evade selection
A PseudoRandom Function (PRF) $f_k(d)$ is a keyed cryptographic hash

- Pseudorandom \( \rightarrow \) Fresh pseudorandom output for each fresh input
- A Function \( \rightarrow \) Identical output for identical input

If (uncoordinated) random sampling is secure
\[ \Rightarrow \text{PRF-based sampling is secure when the adversary sends unique packets} \]

Can use hardware implementation of pipelined, keyed MD5, SHA1, or AES in CBC mode **but not** the CRC $f_k(d) = d \mod k$

But can we prevent adversary from breaking security by **replaying** packets?
PRF-Based Coordinated Sampling (1)

A Pseudorandom Function (PRF) $f_k(d)$ is a keyed cryptographic hash

- Pseudorandom → Fresh pseudorandom output for each fresh input
- A Function → Identical output for identical input

If (uncoordinated) random sampling is secure
⇒ PRF-based sampling is secure when the adversary sends unique packets

Can use hardware implementation of pipelined, keyed MD5, SHA1, or AES in CBC mode but not the CRC $f_k(d) = d \text{ mod } k$

But can we prevent adversary from breaking security by replaying packets?
PRF-Based Coordinated Sampling (2)

\[ \text{Samp}(d) = \begin{cases} 1 & \text{if } f_k(d) \in [R_1, R_2] \\ 0 & \text{else} \end{cases} \]

Can we prevent adversary from breaking security by **replaying** packets?

... without modifying packets at the Samplers...

Prevent adversary from using past sampling outcomes to craft new packets

1. Prevent export packets from leaking sampling outcomes (encrypt, pad to fixed length, send at fixed rate) or (physically secure channel)

2. Change the PRF key frequently (each time e.g. billing info is leaked to hosts)
Fault Detection: **Secure Path Quality Measurement**

1. Use coordinated sampling at sender and receiver
2. Estimate packet loss rates at Collector by comparing records

**Security:** No adversarial router can bias path quality measurement

1. Prevent adversary from selectively dropping non-sampled packets
   - Use PRF-based coordinated sampling
2. Prevent adversary from modifying the receiver’s export packets
   - Cryptographically authenticate the export packets
Conclusions

**Uncoordinated sampling**
- Random sampling with a cryptographic random number generator

**Coordinated sampling**
- Unkeyed hash-based sampling vulnerable even to weak attackers!
- As is sampling with a keyed non-cryptographic hash
- Cryptographic PRF-based sampling
  - Secure when host sends unique packets
  - To prevent replay attacks,
    ... secure the export packets and frequently rekey the PRF

**Path quality measurement**
- Cryptographic PRF-based sampling + authenticated export packets

We need cryptographic hash functions for secure packet sampling!
Secure coordinated sampling is approx as complex as random sampling