Lobster project

Sven Ubik, CESNET
Zurich, 14 April 2005
General purpose of network monitoring:

1. What the network does with user traffic?
2. What is the current state of the network?
3. Is there some problem in the network?
Monitoring questions …

• How long it takes to transport user data?
• How many packets get lost?
• What is available bandwidth and how it fluctuates?
• Why is my TCP connection slow on this fast network?
• What applications are people using most? (is our academic network loaded by sharing movies?)
• What is the performance of the DNS system?
• Are there viruses or worms being spread in the network?
• Is somebody doing some computer network attack?
Active monitoring is a **probe** into the network

- What is travelling next to the probe?
- Is the probe experiencing the same trip as others?

Passive monitoring is a **watch**

- I can see everything, but what can I analyse from it?
Active vs. passive

- How long it takes to transport user data?
- How many packets get lost?
- What is available bandwidth and how it fluctuates?
- Why is my TCP connection slow on this fast network?
- What applications are people using most? (is our academic network loaded by sharing movies?)
- What is the performance of the DNS system?
- Are there viruses or worms being spread in the network?
- Is somebody doing some computer network attack?
Scampi

SCAMPI: Scaleable monitoring platform for the Internet

IST project (April 2002 – March 2005)

• to develop programmable monitoring hardware

• to provide platform for easy writing of portable monitoring applications
LOBSTER: Large Scale Monitoring of Broadband Internet Infrastructure

Specific Support Action (2005-2006)

- Deploy pilot passive monitoring architecture
- Add configurable anonymization
- Add support for distributed monitoring
Lobster network
Passive applications

- Detection of security attacks
  intrusion and DoS attacks
- Characterization of traffic using dynamic ports
  file sharing, etc.
- Short-timescale bandwidth
  continuous, precise, non-intrusive used/available bw.
- Packet loss rate
  for real contracted traffic
- Providing packet trace repository
  for researchers
SCAMPI architecture

Application 1
MAPI stub lib

Application 2
MAPI stub lib

Application 3
MAPI stub lib

mapid daemon

libraries & drivers

libraries & drivers

libraries & drivers

SCAMPI card

DAG card

Ethernet NIC
Scampi card, DAG card with coprocessor:

- Header filtering
- Packet sampling
- Time and length statistics
- Payload searching

Possibilities and software support differ in many details for Scampi card and DAG card
Flows and MAPI functions

- **Flow** - all packets from one monitoring point with applied sequence of monitoring functions

- **Application** - can open multiple flows

- **mapid** - automatically decides between hardware or software implementation

- One mapid can support multiple applications
fd = mapi_create_flow("/dev/scampi/0");

mapi_apply_function(fd, "BPF_FILTER", "src net 192.168.1.0/24");
mapi_apply_function(fd, "STR_SEARCH", "virus");
fid = mapi_apply_function(fd, "PKT_COUNTER");

mapi_connect(fd);

while(1) {
    sleep(5);
    cnt = mapi_read_results(fd, fid, MAPI_REF);
    printf("Packets with virus: %d\n", *cnt);
}
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPF_FILTER</td>
<td>header filtering</td>
</tr>
<tr>
<td>PKT_COUNTER</td>
<td>packet counter</td>
</tr>
<tr>
<td>BYTE_COUNTER</td>
<td>byte counter</td>
</tr>
<tr>
<td>STR_SEARCH</td>
<td>payload searching</td>
</tr>
<tr>
<td>TO_BUFFER</td>
<td>reading packets</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>packet sampling</td>
</tr>
<tr>
<td>HASHSAMP</td>
<td>hash-based packet sampling</td>
</tr>
<tr>
<td>TO_FILE</td>
<td>storing packet trace to disk</td>
</tr>
<tr>
<td>ETHEREAL</td>
<td>ethereal-based filtering</td>
</tr>
<tr>
<td>HASH</td>
<td>packet hash computing</td>
</tr>
<tr>
<td>COOKING</td>
<td>TCP/IP reassembly</td>
</tr>
<tr>
<td>BUCKET</td>
<td>divides packets into buckets based on their timestamps</td>
</tr>
<tr>
<td>THRESHOLD</td>
<td>signals when a threshold is reached</td>
</tr>
</tbody>
</table>
DiMAPI – Distributed MAPI

- **Scope** – a set of remote monitoring sensors

- **Flow** – can span a scope

- DiMAPI monitoring functions:
  a) Can be passed to monitoring sensors  
     e.g., header filtering
  b) Can be applied to a scope as the whole  
     e.g., aggregation or distance-based metrics
Required anonymization
Anonymization options

mapi_apply_function(fd, “ANONYMIZE”, protocol, field, function, args);

STRIP
HASHED
RANDOM
MAP
MAP_DISTRIBUTION
PREFIX_PRESERVING

- first tier anonymization (common to all flows, preferably in HW)
- second tier anonymization (flow-specific)
Anonymization, cont

Admin

Policy Editor → Policy Manager → Policy Repository

User

authd → mapid

MAPI Application

Monitoring Sensor
Credential example

Authorizer: "RSA:abc123" # Admin’s key
Licensees: "RSA:xyz999" # User’s key
Conditions: ANONYMIZE == “defined” &&
   ANONYMIZE.0.pos == 0 &&
   ANONYMIZE.0.param.0 == IP &&
   ANONYMIZE.0.param.1 == SRC_IP &&
   ANONYMIZE.0.param.2 == PREFIX_PRESERVING &&
   ANONYMIZE.1.pos == 1 &&
   ANONYMIZE.1.param.0 == IP &&
   ANONYMIZE.1.param.1 == DST_IP &&
   ANONYMIZE.1.param.2 == PREFIX_PRESERVING &&
   ANONYMIZE.2.pos == 0 &&
   ANONYMIZE.2.param.0 == TCP &&
   ANONYMIZE.2.param.1 == PAYLOAD &&
   ANONYMIZE.2.param.2 == STRIP;
Signature: "RSA-SHA1:213344f9"
Lobster resources

• Project website http://www.ist-lobster.org

• Subscribe to lobster-news@ist-lobster.org

• Lobster tutorial on RIPE 50 meeting, Stockholm, 6 May 2005

• 1st Lobster workshop on TNC2005, Poznan, 7 June 2005

Thank you for your attention