Triggering Execution Environments of Active Networks without ANEP

Eric Y. Chen
NTT Information Sharing Platform Laboratories
Sept 24, 2001
1. An Overview of Active Networks
2. Triggering EEs without ANEP
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Two Categories

In-band
- Embed code into packets
- Code executed in each router on the fly

Out-of-band
- Code pre-loaded into routers
- Packets specify code ID
Triggering the EEs

**Implications**

- Active packets demand active services
- End-applications must be AN-aware
- Conventional packets and applications are not associated with active services
This allows

- Customize the way the network handles conventional packets on a per host or per network basis
- A broader definition for active traffic
Presentation Agenda

1. An Overview of Active Networks
2. Triggering EEs without ANEP
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Presentation Agenda

1. An Overview of Active Networks
2. Triggering EEs without ANEP
   2.1 The Basic Mechanism
   2.2 The Security Model
   2.3 Conflict Management
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks

NTT Information Sharing Platform Laboratories
Active code can manipulate inward and outward packets associated with the code owner.
The Mathematical Model

- I: The entire IP address space
- O: Set of IP addresses owned by a user
- (s,d): a packet from s to d
- An active code can manipulate packets
  \[ A = \{(s,d) \in [(O \times I) \cup (I \times O)] | s \neq d\} \]
- Actual packets manipulated \( C \subseteq A \)
- Definition of active packets: \( (s,d) \in \bigcup_{i=1}^{n} C_i \)
# Routing in an Active Node

<table>
<thead>
<tr>
<th>Type</th>
<th>Destination</th>
<th>Source</th>
<th>Send to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>1.2.3.4</td>
<td>Any</td>
<td>Active Code A</td>
</tr>
<tr>
<td>Active</td>
<td>10.50.0.0</td>
<td>11.12.13.14</td>
<td>Active Code B</td>
</tr>
<tr>
<td>Active</td>
<td>Any</td>
<td>157.2.3.0</td>
<td>Active Code C</td>
</tr>
<tr>
<td>Regular</td>
<td>1.2.0.0</td>
<td></td>
<td>29.15.20.1</td>
</tr>
<tr>
<td>Regular</td>
<td>11.20.0.0</td>
<td>N/A</td>
<td>199.1.1.10</td>
</tr>
<tr>
<td>Regular</td>
<td>199.1.1.0</td>
<td></td>
<td>120.0.0.1</td>
</tr>
</tbody>
</table>

NTT Information Sharing Platform Laboratories
Routing in an Active Node

ACTIVE ROUTER

Active Network Execution Environment

Multiplex

Interface A

Interface B

Interface C

Active Code A

Active Code B

Active Code C

Active Code D

Active Code E

NTT Information Sharing Platform Laboratories
Routing in an Active Node

Active Network Execution Environment

ACTIVE ROUTER

Interface A
Interface B
Interface C

Active Code A
Active Code B
Active Code C
Active Code D
Active Code E

Multiplex
Presentation Agenda

1. An Overview of Active Networks
2. Triggering EEs without ANEP
   2.1 The Basic Mechanism
   2.2 The Security Model
   2.3 Conflict Management
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Security Model

1) Key pair generation
   - public key
   - private key

3) Sign AC by private key
   - Active Code A
     (handles packets destined to IP 1.2.3.4)
   - Signed by user A

4) Dispatch Active Code A

5) Obtain Certificate

6) Verification of signature and access rights

User A
(IP 1.2.3.4)

Node

ISP/CA

Active Code A

Signed by user A

Active Code A
(handles packets destined to IP 1.2.3.4)

Signed by user A

NTT Information Sharing Platform Laboratories
Assumptions

- Active nodes are well known to the user. The user always knows where to inject active code.
- Active nodes always obtain the public key of CA in advance.
1. An Overview of Active Networks

2. Triggering EEs without ANEP
   2.1 The Basic Mechanism
   2.2 The Security Model
   2.3 Conflict Management

3. Prototype Implementation

4. Example Applications

5. Concluding Remarks
(s,d) ∈ C_i ∩ C_j such that (s ∈ O_k ∧ d ∈ O_l) ∧ (k ≠ l)

Receiver’s active code always override sender’s.
(s, d) ∈ C_i ∩ C_j such that (s ∈ O_k ∧ s ∈ O_k) ∨ (d ∈ O_k ∧ d ∈ O_k)

- Old active code overrides new one.

Old Overrides New

Alice's Active Code

Alice's Active Code

Alice's IP

Destination

Source

NTT Information Sharing Platform Laboratories
1. An Overview of Active Networks
2. Triggering EEs without ANEP
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Prototype Implementation

- Active Code
- Execution Environment
- IP
- JVM

JNI
- libpcap
- libnet

process
kernel

packet capture

data link

incoming packets

copy of incoming packets

disable routing (ip_forward=0)

"active packets"

regular packets

Active Code

incoming packets

outgoing packets

NTT Information Sharing Platform Laboratories
1. An Overview of Active Networks
2. Triggering EEs without ANEP
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Distributed Denial of Service

NTT Information Sharing Platform Laboratories
Conventional Firewall

DDoS Attacker

Router

DDoS Attacker

Router

DDoS Attacker

Router

DDoS Attacker

Router

Gateway

LAN

Server

NTP Information Sharing Platform Laboratories
Distributed Firewall

NTT Information Sharing Platform Laboratories
Traffic Converter

Low Bandwidth Connection

VoD Client

Router

Active Code

EE

Legacy VoD Server

Degraded Data

Original Data

NTT Information Sharing Platform Laboratories
1. An Overview of Active Networks
2. Triggering EEs without ANEP
3. Prototype Implementation
4. Example Applications
5. Concluding Remarks
Issues and Future Work

- Need a mechanism to work with DHCP
- Integrate with other existing AN architectures
Questions?

eric@nttmhs.tnl.ntt.co.jp