# Counting Packets Sent Between Arbitrary Internet Hosts

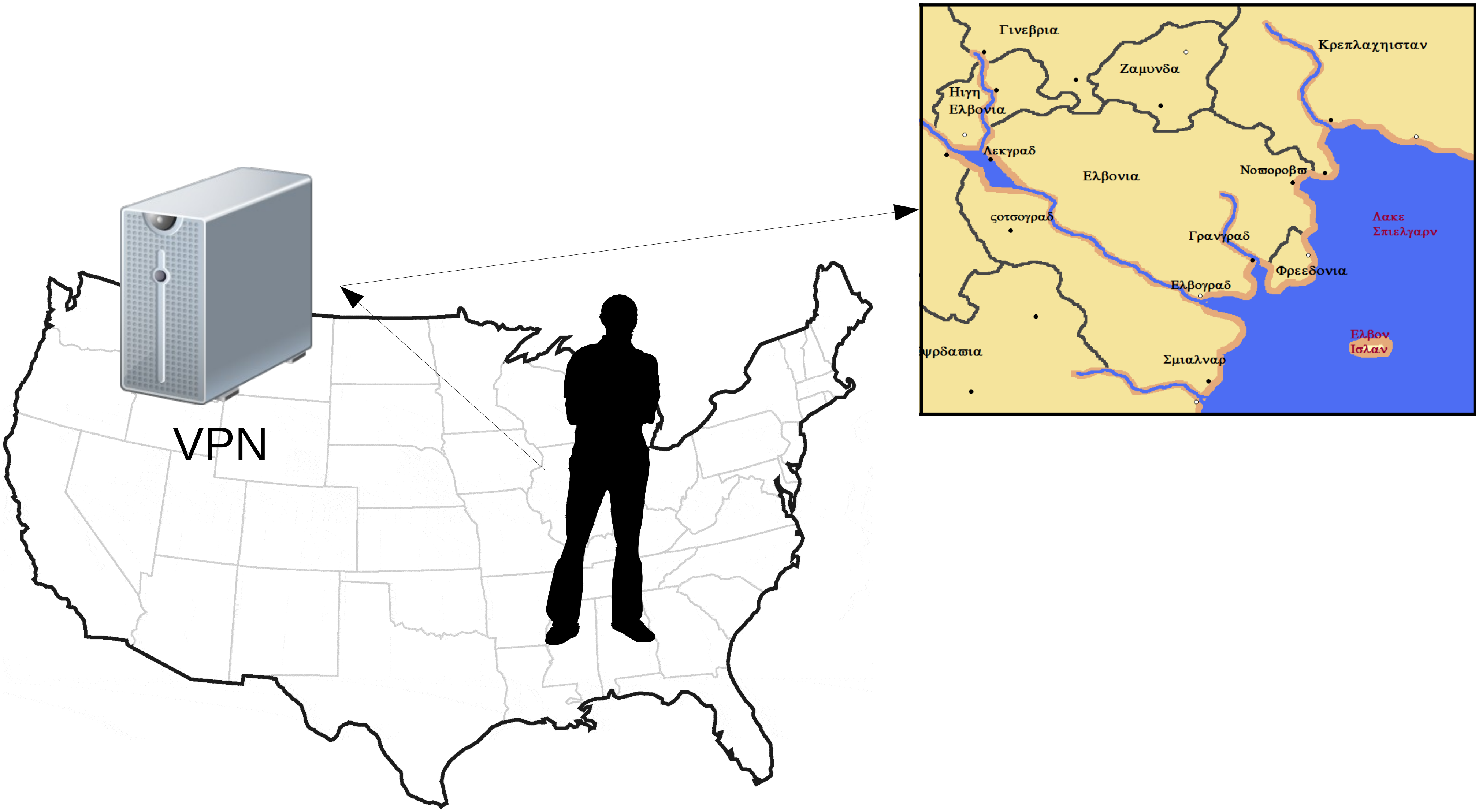
Jeffrey Knockel
Jedidiah R. Crandall

Department of Computer Science
University of New Mexico

# The Side Channel Attack

- We can count # of packets sent between arbitrary hosts on the Internet
- **ICMP/UDP:**
  - Count # of packets a linux machine sends to some other machine
- **TCP:**
  - Determine if some machine is connected to a linux server

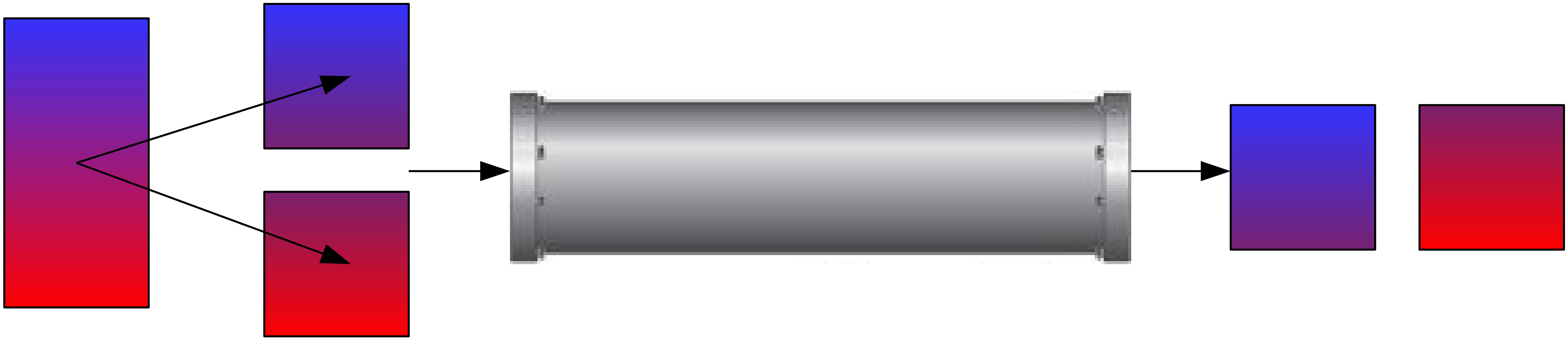
# Scenario 1



VPN

# Background

- **Packet spoofing.** A *spoofed* packet has the return IP address of another machine
- **IP fragmentation.** IP datagrams are split into *fragments* when they are too large to go over a medium



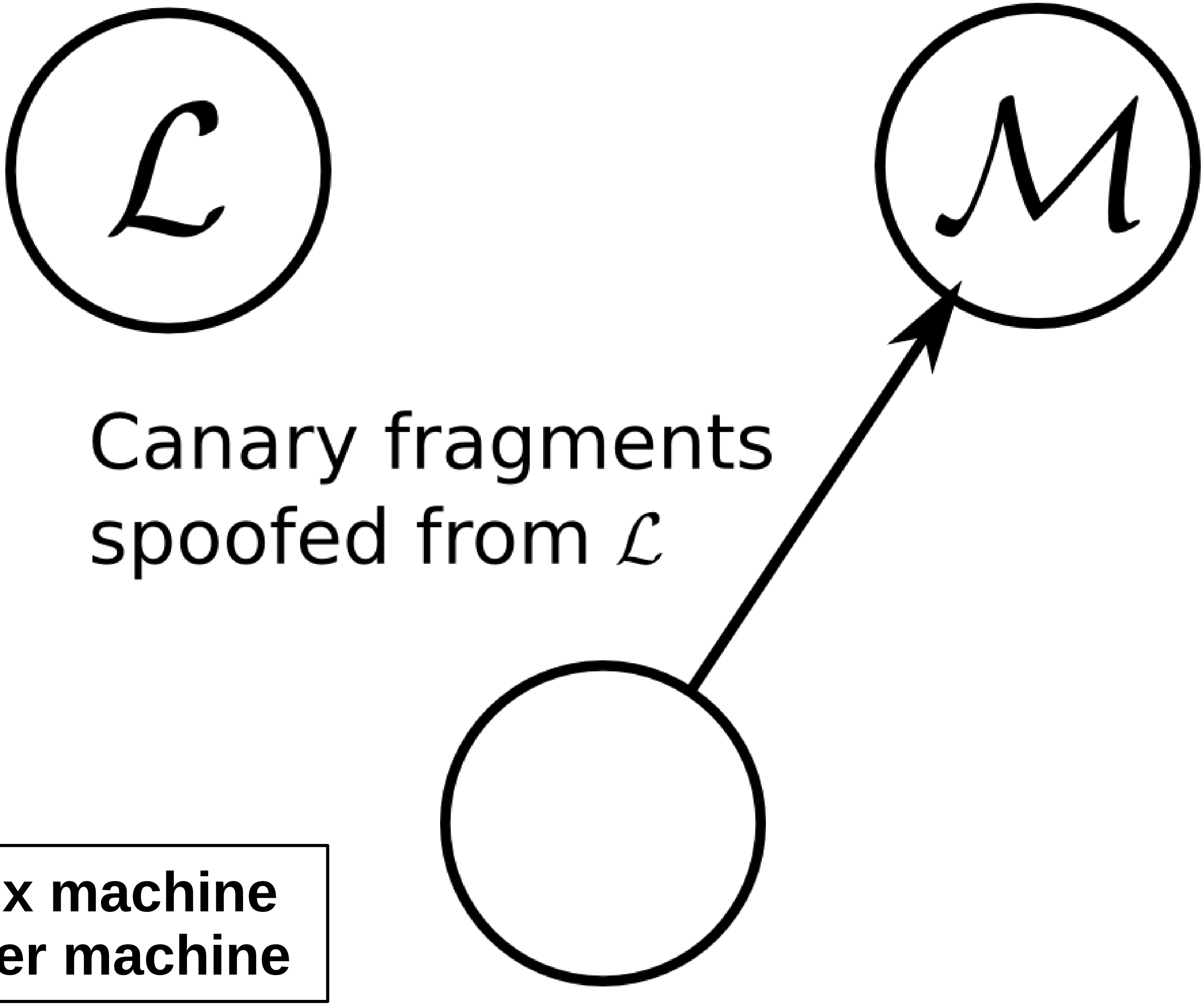
# IP Reassembly

- Some fragments are lost or reordered
- Fragments are kept in a *fragment cache* until all fragments arrive and the datagram is complete
- But they have *finite storage space*
- Side channel!

# IP ID counters

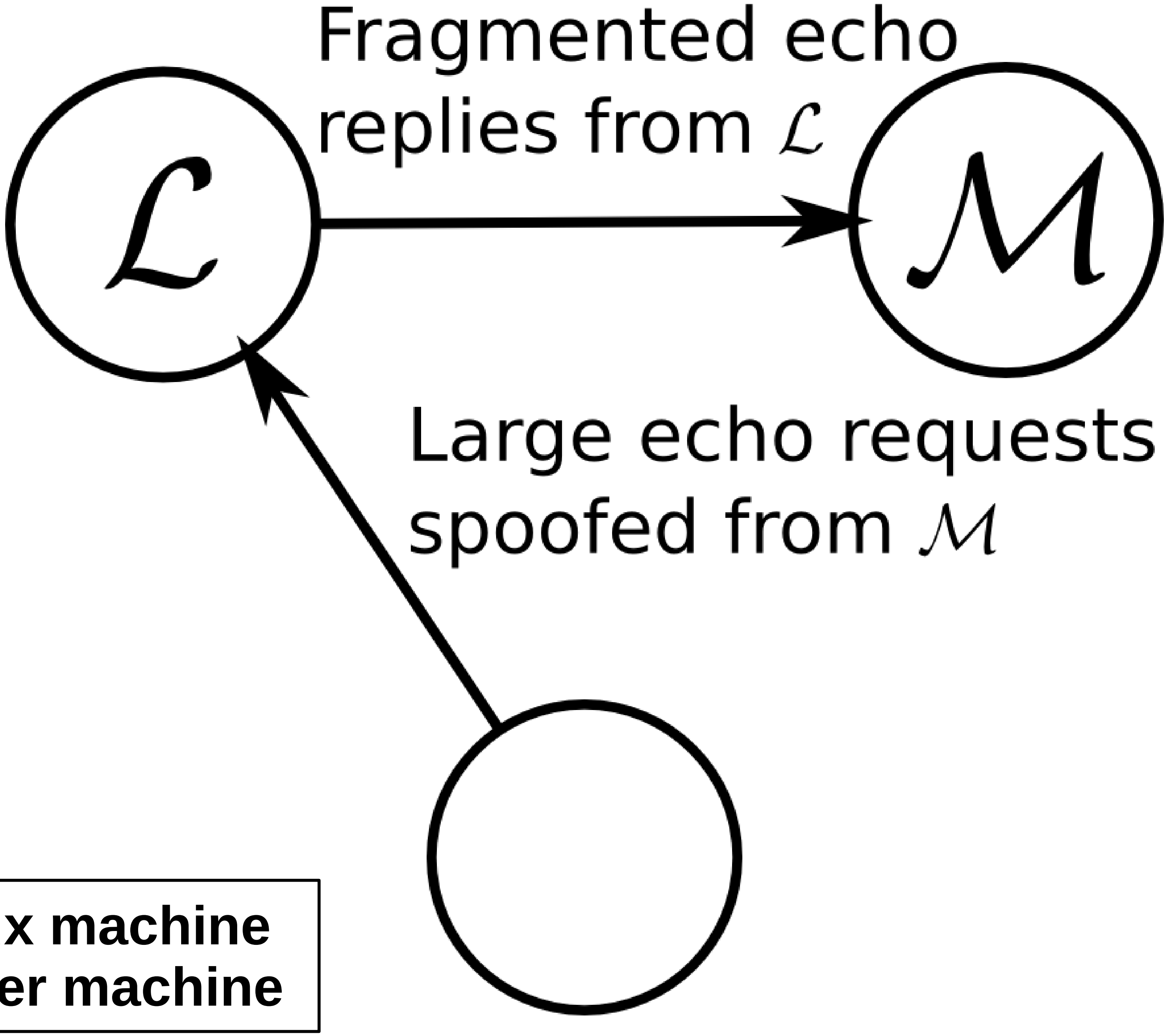
- IP ID's distinguish which datagram fragments belong to
- Global counter → idle scans
  - Port scan from vantage of a “zombie”
- Linux:
  - Per-flow counters (TCP)
  - Per-destination counters (ICMP/UDP, some TCP)
- *We can measure per-destination counters' values*

# Planting canaries

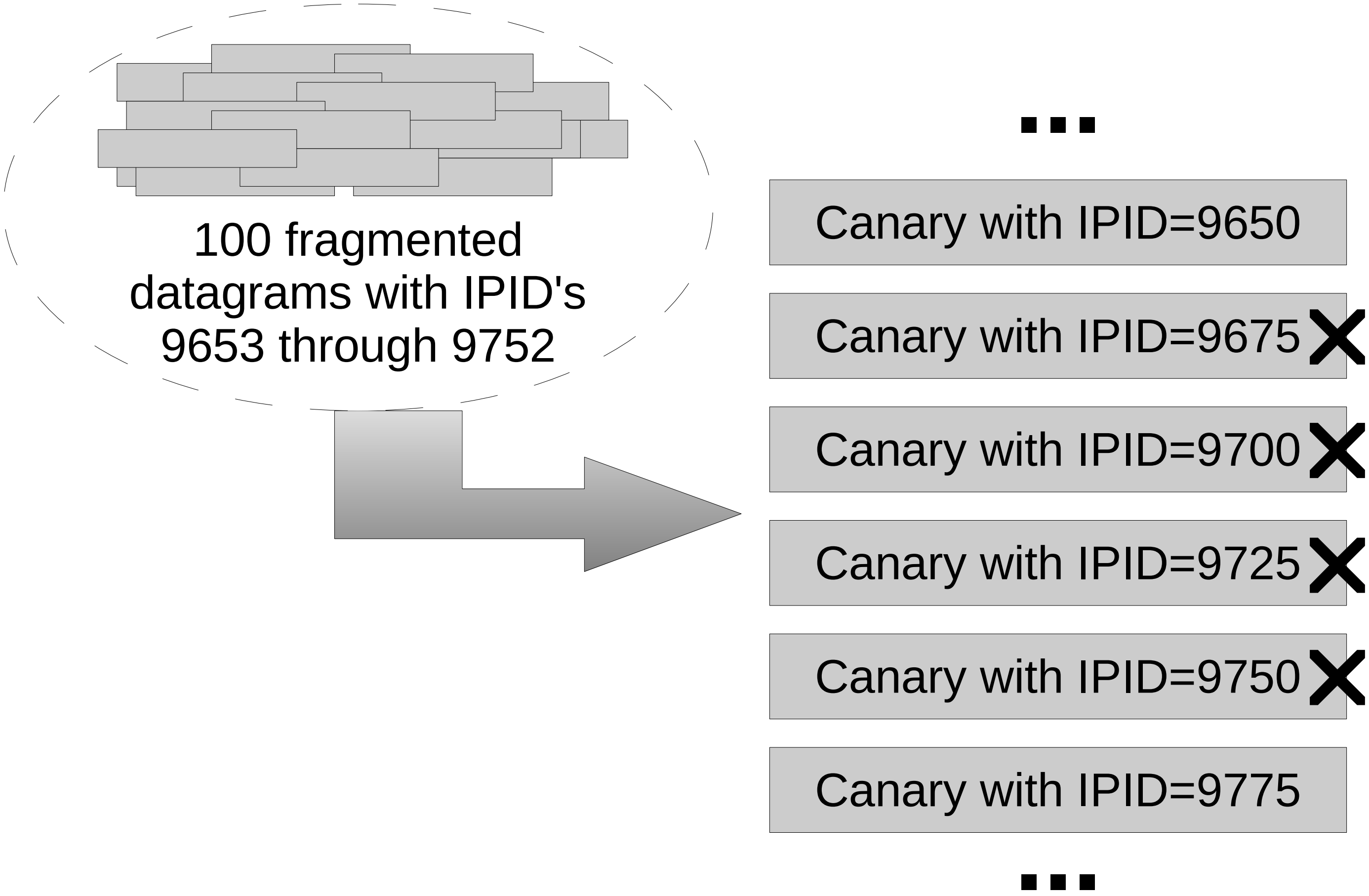


**$\mathcal{L}$: Linux machine**
**$\mathcal{M}$: Other machine**

# Knocking out canaries


Fragmented echo replies from $\mathcal{L}$
$\mathcal{L}$
$\mathcal{M}$
Large echo requests spoofed from $\mathcal{M}$

**$\mathcal{L}$: Linux machine**
**$\mathcal{M}$: Other machine**



100 fragmented datagrams with IPID's 9653 through 9752

...

Canary with IPID=9650

Canary with IPID=9675 ✗

Canary with IPID=9700 ✗

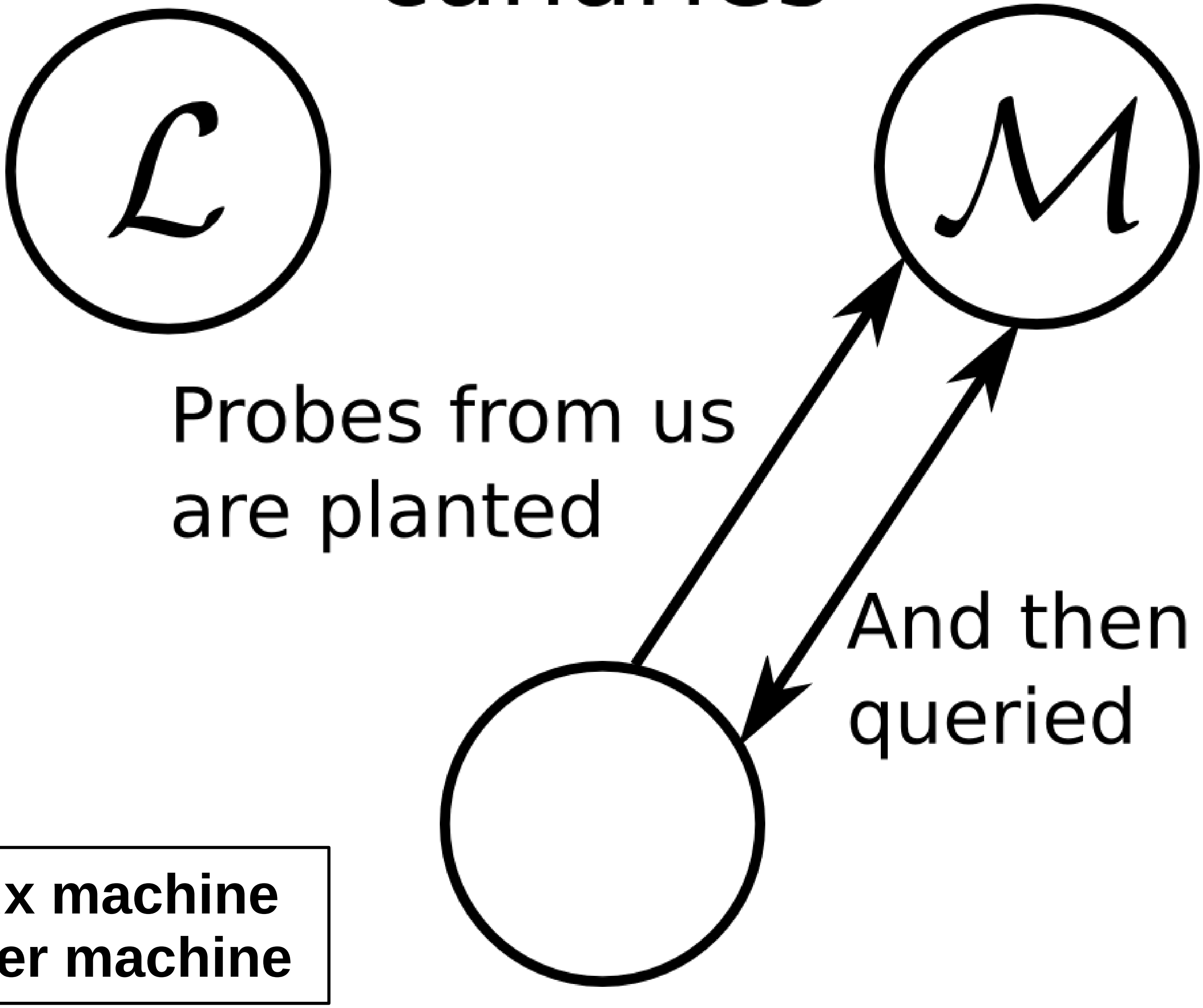
Canary with IPID=9725 ✗

Canary with IPID=9750 ✗

Canary with IPID=9775

...

# Measuring missing canaries



Probes from us are planted

And then queried

$\mathcal{L}$: **Linux machine**
$\mathcal{M}$: **Other machine**

# No Canaries Missing

| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| **Canary** | |
| **Canary** | |
| **Echo request** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **Echo request** | |

- Fill rest of $\mathcal{M}$'s fragment cache with probes

# No Canaries Missing

| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| **Canary** | |
| **Canary** | |
| **Echo request** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **Echo request** | |

- Fill rest of $\mathcal{M}$'s fragment cache with probes
- Query probes

# No Canaries Missing

| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| **Canary** | |
| **Canary** | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

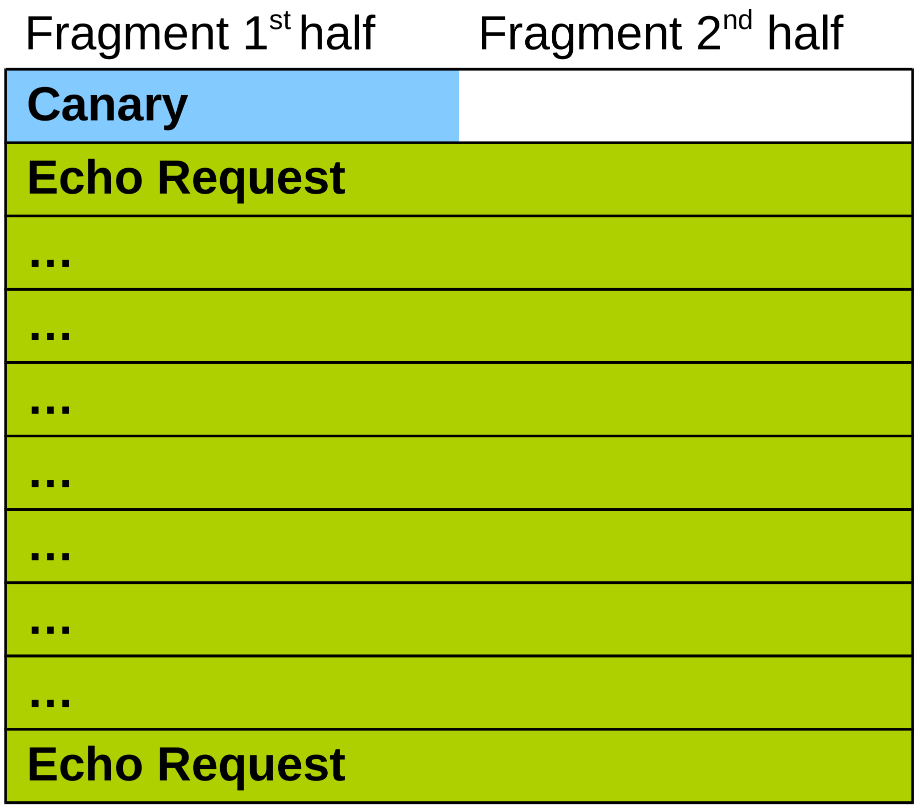
- Fill rest of $\mathcal{M}$'s fragment cache with probes
- Query probes
- Seven responses

# Two Canaries Missing

| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| **Echo Request** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **Echo Request** | |

- Fill rest of $\mathcal{M}$'s fragment cache with probes

# Two Canaries Missing



| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| **Echo Request** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **…** | |
| **Echo Request** | |

- Fill rest of $\mathcal{M}$'s fragment cache with probes
- Query probes

# Two Canaries Missing

| Fragment 1st half | Fragment 2nd half |
|---|---|
| **Canary** | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

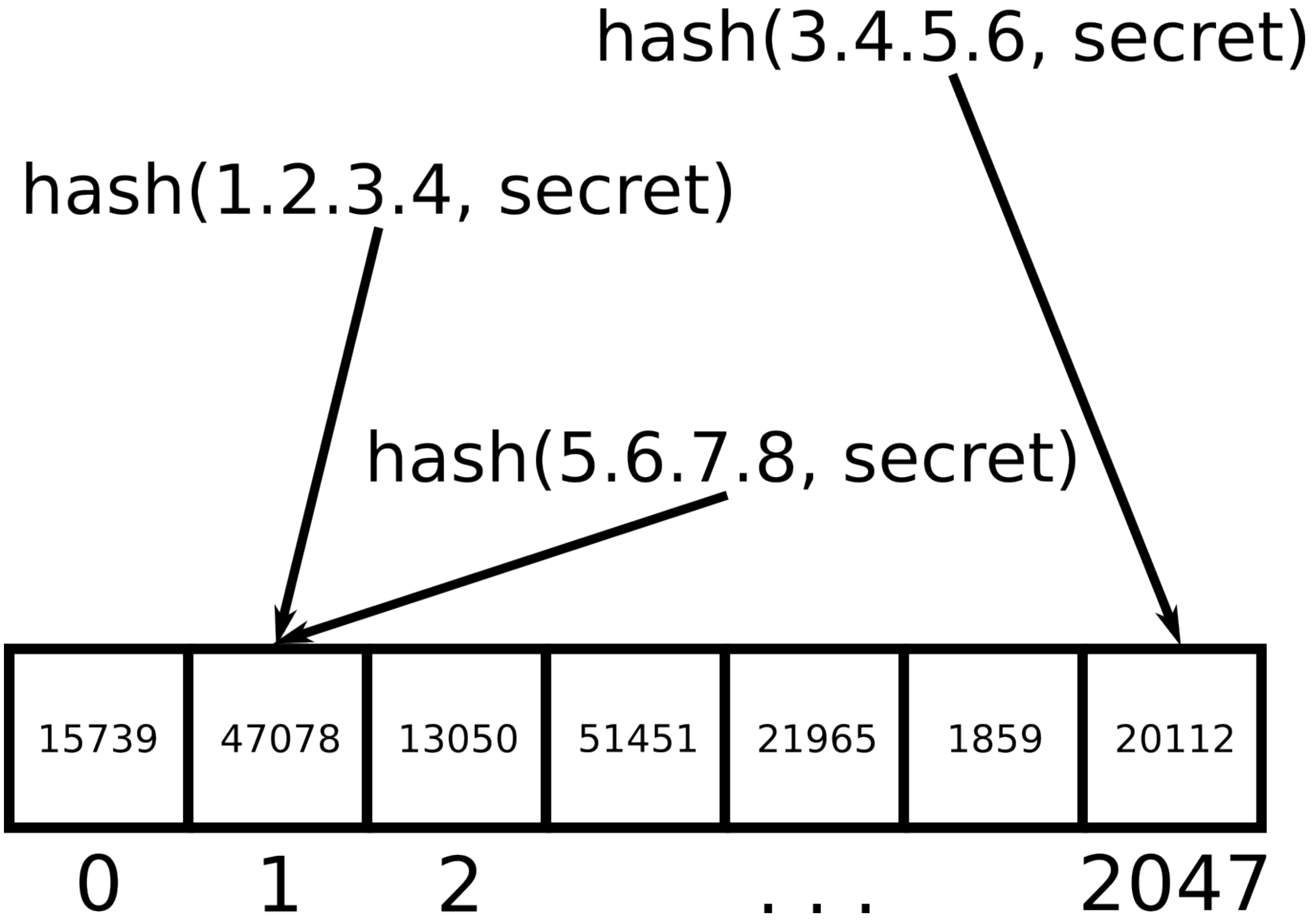
- Fill rest of $\mathcal{M}$'s fragment cache with probes
- Query probes
- Nine responses

# Inferring communication

- Binary search all $2^{16}$ IPv4 ID space
- ICMP/UDP
- TCP
  - **Naive way:** send ACK's
    - Connection → Returns ACK from per-flow counter
    - No connection → Returns RST from per-dst counter
  - **TIME-WAIT way:** send SYN's
    - TIME-WAIT → Returns ACK from per-dst counter
    - No connection → Returns SYNACK with IPID zero

# security@kernel.org

# Hash to one of 2048 counters



hash(3.4.5.6, secret)

hash(1.2.3.4, secret)

hash(5.6.7.8, secret)

| | | | | | | |
|---|---|---|---|---|---|---|
| 15739 | 47078 | 13050 | 51451 | 21965 | 1859 | 20112 |
| 0 | 1 | 2 | | . . . | | 2047 |

# Hash to one of 2048 counters

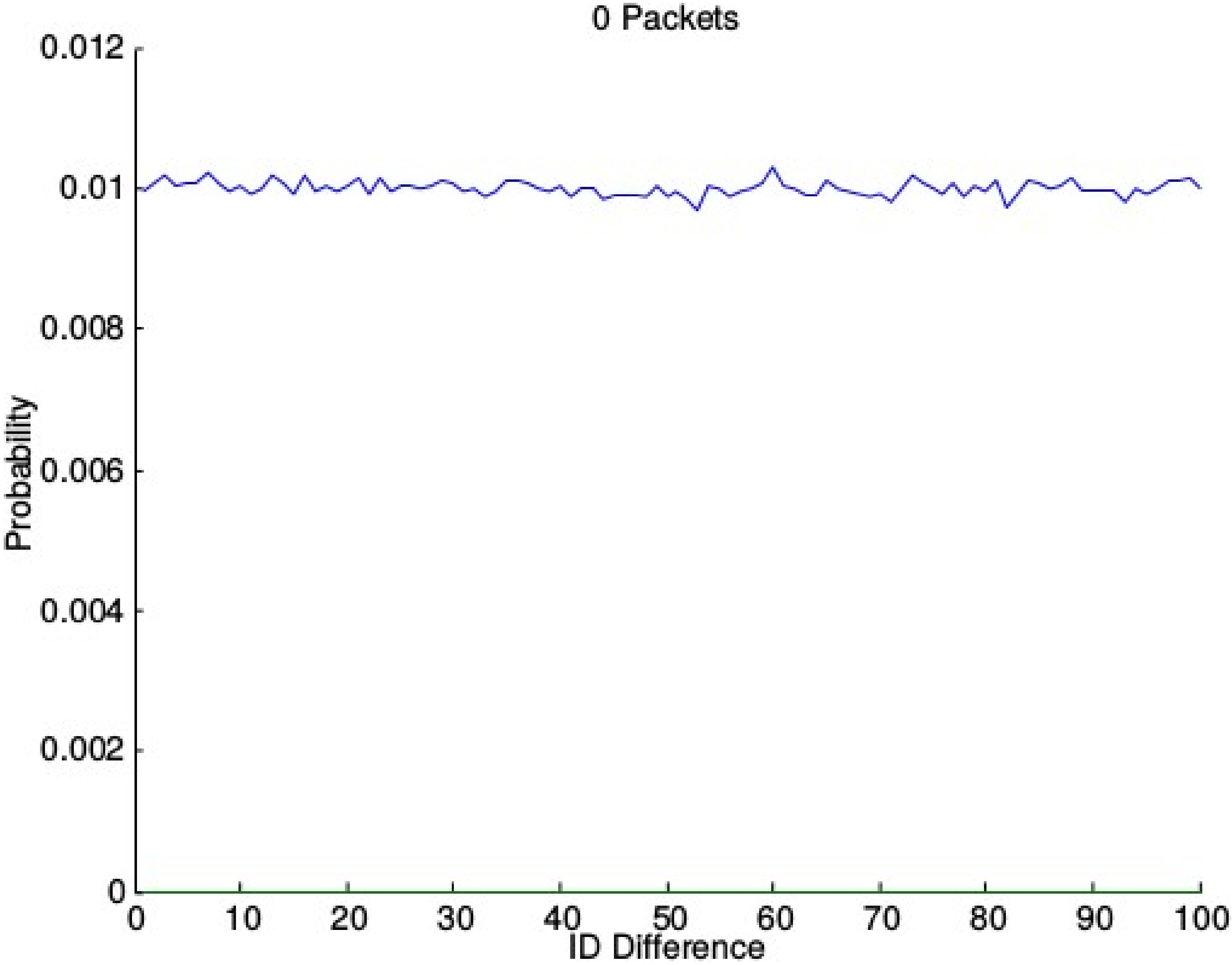
- Committed before we reported issue
- *Performance* reasons, not security reasons
- **Pro anonymity:** Adds noise to counters
  - Good for large number of possible users
- **Con anonymity:** Side channel no longer necessary
  - Bad if attacker can read packets sent to many addresses

# Add randomness

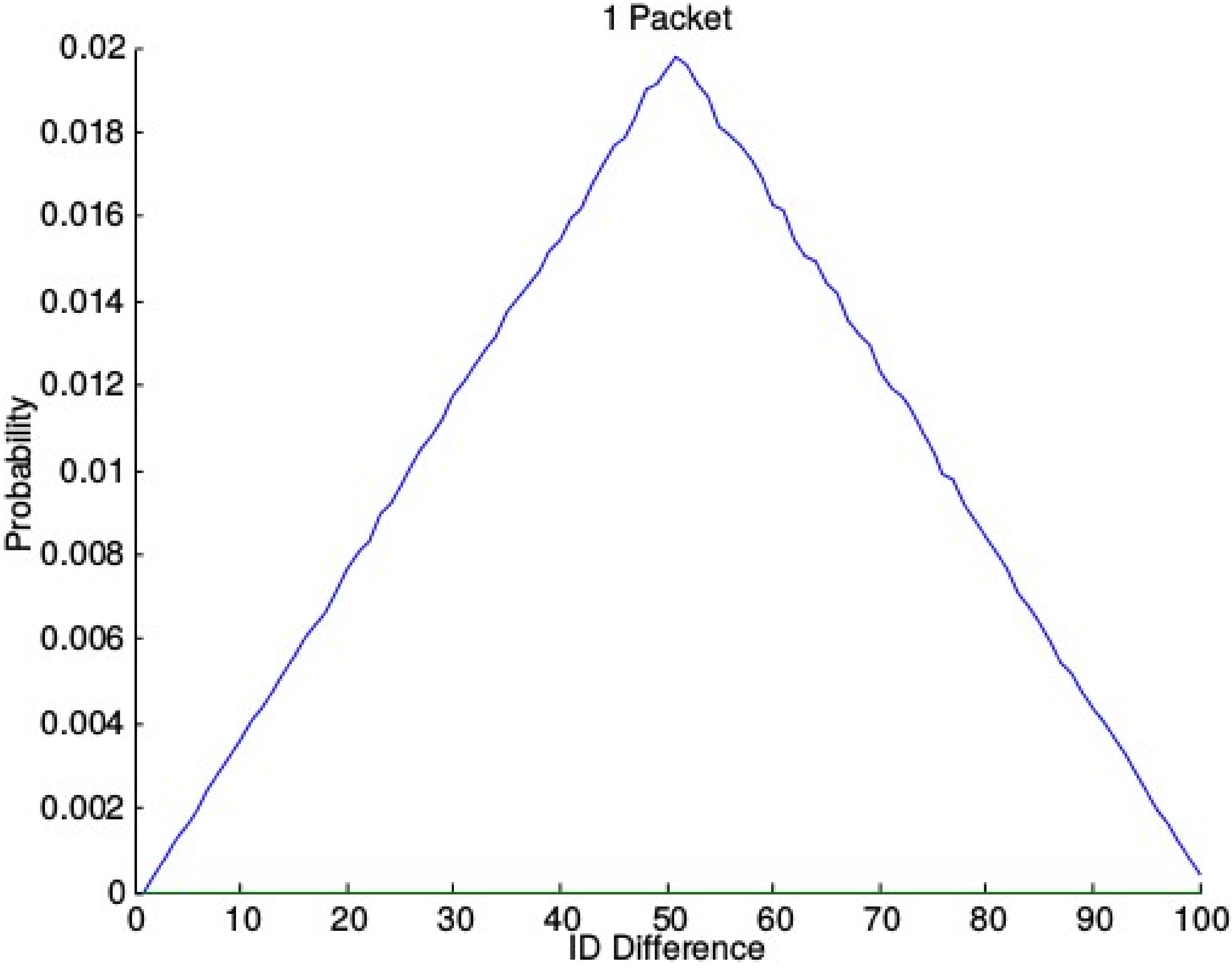
- Hash changed to isolate protocol:
  - hash(dst, src, protocol, secret)
- Add randomness
  - Before every access to counter, add

    *randint*(time since last access)
- Large # of packets can drown out randomness
- Small # of packets still leave a signal…

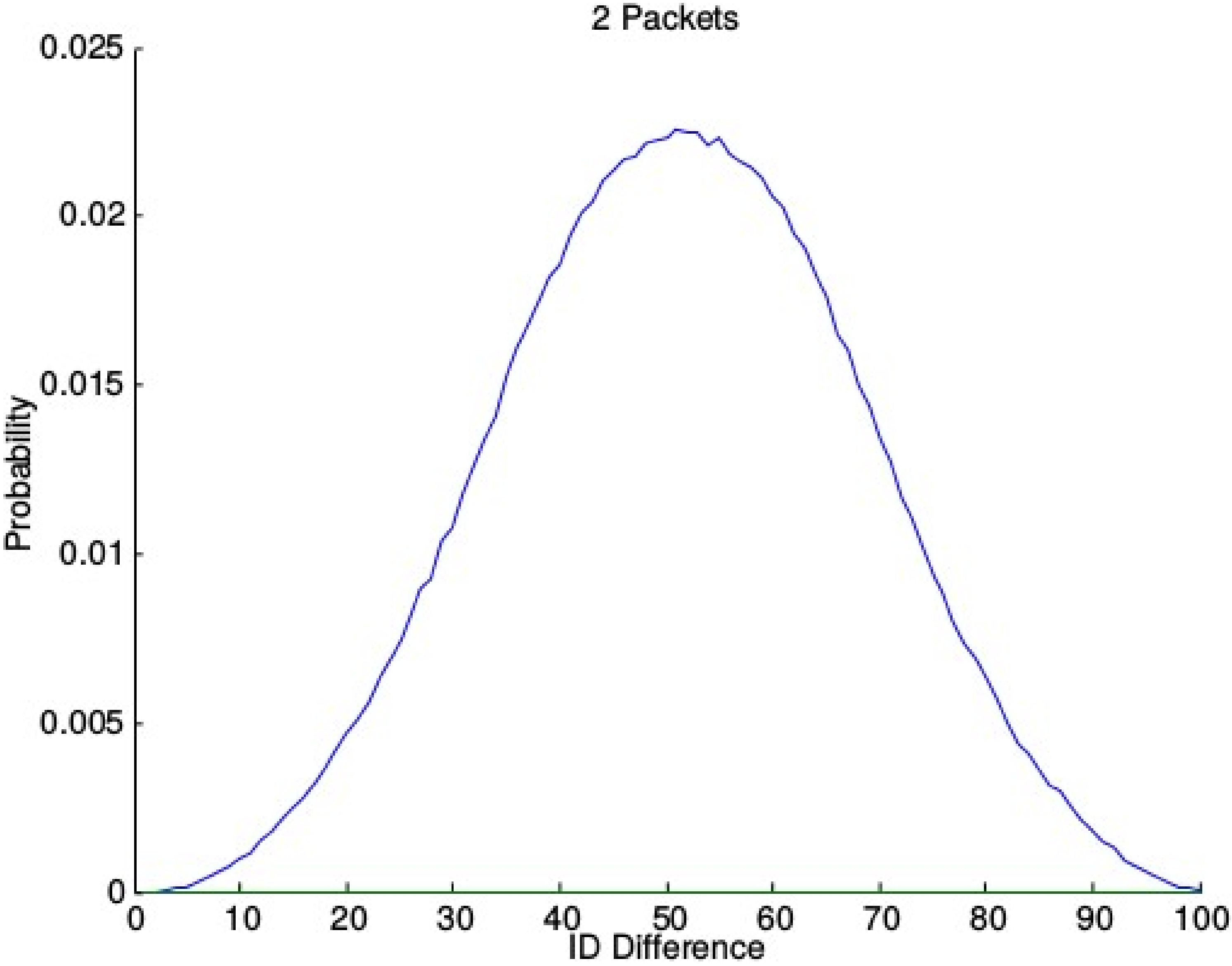
# randint(100)


0 Packets
Probability
ID Difference
0.012
0.01
0.008
0.006
0.004
0.002
0
0
10
20
30
40
50
60
70
80
90
100

# randint(50) + randint(50)

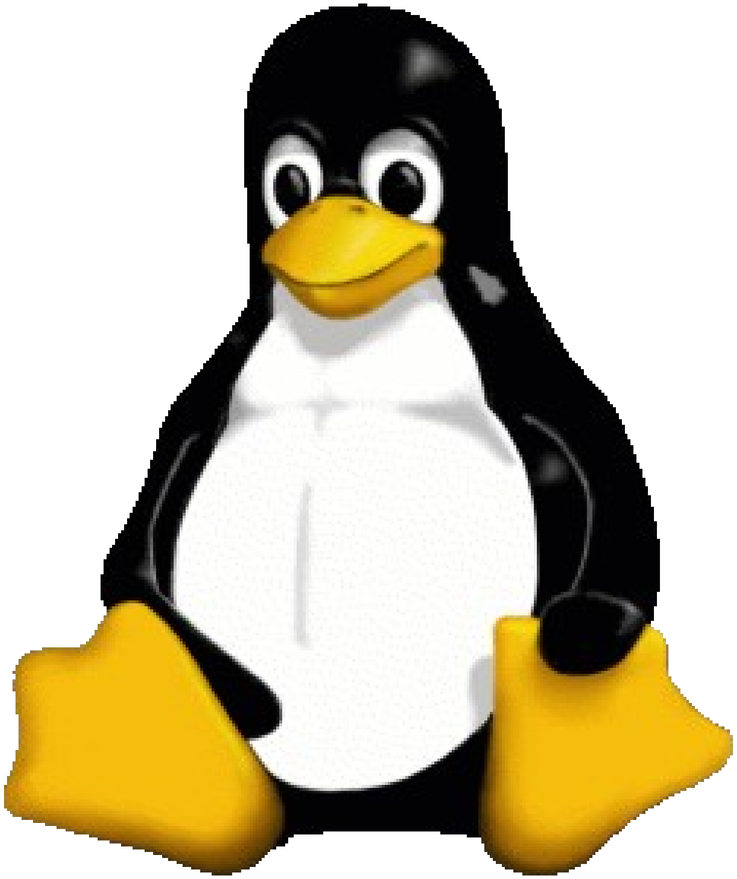


# randint(33) + randint(33) + randint(33)



# Patched kernels

- 3.16+
- 3.15.(10+)
- 3.14.(17+)
- 3.10.(53+)
- 3.4.(103+)
- But vulnerable to multiple addresses!



# Distros: your mileage may vary

# Conclusion

- SSL is broken?

***IP is broken!***

- IPID's must be unique for every in-flight packet
  → **information flow**

# Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. #0844880, #0905177, #1017602, and #1314297.  Jed Crandall is also supported by the Defense Advanced Research Projects Agency CRASH program under grant #P-1070-113237.