Anonymity in XIA

Nicolas Feltman

David Naylor

for Minecfags

Goals

Examine existing methods

DOES XIA BREAK ANYTHING?

Explore new methods

WHAT NEW THINGS DOES XIA LET US DO?

Consider users and developers

HOW CAN WE HELP DEVELOPERS LEVERAGE THESE TOOLS?

HOW CAN WE HELP USERS UNDERSTAND THEM?

(Brief) XIA Review

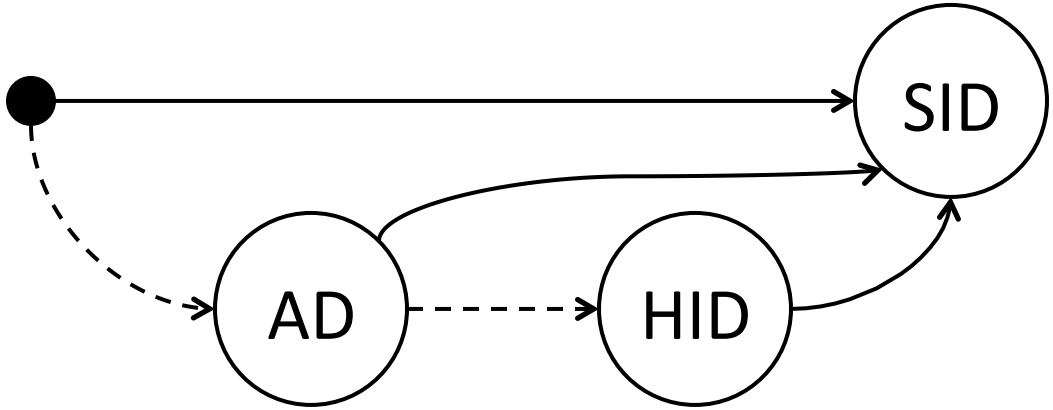
Multiple communication *principals*

E.G., HOSTS, SERVICES, AND CONTENT

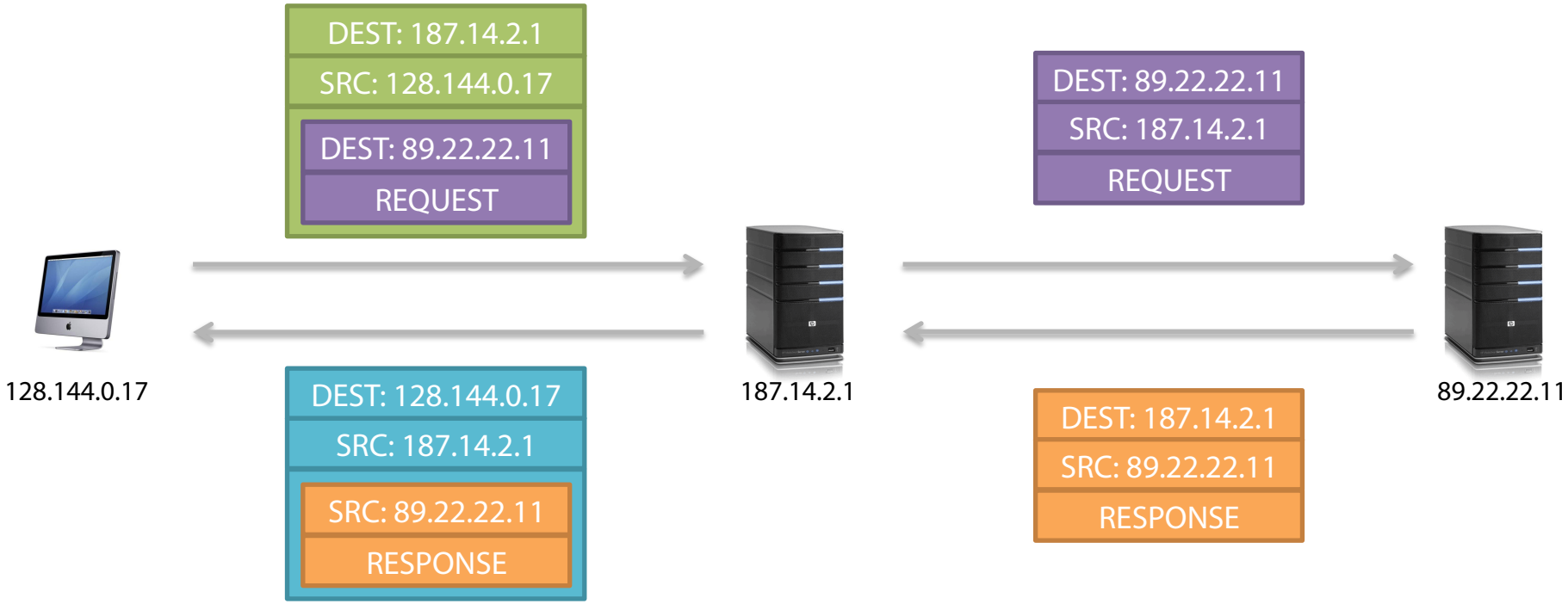
DAG-based addressing

AFFORDS SENDER SOME CONTROL OVER ROUTING

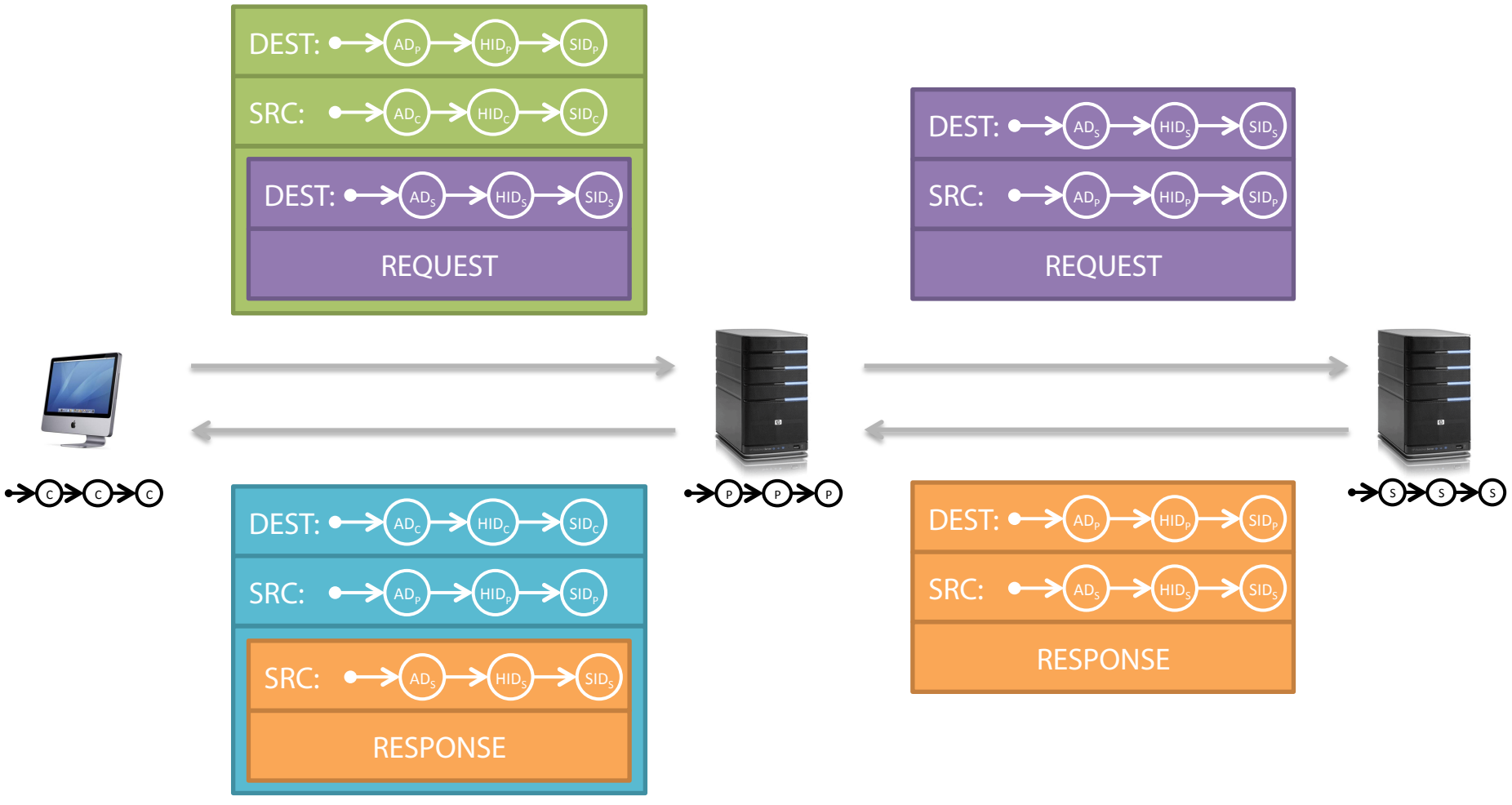
ALLOWS *FALLBACKS* TO BE GRACEFULLY IMPLEMENTED



Existing Approach: Proxy



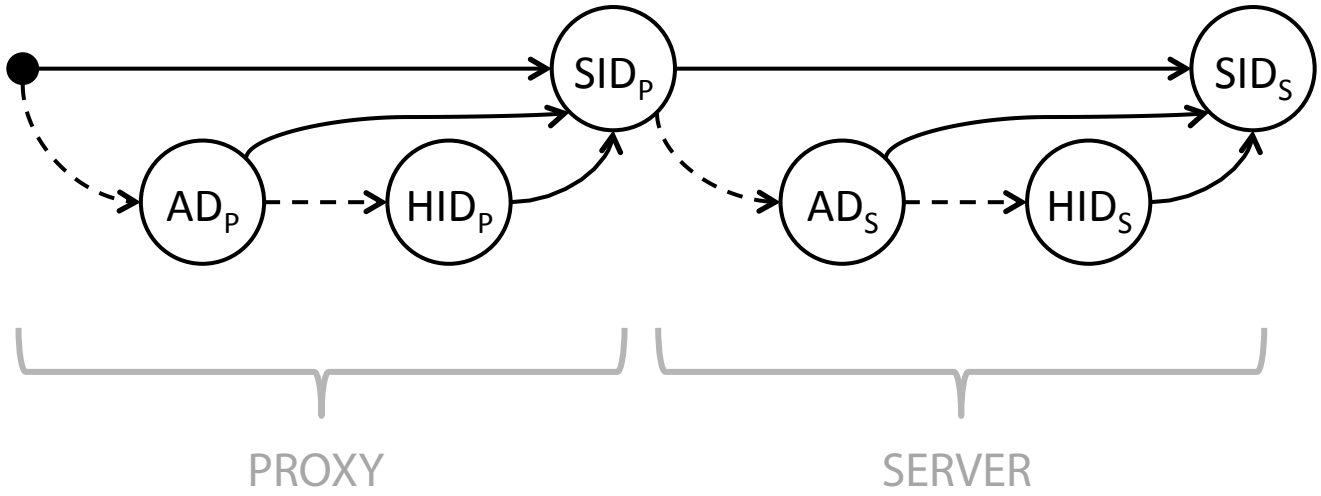
Existing Approach: Proxy



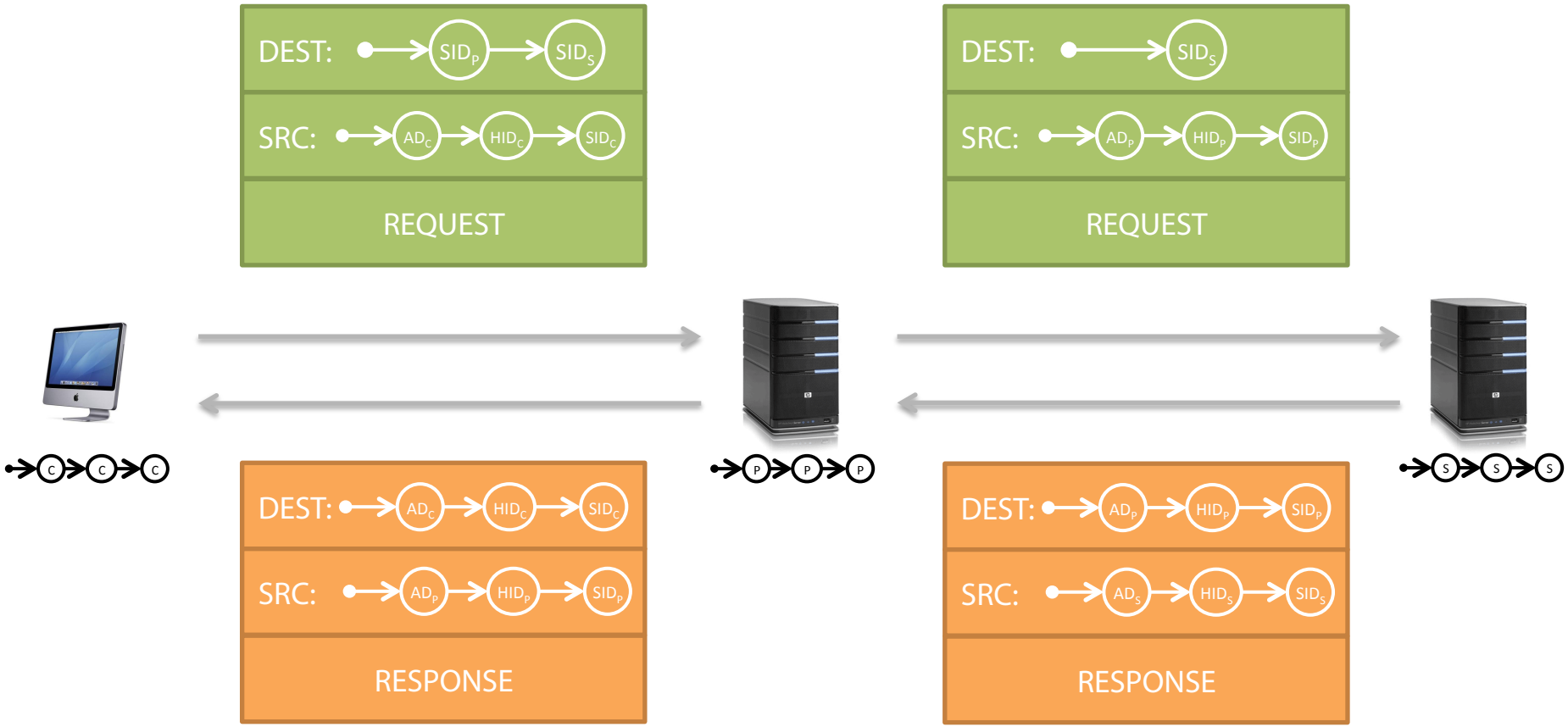
Existing Approach: Proxy

With DAGs, we can be cleaner

WE CAN EXPRESS OUR INTENT TO HAVE A PACKET SENT FIRST THROUGH AN ANONYMIZER, THEN TO THE FINAL DESTINATION

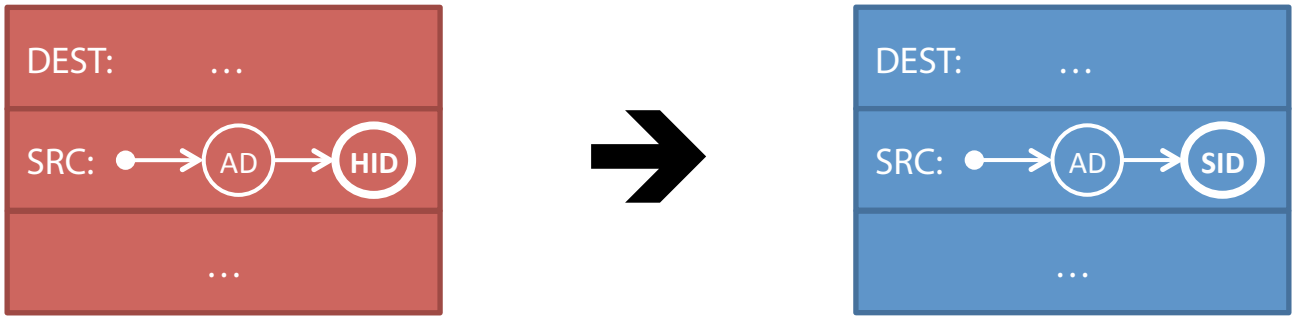


Existing Approach: Proxy



New Approach: Temporary IDs

Use a temp. service ID in place of host ID



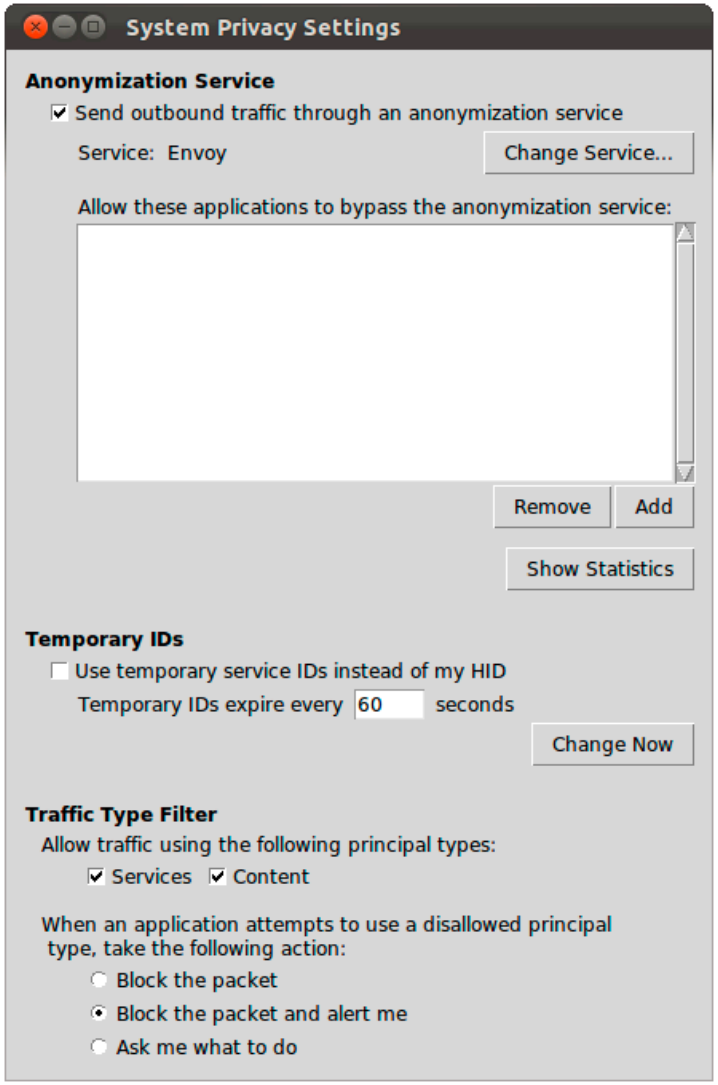
Register temp. SID with local AD

PLACES TRUST IN LOCAL AD INSTEAD OF REMOTE 3RD PARTY

Control and Transparency

Global Controls

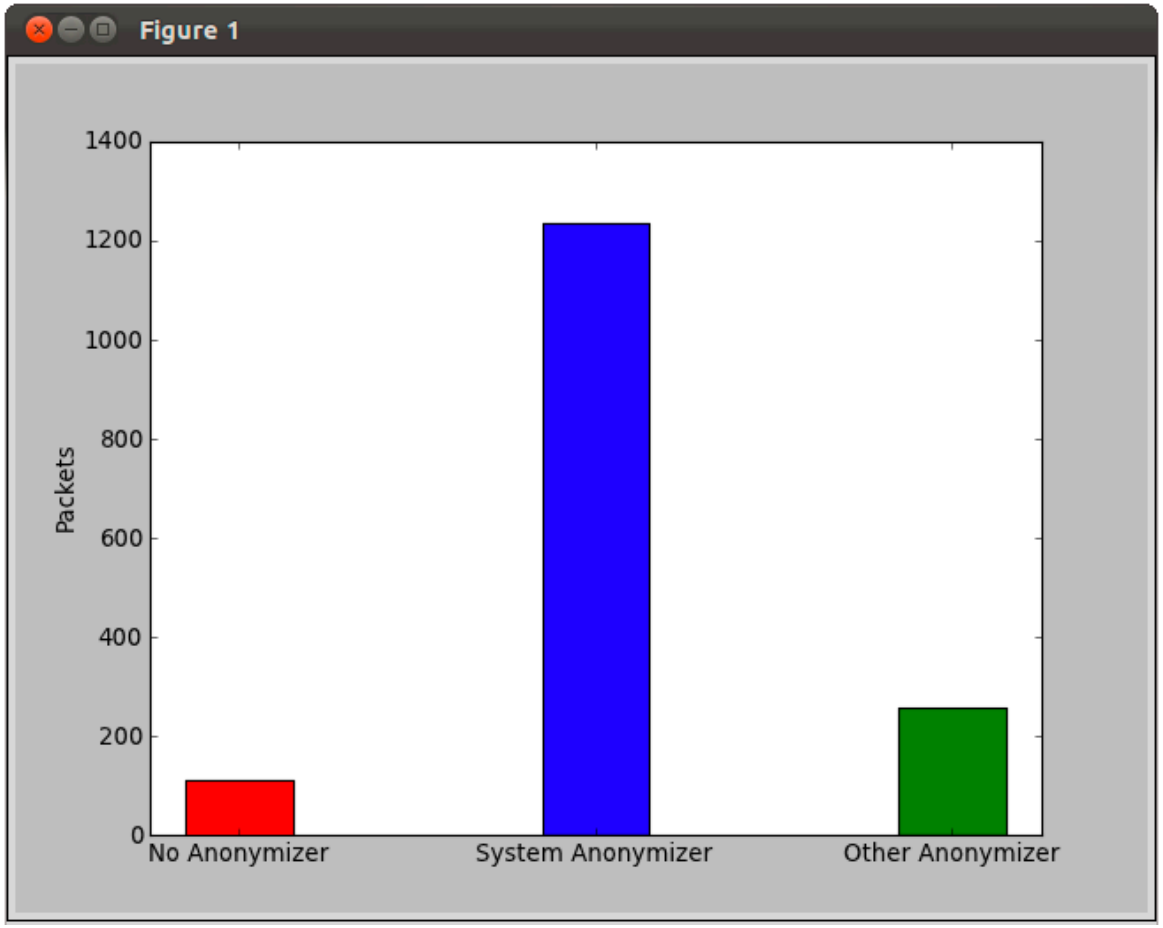
CONTROL ANONYMITY SETTINGS AT THE OS LEVEL



Control and Transparency

Transparency

USERS CAN SEE EXACTLY WHAT'S HAPPENING



Socket API Extension

Using these tools should be dead-easy for developers

EXTEND SOCKET API TO ALLOW SIMPLE SETUP FOR ANONYMOUS COMMUNICATION

Competent developers should have the power to do more

APPLICATIONS CAN REQUEST TO BYPASS SYSTEM ANONYMIZATION SETTINGS

```
Xconnect(sock, dest_DAG)
XconnectWithAnonymizer(sock, dest_DAG, anon_DAG)
XconnectWithoutAnonymizer(sock, dest_DAG)
```

Comparison

	XIA	TCP/IP
Proxy-based anonymizer	▪ In-DAG or next-header	▪ Next-header
Temporary source IDs	▪ Register SID with local AD ▪ Fine-grained: different temp. SID per application	▪ Painful with static IP addresses ▪ Coarse-grained: one IP address for all processes on machine
Principal type filtering	▪ Fine-grained traffic control	▪ N/A

Demo

Questions?