# Towards Automated "Zero Day" Application Characterization: (Can we do better?)


Kostas G. Anagnostakis

I[2]R, Singapore

http://s3g.i2r.a-star.edu.sg/

**Joint work with:**

Asia Slowinska, *VU Amsterdam, NL*

Kirk Jon Khu, *I[2]R, Singapore*

Jonathan M. Smith, *UPenn, USA*

# Problem Statement & Goals

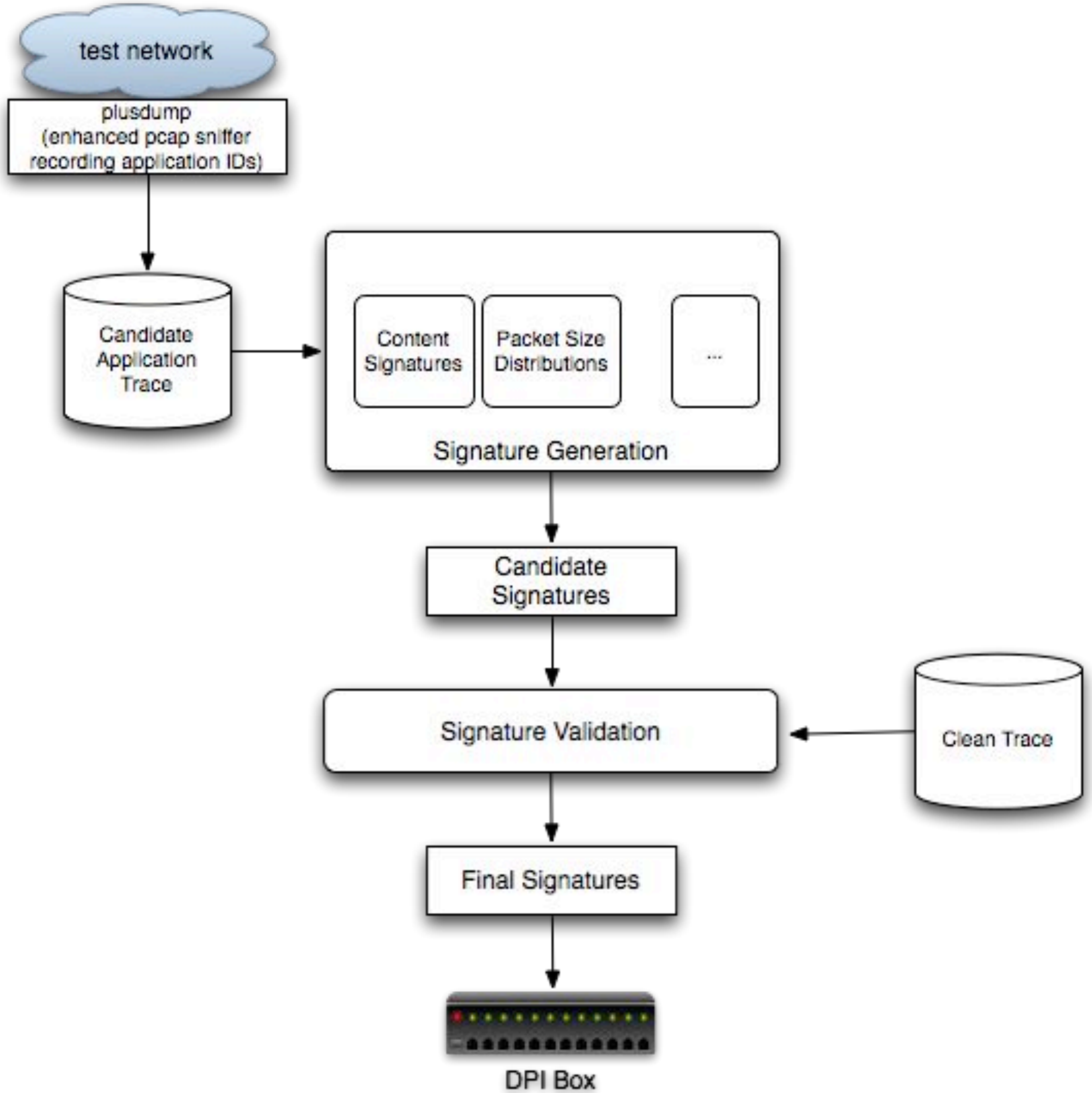
- **Pressing need for application characterization**
  - Maintain performance, control bandwidth usage
  - Protect against misbehaving or undesirable applications
  - Support networking research, or simply satisfy curiosity
- **Application characterization becoming complex**
  - New applications, application versions, new protocols
  - Obfuscation techniques: Skype, Joost, eMule, BitTorrent, etc.
- **Need for automation**
  - Tools to (semi-)automate characterization
  - Capability to rapidly develop+test characterization techniques
- **Our take on the problem**
  - Build upon techniques from zero-day attack sig generation
  - Strategic game between obscuring and revealing party

A*STAR Institute for Infocomm Research



CUB4

# CUB4: Architecture Overview



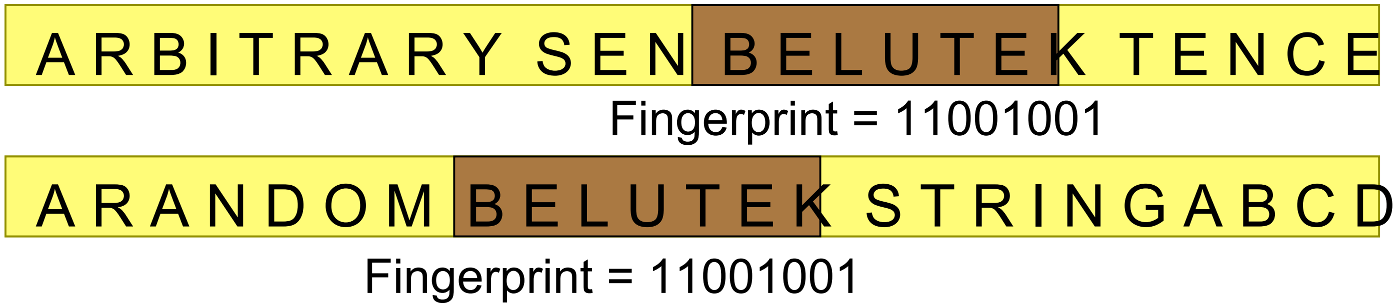

Institute for
Infocomm Research
A*STAR



CUB4

# Application content fingerprinting

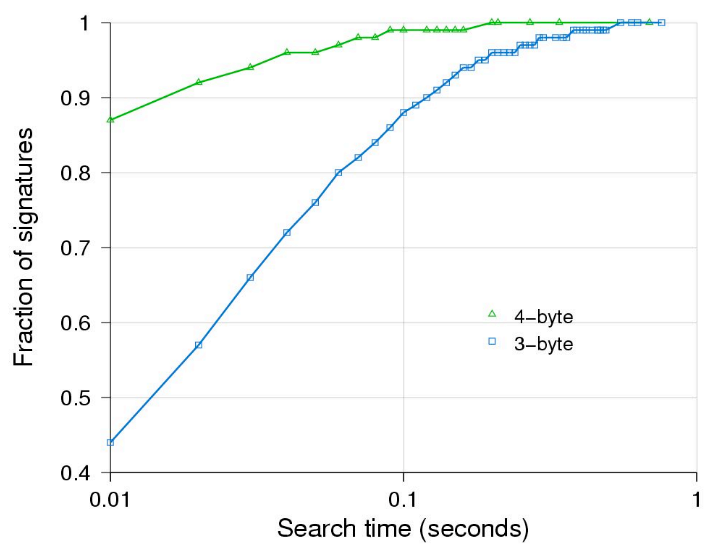
- **Inspired by zero-day worm fingerprinting work**
  - [Akritidis2005,Earlybird2004, Autograph2005]
  - Use of rabin fingerprints over sliding window, with careful max-hit/max-size scoring threshold



# Signature Validation approach

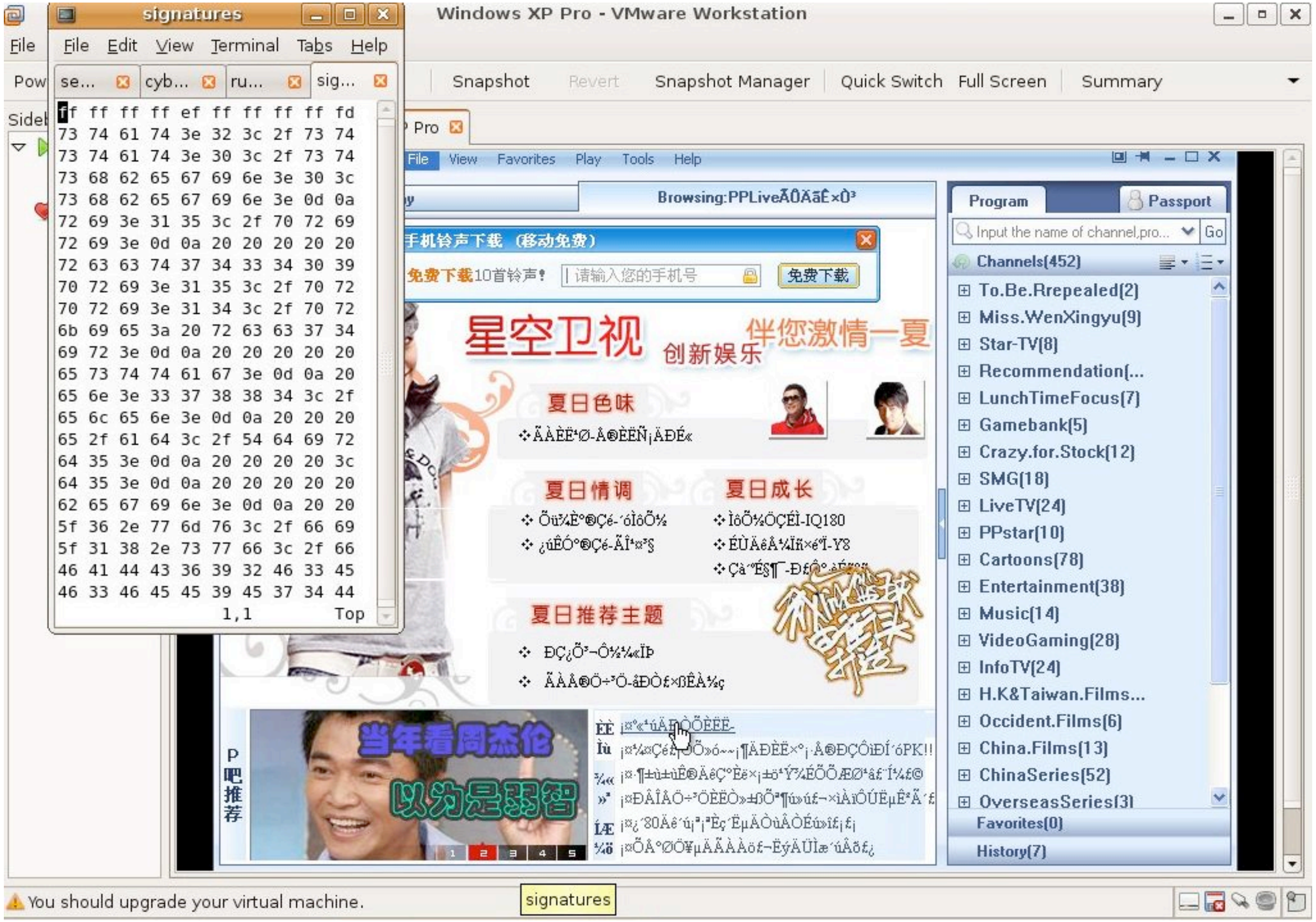
- **Main idea: test candidate signatures against known traffic of other apps**
- **In a nutshell**:
  - Lots of data: for FPs in the order of 10^-6, need >> 10^6 samples (1-100 GB)
  - Assume 1-100 signatures to test, need search time < 1 second
  - Approach: record packet trace, maintain carefully designed index for fast lookups
- **Trade-off:** need around 6x space to perform efficient lookups

**Search time:** 88% < 0.01 sec, 99% < 0.1sec, 99.99% < 1sec, known "bad" strings up to 2 seconds
**Comparison**: testing of PPLive signatures against 100GB trace took 5 hours
**Space:** for N-byte elements, $2^{8N}$ bitmap + 6x trace size




Institute for Infocomm Research

# Example signatures: PPLive






Institute for Infocomm Research A*STAR

CUB4

# Application signature generation: can we go live?

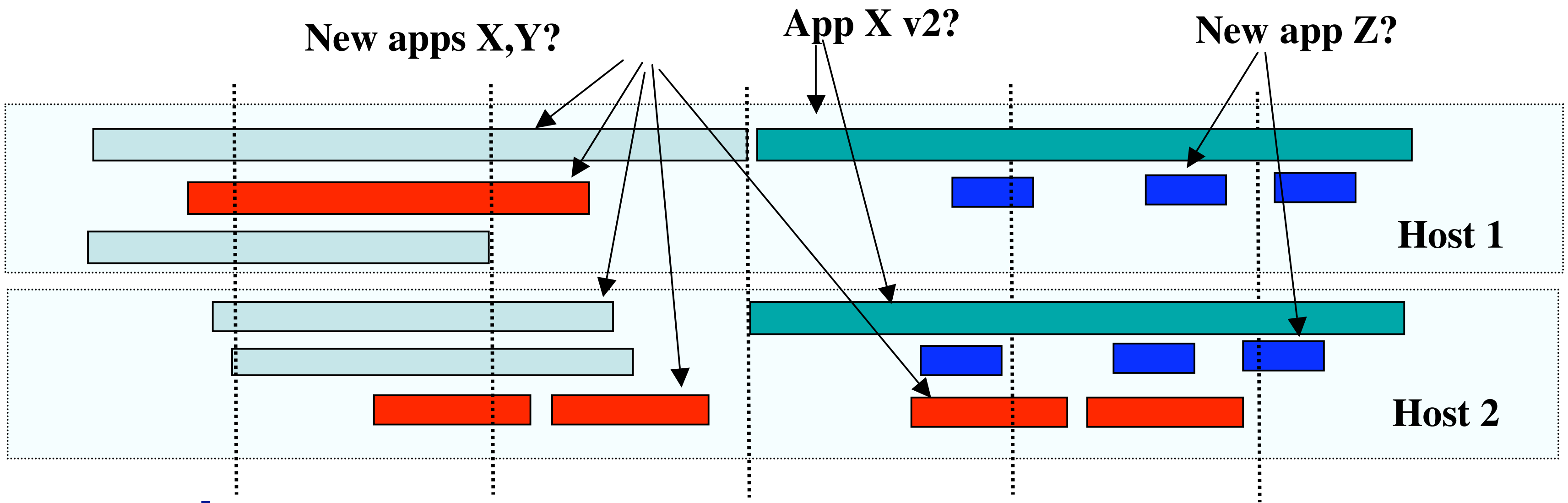
## •Reformulating the problem:

- –So far, feeding our tool with “clean” offline traces
- –Can we apply the same method to “live” traffic?

*“As we know, there are* ***known knowns****. There are things we know we know. We also know there are* ***known unknowns****. That is to say we know there are some things we do not know. But there are also* ***unknown unknowns****, the ones we don't know we don't know.”*

Donald Rumsfeld. Feb. 12, 2002,
Department of Defense news briefing

# Application signature generation: can we go live?




- **Approach:**
  - Split traces per host, and by time
  - Determine joint fingerprints: flows of same host/ among hosts
  - Compute likely set of applications based on the above data
- **High startup cost:**
  - If new applications pop up one at a time, we may have a chance
  - If we're looking at a link with 30-40% unclassified traffic from >20 applications, problem is somewhat more challenging

# Beyond content: packet size

**Joost**

```
11:02:39.249981 IP 4.71.174.175.4166 > 58.185.58.82.49335: UDP, length 941
11:02:39.256917 IP 4.71.174.189.4166 > 58.185.58.82.49335: UDP, length 940
11:02:40.017697 IP 4.71.174.150.4166 > 58.185.58.82.49335: UDP, length 11
11:02:40.026990 IP 4.71.174.175.4166 > 58.185.58.82.49335: UDP, length 11
11:02:40.034641 IP 4.71.174.158.4166 > 58.185.58.82.49335: UDP, length 11
11:02:40.047234 IP 4.71.174.175.4166 > 58.185.58.82.49335: UDP, length 1057
11:02:40.291110 IP 4.71.174.158.4166 > 58.185.58.82.49335: UDP, length 1057
11:02:40.297065 IP 4.71.174.175.4166 > 58.185.58.82.49335: UDP, length 1058
11:02:40.312826 IP 4.71.174.150.4166 > 58.185.58.82.49335: UDP, length 1058
11:02:40.322903 IP 4.71.174.158.4166 > 58.185.58.82.49335: UDP, length 1058
```

**Obfuscated eMule**

```
06:48:13.610291 IP 131.111.218.93.64692 > 221.134.2.109.4662: S 687196497:687196497(0) >
06:48:14.536485 IP 221.134.2.109.4662 > 131.111.218.93.64692: S 1994439214:1994439214(0) >
06:48:14.539900 IP 131.111.218.93.64692 > 221.134.2.109.4662: P 1:121(120)
06:48:15.896222 IP 221.134.2.109.4662 > 131.111.218.93.64692: P 1:107(106)
06:48:16.117231 IP 131.111.218.93.64692 > 221.134.2.109.4662: P 121:230(109)
06:48:16.716196 IP 221.134.2.109.4662 > 131.111.218.93.64692: P 107:215(108)
06:48:16.776250 IP 131.111.218.93.64692 > 221.134.2.109.4662: P 230:271(41)
06:48:17.256531 IP 221.134.2.109.4662 > 131.111.218.93.64692: P 215:336(121)
06:48:17.339697 IP 131.111.218.93.64692 > 221.134.2.109.4662: P 271:293(22)
06:48:18.006485 IP 221.134.2.109.4662 > 131.111.218.93.64692: P 336:342(6)
06:48:18.038964 IP 131.111.218.93.64692 > 221.134.2.109.4662: P 293:339(46)
06:48:18.527099 IP 221.134.2.109.4662 > 131.111.218.93.64692: . 342:1802(1460)
06:48:18.596903 IP 221.134.2.109.4662 > 131.111.218.93.64692: P 1802:2942(1140)
```

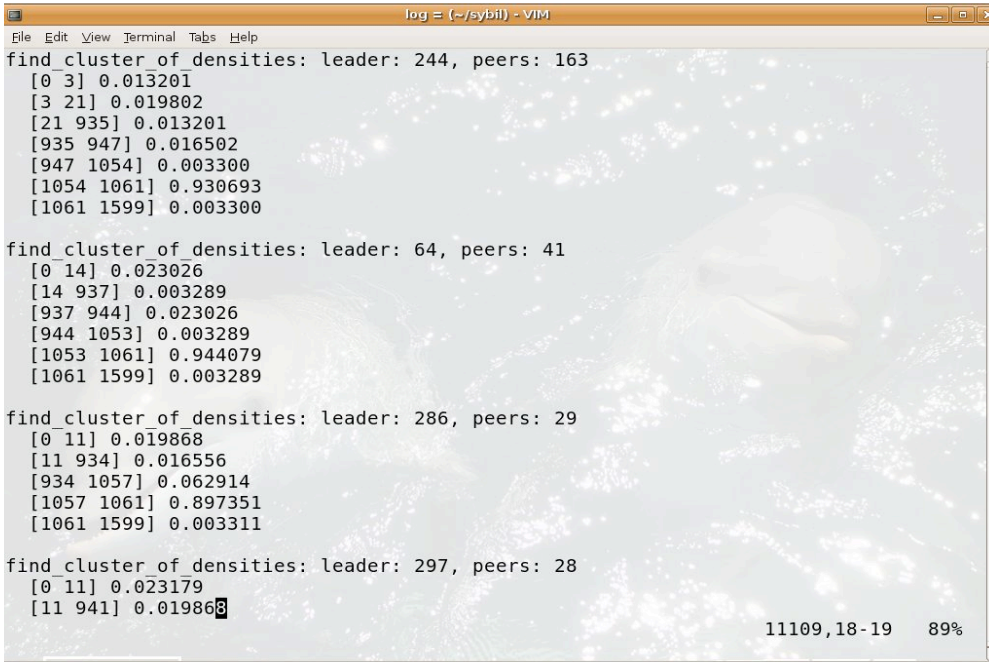
# Behavioral analysis: packet size distributions

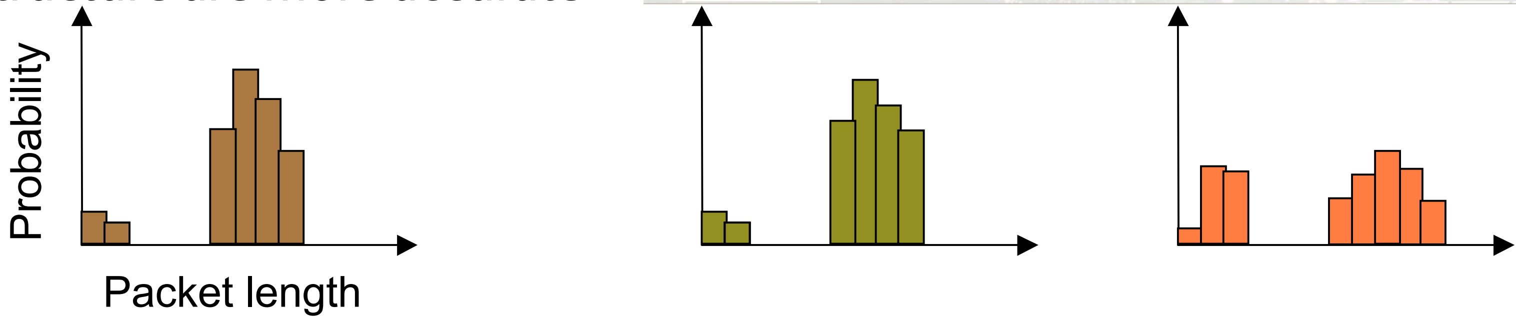
- **Experiment: Packet size histogram over sliding window**
  - Various goodness-of-fit tests
  - Fingerprinted Joost

    Obtained signatures.

    We can see a few representative probability density functions describing packet length.

  - Other heuristics with more structure are more accurate



# Increasing accuracy through taint propagation

- **Basic Idea:**
  - If a flow <src, srcP, dstIP, dstP> previously identified to belong to app X, then any flow <*,*,dstIP,dstP> is very likely to belong to app X
- **Widely applicable:**
  - Any application that operates over TCP and advertises “server”-side ports is relevant
  - Performance gains likely to be higher at aggregation points
  - Reminiscent of blacklisting, but this is not IP based: port # is the key!
- **Implications for heuristic design:**
  - Tainting can compensate for low detection rates as long as the <dstIP,dstPort> pair handles multiple connections
  - But need to be careful with false positive amplification
  - On the other hand, we might tune down sensitivity to avoid FPs, as tainting will compensate for that in terms of detection rate

# Some ongoing work

- **Incorporating structural features in packet size heuristics**
  - Much richer than simple packet size distribution analysis
  - Good preliminary results for obfuscated eMule and encrypted BitTorrent
- **Instrumenting endpoint software**
  - Current focus: use of Argos processor emulator, may need more lightweight
  - Looking for constants in memory that make it into packets
  - One step closer to fully automated fingerprinting
  - This would include cases where an application is updated, etc.
- **Detecting deep architectural properties of applications**
  - Adaptive applications through FEC-blowup or codec-switching
  - P2P through their incentive mechanisms
  - Much more labor-intensive to fingerprint, but also harder to circumvent
- **Active fingerprinting of likely "server" endpoints**
  - Even sending junk sometimes results in well-formed responses

# Going adversarial

- **Two party model:**
  - Obscuring party (app dev) vs. revealing party: (researchers, DPI vendors)
- **This is already happening -- strategies we have seen:**
  - Avoid well known ports (too many to list)
  - Encrypt/obfuscate content (Skype, BitTorrent, eMule)
  - Anti-debugging techniques (Skype)
- **Other strategies we're likely to see:**
  - Content signatures → embedding of other application signatures
  - Packet size heuristics → proper padding
  - Binary instrumentation → more anti-debugging/anti-VM/anti-…
  - Active fingerprinting → only responding to conn requests from "trusted" sources
  - Connection graph/volume analysis → more cover traffic, dummy connections
  - … at the limit, there's stego and Tor-like approaches

**… but for the time being, security through obscurity might have value (?)**

# Towards Automated "Zero Day" Application Characterization: (Can we do better?)

Kostas G. Anagnostakis
I$^2$R, Singapore
http://s3g.i2r.a-star.edu.sg/

**Joint work with:**
Asia Slowinska, *VU Amsterdam, NL*
Kirk Jon Khu, *I$^2$R, Singapore*
Jonathan M. Smith, *UPenn, USA*