The Case for Network Witnesses

Wu-chang Feng

Portland State UNIVERSITY

Travis Schluessler

intel

Supported by:

intel NSF

Internet protocol design (1970s)

- Programmers and users cooperative
- Limited semiconductor capabilities
- Public-key cryptography in a nascent state

- Result
  - Simple design
  - Quickly deployed
  - Immensely successful
  - But, was ultimately and tragically insecure

Fast forward to 2008

- Programmer and user are not trusted
  - Denial-of-service, Botnets, Spam
  - Phishing, DNS poisoning, TCP RST attacks, IP spoofing
  - Cheating in on-line games, Rootkits
- Semiconductor technology explosion
  - Moore’s law over 30+ years
- Widespread use of public-key cryptography
  - Web transactions, IPSec, VPNs, SSL accelerators
  - Trusted hardware and software platforms
    - PS3, Xbox 360 game consoles
    - IBM Trusted Platform Modules (TPM)
    - Intel AMT and TXT
    - Windows Vista

A clean-slate approach

- What if we revisited Internet protocol design in today’s landscape?
  - Users are untrusted
  - Semiconductor technology can support high-speed cryptographic operations in the data-path

Network Witness

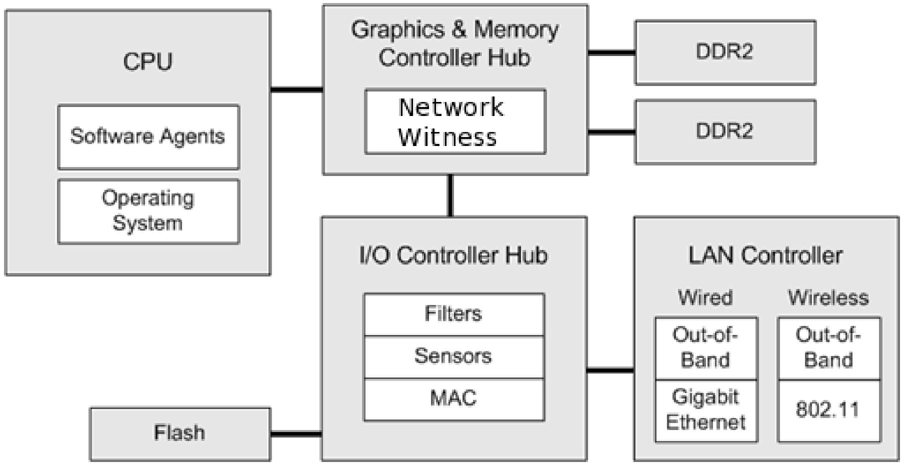
- Tamper-resistant, trusted third party at end-host
  - Our take on Shai Halevi’s “Angel in the Box”
- Functions
  - Provide authenticated measurements of host activity
  - Enforce protocol rules and requirements

Characteristics of a Network Witness

- Reliable introspection
  - Can measure the state of the host and its network usage
- Attestation
  - Can report such measurements in an authenticated manner to other witnesses in the network
- Isolation
  - Measurements are not unduly influenced by host
- Trusted execution
  - Only executes code cryptographically signed by a trusted third party (e.g. the IETF or the manufacturer)
- Tamper-resistance
  - Cost of tampering exceeds value of the witness service

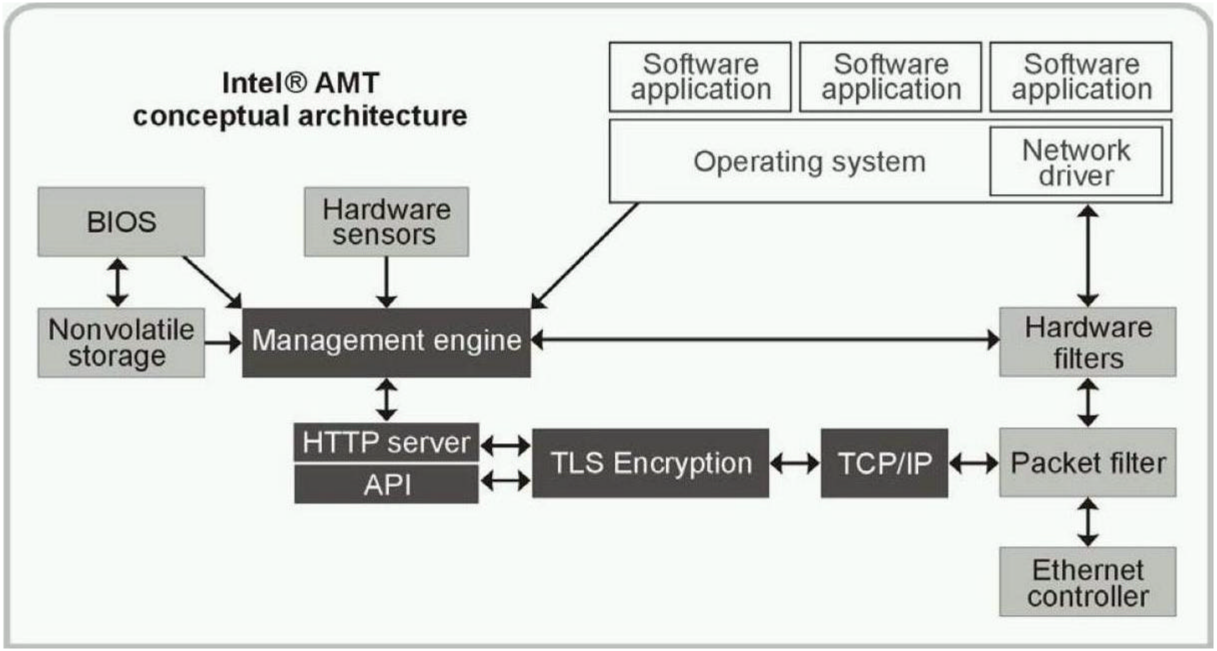
An example witness

- Intel's Active Management Technology platform
  - Introduced in 2005
    - Now, a commodity component on all Intel motherboards
  - Trusted processor in memory controller (iAMT2)
    - Sees all network traffic
    - Sees all peripheral activity
    - Has access to all memory locations
    - OOB channel to communicate across the network



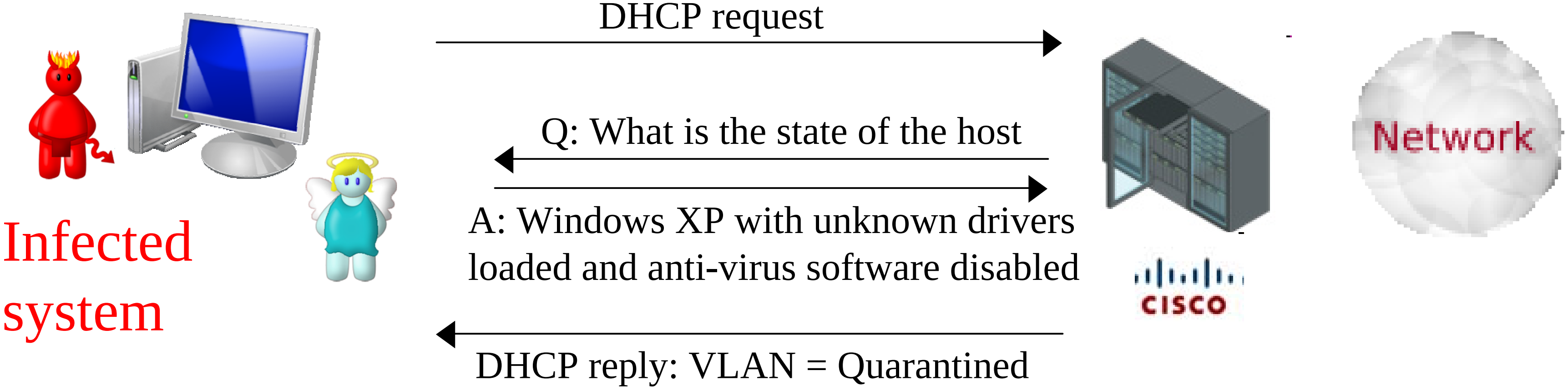
An example witness

- Intel’s Active Management Technology platform
  - Tamper-resistant operation
    - Can not be tampered with from host processor’s software stack
    - Only runs code signed by Intel
    - Equipped with keys to authentically sign host measurements for transmission over the network



Intel AMT with Cisco NAC

- Network access control based on host integrity
  - Measured “security posture” of the running OS and applications determine level of access



DHCP request

Q: What is the state of the host

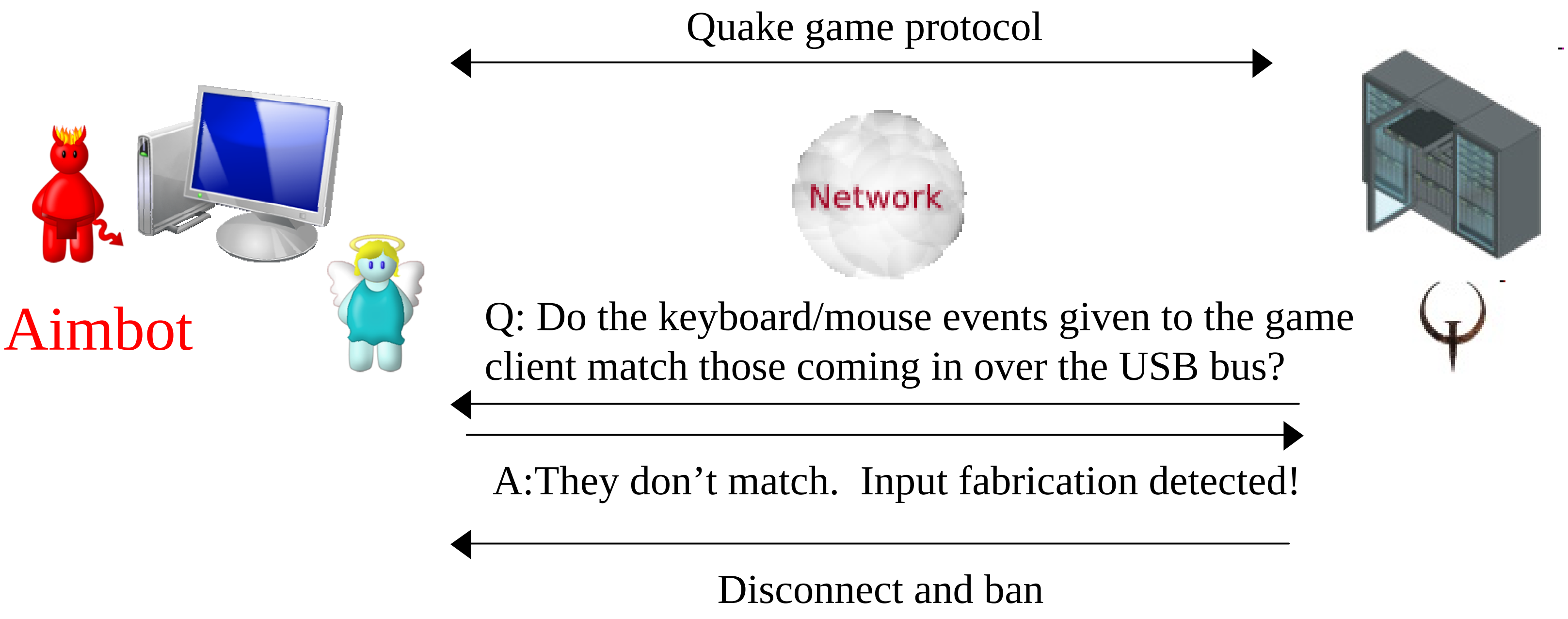
A: Windows XP with unknown drivers loaded and anti-virus software disabled

DHCP reply: VLAN = Quarantined

Infected system

Intel AMT and On-line Games

- On-line game access based on valid host operation
  - Measure that the keyboard/mouse event the game gets
    - Schluessler et. al. “Is a Bot at the Controls?”, NetGames 2007.



Quake game protocol

Aimbot

Q: Do the keyboard/mouse events given to the game client match those coming in over the USB bus?

A:They don’t match. Input fabrication detected!

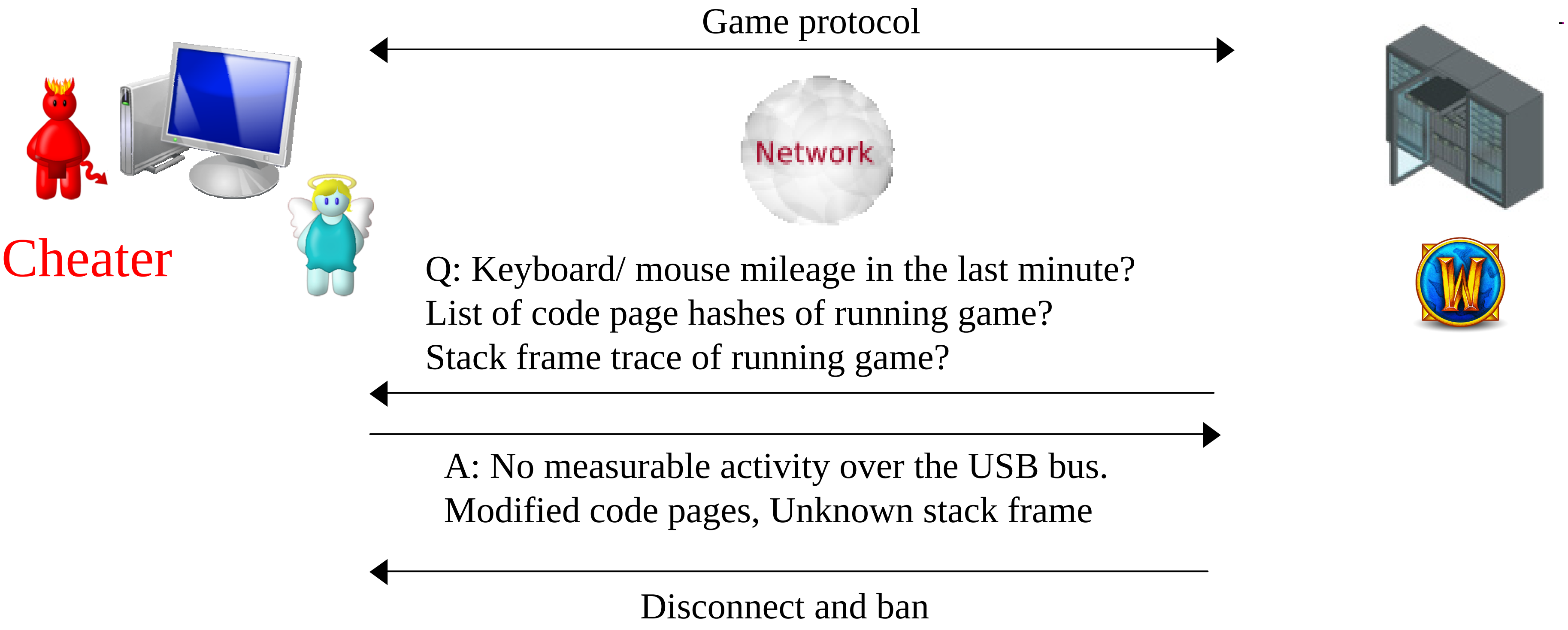
Disconnect and ban

Generalizing the approach

- Observation
  - Trusted third parties greatly simplify network security protocols
- How might this approach be applied to a range of network protocol problems?

Cheating in on-line games

- Use network witness to attest to human activity and game process integrity
  - “Stealth Measurements for Cheat Detection in On-line Games”, NetGames 2008.



Cheater

Game protocol

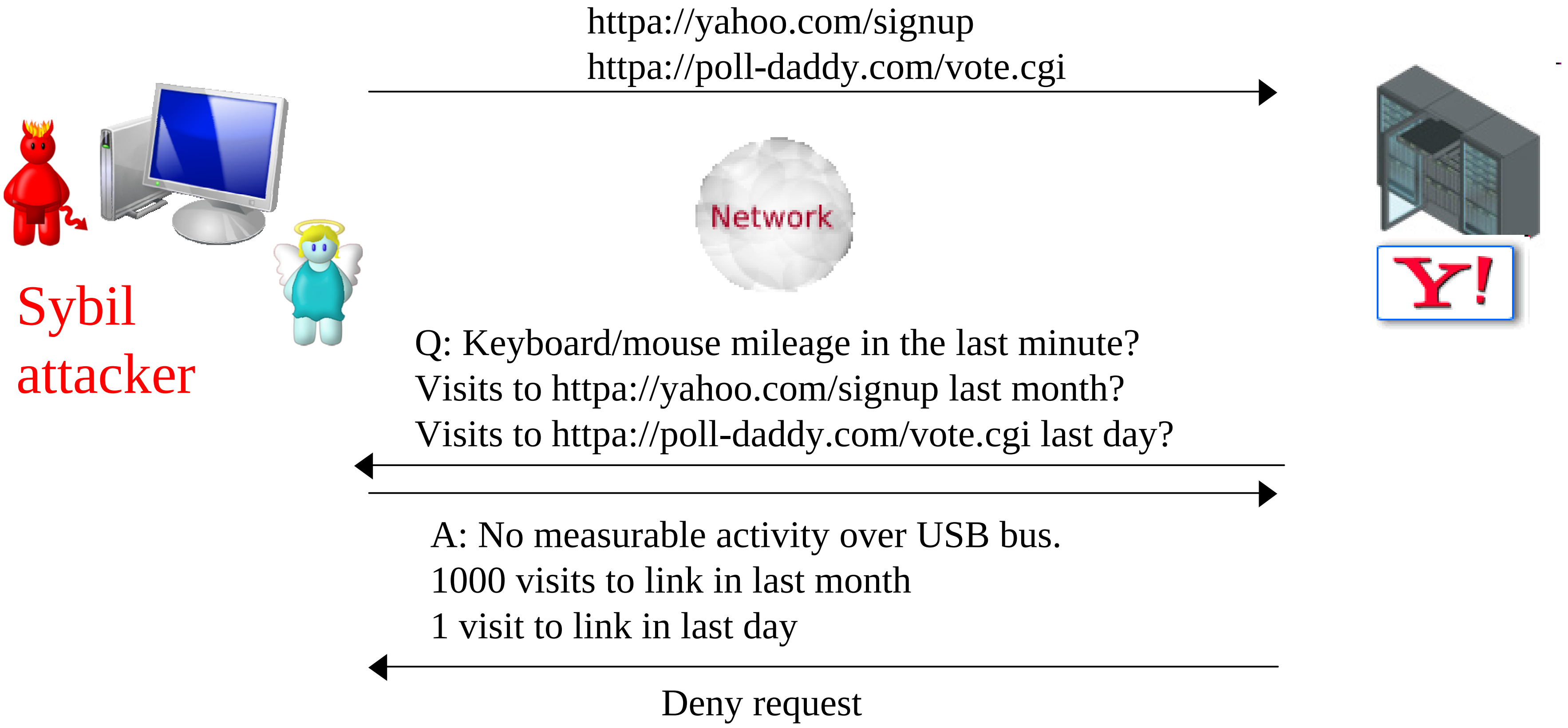
Q: Keyboard/ mouse mileage in the last minute?
List of code page hashes of running game?
Stack frame trace of running game?

A: No measurable activity over the USB bus.
Modified code pages, Unknown stack frame

Disconnect and ban

Sybil attacks

- Use network witness to attest to human activity and prior web account signup or on-line voting activity



httpa://yahoo.com/signup
httpa://poll-daddy.com/vote.cgi

Sybil attacker

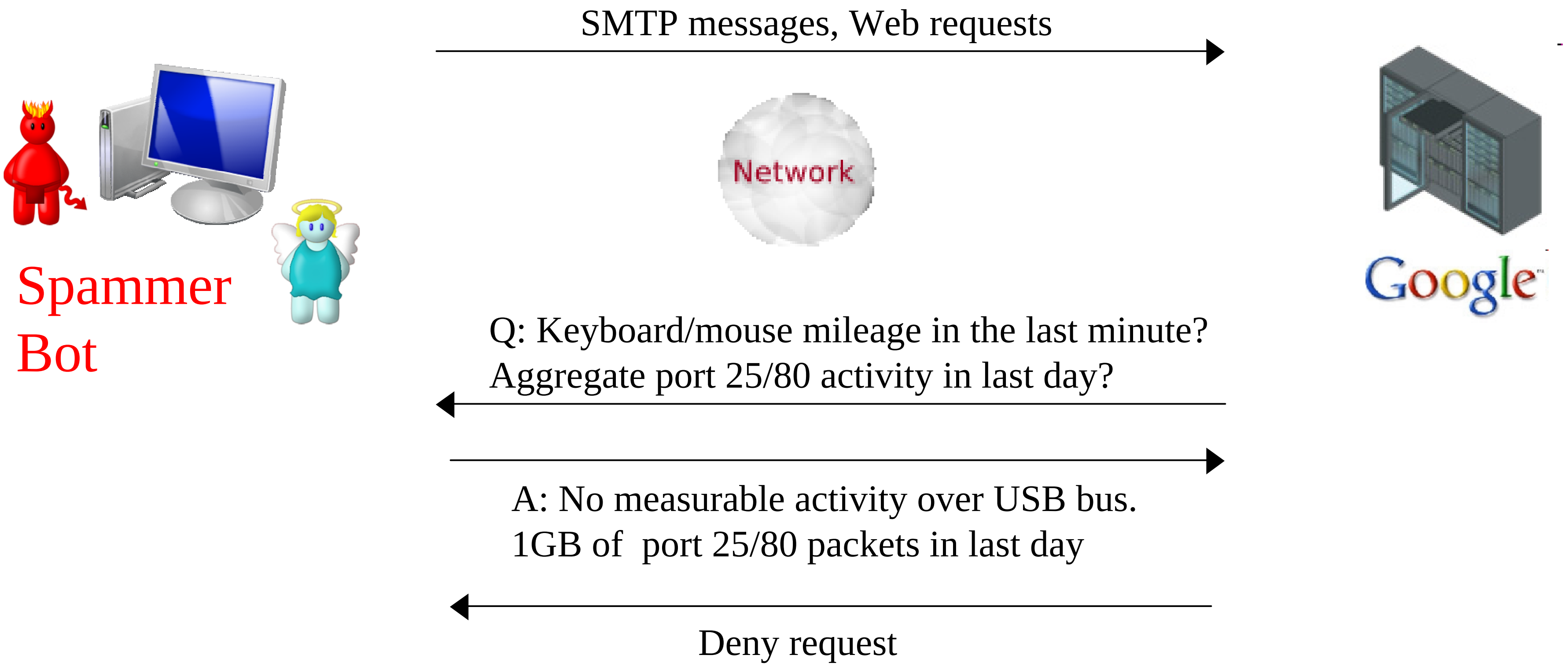
Q: Keyboard/mouse mileage in the last minute?
Visits to httpa://yahoo.com/signup last month?
Visits to httpa://poll-daddy.com/vote.cgi last day?

A: No measurable activity over USB bus.
1000 visits to link in last month
1 visit to link in last day

Deny request

Spam, denial-of-service, botnets

- Use network witness to attest to human activity and prior network usage



Spammer
Bot

SMTP messages, Web requests

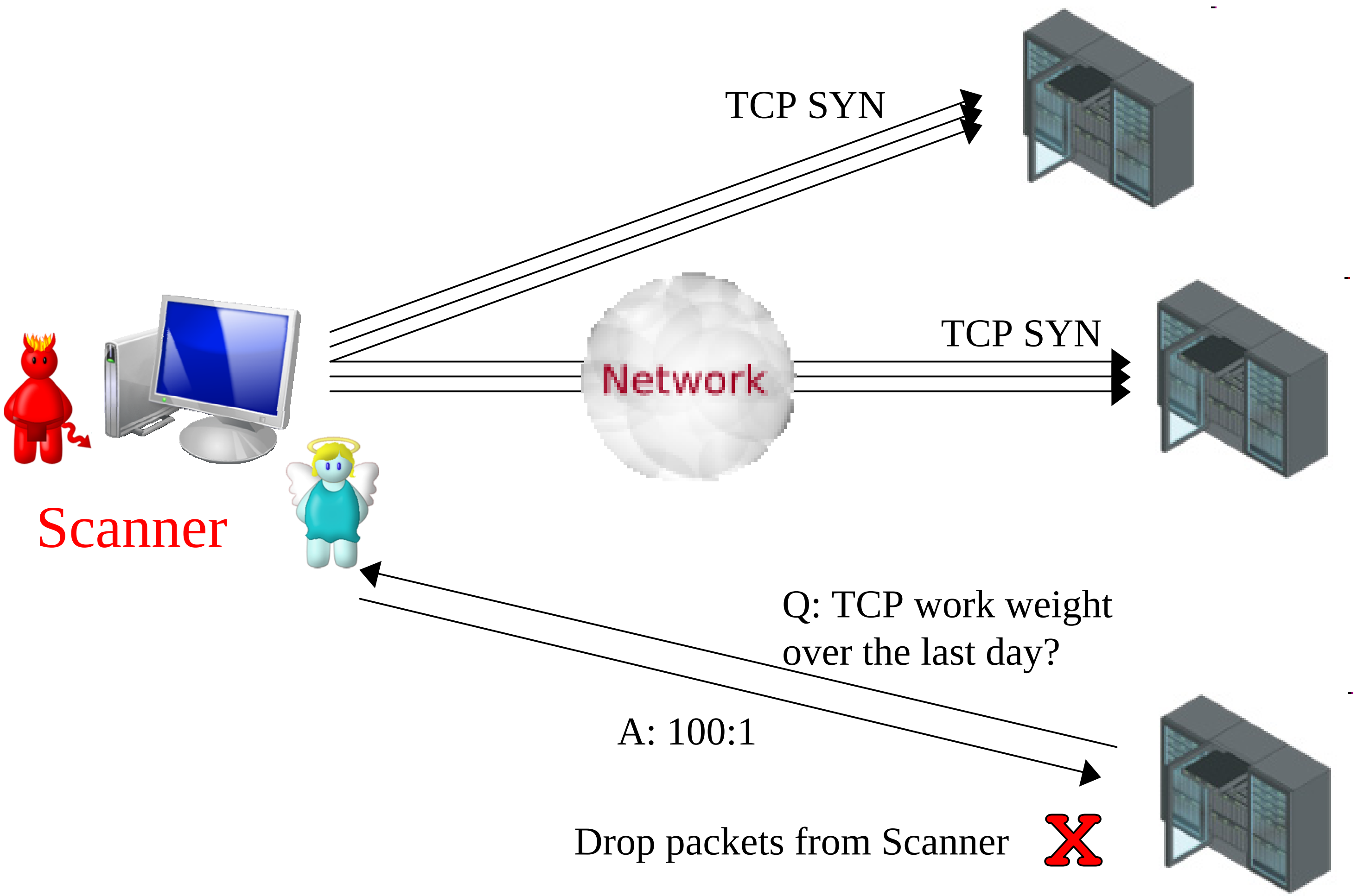
Q: Keyboard/mouse mileage in the last minute?
Aggregate port 25/80 activity in last day?

A: No measurable activity over USB bus.
1GB of port 25/80 packets in last day

Deny request

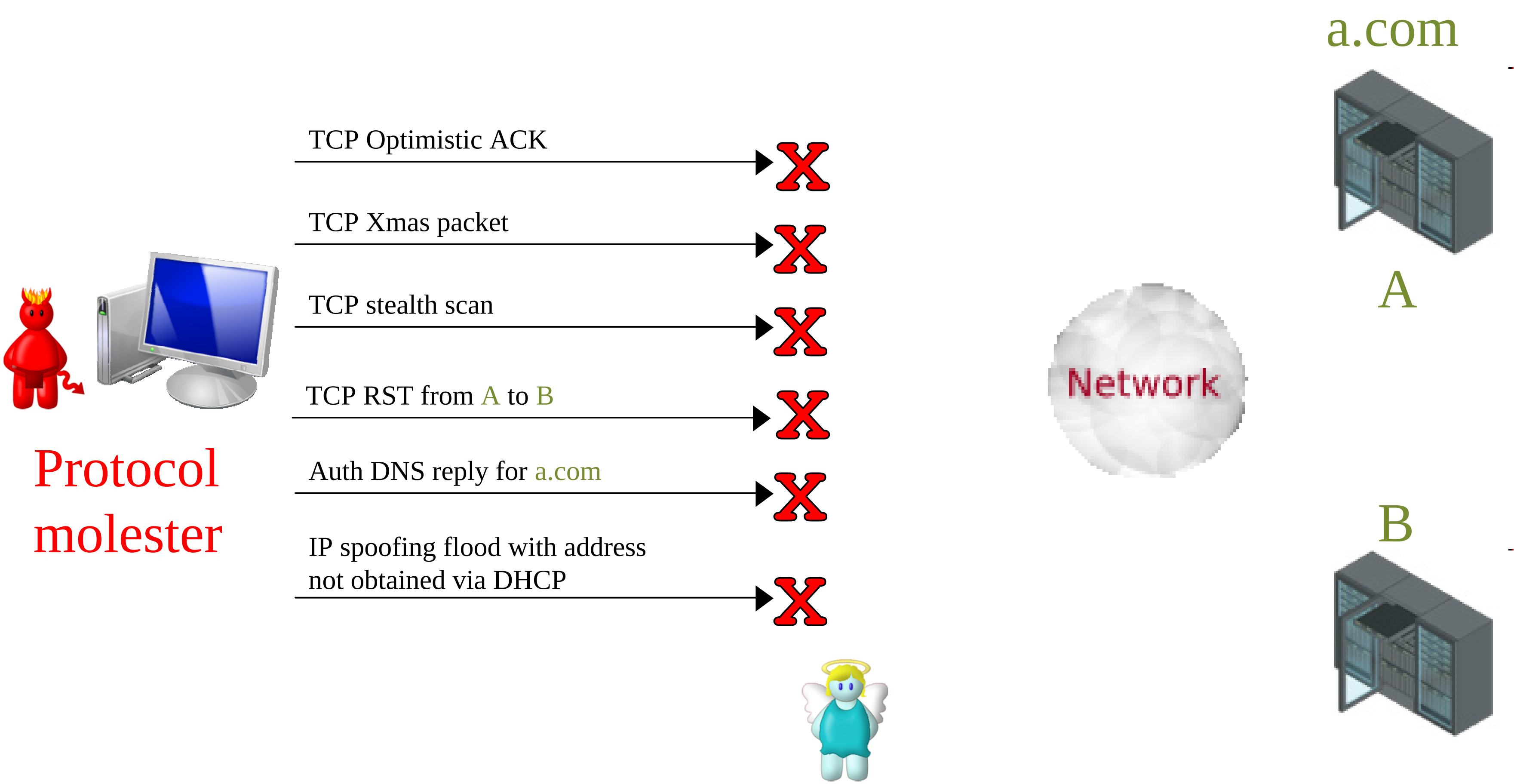
Port scanning

- Use network witness to attest to the ratio of TCP SYN packets sent to TCP SYN/ACK packets received



Protocol enforcement

- Use network witness to ensure packets from the host do not violate protocol rules



Protocol molester

TCP Optimistic ACK

TCP Xmas packet

TCP stealth scan

TCP RST from A to B

Auth DNS reply for a.com

IP spoofing flood with address not obtained via DHCP

Network

a.com

A

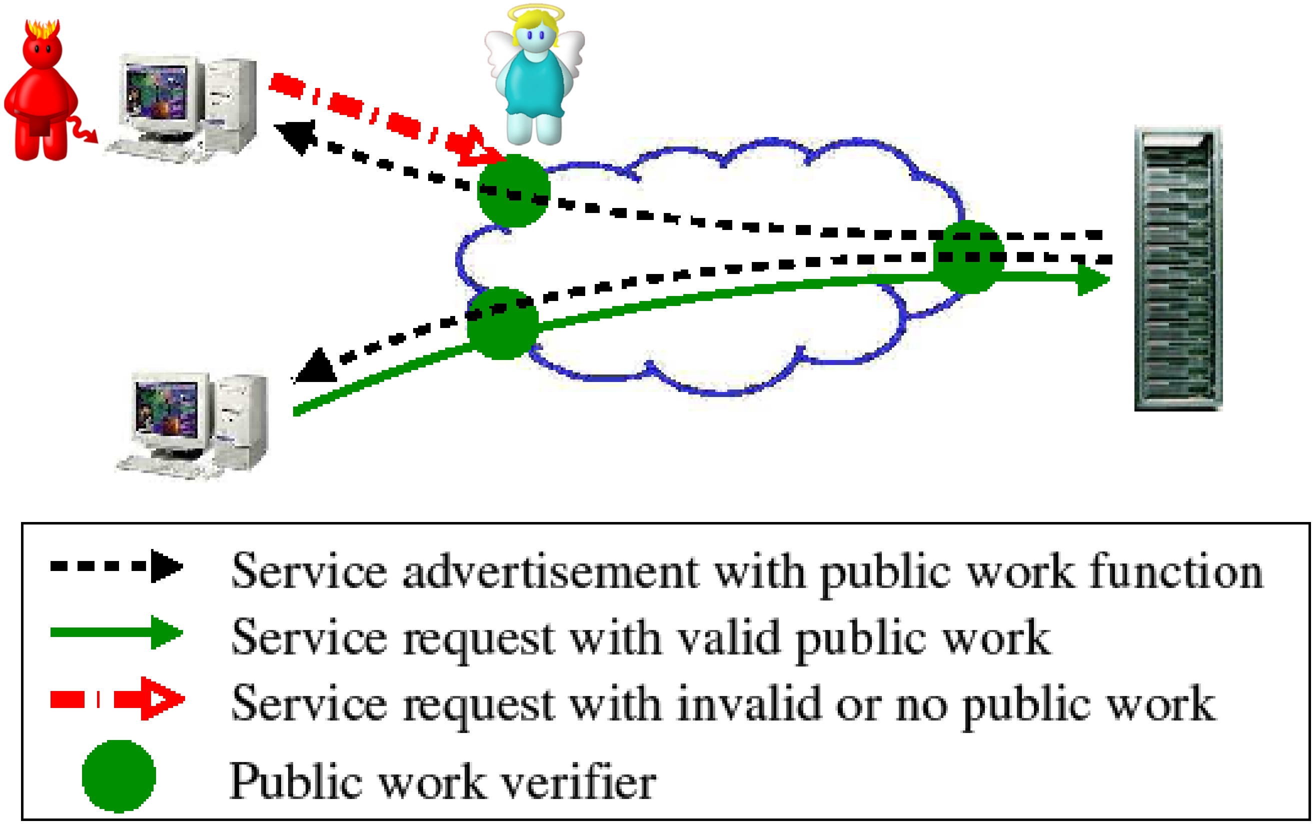
B

Towards new protocols

- Network witnesses can address problems in existing protocols
  - Seems like a waste of our brand new super powers
  - Can we use it to do new things besides cleaning up after an elderly protocol (i.e. TCP)?
  - Maybe…

Public proof-of-work

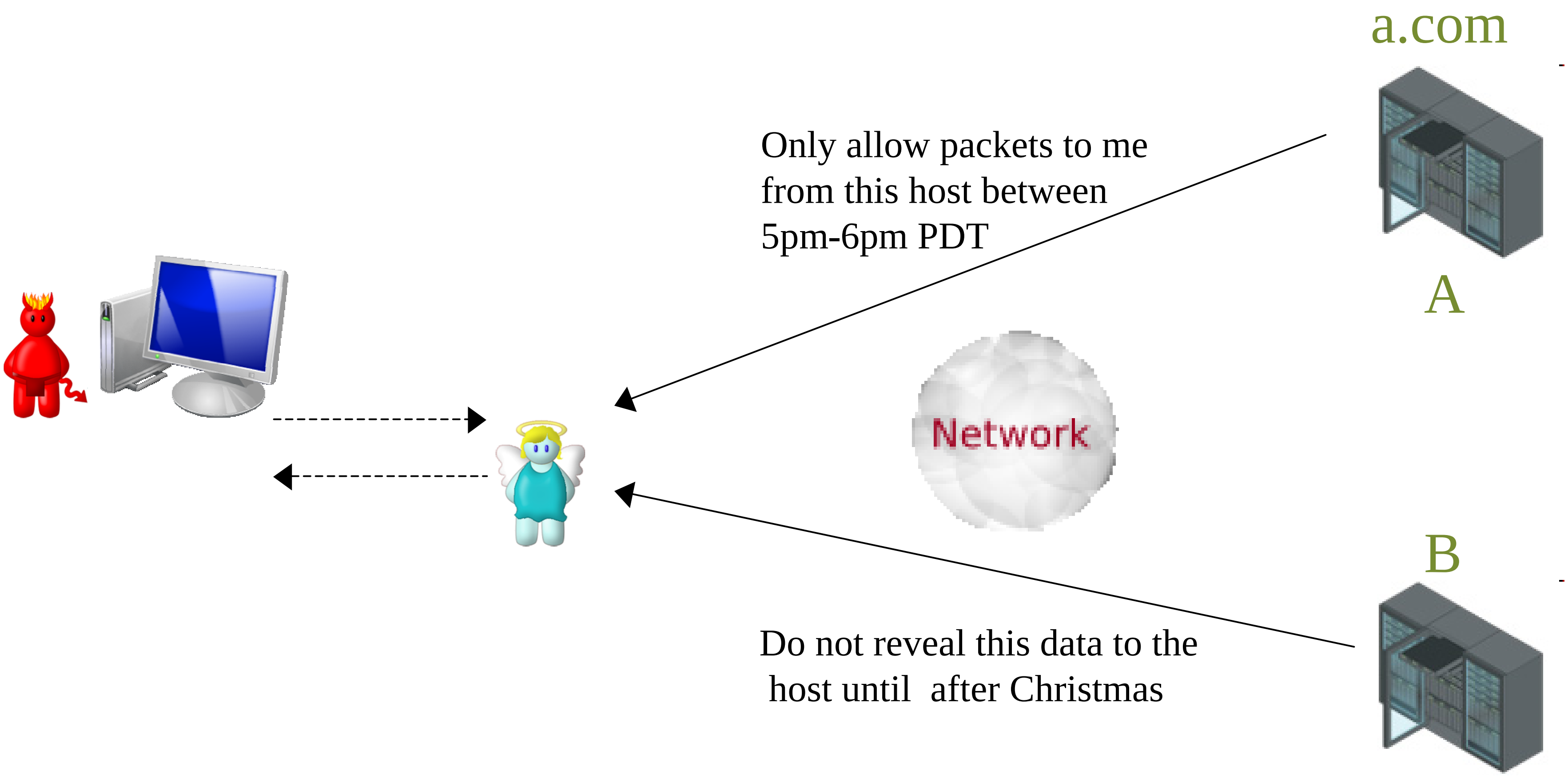
- Use witness to prevent requests with invalid or missing proof-of-work from leaving the end-host
  - “The Case for Public Work”, Global Internet 2007.
  - “Portcullis … ”, SIGCOMM 2007.



Service advertisement with public work function
Service request with valid public work
Service request with invalid or no public work
Public work verifier

Scheduled transmission and reception

- Use witness to ensure
  - Host does not send anything to a site until a scheduled time
  - Host does not receive particular data until a scheduled time



a.com

Only allow packets to me
from this host between
5pm-6pm PDT

A

Network

B

Do not reveal this data to the
host until after Christmas

More half-baked ideas in the paper

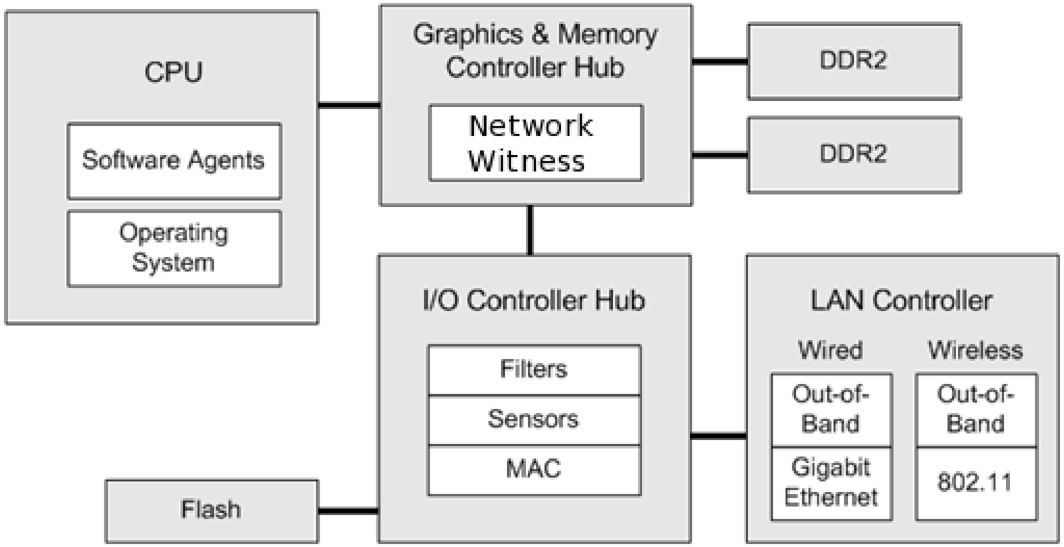
- Attestation-assisted congestion control
- Attested tit-for-tat for peer-to-peer networks
- Data exfiltration prevention
- Execute-once protocols

That was fun, but…

- Devil in the details
- Issues with Network Witnesses
  - Location
  - Measurement fidelity
  - Storage issues
  - Privacy and usability issues
  - Deployment issues

Location

- Network witness location (as defined here) directly determines mitigated threats
  - Current placement in memory controller
    - Drives adversaries (cheaters) into peripherals
  - Placement in end hosts
    - Drives adversaries into the network



Accuracy

- Does the network witness have 20/20 vision?
  - A blind witness can’t attest to much
  - Intel’s ME runs at a fraction of the speed of the FSB
    - Can not implement a “memory watchpoint” to prevent information exposure cheating in on-line games
    - Might not be able to accurately measure what it is asked to attest

Storage issues

- Witness will not have an “elephant file system” for its measurements
  - What happens when witness is unable to attest to the desired measurement due to space limitation?

Privacy and usability

- How can users trust network witnesses not to measure and give away arbitrary data?
  - Attesting all keyboard activity would be a disaster
  - Attesting inter-key timings would also be bad
  - Attesting aggregate keyboard/mouse mileage?

Deployment incentives

- Must give the user some benefit
  - Be able to play on-line games with other players that you can verify are not cheating?
  - Remove CAPTCHA tests for those willing to use hardware that attests keyboard/mouse activity?
  - Others?

Conclusion

- A half-baked approach for building networks around the notion of “network witnesses”
- An approach increasingly being pushed by industry
- Hopefully, we as researchers can influence how industry fully bakes it