An Open Infrastructure For National Performance and Security Monitoring (NIMI)

Andrew K. Adams  
National Laboratory of Applied Network Research (NLANR)  
Pittsburgh Supercomputing Center  
akadams@psc.edu

Vern Paxson  
ICSI Center for Internet Research (ICIR)  
and  
Lawrence Berkeley National Laboratory  
vern@icir.org, vern@ee.lbl.gov

As interpreted by  
Matt Mathis  
mathis@psc.edu  
All errors and omissions are mine

MNI PI Meeting  
16 July 2002
NIMI background:

• National Internet Measurement Infrastructure

• Designed to work across administrative domains, by emphasizing full control of platforms owned by a domain ...

• ... but also ease of delegating some measurement functionality, in a secure fashion.

• Not about a particular type of measurement, but rather facilitating a potentially large class of measurements
Performing measurement:

An infrastructure is built by installing NIMI probes (nimid) throughout the network. All NIMI’s within the same administrative domain are configured by the domain’s Configuration Point of Contact (CPOC).

User invokes the Measurement Client (MC) to scheduled a set of measurements at some point in the future (including "now").
Performing measurement, con’t:

MC contacts different NIMI’s to schedule measurement. Tells platforms where to send the results (DAC).

Data Analysis Client (DAC) eventually receives results (possibly on another host).

User at some point uses MC to delete results from NIMI platforms.
NIMI current status:

65 K lines of C

Runs under FreeBSD, NetBSD, Linux, Solaris

14 measurement tools supported

Global operation at 50+ sites

Current architecture at its administrative scaling limit ...
Lessons learned:

Heterogeneity issues:
• remote administration headaches
• platform & administrative inconsistencies

• Attempt to encapsulate areas of differences

Large systems complications:
• incongruence within large distributed systems
• routine maintenance headaches
• managing large-scale studies

• Solve with incremental design refinements.
Future directions

Key infrastructure for a dozen large-scale measurement studies.

A great resource:

Good news, we got the majority of the architecture right!
Bring NIMI up to current standards

Use X.509 certificates (over OpenSSL):
  • Replace NIMI home-grown RSA certificates
  • Long-lived certificates (keys) map to individual people
    • Use existing signing & distribution servers (e.g., Verisign)
  • NIMI CA will use X.509 certificates to issue measurement group certificates
  • Standard certificate management framework - familiar to admins

Replace current messaging with XML
1st hard problem to solve -- secure upload:

Experience shows that large measurement projects frequently need to update measurement tools ("user" code).

Supporting diverse measurements also needs this.

Problem: how can you trust the code?

Threats:
- subverting the platform (use it as stepping stone)
- using NIMI for attacks (e.g., distributed DoS)
- affecting integrity of other measurements
2nd hard problem to solve -- resource management:

Current granularity of control does not address concurrent measurements

Need fine-grained access to resources such as:

- CPU, memory, disk space
- network utilization
- well-known ports
- packet filter
- destination

Observation: if you can solve this problem, you probably have the right hooks to solve the secure upload problem, too.
Need to protect system & network from corrupt tools.

- "Manual" trust doesn’t scale

Sandboxes
- Enforcement daemon (monitoring/policing)
- Network proxy (generic packet filter & raw I/O daemon)
- Safe languages
- Trusted library
- Kernel modifications

Perhaps different solutions for different tools or environments
Matt’s observations on measurements and grids

Superficially they are "the same"
• Distributed command, control and data management
• to support delayed remote execution

But
• Grids provide homogenized resources
• Measurements MUST be anchored in time and space

Explicit consideration of "anchored" vs "homogenized"
• Consider separate APIs to a common infrastructure
• May lead to better APIs and models for both
Parallels in Transport Protocols

Transport protocols are failing to solve real problems

Application developers "help" with network aware applications

Which are mostly (non-portable) layering violations
  • Exception: measures of abstract properties - delay, throughput, etc

Web100 adds a separate "management" interface
  • To diagnose and adjust TCP performance

Primary API becomes simple again
  • All applications benefit
In Conclusion:

- Replace current messaging with XML to allow inter-operability (near-term).
- Replace current authentication framework with X.509 over OpenSSL (near-term).
- Tackle secure upload and resource management:
  - packet capture & sending (pcapd)
  - need language for specifying resource access & control policies
- Support for adaptive measurement.
The End
Web100 Key Technical Components

When there is a problem, just ask TCP
• TCP has an ideal vantage point
• TCP can identify the bottleneck subsystem
• TCP already measures the network
• TCP can measure the application

Autotune TCP/IP
• Require less expertise from users
• Net100 formalizing "Workarounds" in a deamon

Global impact through
• Standards
  • draft-ietf-tsvwg-tcp-mib-extension-01.txt
• Commercial adoption