Scriptroute

Authors: Neil Spring
David Wetherall
Tom Anderson

Published: USITS, 2003

Problem

• Internet measurement
  – diagnosing problems
    • connectivity between hosts
    • routing / network paths
  – performance
    • available bandwidth
    • congestion
Two Solutions

• NIMI
  - lots of functionality
  - 'closed' system
  - seems to be the same for SAMI

• traceroute
  - restricted to essentially one function
  - 'open' system

New Solution

• Ideal solution should:
  - have locally maintained infrastructure
  - be easily extendable
    • infrastructure
    • functionality
  - be deployed widely
    • open system
    • some benefit for providers
New Solution

- Real solution needs to balance security with the ideal solution
  - can not facilitate malicious behavior
  - can not open local host to additional attacks
- Have to put security first, but not in an unreasonable way
  - ensure that no amplifications of attack traffic are possible
  - malicious users get no benefit

Scriptroute

- Three basic components
  - front-end
  - interpreter
  - network guardian

Figure from "Scriptroute: A Public Internet Measurement Facility"
Front-end

- Public web server, thttpd
  - limits execution time and size of script
  - limits number of executing scripts
- Users submit scripts via a POST form
- Script is processed
- Results are displayed to user

Interpreter

- Takes submitted script and runs it in a sandboxed environment
- Resource limits are placed on each instance of the interpreter
- Interpreter communicates with network guardian via the Send-train API (sockets)
  - request: set of (delay,probe)
  - reply: set of (time,probe,time,response)
Interpreter

- Resource limits
  - kernel limit on processor time and memory usage
  - one process per running script
  - total number of interpreters (also, per user)
- Each script instance runs in a chroot environment
  - chroot only contains interpreter and logs
  - interpreter runs as nobody in safe-mode

Network Guardian

- Takes packet transmission requests from the interpreter
- Regulates the traffic sent based on configurable filters and limits
- Gathers responses to probes and sends data back to interpreter
- Uses raw sockets and libpcap
**Network Guardian**

- **Resource Limits**
  - probes must be well-formed and meet source and destination requirements
  - rate-limiting of packets based on token buckets (sending rate and number of packets)
- **All packets also logged**
  - provides some level of accountability and reference if problems occur

*Figure from “Scriptroute: A Public Internet Measurement Facility”*
Operation

• Public servers exist, mostly on Planetlab right now (~400 running public servers)
• Users can find a server via the scriptroute homepage and upload scripts via http to run on that server
• Alternatively, a local server can be built
  - encouraged to do this because local (i.e., shell) users have more access to the system, which can facilitate debugging

Examples using Scriptroute

• Reverse Path Tree
  - build a tree representing paths from active scriptroute servers to some destination
  - uses scriptroute version of traceroute combined with IP aliasing and path memory
• See webpage and example
Examples using Scriptroute

- Basic tools (traceroute, ping, etc) provided in distribution of scriptroute
- Quick example of traceroute and ping running from a local install of the server
- Also able to run these from another server
  - web interface
  - local ruby interface
  - need an authentication cookie for Planetlab

Conclusion

- Open, flexible system
- Server admins retain control
- Has been implemented
- Many servers already available
- Appears to maintain security