Netalyzr Updates

Christian Kreibich (ICSI),
Nicholas Weaver (ICSI),
and Vern Paxson (ICSI & UC Berkeley)

Acknowledgements and Important Disclaimers

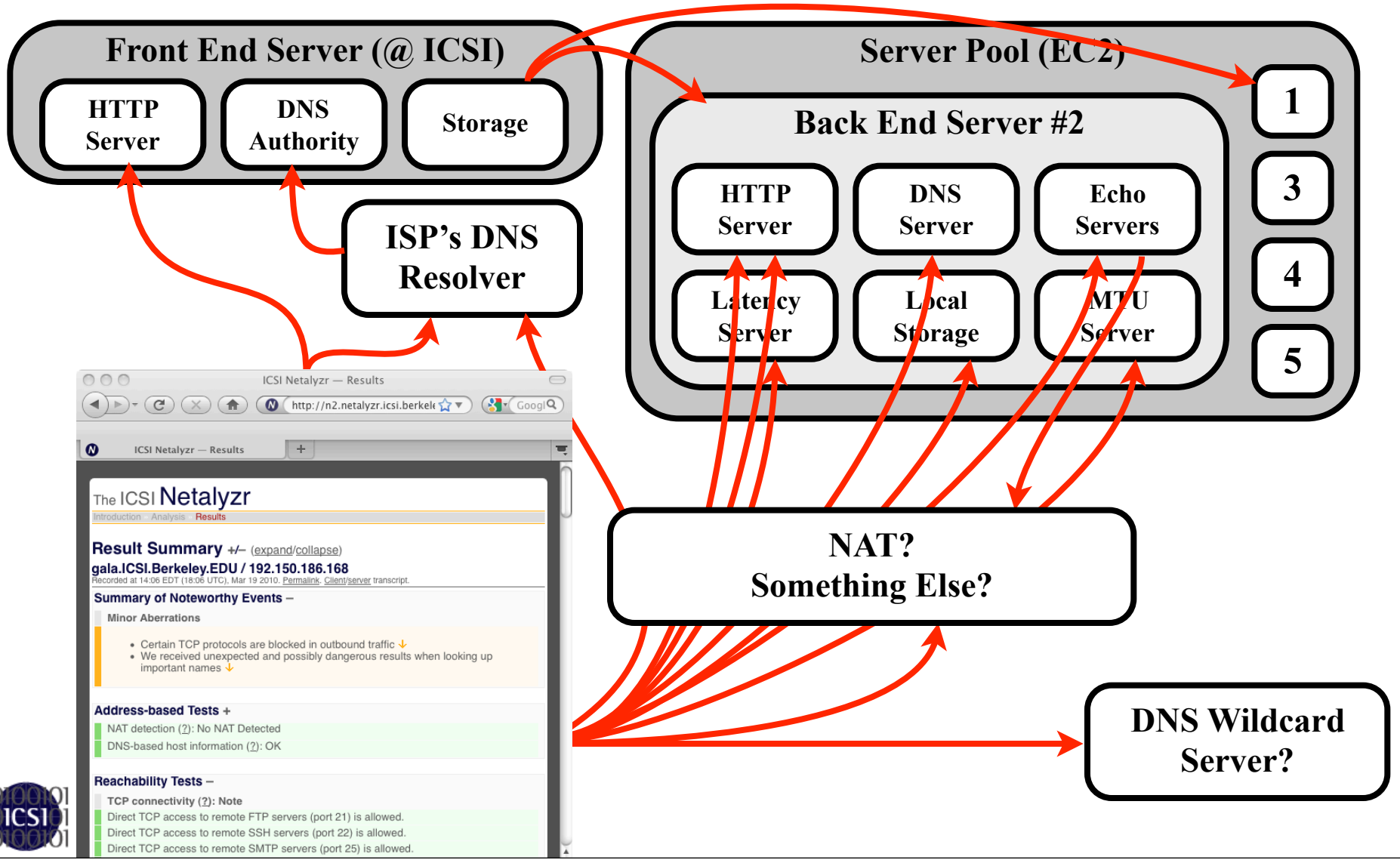
- This work sponsored by the National Science Foundation
  - With additional support from Comcast and Google
  - All opinions are those of the presenter, not those of the sponsors

- This talk is describing our additions in our next release
  - Hopefully Really Soon Now(TM)

Network Transparency And Network Debugging

- How do you know what the network actually is?
  - Network ***Transparency***: What does the network really do to the data?
- What is not working?
  - Network ***Debugging***: Is there something wrong that needs to be fixed
- We desired a comprehensive tool for multiple roles
  - An easy to use network survey for everyone
  - A detailed diagnostic and debugging tool for experts
- Thus we built ***Netalyzr***, a network debugging and diagnostic tool which runs in the web browser
  - Just two mouseclicks

Netalyzr's Architecture



So What's New?

- General Improvements
  - Now substantially faster
  - Subtle GUI improvements
- Improvements for Researchers
  - How to use Netalyzr in your own research
- Improvements for Educators
  - Sorted transcript makes Netalyzr an excellent classroom exercise
- New and improved tests

Speedups & Improvements

- We kept adding tests, which slowed down execution
- No one change, rather a lot of little optimizations
  - Changing timeouts, more aggressive parallelization, and other such changes
- A test from my home Internet connection now takes 160 seconds
  - Compared with 315 seconds for the older version
- Also subtle graphical improvements
  - Focuses on problems
- And a robustness tweak
  - The bandwidth test was causing network connections to ***crash***!?!??
    - Now we still allow it to crash, but we just keep retrying the results upload until it succeeds

Embedding Netalyzr

- Netalyzr has become a debugging service for others
  - League of Legends uses Netalyzr for user network debugging
- We've cooperated with others using Netalyzr in larger projects
  - HomeNet Profiler invokes Netalyzr
  - Multiple V6 deployment trials refer to Netalyzr
- And were notified of a spontaneous use
  - A researcher who was *parsing our debugging output* to capture the desired parameters!
- We want to improve this process
  - A now supported ***JSON*** representation
  - Mode flags for the command line client
  - A ***JSON*** representation of all sessions associated with a mode

The JSON representation

- JavaScript Object Notation is a standard, portable data format
  - Basic data types: strings, integers, lists, objects/ dictionaries (key/value pairs)
  - Libraries exist for many programming languages
- Replace ***result*** with ***json*** in the rendering:
  - http://netalyzr.icsi.berkeley.edu/***json***/id=example
  - Dumps a JSON object (with pretty-printing enabled)
    - The JSON object is thus “Geek Readable” (but not human readable

The JSON Object

- "id" : session-id
- "transcript":
  - Key/value pair of transcript locations
    - Assumed to be on the server where you fetch the JSON object from
- "uploads":
  - Key/value pairs of uploaded data locations
- "formdata":
  - Key/value pairs of any data input into the form by the user
- "args":
  - An object containing uploaded arguments:
    key = argument name
    value = ***list*** of results as text strings
- "results":
  - A list of the results that would be rendered
    - Only some test results are actually interpreted, however

But how does one *get* the transaction IDs?

- All Netalyzr sessions have an associated mode flag
  - http://netalyzr.icsi.berkeley.edu/m=X
  - java -jar NetalyzrCLI.jar -m X
    - Command line clients automatically have “cli” appended to the mode
- To fetch all sessions associated with a given mode
  - Arrange with us to specify a mode-key
  - http://netalyzr.icsi.berkeley.edu/modes/mode-key=Y
  - Returns a JSON object for the appropriate modes
    - A comment field
    - The regular expression specifying the modes
    - A list of matching modes, containing the session IDs, the IPs, and the mode strings
  - Updates approximately once an hour

So what does this mean?

- Getting session results:
  - You can use the command line client to generate individual results links
  - You can have others submit results links
  - You can arrange to have a unique mode flag with us, to get all results associated with that mode flag
    - Netalyzr now also supports autolaunch http://netalyzr.icsi.berkeley.edu/dispatch.html/m=MYMODE
      - Please, don't do this without explicit user consent
- Analyzing session results:
  - JSON download contains or points to ***all*** data captured by the applet and uploaded to the server
  - Some results from server-side analyses:
    - Buffer capacity estimation
    - Server-side analysis of the "Lookup of important names" test

A good networking class exercise

- Go to a network, run Netalyzr, save the results URL, and explain how tests X, Y, and Z operate
  - The client debugging transcript link on the top of the page conveys a huge amount of information
- Previously, this was somewhat annoying for students, as the output was unsorted
  - Extensive multithreading would interleave the results of various tests
- Now its sorted
  - The students can now easily get all output associated with an individual test
- If you do this, contact us for a custom mode flag...

Some New Tests

- UPnP Probing
  - Discover the NAT and other network devices using UPnP
    - If present, download the XML description
  - Combined with probing the resolver in the NAT with CHAOS queries, this should enable us to develop a good picture of which NATs are associated with which problems
  - Bandwidth tests now probe for UPnP byte counters
- Paxfire detection
  - Some ISPs were using DNS to MITM search engines
    - Replace search results with “to final destination through affiliate program” redirection
- 404 rewriting
    - Some ISPs will MITM HTTP to change 404 errors to redirect to advertisement servers

Some New Tests

- ***Removed*** the virus filter test
  - Symantec AV now blocks Java's network connection after it attempts to download the EICAR test
    - Effectively killing Netalyzr
- Detailed probing of all configured DNS resolvers
  - CHAOS queries to obtain version information
  - Check for "chinese root" problems
    - Where a query to a root server passes through the Great Firewall of China, and therefore Chinese censorship
- Probing of DNS root authorities
  - Identify each root instance using CHAOS queries
  - Check to see if the path passes through the Great Firewall

So What Else Should We Do?

- Are there new tests you'd like to see?
- Can we make it easier for others to use Netalyzr?