Domain Statistics Collector Tutorial

Duane Wessels

DNS-OARC

Advanced ccTLD Workshop

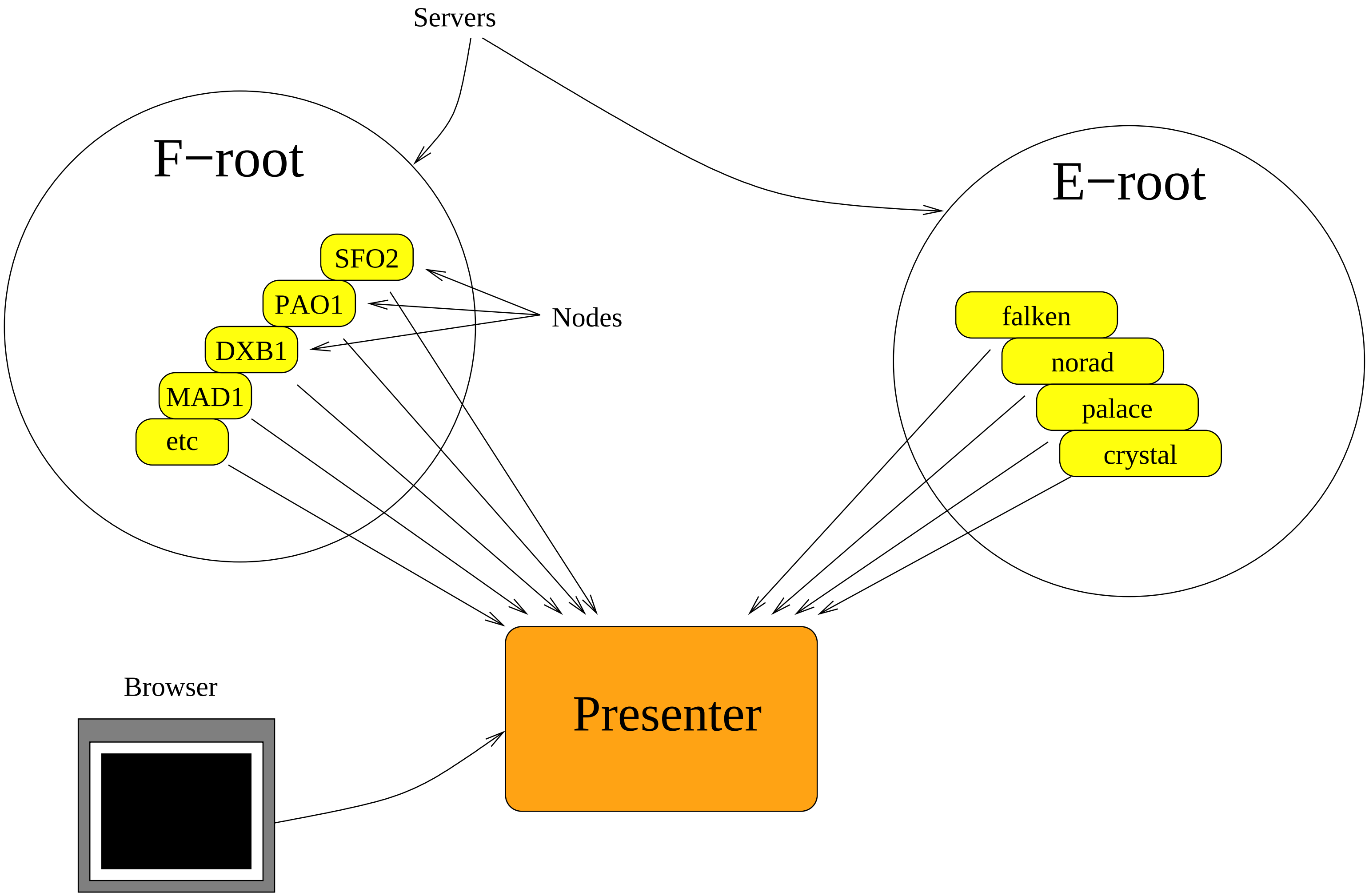
September 16, 2008

What is DSC?

- A system for collecting, transferring, viewing, and storing a variety of measurements taken from DNS servers.
- Open source (BSD license) software that runs on BSD, Linux, and Solaris.
- Used by Root, TLD operators (and others) to visualize DNS traffic characteristics and share data.
- `http://dns.measurement-factory.com/tools/dsc/`

Architecture

DSC Architecture



Collector

- A DSC Collector process runs on (or near) a DNS server node.

- Uses libpcap, just like tcpdump.

- Works with Ethernet taps or port mirroring if you don't want to run it *on* the server itself.

- Can be configured to collect a number of different *Datasets*.

- Writes XML files to disk every 60 seconds for transfer to the Presenter.

Data Transfer

- A cron job runs every minute to transfer XML files from Collector to Presenter.
- Can send to multiple Presenters.
- Usually data is *pushed* rather than *pulled*.
- Scripts are provided to use rsync/SSH.
- Can also use HTTPS and client-side X.509 certificates.

Presenter

- A cron job processes incoming XML files (and stores the data in a format that is faster to read).
- Apache and a CGI script are used to view the data.
- CGI and XML processing can be on different machines if you use NFS.

Storage

- XML files are removed by cron job (for example, after 3 days)
- Other data files remain permanently.
- Data files are stored in SERVER/NODE/YYYYMMDD/*.dat
- Estimate about 500–800 MB to store 1 year of data.

Indexers and Datasets

How DSC Stores Data

- Data is stored in 1- or 2-dimensional arrays of counters.
- The arrays count the number of times that the collector sees packets with certain values, parameters, or characteristics.
- Each array is called a Dataset.
- Here is a simple dataset:

Qtype	1	2	5	12	15	28	38
Count	201	5	9	89	117	52	33

- Note that while (in this example) we could use Qtype as the array index, that doesn't work in general because we also want to count non-numeric things like domain names and IP addresses.
- Thats where Indexers come in...

Indexers

- An Indexer turns some value in a DNS message into an array index.
- Sort of like the way associative arrays work in perl/awk/php/etc.
- Some indexers are small
  - For example, the single-bit Recursion Desired flag
- Some indexers are large
  - For example, the query name or client IP address

Value	Index
www.isoc.org	0
www.icann.org	1
www.google.com	2
www.microsoft.com	3
www.yahoo.com	4
...	...

○ If you want to add a new Indexer, you have to write some C code.

Datasets

- A dataset is an 1D or 2D array of counters.
- Defined by one or two indexers, and given a name.
- Some filters and other options can be applied to Datasets.
- In most cases there is a one-to-one mapping between a Dataset and a graph on the Presenter. Sometimes there is more than one way to display the data.
- Datasets are written to disk every 60 seconds as an XML file.
- If you want to add a new Dataset, add a line to the configuraiton file.

Dataset Examples

```
dataset qtype dns All:null Qtype:qtype queries-only;

dataset rcode_vs_replylen dns Rcode:rcode ReplyLen:msglen
    replies-only;

dataset client_subnet2 dns Class:query_classification
    ClientSubnet:cip4_net quer ies-only max-cells=200;
```

Data Transfer

Getting XML from Collector to Presenter

- DSC doesn’t really care how the XML files get from the Collector to the Presenter.
- Designed for store-and-forward so that data will be queued on the collectors if presenter is unreachable.
- Some scripts are provided that use rsync and X509.
- Also a script to send data to DNS-OARC (using SSH without rsync).
- You could write your own, use NFS, etc.

rsync/SSH

- Probably the best balance between security and simplicity.
- Create a separate SSH key for each NODE.
- Place the NODE's keys in the presenter authorized_keys file.

X509

- Perhaps more secure than SSH, but a hassle to maintain.
- Create X509 keys/certificates for each NODE
- Upload through Apache with custom CGI script.

Demo

Installation

Installing Collector

- Download DSC software from workshop FTP server

```
$ cd
$ fetch ftp://193.0.24.110/pub/dsc-200808221554.tar.gz
$ fetch ftp://ftp.bert/pub/dsc-200808221554.tar.gz
$ tar xzf dsc-200808221554.tar.gz
$ cd dsc-200808221554
$ cd collector
$ make
```

- Oops, we need a Perl module...

```
$ (cd /usr/ports/devel/p5-Proc-PID-File ; sudo make all install)
$ make
$ sudo make install
```

Configuring Collector

```
$ cd /usr/local/dsc/etc
$ cp dsc.conf.sample dsc.conf
$ vi dsc.conf
```

- Can leave most of the defaults as they are.
- Today, pay special attention to:

```
run_dir /usr/local/dsc/run/ns1;
local_address 193.0.__.__;
interface em0;
```

- Create the run_dir

```
$ sudo mkdir -p /usr/local/dsc/run/ns1
```

dsc Test Run

```
$ cd /usr/local/dsc
$ sudo bin/dsc -f -d etc/dsc.conf
$ ls -l run
$ less run/*.xml
```

Running dsc normally

- DSC source distribution includes a BSD-style rc script, but you have to install it manually.

```
$ cd dsc-200808221554
$ sudo install -m 755 collector/dsc/dsc.sh \
  /usr/local/etc/rc.d/dsc

$ sudo /usr/local/etc/rc.d/dsc start
```

Collector Cron Jobs

- upload-prep.pl moves files from dsc run_dir to one or more upload directories.

```
* * * * * /usr/local/dsc/libexec/upload-prep.pl
```

- upload-rsync.sh (or similar) copies XML files from the upload directory to the presenter system.

```
* * * * * /usr/local/dsc/libexec/upload-rsync.sh ns1 \
  noc dsc-pc1@193.0.24.110:/usr/local/dsc/data/pc1/ns1
```

- But don't save the crontab file yet...!

How does upload-prep.pl work?

- upload-prep.pl moves files from dsc run_dir to one or more upload directories.

- You must create these upload directories

```
$ cd /usr/local/dsc/run/ns1
$ sudo mkdir upload
$ sudo mkdir upload/noc
$ sudo mkdir upload/presenter2   # you could have more than one
```

- XML files will stay in these upload directories until they are uploaded and removed.

- Can run out of disk space if not careful.

How does upload-rsync.sh work?

- Takes three arguments: NODENAME UPDIR DESTINATION
- NODENAME is the name of this collector node and must be unique.
  - ▷ example: ns1
- UPDIR is the name of the upload directory
  - ▷ example: noc
- DESTINATION is an rsync-style destination
  - ▷ example: dsc-pc1@193.0.24.110:/usr/local/dsc/data
- Looks for a SSH key at $HOME/.ssh/dsc_uploader_id
  - ▷ `ssh-keygen -d -f $HOME/.ssh/dsc_uploader_id`
- Send your SSH public key to the presenter when its ready and test that it works.

Back to crontab

- Save the new cron jobs
- Wait 60 seconds or less
- Check your mailbox for cron job errors

Installing Presenter

Dependencies

```
$ (cd /usr/ports/*/p5-CGI-Untaint; sudo make all install)
$ (cd /usr/ports/*/p5-File-Flock; sudo make all install)
$ (cd /usr/ports/*/p5-File-NFSLock; sudo make all install)
$ (cd /usr/ports/*/p5-Hash-Merge; sudo make all install)
$ (cd /usr/ports/*/p5-IP-Country; sudo make all install)
$ (cd /usr/ports/*/p5-Math-Calc-Units; sudo make all install)
$ (cd /usr/ports/*/p5-Net-DNS; sudo make all install)
$ (cd /usr/ports/*/p5-Text-Template; sudo make all install)
$ (cd /usr/ports/*/p5-Proc-PID-File ; sudo make all install)
$ (cd /usr/ports/www/apache22; sudo make all install)
$ (cd /usr/ports/math/ploticus; sudo make all install)
```

Install

```
$ cd dsc-200808221554/presenter
$ cd perllib
$ perl Makefile.PL
$ make && sudo make install
$ cd ..
$ make && sudo make install
```

Cron Jobs

```
*/5 * * * * exec /usr/local/dsc/libexec/refile-and-grok.sh
@midnight find /usr/local/dsc/data/*/*/done \
  | /usr/local/dsc/libexec/remove-xmls.pl 3
17 * * * * cd /usr/local/dsc/cache; /bin/ls -t \
  | /usr/bin/tail +500 \
  | /usr/bin/xargs /bin/rm
```

- refile-and-grok.sh processes the incoming XML files
- remove-xmls.pl ... removes old XML files
- Lastly, a job to keep the image cache to a finite size.

The Grapher

- Copy or symlink the dsc-grapher.pl to Apache's cgi-bin directory

- Might need

```
AddHandler cgi-script .pl
Options ExecCGI Includes FollowSymlinks
```

-
  ```
  $ cd htdocs
  $ sudo ln -s /usr/local/dsc/share/html dsc
  ```

- dsc-grapher.cfg

```
$ cd /usr/local/dsc/etc
$ cp dsc-grapher.cfg.sample dsc-grapher.cfg
$ vi dsc-grapher.cfg

server TLD ns1 ns2 ...
```