

Small Scale Data Mining

Fun and Games With Open Source

Dr E.W. Lisse

Namibian Network Information Centre (Pty) Ltd

2013-07-15



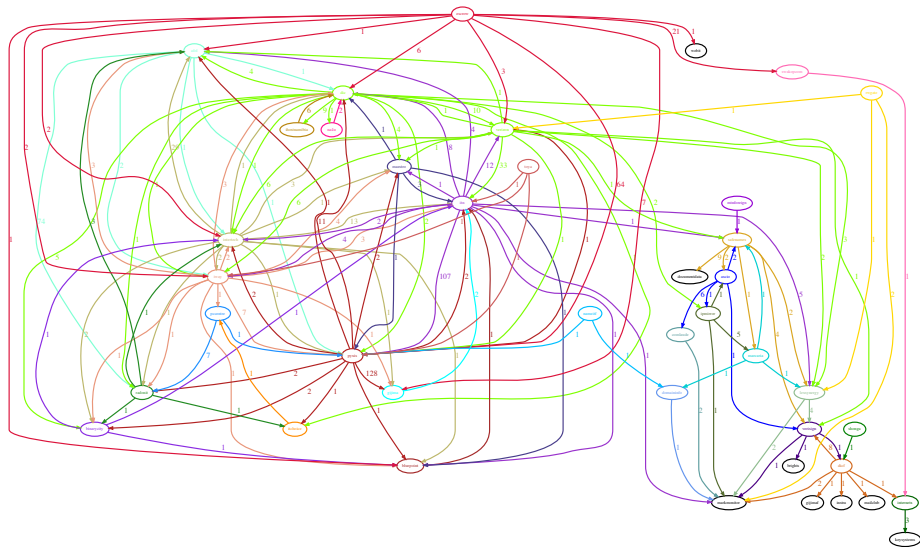
Outline

What is this Talk about?

- 1 Domain Name Transfers
- 2 Transaction Patterns
- 3 Registration Patterns
- 4 Discussion

So, What's This?

.NA[®] Domain Name Transfers



.NA[®] Domain Name Transfers

Background

Background

One Reseller moved between Registrars repeatedly.

.NA[®] Domain Name Transfers

Background

Background

One Reseller moved between Registrars repeatedly.

Issue

Are there Transfer Patterns?

.NA[®] Domain Name Transfers

Background

Background

One Reseller moved between Registrars repeatedly.

Issue

Are there Transfer Patterns?

Tools Used

- Graphviz
- Perl
 - DBI
 - GraphViz2
- SSH Tunnel Manager
- LyX/L^AT_EX
 - Beamer
- librsvg

.NA[®] Domain Name Transfers

Perl Code

Define some \$variables, an @array and a %hash

```
my $i = 0;
my $registrar = '';
my @colors = ("aquamarine", "blue", "blueviolet", "brown",
...
            "violetred", "wheat", "yellow", "yellowgreen");
my %farbe;
```

.NA[®] Domain Name Transfers

Perl Code

Define some \$variables, an @array and a %hash

```
my $i = 0;
my $registrar = '';
my @colors = ("aquamarine", "blue", "blueviolet", "brown",
...
              "violetred", "wheat", "yellow", "yellowgreen");
my %farbe;
```

Prepare GraphViz2

```
use GraphViz2;
my $PENWIDTH = 3;
my $driver = 'dot';
my($graph) = GraphViz2 -> new
(
    edge    => {penwidth => $PENWIDTH, fontsize => 20 },
    global  => {driver => $driver, directed => 1 },
    node    => {penwidth => $PENWIDTH, fontsize => 11 },
);
```


.NA[®] Domain Name Transfers

Perl Code

Connect to the database

```
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```

.NA[®] Domain Name Transfers

Perl Code

Connect to the database

```
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```

Construct the Query

```
my $TRANSFERS = 5;
my $sql = qq|SELECT t.owner_clid, t.transfer_to_clid, COUNT(t.*)
FROM transfer_request t
WHERE t.response LIKE '%APPROVED'
GROUP BY t.transfer_to_clid, t.owner_clid
HAVING COUNT(t.*) > $TRANSFERS
ORDER BY t.owner_clid, t.transfer_to_clid|;
```

.NA[®] Domain Name Transfers

Perl Code

Connect to the database

```
use DBI; my $EPP = DBI->connect("dbi:Pg:dbname=epp");
```

Construct the Query

```
my $TRANSFERS = 5;
my $sql = qq|SELECT t.owner_clid, t.transfer_to_clid, COUNT(t.*)
FROM transfer_request t
WHERE t.response LIKE '%APPROVED'
GROUP BY t.transfer_to_clid, t.owner_clid
HAVING COUNT(t.*) > $TRANSFERS
ORDER BY t.owner_clid, t.transfer_to_clid|;
```

Execute the Query

```
my $dbh = $EPP->prepare($sql);
$dbh->execute or die;
```

.NA[®] Domain Name Transfers

Result of the Query, abbreviated (\$TRANSFERS = 5) and sorted

<i>owner_clid</i>	<i>transfer_to_clid</i>	<i>count</i>
afol	radonit	75
ascio	comlaude	6
dic	ihostnamibia	9
dic	verizon	10
dicf	verisign	8
escrow	dic	6
escrow	gijima	7
escrow	pyxis	64
escrow	swakopcom	21
...		
verizon	intertech	6
verizon	itn	33

.NA[®] Domain Name Transfers

Perl Code

Fill up the Registrar hash, recycle colors, and 'draw' the 'edges'

```
while ( my @rec = $dbh->fetchrow_array ) {
    if ( $registrar ne $rec[0] ) {
        $farbe{$rec[0]} = $colors[$i];
        $registrar = $rec[0];

        if ( $i > $#colors ) {
            $i = 0;
        } else {
            $i++;
        }
    }
    $graph -> add_edge(from => $rec[0], to => $rec[1],
        label => "$rec[2]",
        color => $farbe{$rec[0]}, fontcolor => $farbe{$rec[0]});
}
```

.NA[®] Domain Name Transfers

Perl Code

'Draw' all loosing Registrars from the Hash

```
foreach my $looser ( sort keys %farbe ) {  
    $graph -> add_node( name => $looser, color => $farbe{$looser},  
                        fontcolor => $farbe{$looser} );  
}
```

.NA[®] Domain Name Transfers

Perl Code

'Draw' all loosing Registrars from the Hash

```
foreach my $looser ( sort keys %farbe ) {  
    $graph -> add_node( name => $looser, color => $farbe{ $looser },  
                        fontcolor => $farbe{ $looser } );  
}
```

Run Graphviz, Writing the SVG file (here 't5.svg')

```
$graph -> run( format => 'svg', output_file => "t$TRANSFERS.svg" );
```

.NA[®] Domain Name Transfers

Perl Code

'Draw' all loosing Registrars from the Hash

```
foreach my $looser ( sort keys %farbe ) {  
    $graph -> add_node( name => $looser, color => $farbe{$looser},  
                        fontcolor => $farbe{$looser} );  
}
```

Run Graphviz, Writing the SVG file (here 't5.svg')

```
$graph -> run( format => 'svg', output_file => "t$TRANSFERS.svg" );
```

Write the DOT file (here 't5.dot') (Optional)

```
open( DOT, ">t$TRANSFERS.dot" ) or die;  
print DOT $graph -> dot_input();  
close DOT;
```


.NA[®] Domain Name Transfers

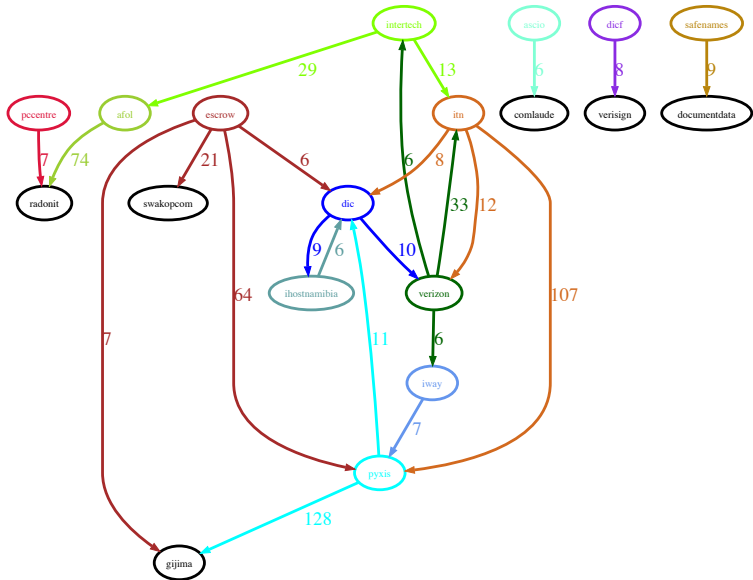
Graphviz Code, abbreviated ('t5.dot')

```
digraph Perl
```

```
{
  node [fontsize="11" penwidth="3"];
  edge [fontsize="20" penwidth="3"];
  "afol" -> "radonit" [color="aquamarine" fontcolor="aquamarine" label="75"];
  "afol" [color="aquamarine" fontcolor="aquamarine"];
  "ascio" -> "comlaude" [color="blue" fontcolor="blue" label="6"];
  "ascio" [color="blue" fontcolor="blue"];
  "dic" -> "ihostnamibia" [color="blueviolet" fontcolor="blueviolet" label="9"];
  "dic" -> "verizon" [color="blueviolet" fontcolor="blueviolet" label="10"];
  "dic" [color="blueviolet" fontcolor="blueviolet"];
  "dicf" -> "verisign" [color="cadetblue" fontcolor="cadetblue" label="8"];
  "dicf" [color="cadetblue" fontcolor="cadetblue"];
  "escrow" -> "dic" [color="chartreuse" fontcolor="chartreuse" label="6"];
  "escrow" -> "gijima" [color="chartreuse" fontcolor="chartreuse" label="7"];
  "escrow" -> "pyxis" [color="chartreuse" fontcolor="chartreuse" label="64"];
  "escrow" -> "swakopcom" [color="chartreuse" fontcolor="chartreuse" label="21"];
  "escrow" [color="chartreuse" fontcolor="chartreuse"];
  ...
  "verizon" -> "intertech" [color="darkviolet" fontcolor="darkviolet" label="6"];
  "verizon" -> "itn" [color="darkviolet" fontcolor="darkviolet" label="33"];
  "verizon" -> "iway" [color="darkviolet" fontcolor="darkviolet" label="6"];
  "verizon" [color="darkviolet" fontcolor="darkviolet"];
}
```

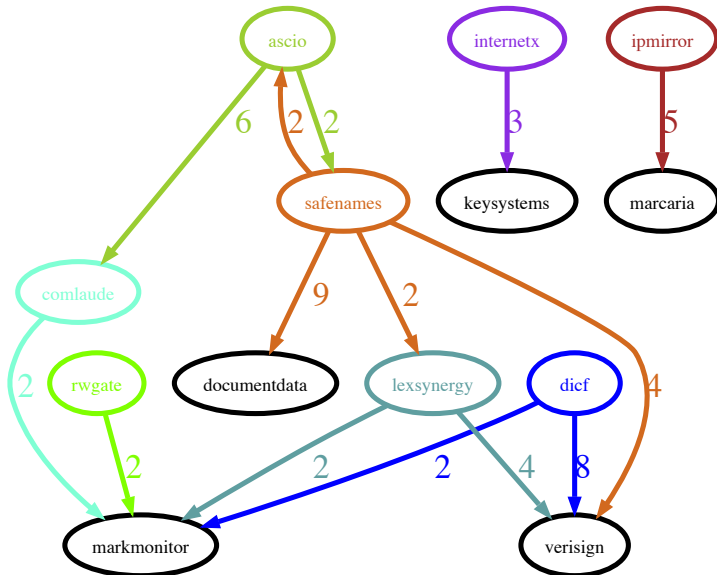
.NA[®] Domain Name Transfers

('t5.svg')



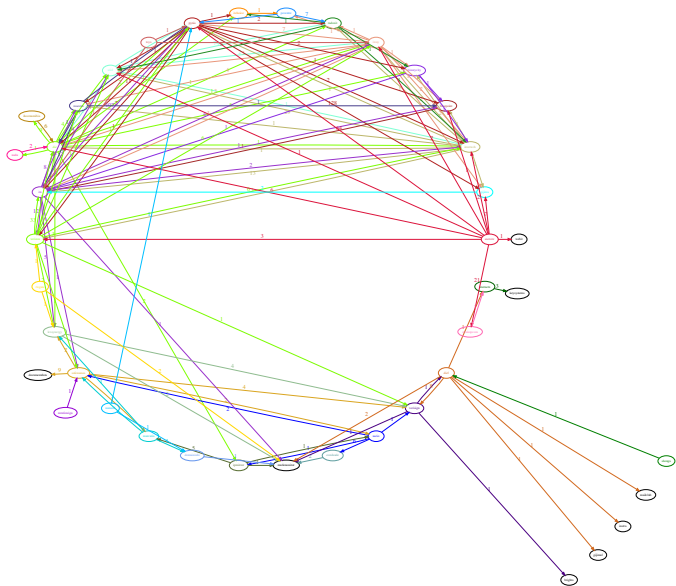
.NA[®] Domain Name Transfers

More than 1 Transfer (from Foreign Registrar)



.NA[®] Domain Name Transfers

All Transfers (using circo)



Background

- Local Resellers Requested 24/7 Availability (2005)
- CoCCATools was deployed (2007)
- Resellers were promoted to Registrars (2008)

Transaction Patterns

24/7?

Background

- Local Resellers Requested 24/7 Availability (2005)
- CoCCATools was deployed (2007)
- Resellers were promoted to Registrars (2008)

Issue

Can Registrar Activity be displayed by the hour?

Transaction Patterns

24/7?

Background

- Local Resellers Requested 24/7 Availability (2005)
- CoCCATools was deployed (2007)
- Resellers were promoted to Registrars (2008)

Issue

Can Registrar Activity be displayed by the hour?

Tools Used

- R
 - RPostgreSQL
 - plotrix
- SSH Tunnel Manager

Transaction Patterns

R Code

Set Up the Dates

```
mindate <- as.Date("2012-06-01")  
maxdate <- as.Date("2013-05-31")
```

Transaction Patterns

R Code

Set Up the Dates

```
mindate <- as.Date("2012-06-01")  
maxdate <- as.Date("2013-05-31")
```

Pull in the Packages

```
require(RPostgreSQL)  
require(plotrix)
```

Transaction Patterns

R Code

Set Up the Dates

```
mindate <- as.Date("2012-06-01")  
maxdate <- as.Date("2013-05-31")
```

Pull in the Packages

```
require(RPostgreSQL)  
require(plotrix)
```

Configure the Driver and the Connection

```
drv <- dbDriver("PostgreSQL")  
con <- dbConnect(drv, dbname = "epp")
```

Transaction Patterns

R Code

Run the Query (for Nambian Registrars)

```
hourNA <- dbGetQuery(con, paste("
SELECT
  DATE_PART('hour',l.created) AS hour, COUNT(*) AS domains
FROM ledger l, domain d, client c
WHERE l.domain_name = d.name
      AND d.clid = c.clid AND c.country = 'NA'
      AND l.trans_type ilike 'Re%'
      AND DATE(l.created) BETWEEN '" , mindate , "' AND '" , maxdate , "'
GROUP BY DATE_PART('hour',l.created)
ORDER BY DATE_PART('hour',l.created) ASC
" , sep = ""))
```

Transaction Patterns

R Code

Run the Query (for Nambian Registrars)

```
hourNA <- dbGetQuery(con, paste("
SELECT
  DATE_PART('hour',l.created) AS hour, COUNT(*) AS domains
FROM ledger l, domain d, client c
WHERE l.domain_name = d.name
      AND d.clid = c.clid AND c.country = 'NA'
      AND l.trans_type ilike 'Re%'
      AND DATE(l.created) BETWEEN '" , mindate , "' AND '" , maxdate , "'
GROUP BY DATE_PART('hour',l.created)
ORDER BY DATE_PART('hour',l.created) ASC
" , sep = ""))
```

Make Data Frames, count Transactions, set up Y-Axis

```
hourlyNA <- data.frame(hour = hourNA$hour, domains = hourNA$domains)
hourlyFOR <- data.frame(hour = hourlyFOR$hour, domains = hourlyFOR$domains)
totNA <- sum(hourlyNA$domains)
totFOR <- sum(hourlyFOR$domains)
maxhoursNA <- round(max(hourlyNA$domains) + 50, -2)
```

Transaction Patterns

R Code

Plot the Lines in Green

```
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green",  
     lwd = 5, ylab = "Domain Names", xlab = "", ylim = c(0,maxhoursNA),  
     xlim = c(-0.1,23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

Transaction Patterns

R Code

Plot the Lines in Green

```
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green",  
     lwd = 5, ylab = "Domain Names", xlab = "", ylim = c(0,maxhoursNA),  
     xlim = c(-0.1,23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

Plot the Bars in Blue

```
lines(hourlyNA$hour, hourlyNA$domains, col = "blue", type = "h", lwd = 10)
```

Transaction Patterns

R Code

Plot the Lines in Green

```
plot(hourlyFOR$hour, hourlyFOR$domains, type = "l", col = "green",  
     lwd = 5, ylab = "Domain Names", xlab = "", ylim = c(0,maxhoursNA),  
     xlim = c(-0.1,23.2), yaxs = "i", xaxs = "i", xaxt = "n", frame.plot = FALSE)
```

Plot the Bars in Blue

```
lines(hourlyNA$hour, hourlyNA$domains, col = "blue", type = "h", lwd = 10)
```

Title, Axis, Legend

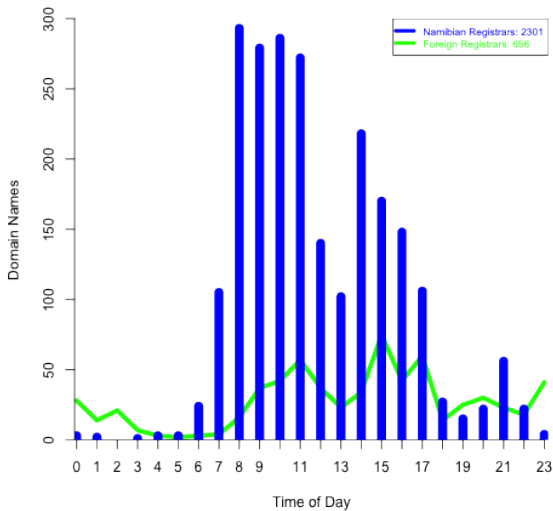
```
title(xlab = "Time of Day")
```

```
axis(1, 0:23)
```

```
legend('topright', c(paste("Namibian Registrars:",totNA),  
                    paste("Foreign Registrars:", totFOR)), col = c("blue", "green", 0),  
      text.col = c("blue", "green", "black"), pch = NA, lwd = 4, cex = 0.7, bg = 0)
```


Transaction Patterns

Results (not 24/7)



Registration Patterns

Abbreviated

Issue

What is the ratio between 2nd vs 3rd level domain names?

Registration Patterns

Abbreviated

Issue

What is the ratio between 2nd vs 3rd level domain names?

Run the query

```
dpct <- dbGetQuery(con, paste("SELECT
  DISTINCT UPPER(d.zone) AS zone,
  COUNT(d.*) AS domains,
  ROUND(COUNT(d.*) * 100.0/(SELECT COUNT(*) FROM domain), 2) AS percent
FROM domain d, ledger l
WHERE l.domain_name = d.name
GROUP BY zone
ORDER BY percent DESC"))
```

Date Constraint not displayed

Registration Patterns

Abbreviated

Issue

What is the ratio between 2nd vs 3rd level domain names?

Run the query

```
dpct <- dbGetQuery(con, paste("SELECT
  DISTINCT UPPER(d.zone) AS zone,
  COUNT(d.*) AS domains,
  ROUND(COUNT(d.*) * 100.0/(SELECT COUNT(*) FROM domain), 2) AS percent
FROM domain d, ledger l
WHERE l.domain_name = d.name
GROUP BY zone
ORDER BY percent DESC"))
```

Date Constraint not displayed

Draw the Chart

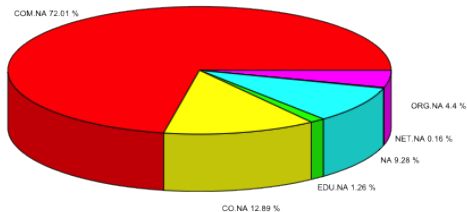
```
pie3D(dpct$domains, labels = paste(dpct$zone, dpct$percent, sep=" "),
  col = c(2,3,4,5,6,7,8,9,10), labelcol = "black", explode = 0,
  labelcex = 0.6, height = 0.15, radius = 1)
```

Title and Legend not displayed

Registration Patterns

Result of the Query

<i>zone</i>	<i>percent</i>
COM.NA	72.01
CO.NA	12.89
NA	9.28
ORG.NA	4.40
EDU.NA	1.26
NET.NA	0.16



Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
 - On any SQL database, in fact
 - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
 - SELECT is safe
- Automatic Report Generation
 - SWEAVE
 - \LaTeX
 - RGraphviz
- Report / Presentation Writing
 - \LaTeX / \LyX
 - Beamer
 - (SWEAVE)
 - Open Office?
- Kolophonium
 - Open Source
 - Quality Output

Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
 - On any SQL database, in fact
 - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
 - SELECT is safe
- Automatic Report Generation
 - SWEAVE
 - L^AT_EX
 - RGraphviz
- Report / Presentation Writing
 - L^AT_EX/L^AT_EX
 - Beamer
 - (SWEAVE)
 - Open Office?
- Kolophonium
 - Open Source
 - Quality Output

Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
 - On any SQL database, in fact
 - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
 - SELECT is safe
- Automatic Report Generation
 - SWEAVE
 - \LaTeX
 - RGraphviz
- Report / Presentation Writing
 - \LaTeX / \LyX
 - Beamer
 - (SWEAVE)
 - Open Office?
- Kolophonium
 - Open Source
 - Quality Output

Discussion

Much More is Possible

- Can be done on FRED and CoCCATools
 - On any SQL database, in fact
 - Any Aggregate (GROUP BY, COUNT, etc) SQL Query can be graphed
 - SELECT is safe
- Automatic Report Generation
 - SWEAVE
 - \LaTeX
 - RGraphviz
- Report / Presentation Writing
 - \LaTeX / \LyX
 - Beamer
 - (SWEAVE)
 - Open Office?
- **Kolophonium**
 - Open Source
 - Quality Output