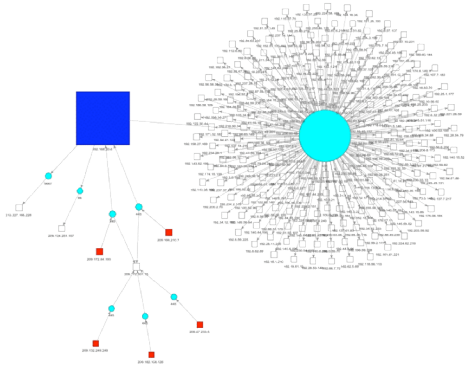
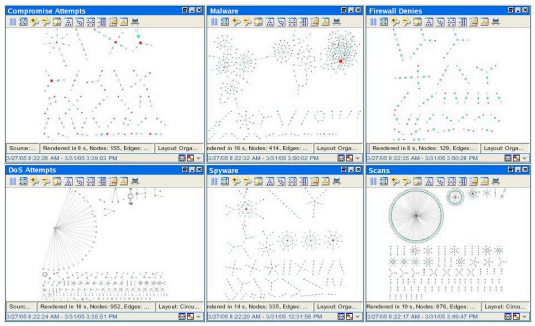
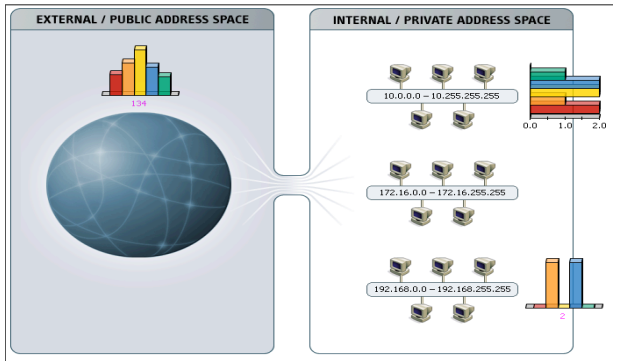
Visual Log Analysis – The Beauty of Graphs

DefCon 2006, Las Vegas





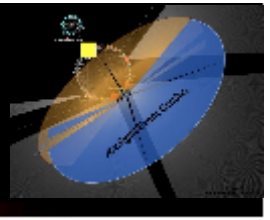


Raffael Marty, GCIA, CISSP
Manager Solutions @ ArcSight

August 5th, 2006

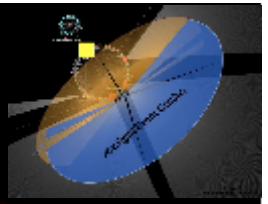
*

Raffael Marty, GCIA, CISSP

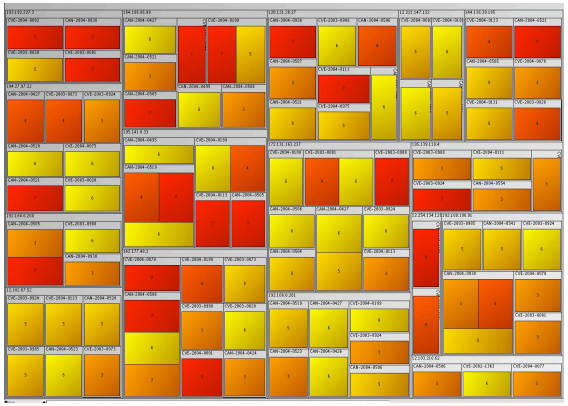


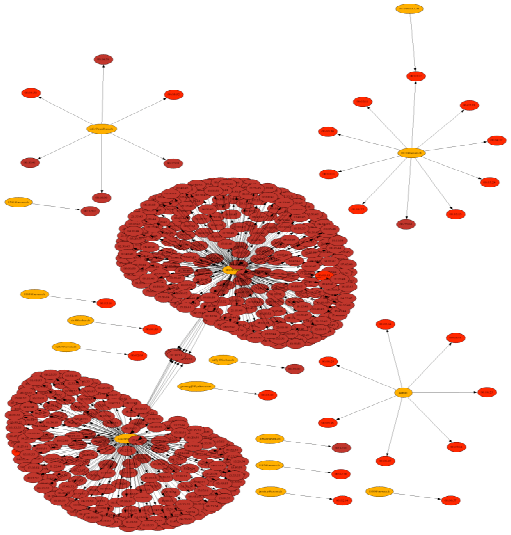
- Enterprise Security Management (ESM) specialist
- Strategic Application Solutions @ ArcSight, Inc.
- Intrusion Detection Research @ IBM Research
  - See http://**thor.cryptojail.net**
- IT Security Consultant @ PriceWaterhouse Coopers
- Open Vulnerability and Assessment Language (OVAL) board member
- Passion for Visual Security Event Analysis

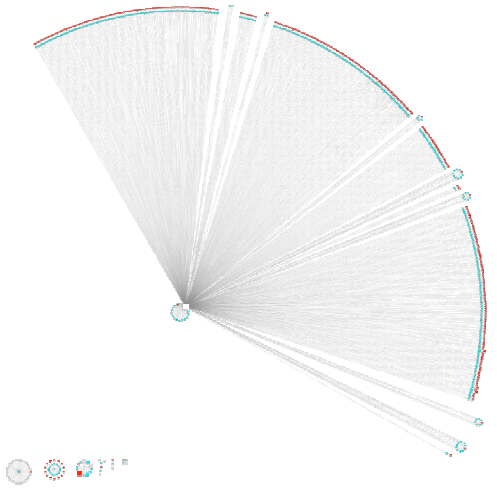
Table Of Contents



- Introduction
- Graphing Basics
- AfterGlow
- Firewall Log File Analysis

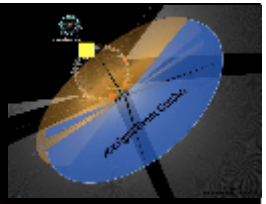






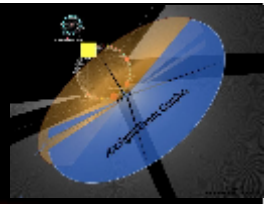
Introduction

Disclaimer



> IP addresses and host names showing up in event graphs and descriptions were obfuscated/changed. The addresses are completely random and any resemblance with well-known addresses or host names are purely coincidental.

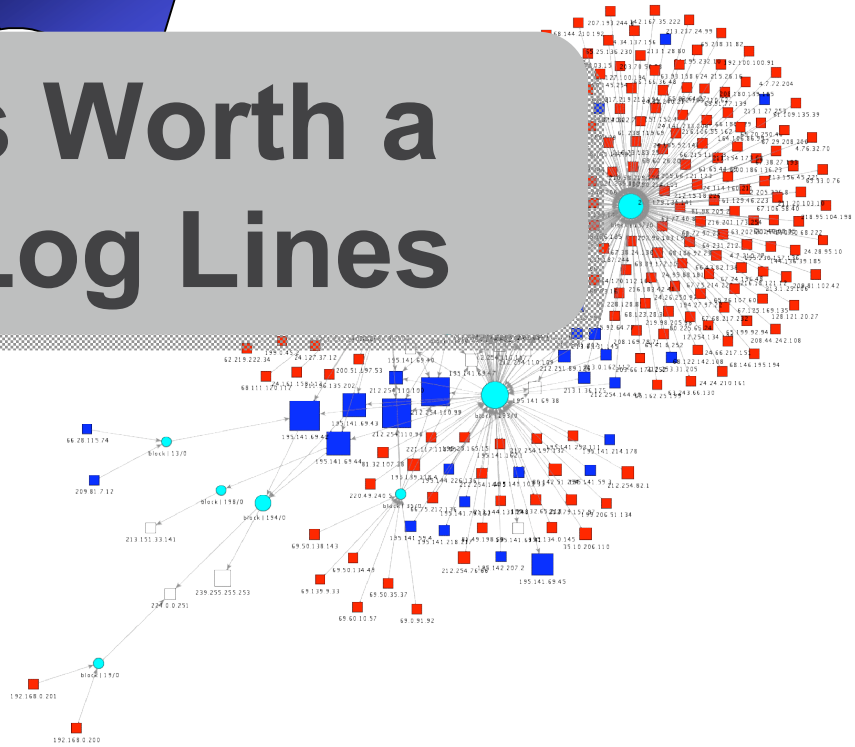
Text or Visuals?

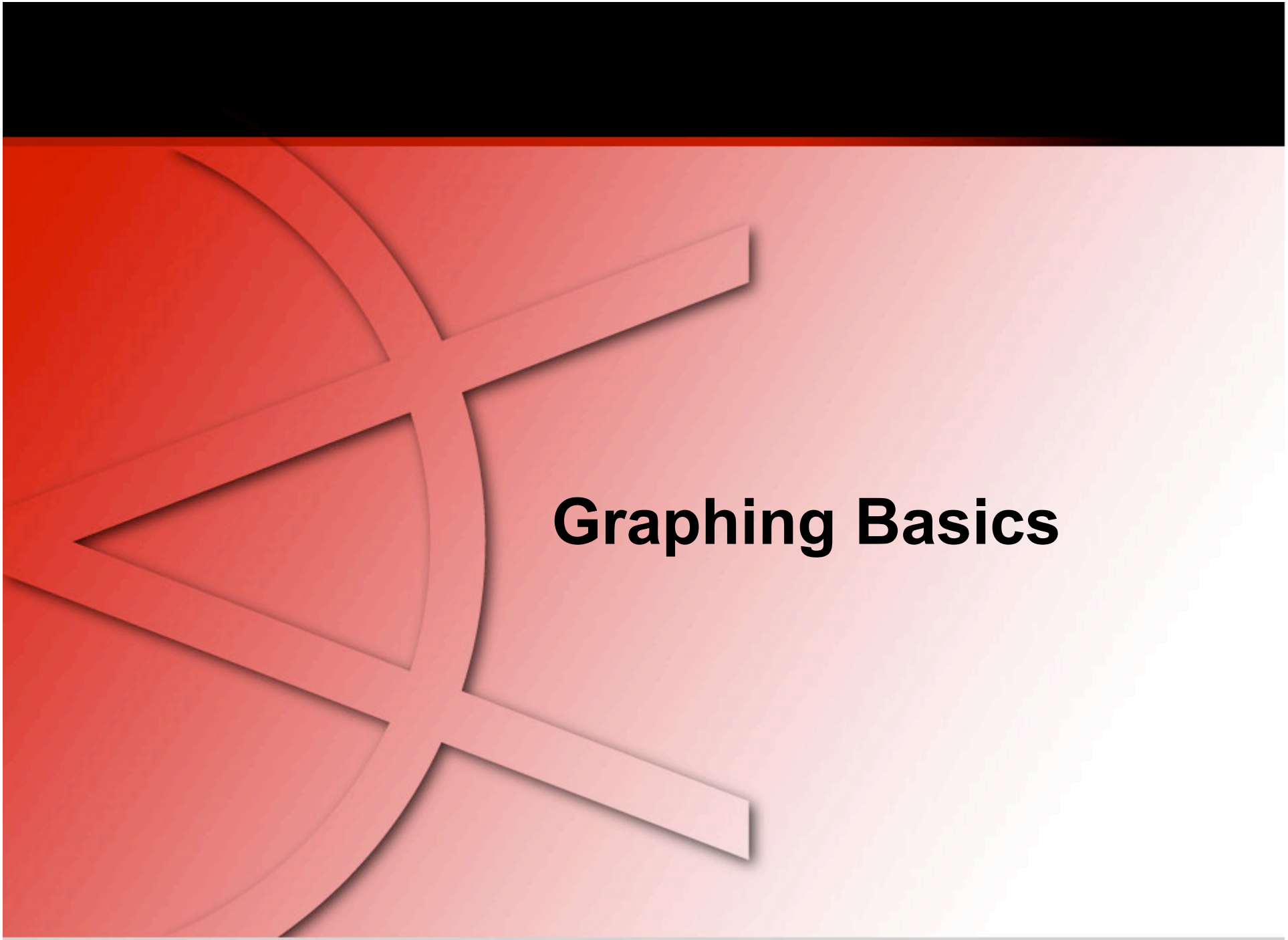


- What would you rather look at?

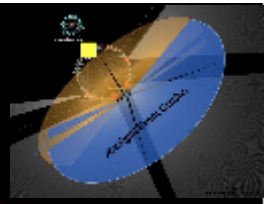
```
Jun 17 09:42:30 rmarty ifup: Determining IP information for eth0...
Jun 17 09:42:35 rmarty ifup:  failed; no link present.  Check cable?
Jun 17 09:42:35 rmarty network: Bringing up interface eth0:  failed
Jun 17 09:42:38 rmarty sendmail: sendmail shutdown succeeded
Jun 17 09:42:38 rmarty sendmail: sm-client shutdown succeeded
Jun 17 09:42:39 rmarty sendmail: sendmail startup succeeded
Jun 17 09:42:39 rmarty sendmail: sm-client startup succeeded
Jun 17 09:43:39 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 09:45:42 rmarty last message repeated 2 times
Jun 17 09:45:47 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 09:56:02 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPOFFER on 172.16.48.128 to 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPREQUEST for 172.16.48.128 (172.16.48.254) from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPACK on 172.16.48.128 to 00:0c:29:b7:b2:47 via vmnet8
Jun 17 10:00:03 rmarty cro
Jun 17 10:00:10 rmarty cro
Jun 17 10:01:02 rmarty cro
Jun 17 10:01:07 rmarty cro
Jun 17 10:05:02 rmarty cro
Jun 17 10:05:05 rmarty cro
Jun 17 10:13:05 rmarty por
Jun 17 10:13:05 rmarty por
Jun 17 10:14:09 rmarty por
Jun 17 10:14:09 rmarty por
Jun 17 10:14:09 rmarty por
Jun 17 10:14:09 rmarty por
Jun 17 10:21:30 rmarty por
Jun 17 10:21:30 rmarty por
Jun 17 10:28:40 rmarty vmn
Jun 17 10:28:41 rmarty vmn
Jun 17 10:28:41 rmarty vmnet
Jun 17 10:28:45 rmarty vmnet-dhcpd: DHCPREQUEST for 172.16.48.128 (172.16.48.254) from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 10:30:47 rmarty portsentry[4797]: attackalert: UDP scan from host: 192.168.80.8/192.168.80.8 to UDP port: 68
Jun 17 10:30:47 rmarty portsentry[4797]: attackalert: Host: 192.168.80.8/192.168.80.8 is already blocked Ignoring
Jun 17 10:30:47 rmarty portsentry[4797]: attackalert: UDP scan from host: 192.168.80.8/192.168.80.8 to UDP port: 68
Jun 17 10:30:47 rmarty portsentry[4797]: attackalert: Host: 192.168.80.8/192.168.80.8 is already blocked Ignoring
Jun 17 10:35:28 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 10:35:31 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 10:38:51 rmarty vmnet-dhcpd: DHCPREQUEST for 172.16.48.128 from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 10:38:52 rmarty vmnet-dhcpd: DHCPACK on 172.16.48.128 to 00:0c:29:b7:b2:47 via vmnet8
Jun 17 10:42:35 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 10:42:38 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
```

A Picture is Worth a Thousand Log Lines

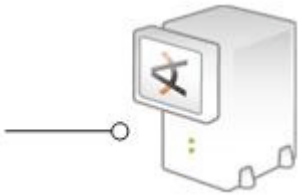



Graphing Basics

How To Generate A Graph

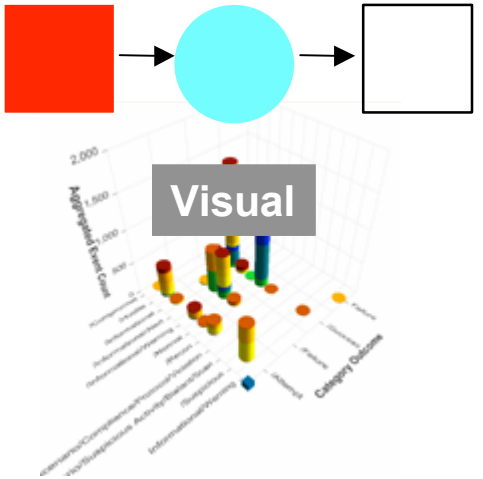


... | Normalization | ...



Device **Parser** **Event Visualizer**

```
Jun 17 09:42:30 rmarty ifup: Determining IP information for eth0...
Jun 17 09:42:35 rmarty ifup:  failed; no link present.  Check cable?
Jun 17 09:42:35 rmarty network: Bringing up interface eth0:  failed
Jun 17 09:42:38 rmarty sendmail: sendmail shutdown succeeded
Jun 17 09:42:38 rmarty sendmail: sm-client shutdown succeeded
Jun 17 09:42:39 rmarty sendmail: sendmail startup succeeded
Jun 17 09:42:39 rmarty sendmail: sm-client startup succeeded
Jun 17 09:43:39 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 09:45:42 rmarty last message repeated 2 times
Jun 17 09:45:47 rmarty vmnet-dhcpd: DHCPINFORM from 172.16.48.128
Jun 17 09:56:02 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPOFFER on 172.16.48.128 to 00:0c:29:b7:b2:47 via vmnet8
NH
```

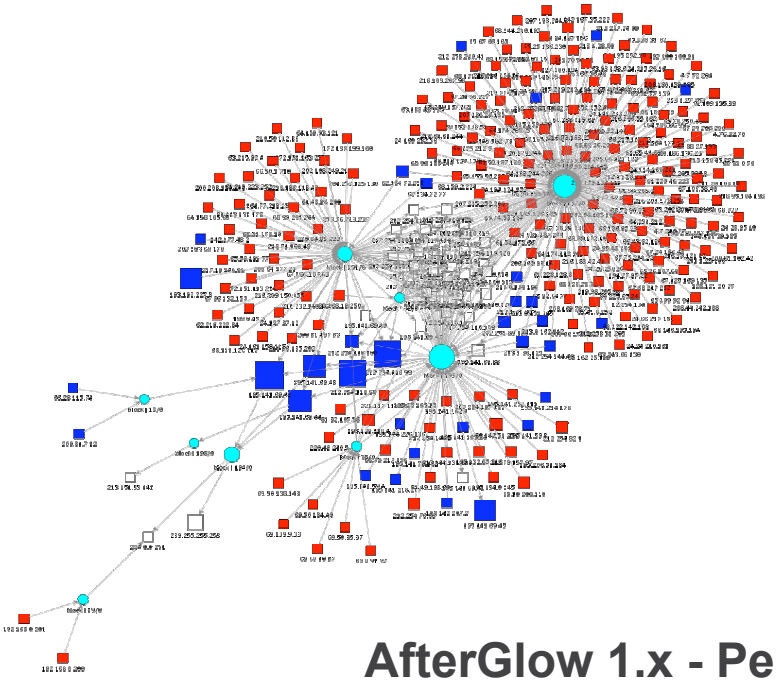


Log File

Visual Types

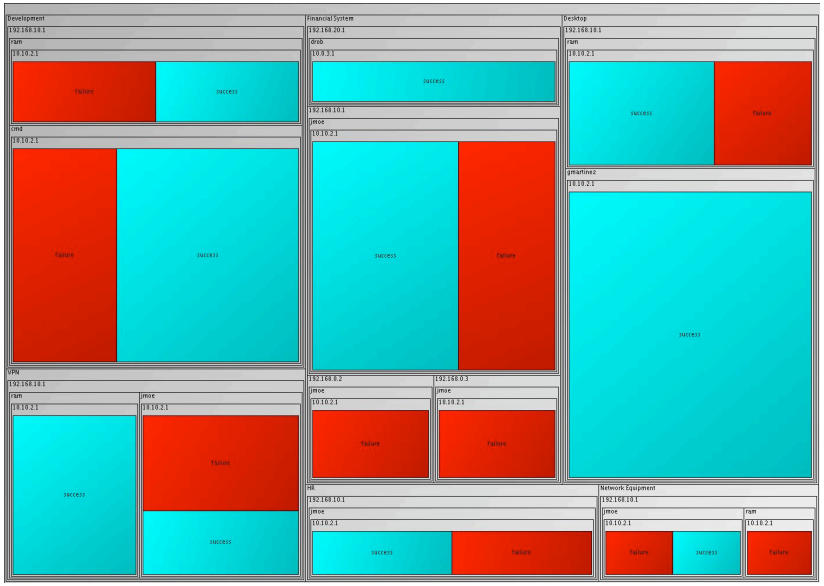
- Visuals that AfterGlow supports:

Link Graphs



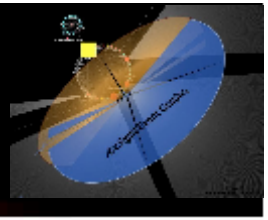
AfterGlow 1.x - Perl

TreeMaps



AfterGlow 2.0 - JAVA

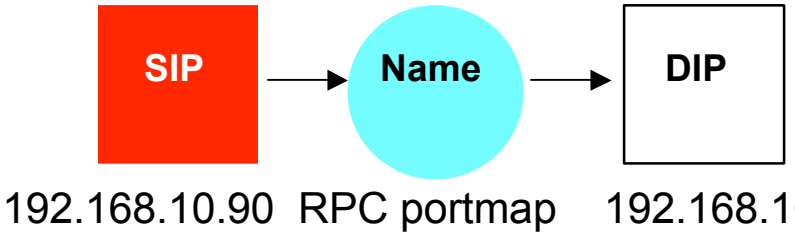
Link Graph Configurations

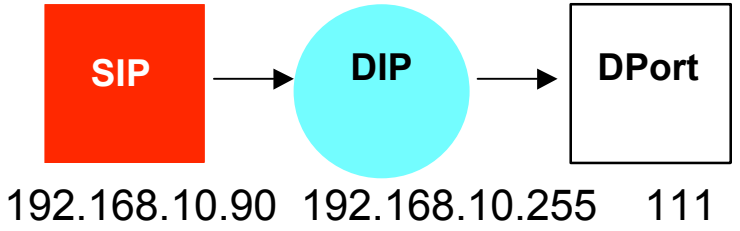


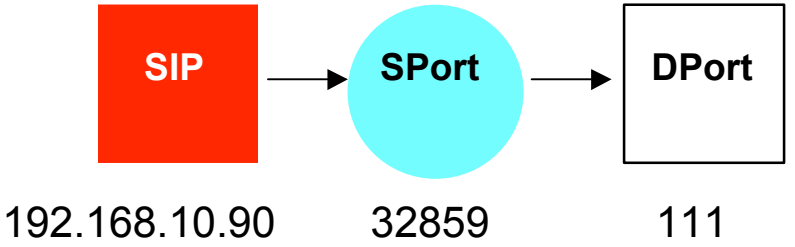
Raw Event:

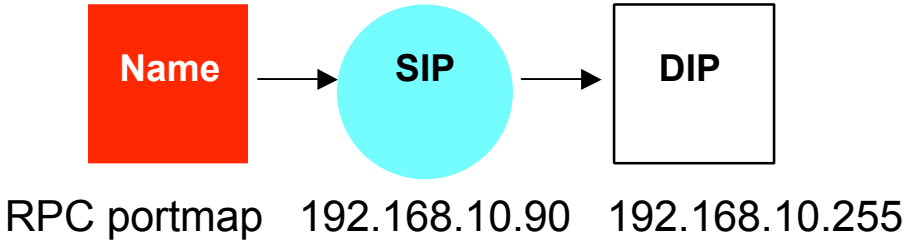
```
[**] [1:1923:2] RPC portmap UDP proxy attempt [**]
[Classification: Decode of an RPC Query] [Priority: 2]
06/04-15:56:28.219753 192.168.10.90:32859 ->
192.168.10.255:111
UDP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:148 DF
Len: 120
```

Different node configurations:

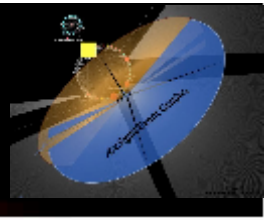


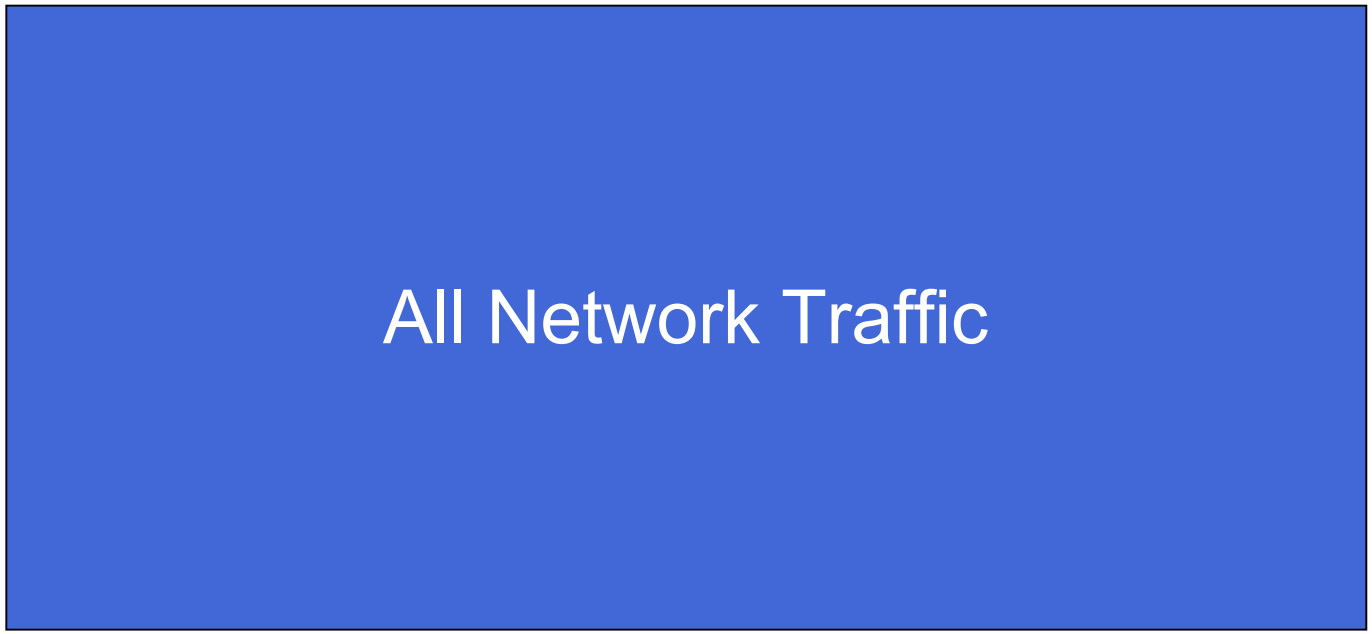




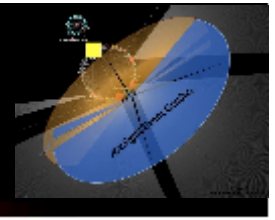


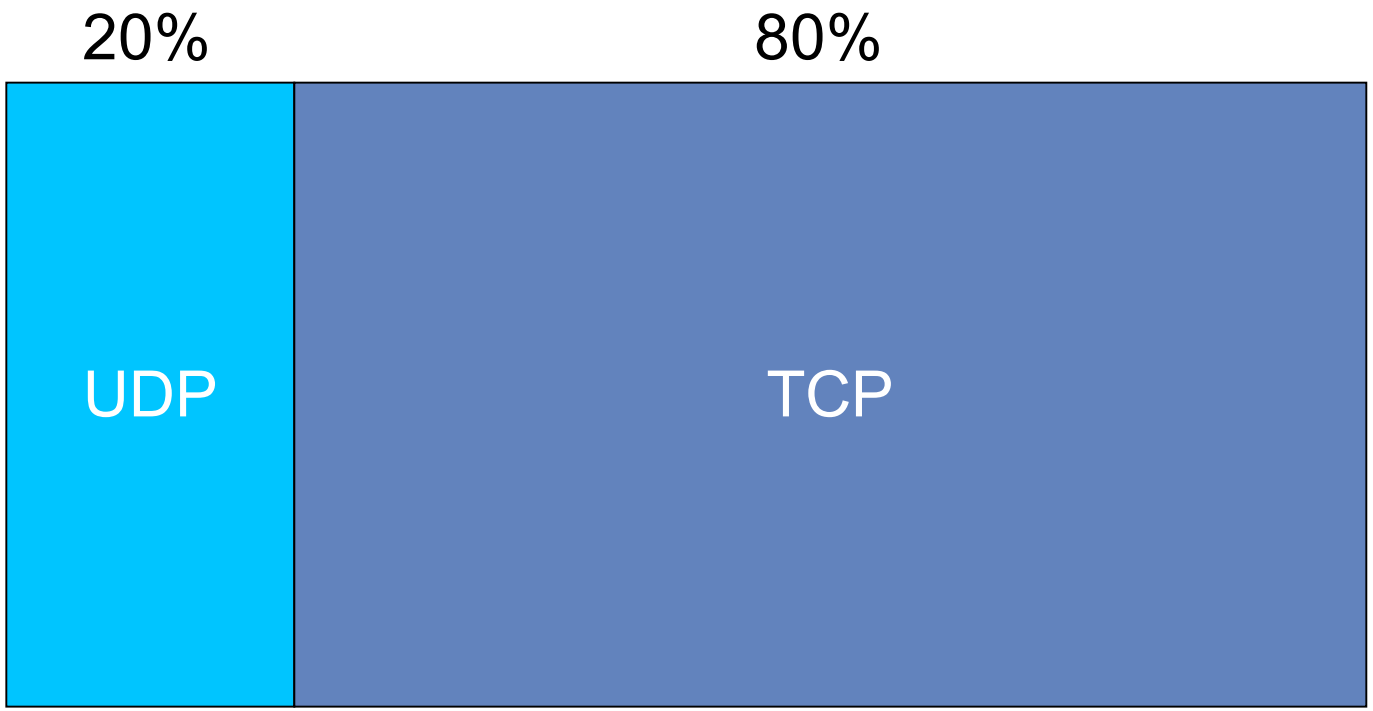
Tree Maps





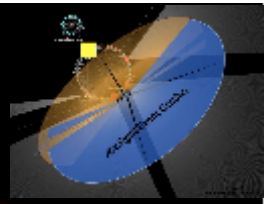
Tree Maps

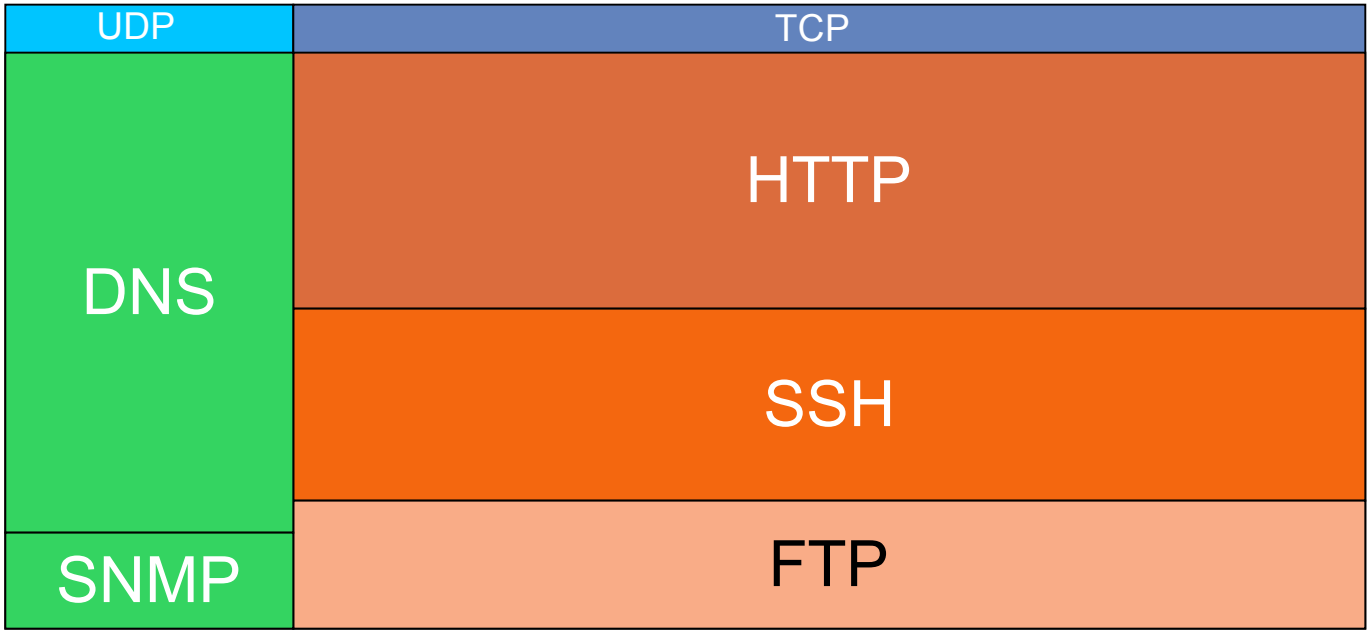




Configuration (Hierarchy): Protocol

Tree Maps

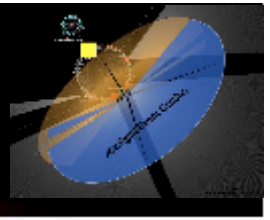




Configuration (Hierarchy): Protocol -> Service


AfterGlow
afterglow.sourceforge.net

AfterGlow

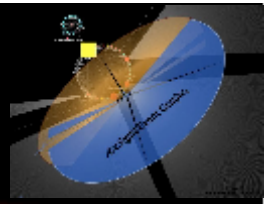


http://afterglow.sourceforge.net

- Two Versions:
  - AfterGlow 1.x – Perl for **Link Graphs**
  - AfterGlow 2.0 – Java for **TreeMaps**

- Collection of Parsers:
  - pf2csv.pl BSD PacketFilter (pf)
  - tcpdump2csv.pl tcpdump 3.9
  - sendmail2csv.pl Sendmail transaction logs

AfterGlow
Parsers



- tcpdump2csv.pl
  - Takes care of swapping response source and targets

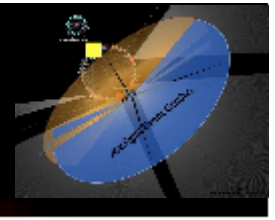
```
tcpdump -vttttnnelr /tmp/log.tcpdump |
./tcpdump2csv.pl "sip dip sport"
```

- sendmail_parser.pl
  - Reassemble email conversations:

```
Jul 24 21:01:16 rmarty sendmail[17072]: j6P41Gqt017072:
from=<root@localhost.localdomain>, size=650, class=0, nrcpts=1,
Jul 24 21:01:16 rmarty sendmail[17073]: j6P41Gqt017072: to=ram,
ctladdr=<root@localhost.localdomain> (0/0), delay=00:00:00,
xdelay=00:00:00, mailer=local, pri=30881, dsn=2.0.0, stat=Sent
```

- pf2csv.pl
  - Parsing OpenBSD pf output

AfterGlow 1.x - Perl



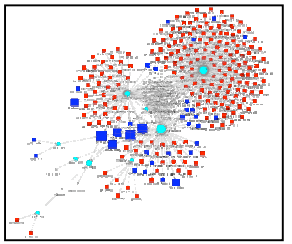
Parser

CSV File

AfterGlow

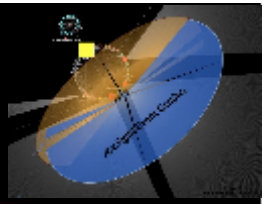
Graph LanguageFile

Grapher



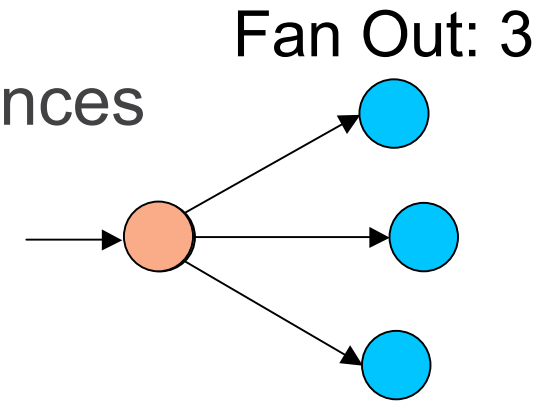
- Supported graphing tools:
  - GraphViz from AT&T (dot, neato, circo, twopi) http://www.graphviz.org
  - LGL (Large Graph Layout) by Alex Adai http://bioinformatics.icmb.utexas.edu/lgl/

Features

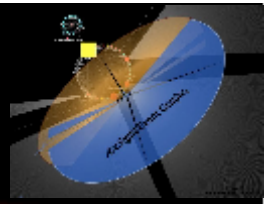


- Generate Link Graphs

- Filtering Nodes
  - Based on name
  - Based on number of occurrences
- Fan Out Filtering
- Coloring
  - Edges
  - Nodes
- Clustering



Command Line Parameters



- Some command line arguments:

  -h : help

  -t : two node mode

  -d : print count on nodes

  -e : edge length

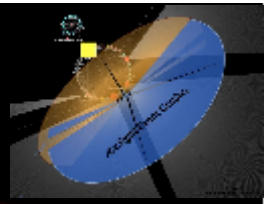
  -n : no node labels

  -o *threshold* : omit threshold (fan-out for nodes to be displayed)

  -f *threshold* : fan out threshold for source node

  -c *configfile* : color configuration file

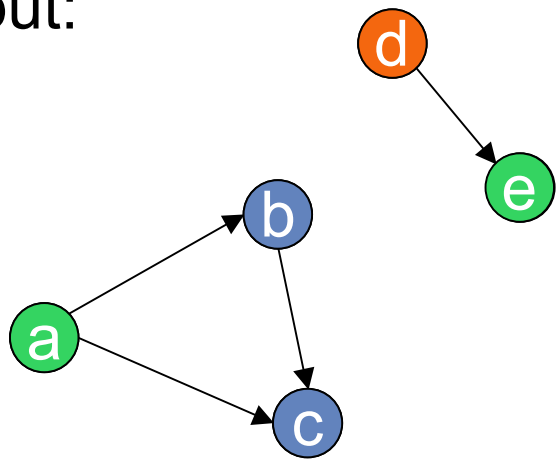
Hello World



Input Data:

```
a,b
a,c
b,c
d,e
```

Output:



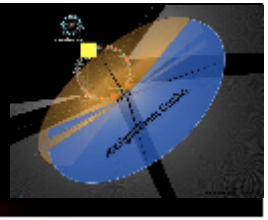
Command:

```
cat file | ./afterglow -c simple.properties -t \
neato -Tgif -o test.gif
```

simple.properties:

```
color.source="green" if ($fields[0] ne "d")
color.target="blue" if ($fields[1] ne "e")
color.source="red"
color="green"
```

Property File – Color Definition



- Coloring:

```
color.[source|event|target|edge]=
    <perl expression returning a color name>
```

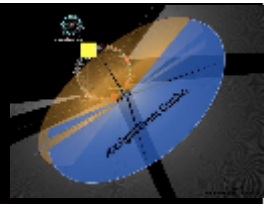
- Array `@fields` contains input-line, split into tokens:

```
color.event="red" if ($fields[1] =~ /^192\..*)
```

- Filter nodes with "`invisible`" color:

```
color.target="invisible" if ($fields[0] eq
 "IIS Action")
```

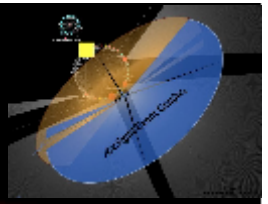
Property File - Clustering



- Clustering:

```
cluster.[source|event|target]=
    <perl expression returning a cluster name>
```

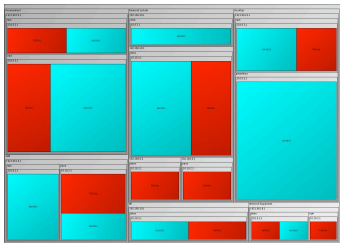
AfterGlow 2.0 - Java



Parser

CSV File

AfterGlow - Java



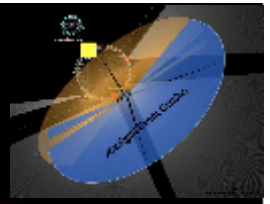
- Command line arguments:

  -h : help

  -c file : property file

  -f file : data file

AfterGlow 2.0
Example



- Data:

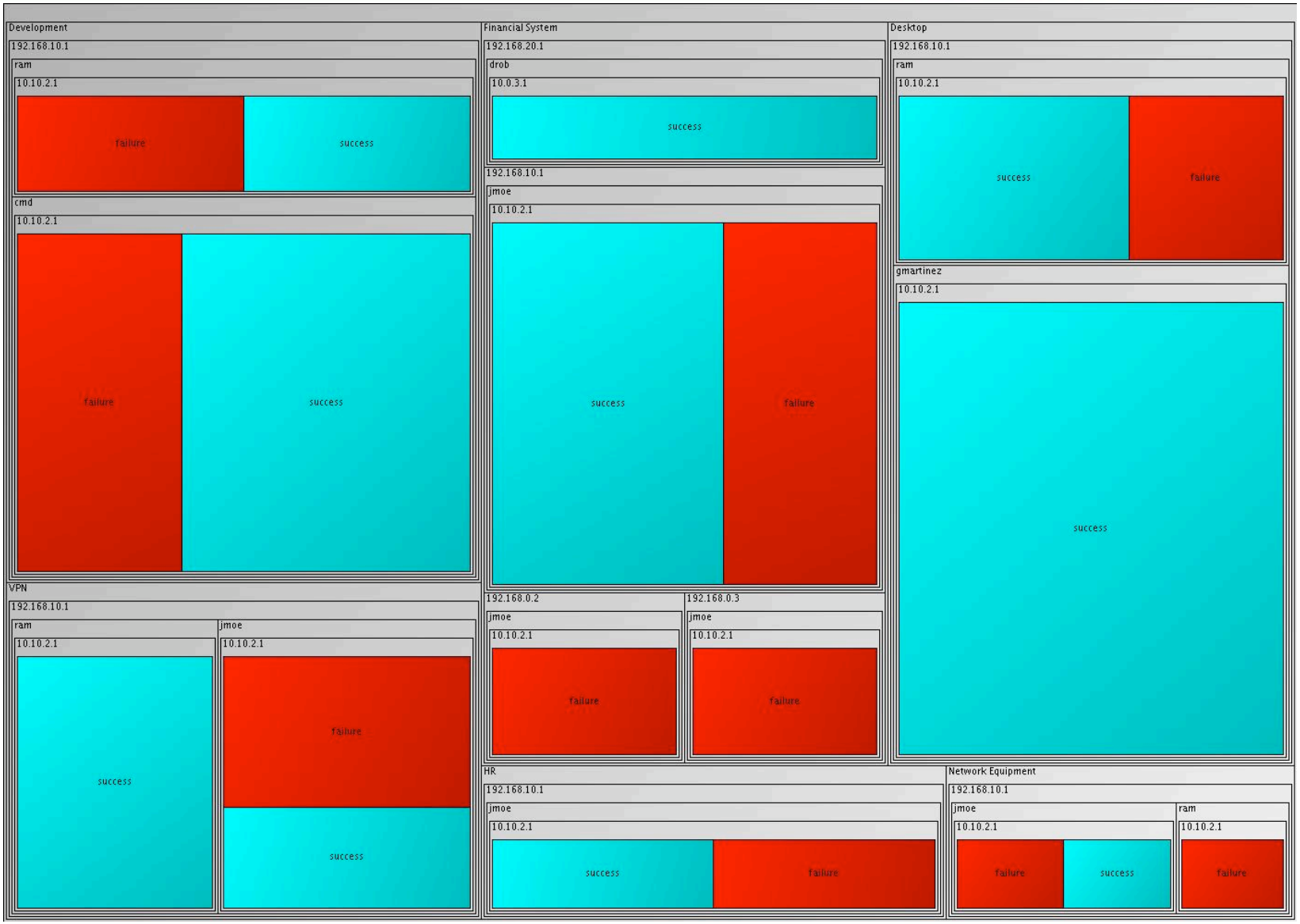
```
Target Syste
Development,
VPN,192.168
Financial Sy
VPN,192.168
VPN,192.168
Financial Sy
Financial Sy
```

- Launch:

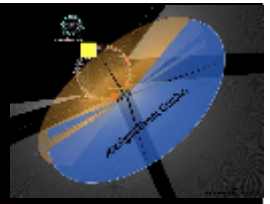
./afterglo

```
# AfterGlow - JAVA 2.0
# Properties File

# File to load
file.name=/home/ram/afterglow/data/sample.csv

# Column Types (default is STRING), start with 0!
#   Valid values:
#     STRING
#     INTEGER
#     CATEGORICAL

column.type.count=4
column.type[0].column=0
column.type[0].type=INTEGER
column.type[1].column=1
column.type[1].type=CATEGORICAL
column.type[2].column=2
column.type[2].type=CATEGORICAL
column.type[3].column=3
column.type[3].type=CATEGORICAL

# Size Column (default is 0)
size.column=0

# Color Column (default is 0)
color.column=2
```

AfterGlow 2.0
Output

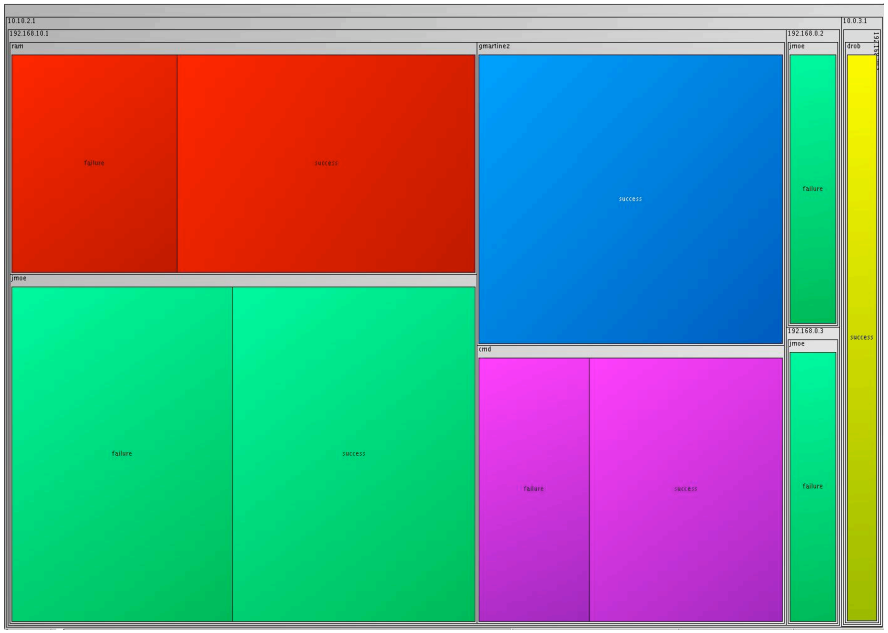


AfterGlow 2.0
Interaction



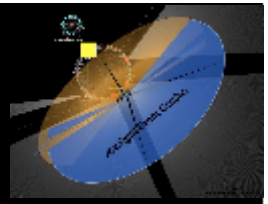
- Left-click:
  - Zoom in
- Right-click:
  - Zoom all the way out
- Middle-click
  - Change Coloring to current depth

    (Hack: Use SHIFT for leafs)



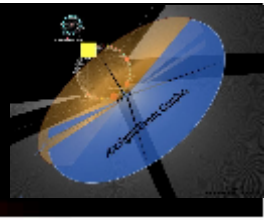
Firewall Log File Analysis

Firewall Log File Analysis
Overview



1. Parse Firewall Log
2. Investigate allowed incoming traffic
   - Do you know what you are dealing with?
3. Investigate allowed outgoing traffic
   - What is leaving the network?
4. Investigate blocked outgoing traffic
   - Mis-configured or compromised internal machines OR ACL problem
5. Investigate blocked incoming traffic
   - What is trying to attack me?

Parsing PF Firewall Log



Input (pflog):

```
Feb 18 13:39:15.598491 rule 71/0(match): pass in on xl0: 195.27.249.139.63263 >
195.141.69.42.80: S 492525755:492525755(0) win 32768 <mss 1460,nop,wscale
0,nop,nop,timestamp 24053 0> (DF)
Feb 18 13:39:15.899644 rule 71/0(match): pass in on xl0: 195.27.249.139.63264 >
195.141.69.42.80: S 875844783:875844783(0) win 32768 <mss 1460,nop,wscale
0,nop,nop,timestamp 24054 0> (DF)
```

Command:

```
cat pflog | pf2csv.pl "sip dip dport"
```

Output:

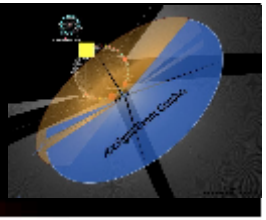
```
195.27.249.139,195.141.69.42,80
195.27.249.139,195.141.69.42,80
```

AfterGlow Input

Visualization:

```
cat pflog | pf2csv.pl "sip dip dport" | \
afterglow -c properties | neato -Tgif -o foo.gif
```

Passed Incoming Traffic



Command:

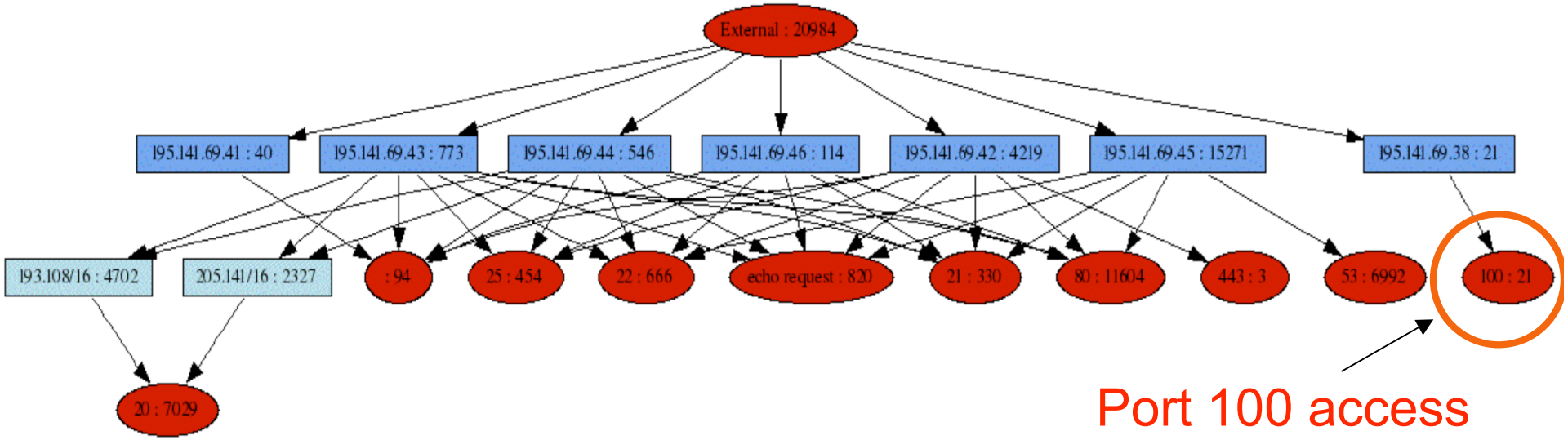
```
cat log | grep pass_in | ./afterglow -c properties -d | dot -Tgif -o foo.gif
```

Properties:

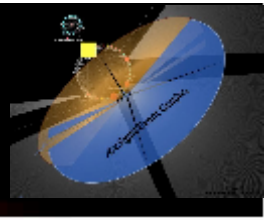
```
cluster.source="External" if (!match("^195\.141\.69"))
color="red" if (field() eq "External")
color.event="blue" if (regex("^195\.141\.69"))
color.event="lightblue"
color="red"
```

Features/Functions:

```
field()
cluster
match()
```

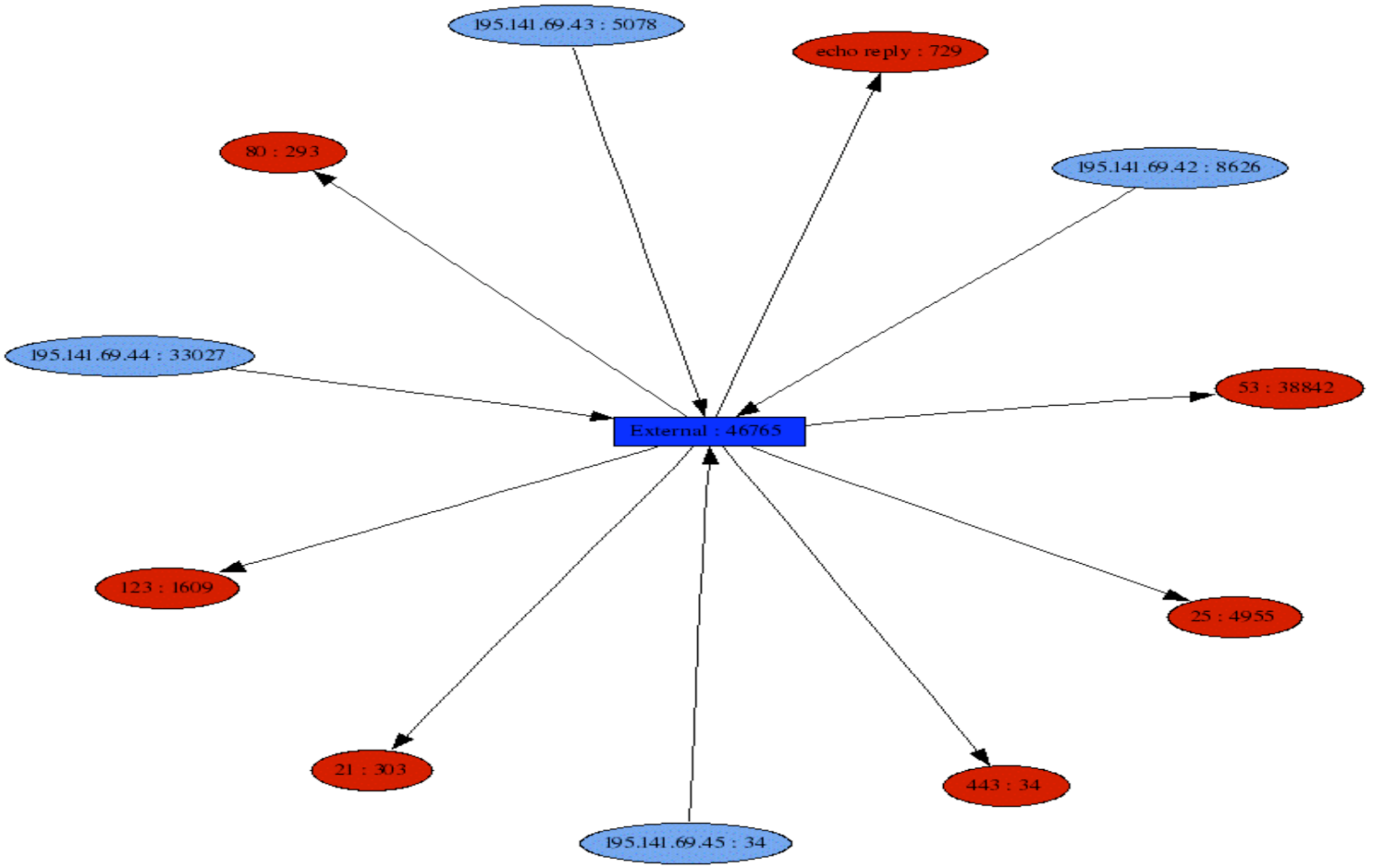


Passed Outgoing Traffic

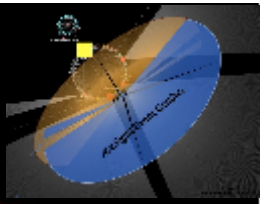


Command:

```
cat log | grep pass_out | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```

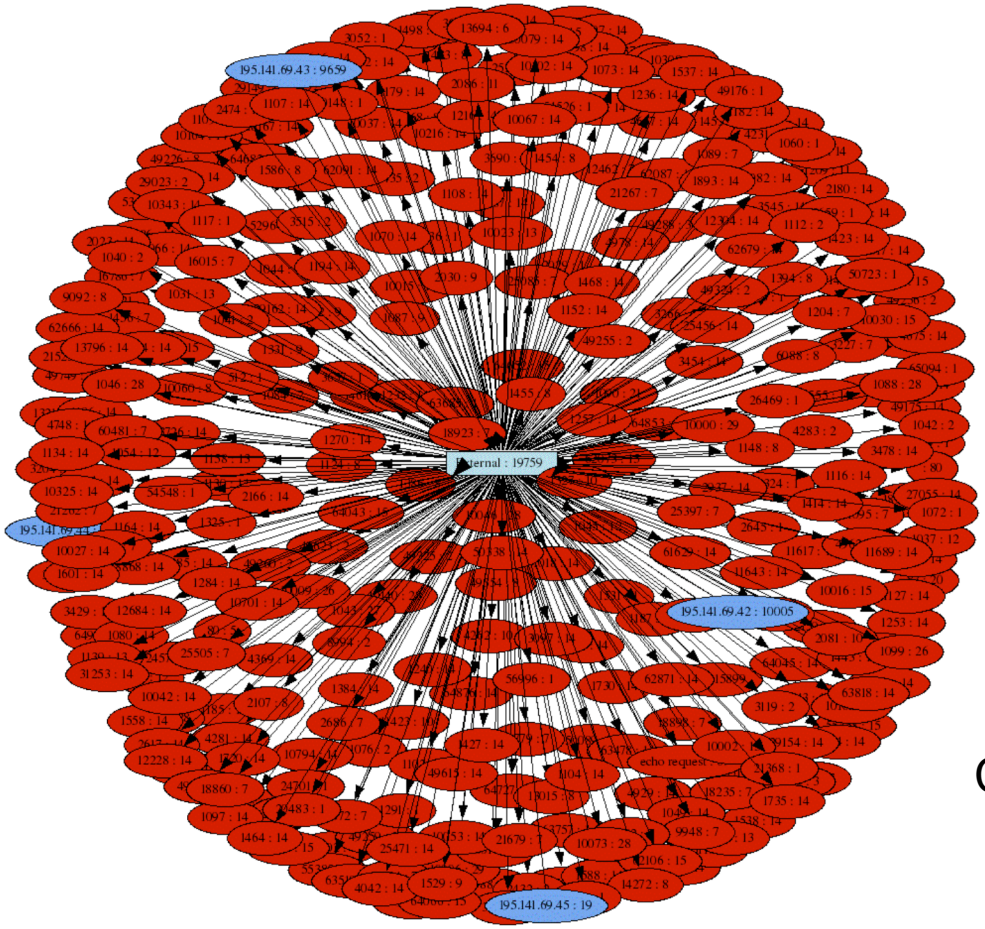


Firewall Log File Analysis
Blocked Outgoing Traffic



Command:

```
cat log | grep block_out | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```



What happened?

Rule-set logs on response ☹

block in on xl1

Firewall

request

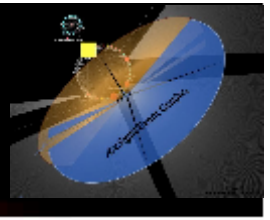
response

xl0 xl1

Client

Server

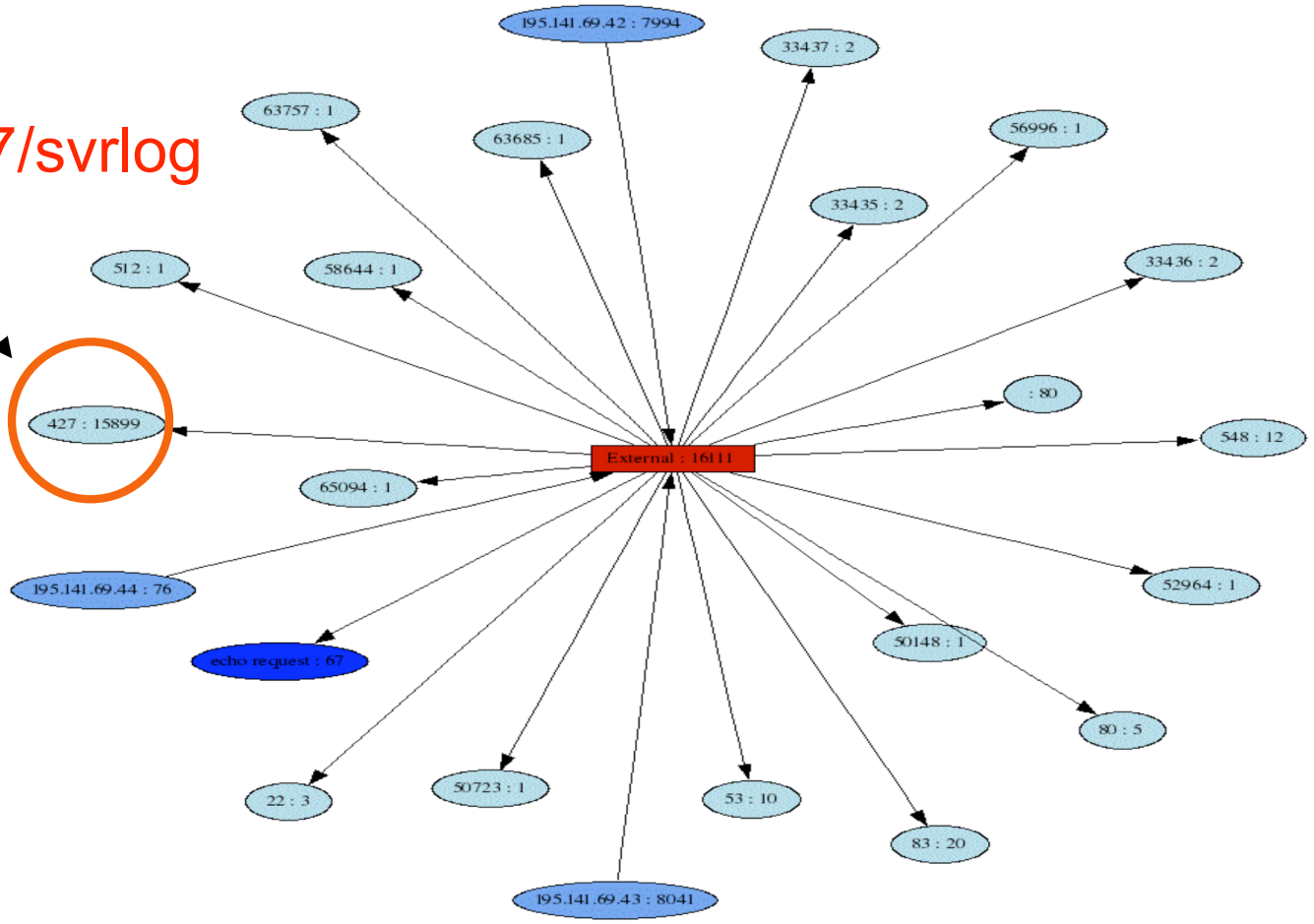
Blocked Outgoing Traffic – 2nd Attempt



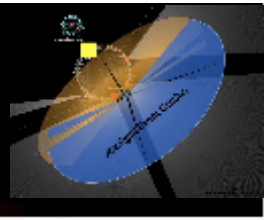
```
cat log | pf2csv.pl "sip dip dport reversed" | grep -v "R$"
```

Uses heuristics to filter responses out

Port 427/svrlog



Blocked Incoming Traffic

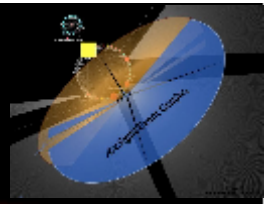


Command:

```
cat log | grep block_in | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```

You guessed right:
WAY TOO MESSY!

Blocked Incoming Port-Scans

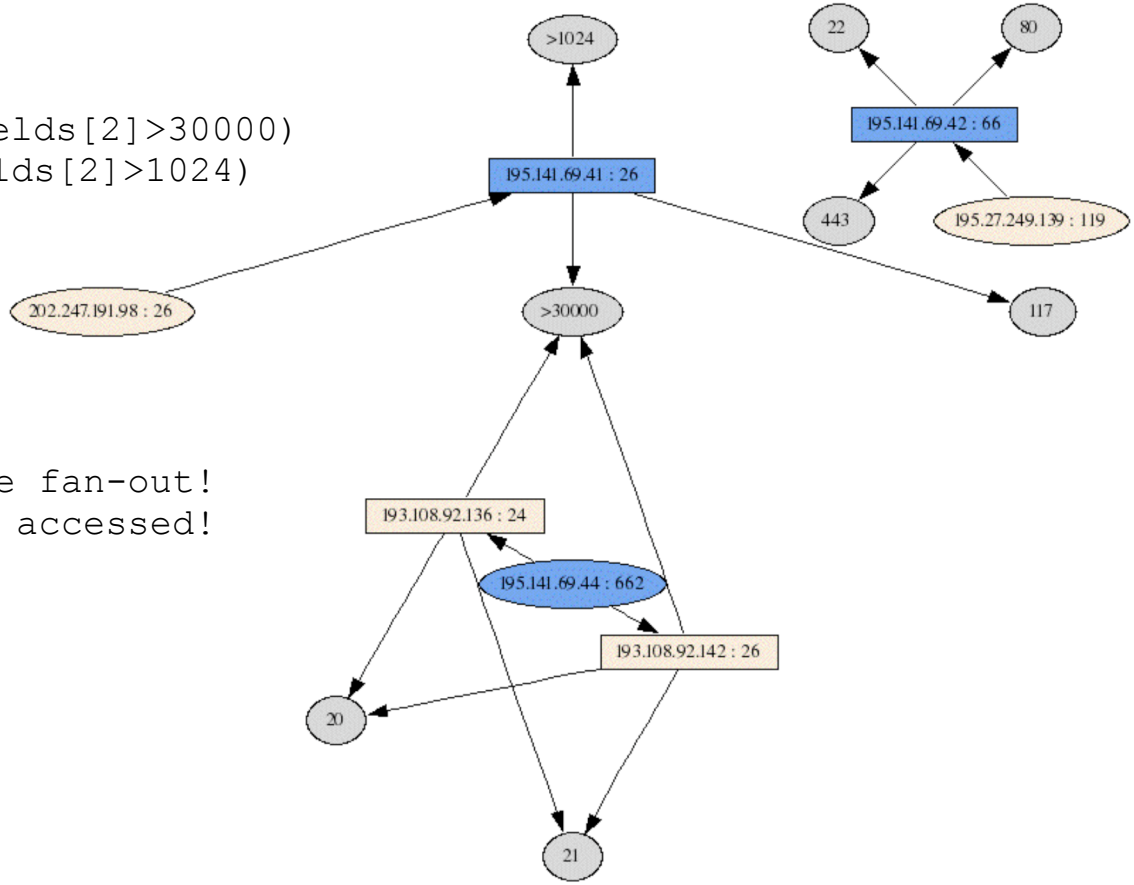


Command:

```
cat log |grep block_in |./afterglow -c properties -d -g 2 | neato -Tgif -o foo.gif
```

Properties:

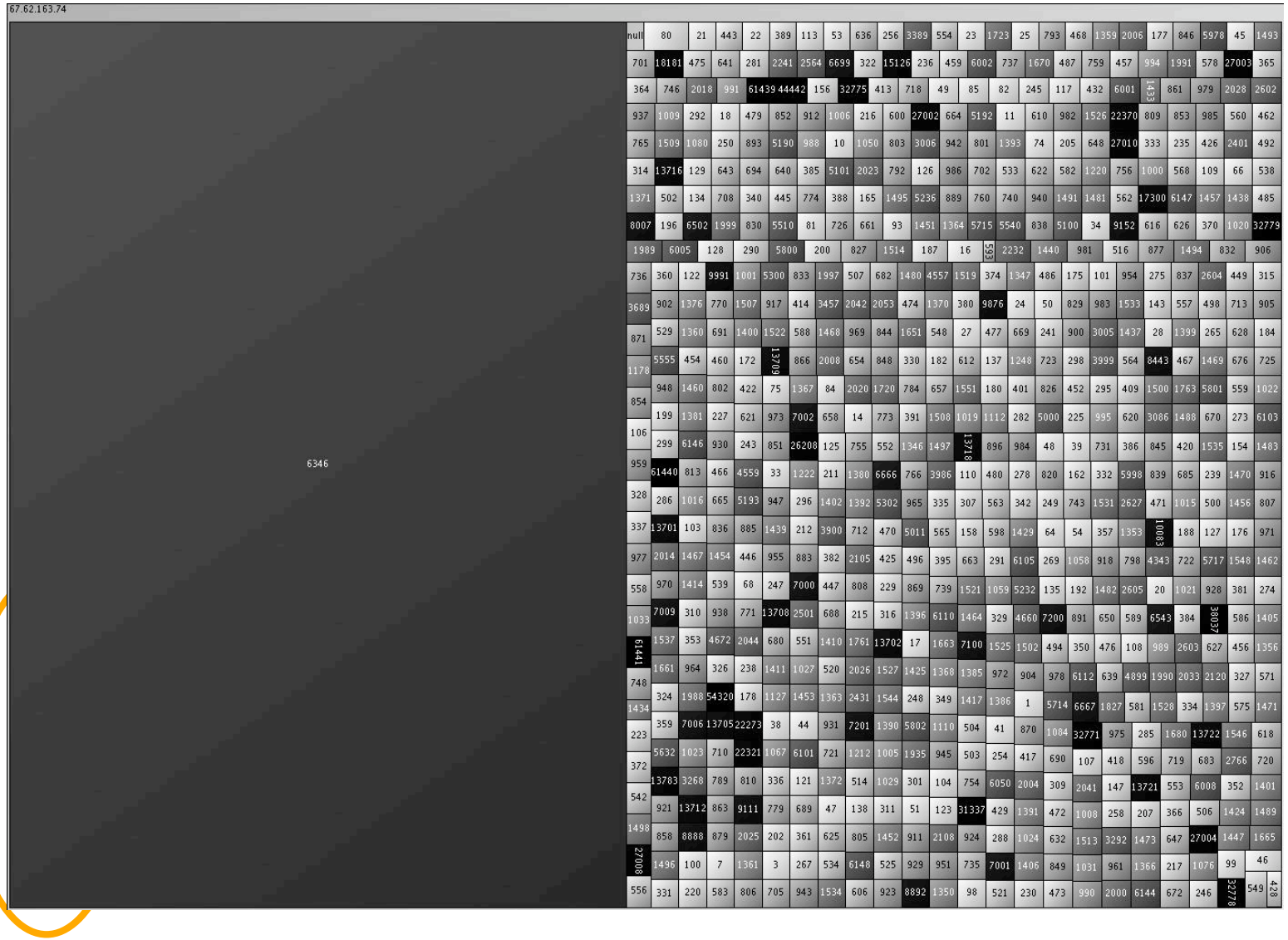
```
cluster.target="">30000" if ($fields[2]>30000)
cluster.target="">1024" if ($fields[2]>1024)
color= . . .
```



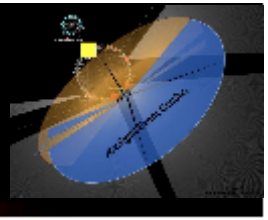
Feature:

```
-g 2 : Filter based on event-node fan-out!
       i.e., more than two ports accessed!
```

Firewall Log File Analysis
Blocked Incoming Port-Scans



Blocked Incoming Bogon Addresses

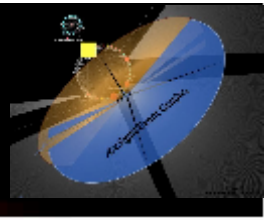


Command:

```
cat log | grep block_in |./afterglow -c properties -d | neato -Tgif -o foo.gif
```

This is going to be crazy!

Firewall Log File Analysis
Blocked Incoming Bogon Addresses



Command:

```
cat log |grep block_in |./afterglow -c properties -d | neato -Tgif -o foo.gif
```

Properties:

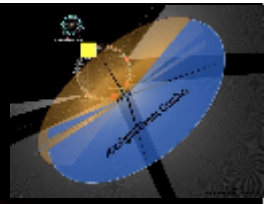
Bogon Address Space

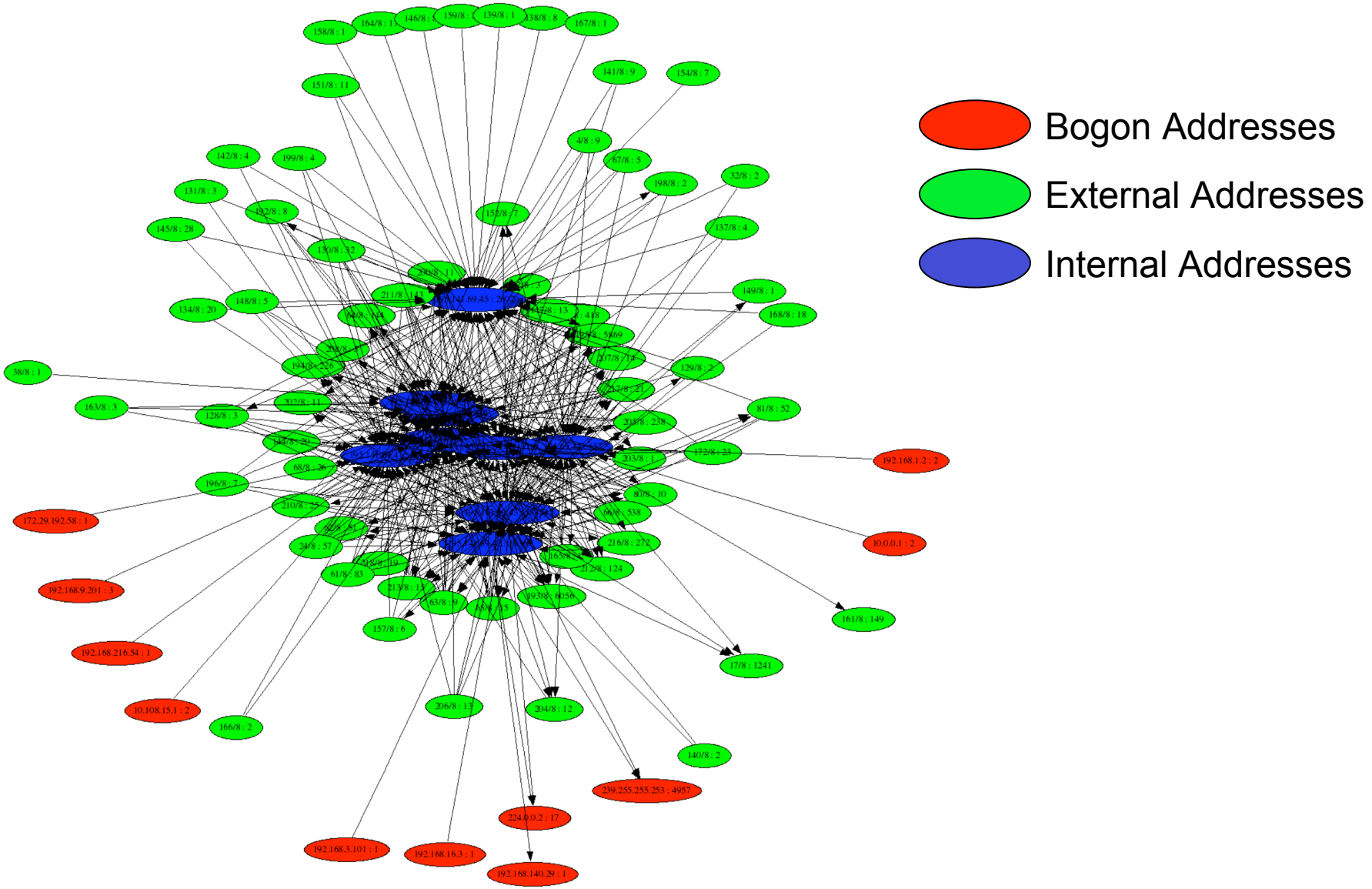
```
variable=@ranges=qw{0.0.0.0/7 2.0.0.0/8 5.0.0.0/8 7.0.0.0/8 10.0.0.0/8 23.0.0.0/8 27.0.0.0/8
31.0.0.0/8 36.0.0.0/7 39.0.0.0/8 42.0.0.0/8 49.0.0.0/8 50.0.0.0/8 77.0.0.0/8 78.0.0.0/7 92.0.0.0/6 96.0.0.0/4
112.0.0.0/5 120.0.0.0/8 127.0.0.0/8 169.254.0.0/16 172.16.0.0/12 173.0.0.0/8 174.0.0.0/7 176.0.0.0/5 184.0.0.0/6
192.0.2.0/24 192.168.0.0/16 197.0.0.0/8 198.18.0.0/15 223.0.0.0/8 224.0.0.0/3};

color="blue" if ($value=0; map{ $value+=subnet(field(),$_) }
  @ranges; regex_replace("(\\d+)")."/8" if
  (!match("^(195\.141\.69)") && !$value);
```

Features:

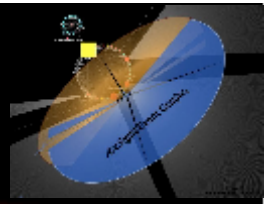
```
variable=
regex_replace()
subnet(IP,range) e.g., subnet("10.0.0.2","10.0.0.0/8") ➔ 1 (true)
```

Firewall Log File Analysis
Blocked Incoming Bogon Addresses





Summary



- Introduced AfterGlow
  - Filtering
  - Coloring
  - Clus
- Quicl
  - Und
  - Find Outliers
  - Spot suspicious activity

Don't **Read** Log Files

Visualize Them!!

THANKS!

raffy@arcsight.com

by ram
ContentTeam@ArcSight, Inc