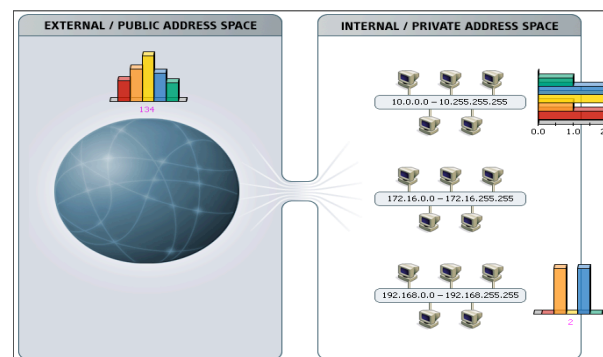
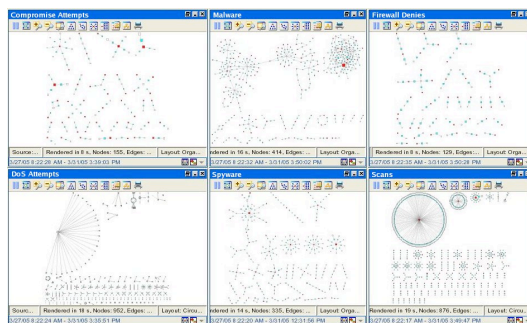
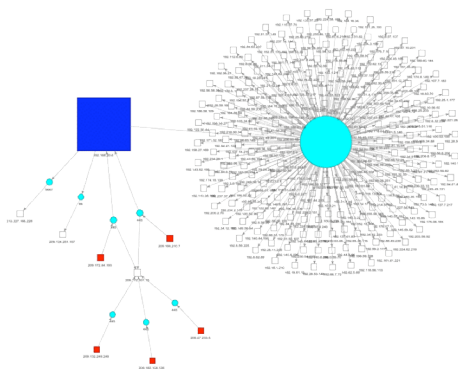


Visual Log Analysis – The Beauty of Graphs

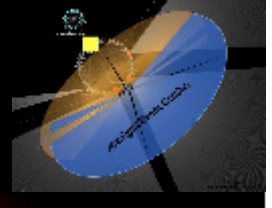
DefCon 2006, Las Vegas



Raffael Marty, GCIA, CISSP
Manager Solutions @ ArcSight

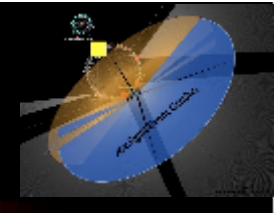
August 5th, 2006

Raffael Marty, GCIA, CISSP

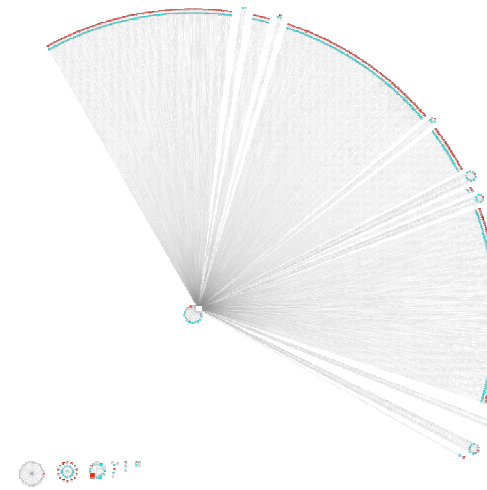
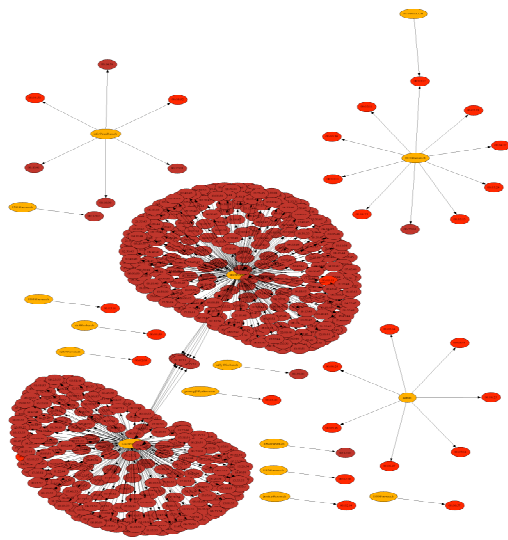
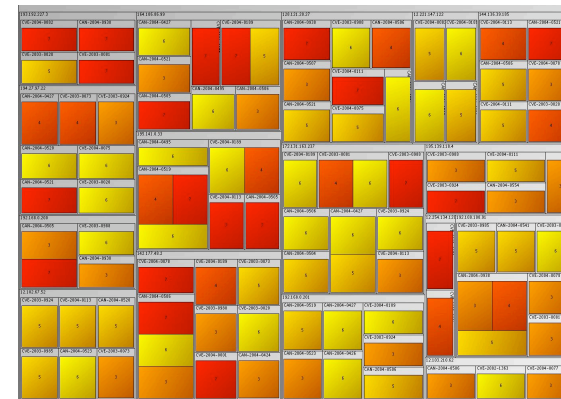


- Enterprise Security Management (ESM) specialist
- Strategic Application Solutions @ ArcSight, Inc.
- Intrusion Detection Research @ IBM Research
 - See <http://thor.cryptojail.net>
- IT Security Consultant @ PriceWaterhouse Coopers
- Open Vulnerability and Assessment Language (OVAL) board member
-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

Table Of Contents



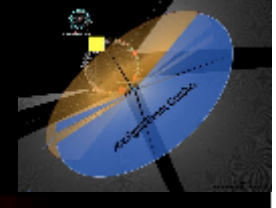
- Introduction
- Graphing Basics
- AfterGlow
- Firewall Log File Analysis





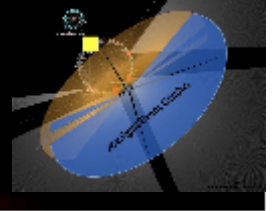
Introduction

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. ”

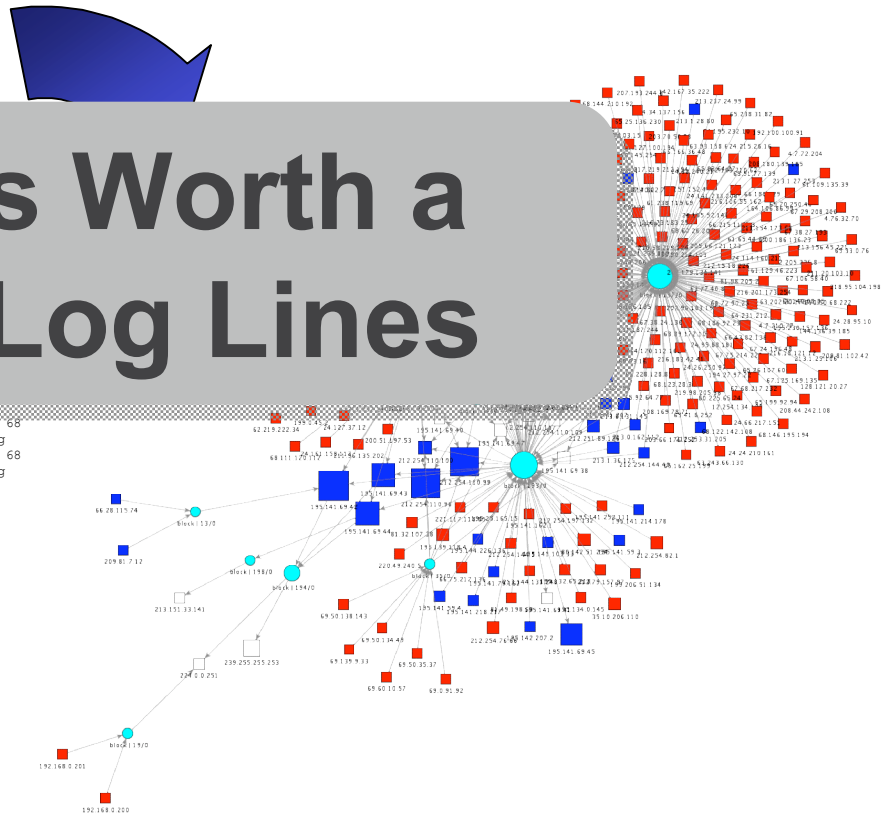
Text or Visuals?



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 09:45:02 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 09:56:03 rmarty vmnet-dhcpd: DHCPDISCOVER from 00:0c:29:b7:b2:47 via vmnet8
Jun 17 10:00:03 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 poi
Jun 17 10:13:05 rmarty poi
Jun 17 10:14:09 rmarty poi
Jun 17 10:14:09 rmarty poi
Jun 17 10:14:09 rmarty poi
Jun 17 10:21:30 rmarty poi
Jun 17 10:21:30 rmarty poi
Jun 17 10:28:40 rmarty vnn
Jun 17 10:28:41 rmarty vnn
Jun 17 10:28:45 rmarty vnn
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
```

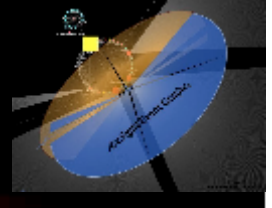
A Picture is Worth a Thousand Log Lines





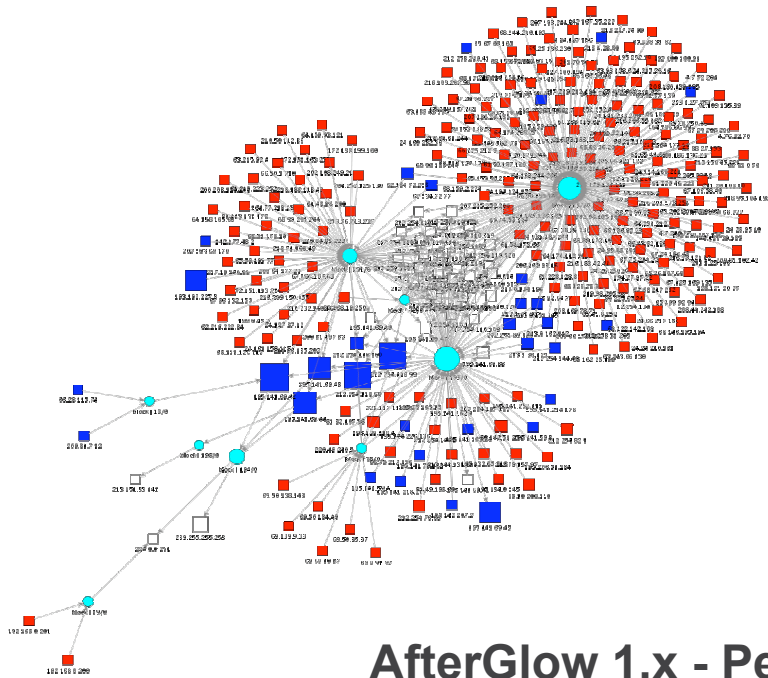
Graphing Basics

Visual Types



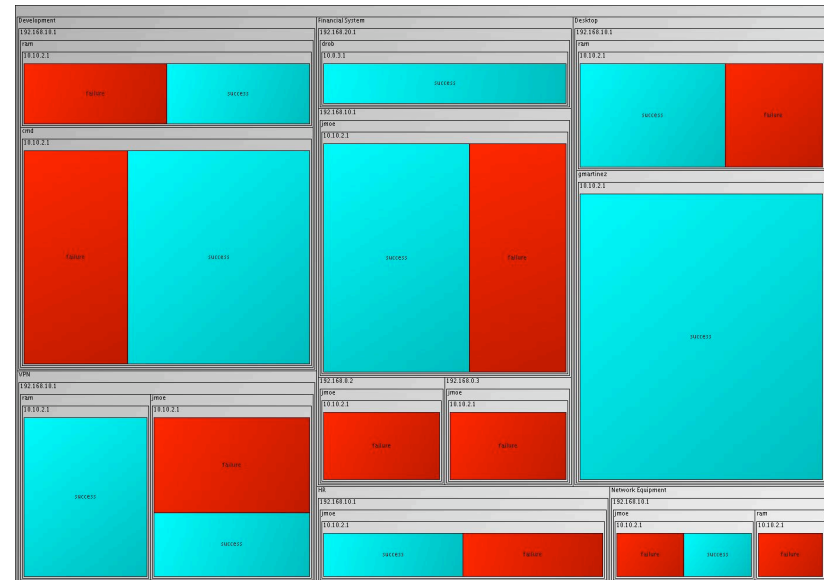
— Visuals that AfterGlow supports:

Link Graphs



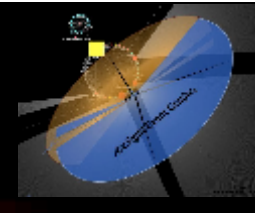
AfterGlow 1.x - Perl

TreeMaps



AfterGlow 2.0 - JAVA

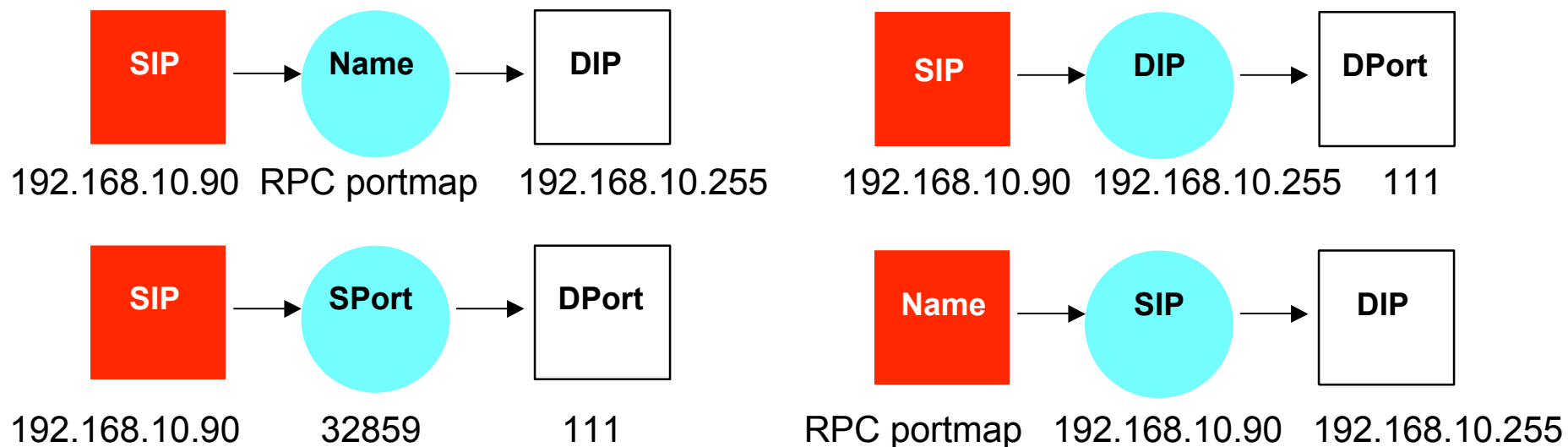
Link Graph Configurations



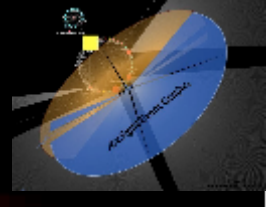
Raw Event:

```
[**] [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:

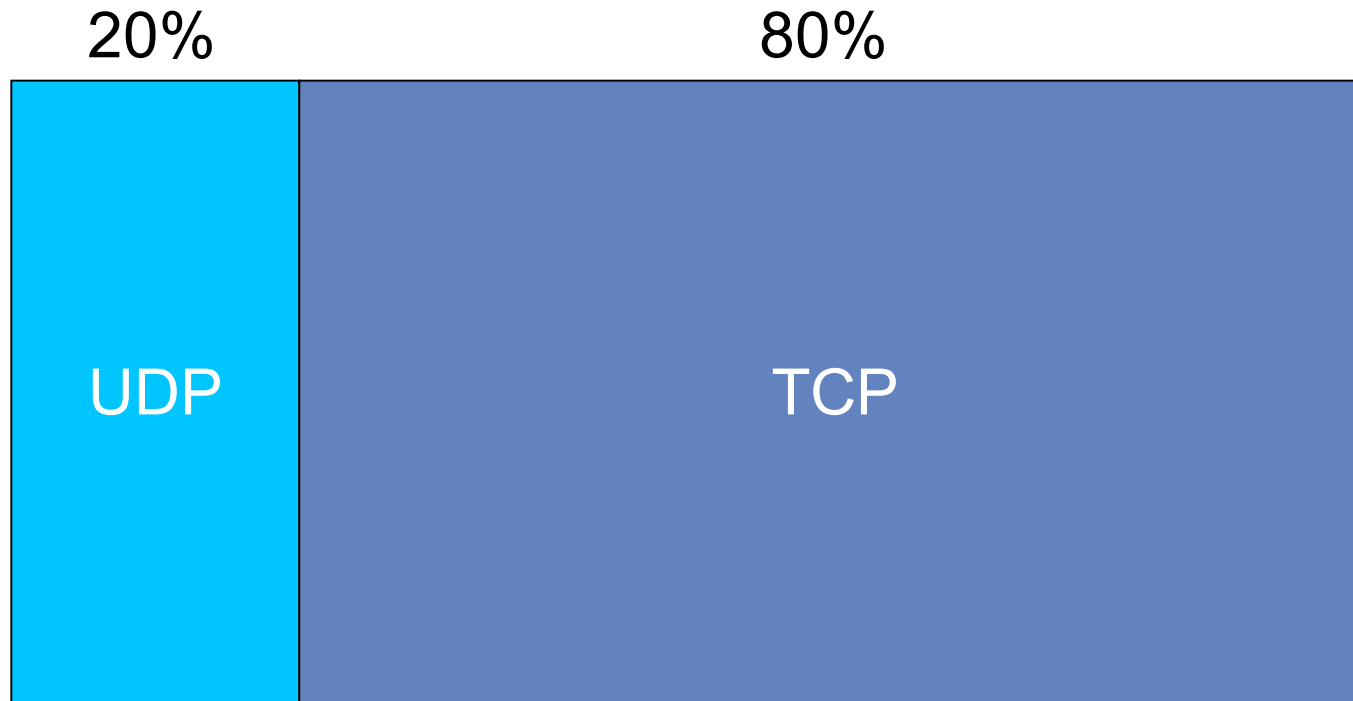
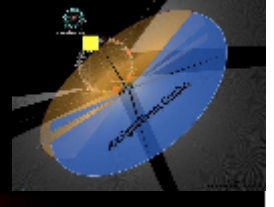


Tree Maps



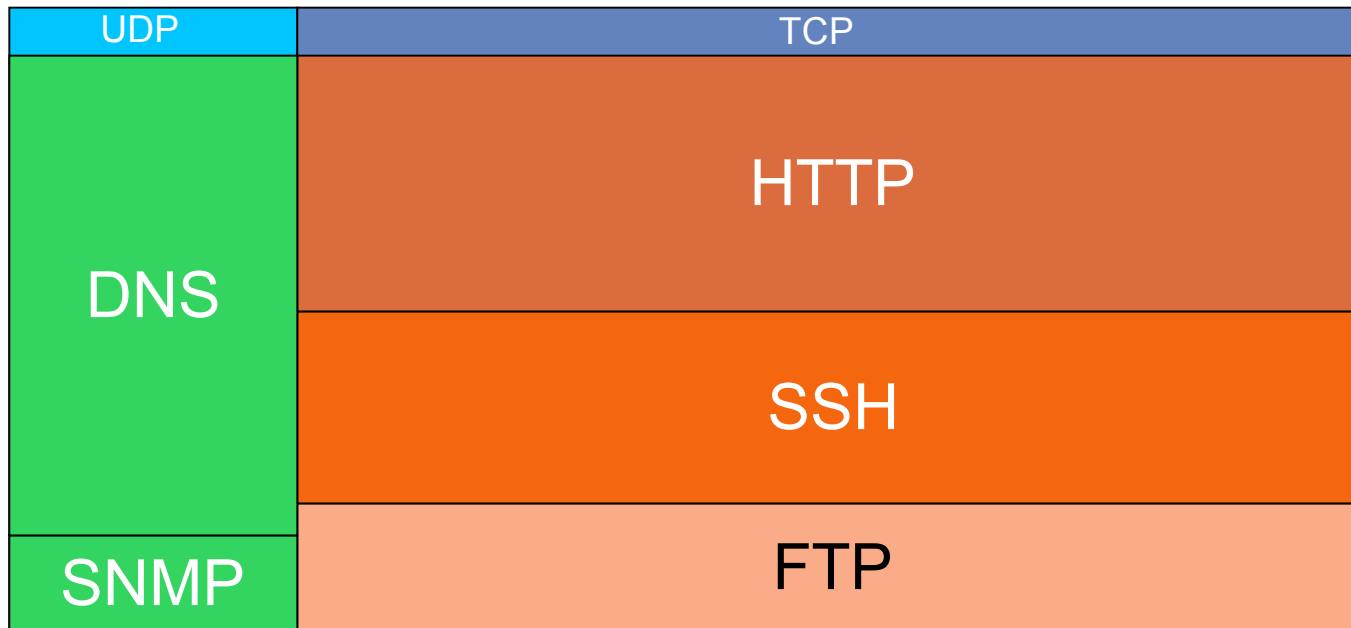
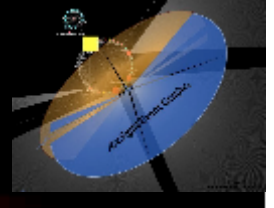
All Network Traffic

Tree Maps



Configuration (Hierarchy): Protocol

Tree Maps

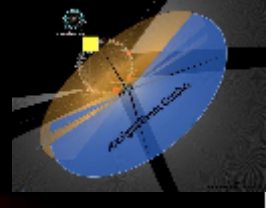


Configuration (Hierarchy): Protocol -> Service



AfterGlow

afterglow.sourceforge.net



<http://afterglow.sourceforge.net>

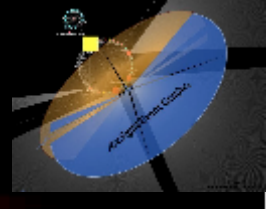
Two Versions:

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

– sendmail_parser.pl

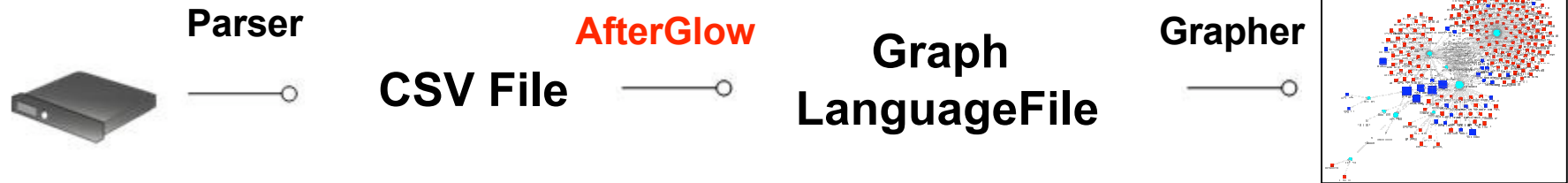
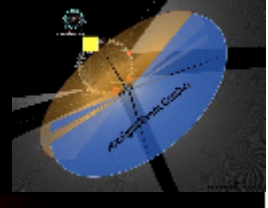
- Reassemble email conversations:

```
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

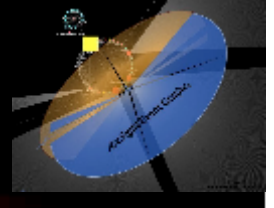
AfterGlow 1.x - Perl



_ 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/>

AfterGlow 1.x Features



— Generate Link Graphs

— Filtering Nodes

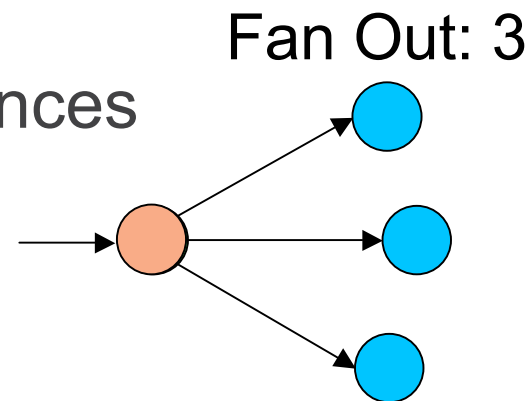
- Based on name
- Based on number of occurrences

— Fan Out Filtering

— Coloring

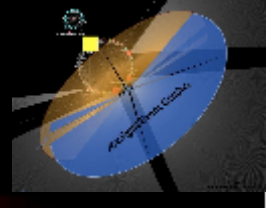
- Edges
- Nodes

— Clustering



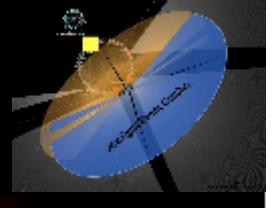
AfterGlow 1.x

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

AfterGlow 1.x Hello World



Input Data:

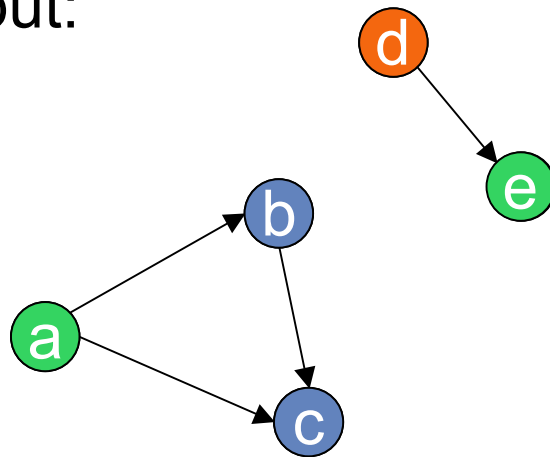
a, b

a, c

b, c

d, e

Output:



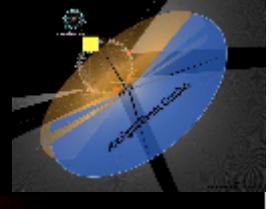
Command:

```
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"
```

AfterGlow 1.x Property File – Color Definition



- **Coloring:**

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

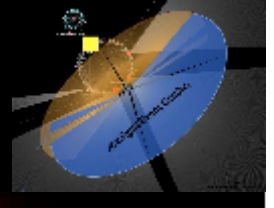
- **Array @fields contains input-line, split into tokens:**

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

- **Filter nodes with "invisible" color:**

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

AfterGlow 1.x Property File - Clustering

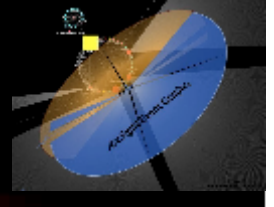


- Clustering:

```
cluster.[source|event|target]=
```

```
<perl expression returning a cluster name>
```

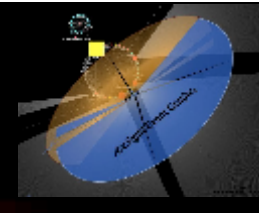
AfterGlow 2.0 - Java



_ Command line arguments:

- h : help
- c file : property file
- f file : data file

AfterGlow 2.0 Example



Data:

Target System

Development

VPN,192.168

Financial S

VPN,192.168

VPN,192.168

Financial S

Financial S

```
# 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
```

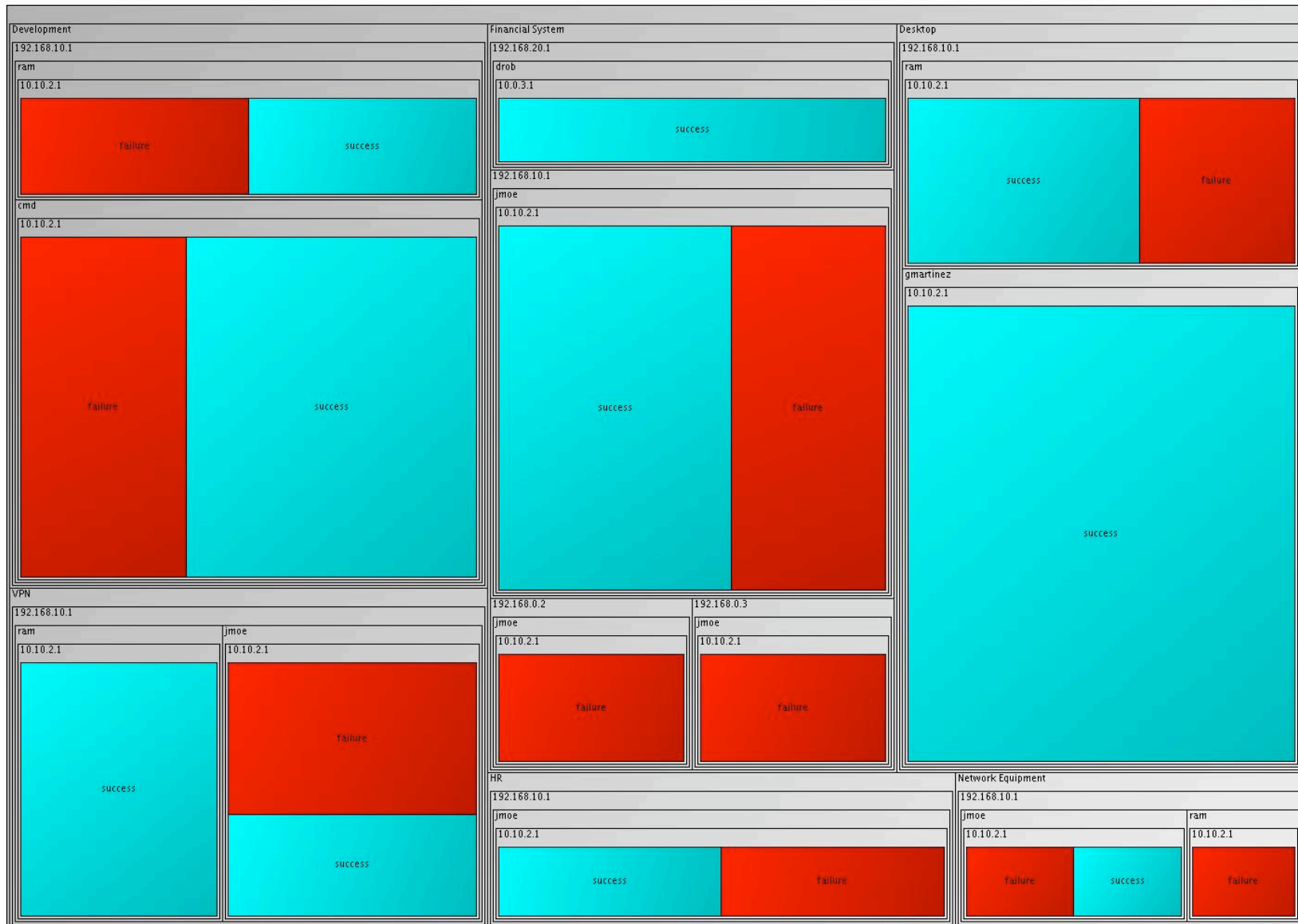
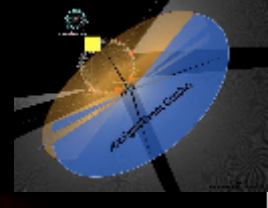
```
# Size Column (default is 0)
size.column=0
```

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

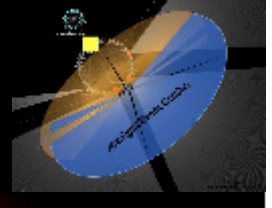
Launch:

`./afterglow`

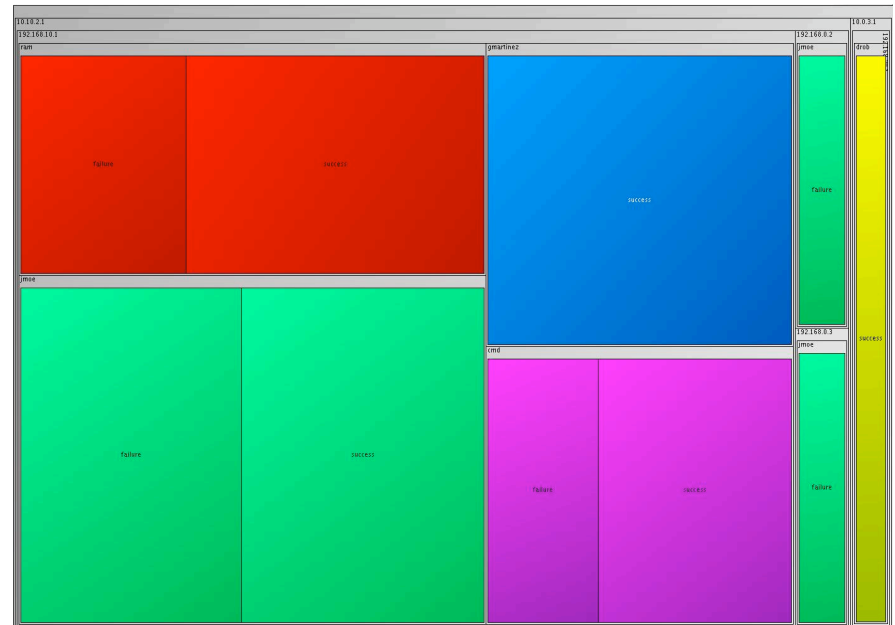
AfterGlow 2.0 Output



AfterGlow 2.0 Interaction



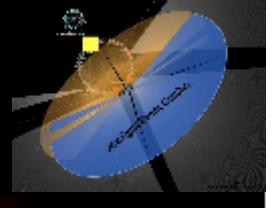
- Left-click:
 - Zoom in
- Right-click:
 - Zoom all the way out
- Middle-click
 - Change Coloring to current depth
(Hack: Use SHIFT for leafs)





Firewall Log File Analysis

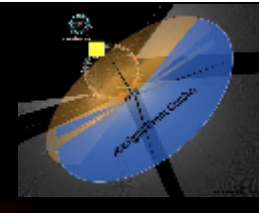
Firewall Log File Analysis Overview



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?

Firewall Log File Analysis

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:

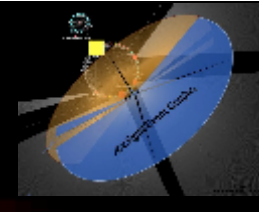
```
195.27.249.139,195.141.69.42,80
195.27.249.139,195.141.69.42,80
```

← AfterGlow Input

Visualization:

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

Firewall Log File Analysis Passed Incoming Traffic



Command:

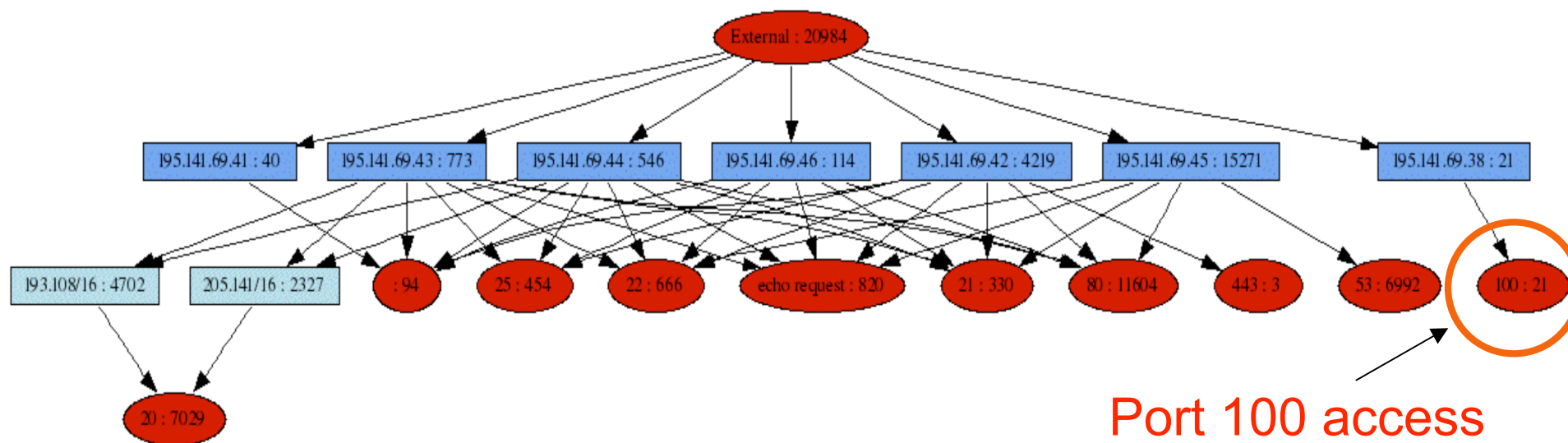
```
cat log | grep pass_in | ./afterglow -c properties -d | dot -Tgif -o foo.gif
```

Properties:

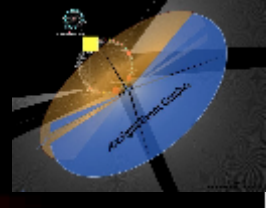
```
cluster.source="External" if (!match("^195\.141\.69"))  
color="red" if (field() eq "External")  
color.event="blue" if (regex("^195\.141\.69"))  
color.event="lightblue"  
color="red"
```

Features/Functions:

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

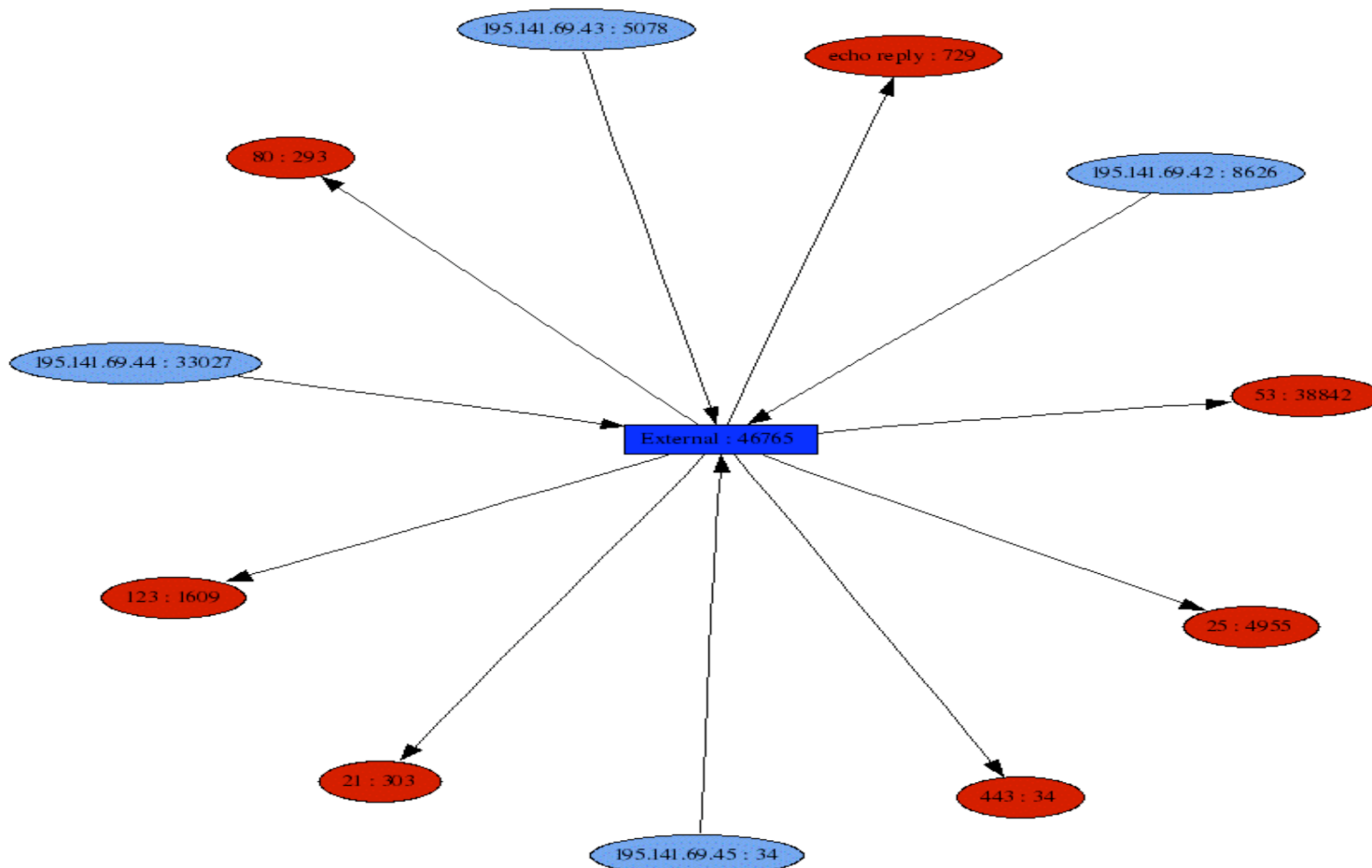


Firewall Log File Analysis Passed Outgoing Traffic

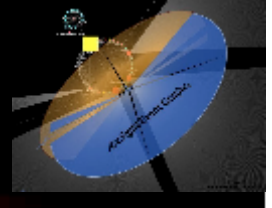


Command:

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

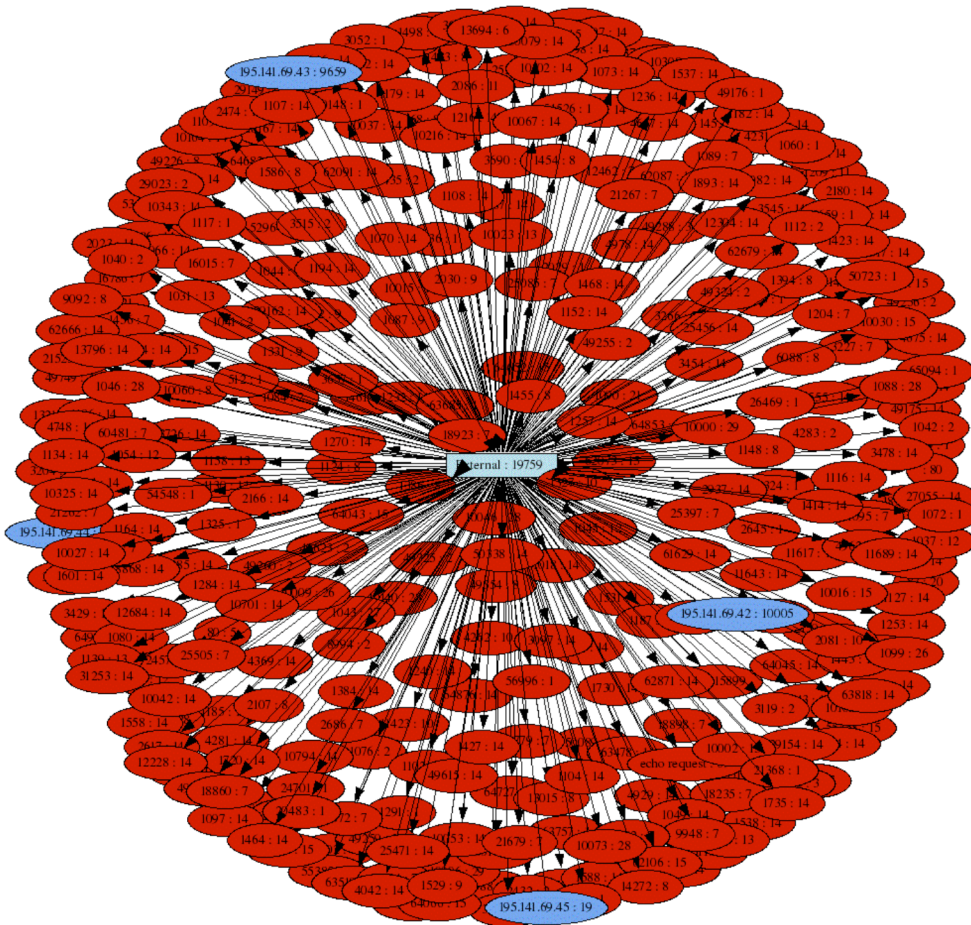


Firewall Log File Analysis Blocked Outgoing Traffic



Command:

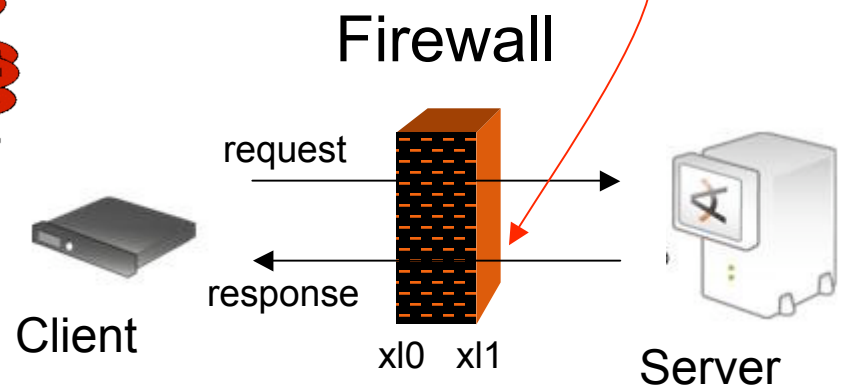
```
cat log | grep block_out | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```



What happened?

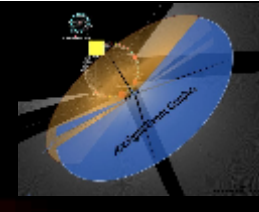
Rule-set logs on response ☹️

block in on xl1



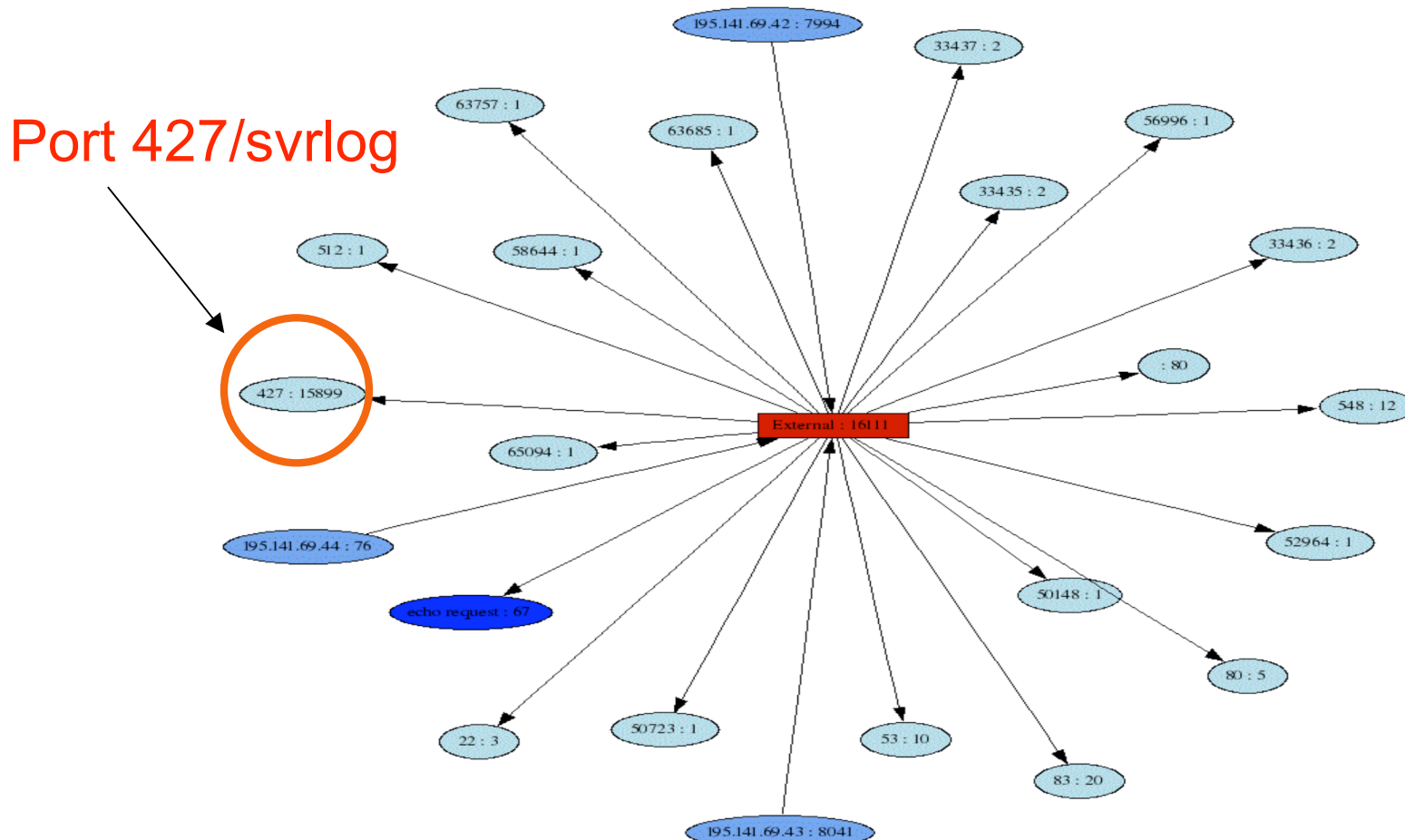
Firewall Log File Analysis

Blocked Outgoing Traffic – 2nd Attempt



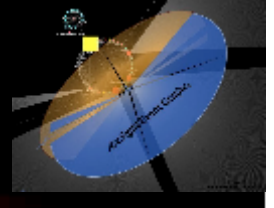
```
cat log | pf2csv.pl "sip dip dport reversed" | grep -v "R$"
```

Uses heuristics to filter responses out



Firewall Log File Analysis

Blocked Incoming Traffic



Command:

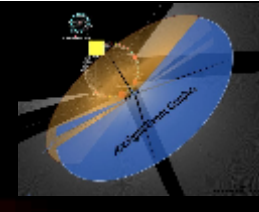
```
cat log | grep block_in | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```

You guessed right:
WAY TOO MESSY!



Firewall Log File Analysis

Blocked Incoming Port-Scans



Command:

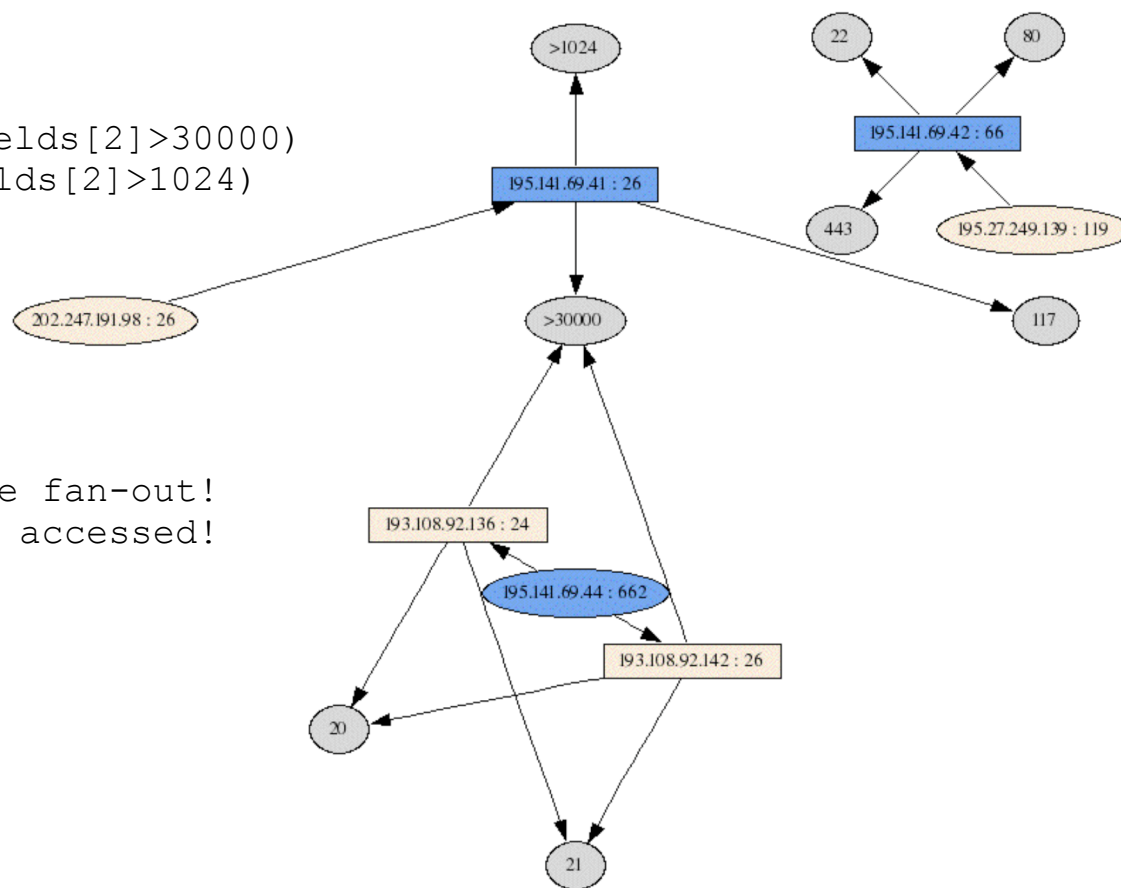
```
cat log |grep block_in |./afterglow -c properties -d -g 2 | neato -Tgif -o foo.gif
```

Properties:

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

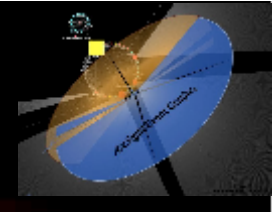
Feature:

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



Firewall Log File Analysis

Blocked Incoming Port-Scans



67.62.163.74

	null	80	21	443	22	389	113	53	636	256	3389	554	23	1723	25	793	468	1359	2006	177	846	5978	45	1493
701	18181	475	641	281	2241	2564	6699	322	15126	236	459	6002	737	1670	487	759	457	994	1991	578	27003	365		
364	746	2018	991	61439	44442	156	32775	413	718	49	85	82	245	117	432	6001	143	861	979	2028	2602			
937	1009	292	18	479	852	912	1006	216	600	27002	664	5192	11	610	982	1526	22370	809	853	985	560	462		
765	1509	1080	250	893	5190	988	10	1050	803	3006	942	801	1393	74	205	648	27010	333	235	426	2401	492		
314	13716	129	643	694	640	385	5101	2023	792	126	986	702	533	622	582	1220	756	1000	568	109	66	538		
1371	502	134	708	340	445	774	388	165	1495	5236	889	760	740	940	1491	1481	562	17300	6147	1457	1438	485		
8007	196	6502	1999	830	5510	81	726	661	93	1451	1364	5715	5540	838	5100	34	9152	616	626	370	1020	32779		
1989	6005	128	290	5800	200	827	1514	187	16	2232	1440	981	516	877	1494	832	906							
736	360	122	9991	1001	5300	833	1997	507	682	1480	4557	1519	374	1347	486	175	101	954	275	837	2604	449	315	
3689	902	1376	770	1507	917	414	3457	2042	2053	474	1370	380	9876	24	50	829	983	1533	143	557	498	713	905	
871	529	1360	691	1400	1522	588	1468	969	844	1651	548	27	477	669	241	900	3005	1437	28	1399	265	628	184	
1178	5555	454	460	172	13708	866	2008	654	848	330	182	612	137	1248	723	298	3999	564	8443	467	1469	676	725	
854	948	1460	802	422	75	1367	84	2020	1720	784	657	1551	180	401	826	452	295	409	1500	1763	5801	559	1022	
106	199	1381	227	621	973	7002	658	14	773	391	1508	1019	1112	282	5000	225	995	620	3086	1488	670	273	6103	
959	299	6146	930	243	851	26208	125	755	552	1346	1497	13718	896	984	48	39	731	386	845	420	1535	154	1483	
328	61440	813	466	4559	33	1222	211	1380	6666	766	3986	110	480	278	820	162	332	5998	839	685	239	1470	916	
337	286	1016	665	5193	947	296	1402	1392	5302	965	335	307	563	342	249	743	1531	2627	471	1015	500	1456	807	
977	13701	103	836	885	1439	212	3900	712	470	5011	565	158	598	1429	64	54	357	1353	10083	188	127	176	971	
558	2014	1467	1454	446	955	883	382	2105	425	496	395	663	291	6105	269	1058	918	798	4343	722	5717	1548	1462	
1033	970	1414	539	68	247	7000	447	808	229	869	739	1521	1059	5232	135	192	1482	2605	20	1021	928	381	274	
61441	7009	310	938	771	13708	2501	688	215	316	1396	6110	1464	329	4660	7200	891	650	589	6543	384	38037	586	1405	
748	1537	353	4672	2044	680	551	1410	1761	13702	17	1663	7100	1525	1502	494	350	476	108	989	2603	627	456	1356	
1434	1661	964	326	238	1411	1027	520	2026	1527	1425	1368	1385	972	904	978	6112	639	4899	1990	2033	2120	327	571	
223	824	1988	54320	178	1127	1453	1363	2431	1544	248	349	1417	1386	1	5714	6667	1827	581	1528	334	1397	575	1471	
372	359	7006	13705	22273	38	44	931	7201	1390	5802	1110	504	41	870	1084	32771	975	285	1680	13722	1546	618		
542	5632	1023	710	22321	1067	6101	721	1212	1005	1935	945	503	254	417	690	107	418	596	719	683	2766	720		
1498	13783	3268	789	810	336	121	1372	514	1029	301	104	754	6050	2004	309	2041	147	13721	553	6008	352	1401		
27008	921	13712	863	9111	779	689	47	138	311	51	123	31337	429	1391	472	1008	258	207	366	506	1424	1489		
556	858	8888	879	2025	202	361	625	805	1452	911	2108	924	288	1024	632	1513	3292	1473	647	27004	1447	1655		
428	1496	100	7	1361	3	267	534	6148	525	929	951	735	7001	1406	849	1031	961	1366	217	1076	99	46		
556	331	220	583	806	705	943	1534	606	923	8892	1350	98	521	230	473	990	2000	6144	672	246	32778	549	428	

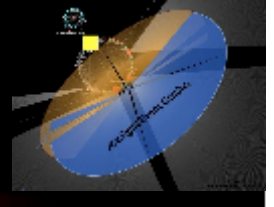
SIP

DIP

DPort

Firewall Log File Analysis

Blocked Incoming Bogon Addresses



Command:

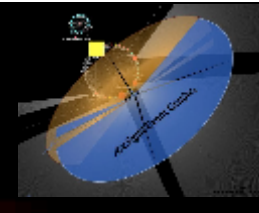
```
cat log | grep block_in | ./afterglow -c properties -d | neato -Tgif -o foo.gif
```



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};
```

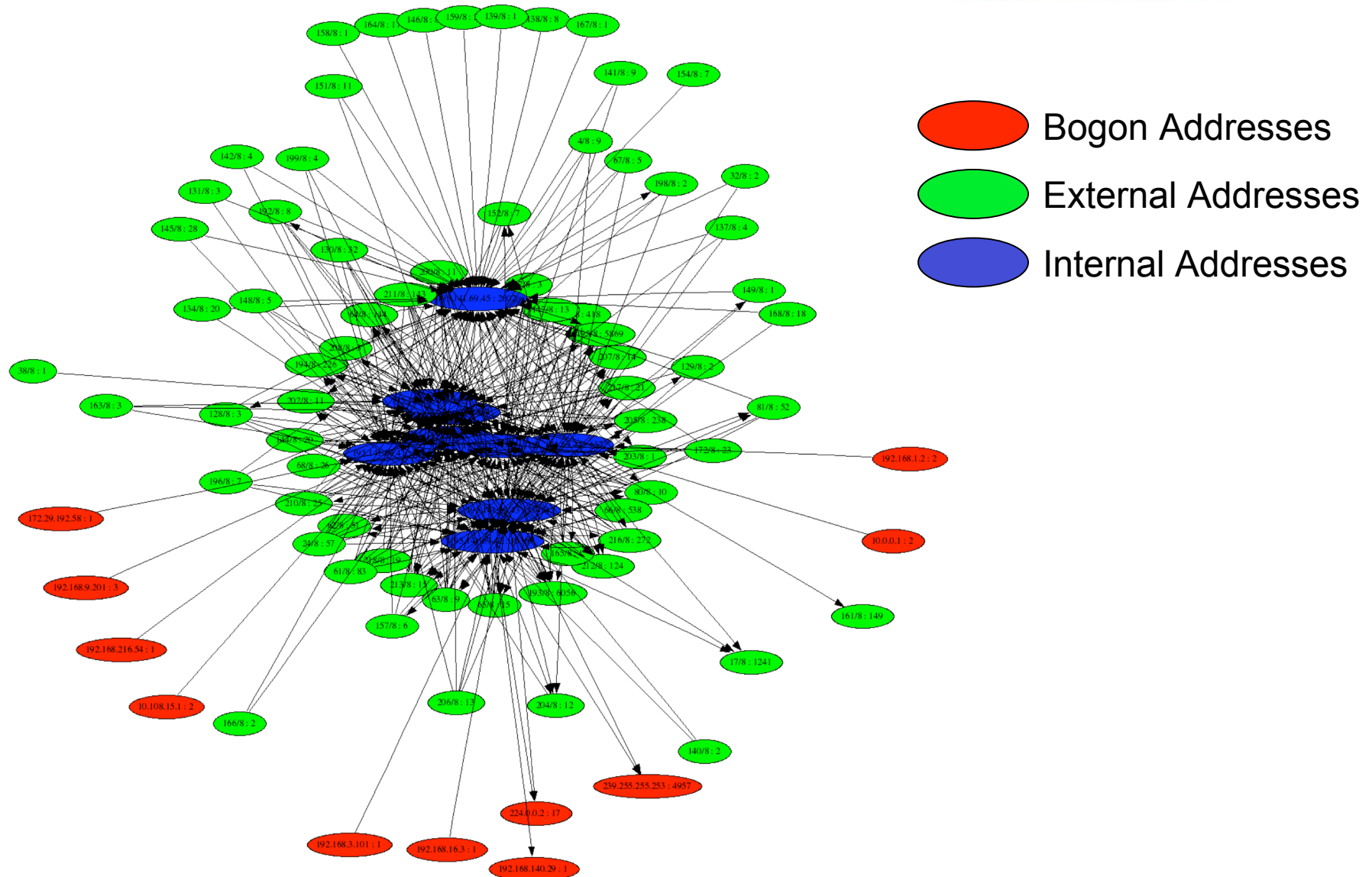
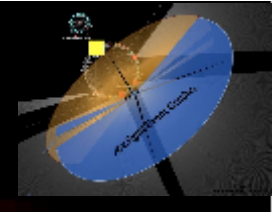
```
color $value=0; map { $value+=subnet(field(), $_) }  
@ranges; regex_replace("(\\d+)."/8" if  
color match("^195\\.(141|69)\\.105\\.!$value)");  
color $value=0; map { $value+=subnet(field(), $_) }  
@ranges; regex_replace("(\\d+)."/8" if  
(!match("^195\\.141\\.69)")) && !$value);
```

Features:

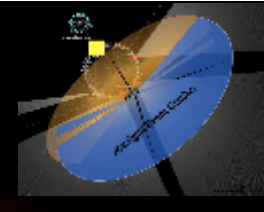
```
variable=  
regex_replace()  
subnet(IP,range) e.g., subnet("10.0.0.2","10.0.0.0/8") → 1 (true)
```

Firewall Log File Analysis

Blocked Incoming Bogon Addresses



Summary



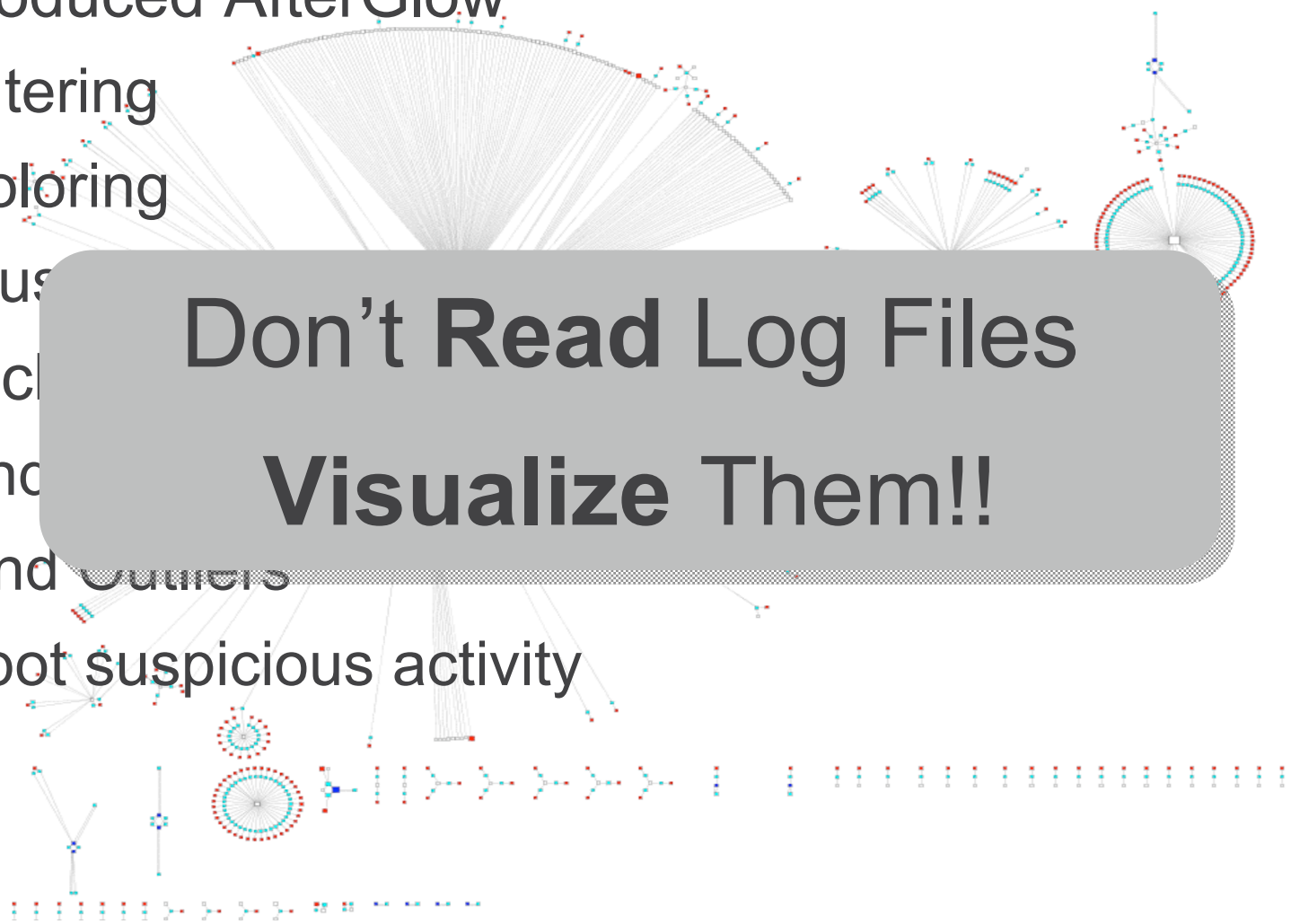
Introduced AfterGlow

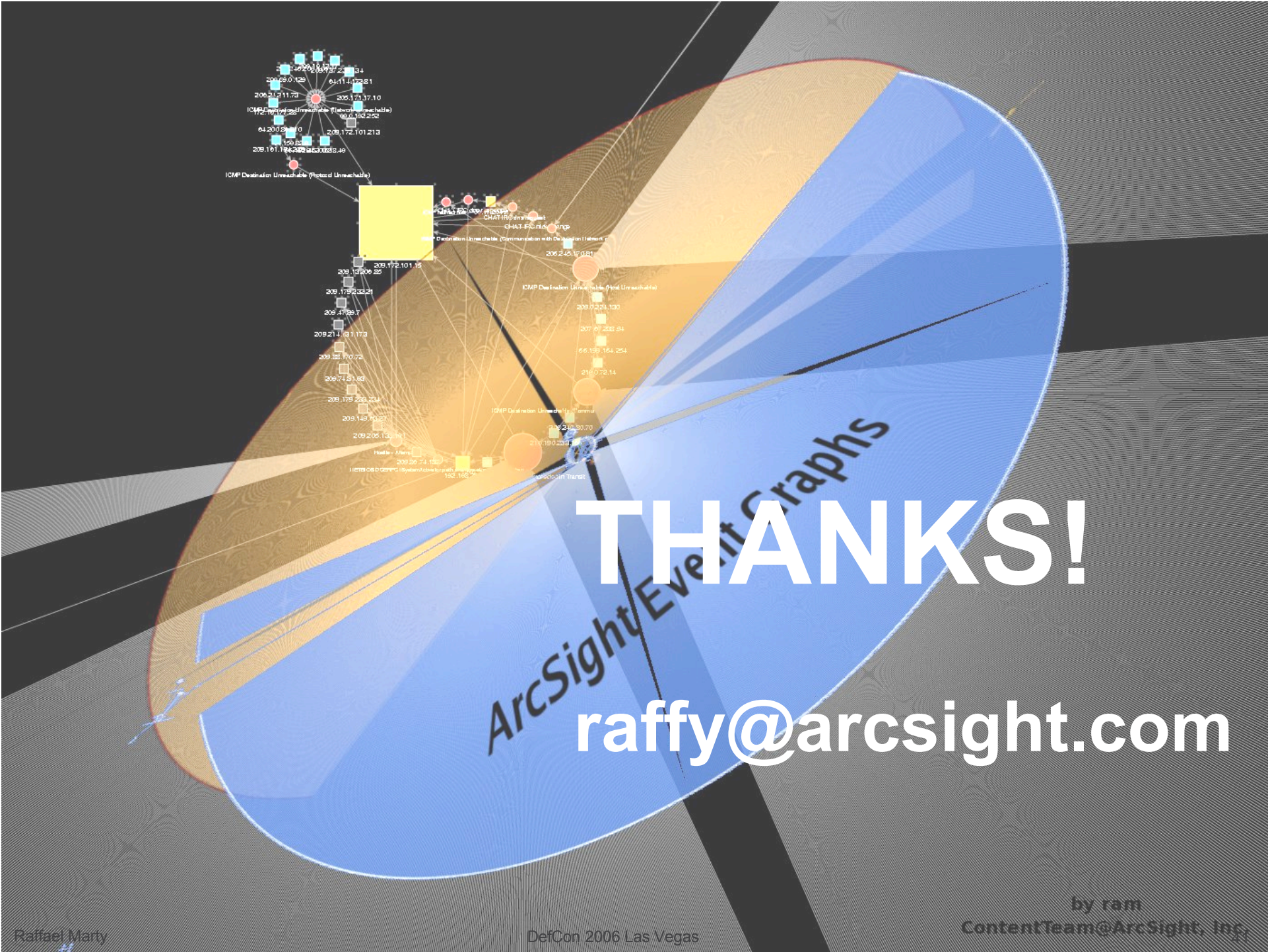
- Filtering
- Coloring
- Clustering

Quick

- Undo
- Find Outliers
- Spot suspicious activity

**Don't Read Log Files
Visualize Them!!**





THANKS!

raffy@arcsight.com

ArcSight Event Graphs