Cooperative Infrastructure for Security and CTF Teams

By Dan Borges

whoami

- Casual CTF Player and Host
- CCDC Red Teamer
- Ex-Mandiant Consultant
- CEO of a startup
- http://attack.tools

This is not an intro, this is for ctf teams

trailofbits.github.io/ctf/

CTF Field Guide

About the author
Questions and Issues
Edit and Contribute
Introduction
1. Capture the Flag
1.1. Find a CTF
1.2. Find a Job
2. Vulnerability Discovery
2.1. Auditing Source
2.2. Auditing Binaries
2.3. Auditing Webapps
3. Exploit Creation
3.1. Binary Exploits 1

CTF Field Guide

> "Knowing is not enough; we must apply. Willing is not enough; we must do." - Johann Wolfgang von Goethe

Welcome!

We're glad you're here. We need more people like you.

If you're going to make a living in defense, you have to think like the offense.

So, learn to win at Capture The Flag (CTF). These competitions distill major disciplines of professional computer security work into short, objectively measurable exercises. The focus areas that CTF competitions tend to measure are vulnerability discovery, exploit creation, toolkit creation, and operational tradecraft.

Whether you want to succeed at CTF, or as a computer security professional, you'll need to become an expert in at least one of these disciplines. Ideally in all of them.

That's why we wrote this book.

In these chapters, you'll find everything you need to win your next CTF competition:

GitHub, Inc. [US] https://github.com/ctfs

CTFs

Repositories | People 3

Filters | Find a repository...

write-ups-tools — Python ★ 24 ⑂ 19
A collection of tools used to maintain and create CTF writeup folders
Updated 14 hours ago

write-ups-2015 — CSS ★ 767 ⑂ 331
Wiki-like CTF write-ups repository, maintained by the community. 2015
Updated 16 hours ago

write-ups-2013 — Python ★ 66 ⑂ 38
Wiki-like CTF write-ups repository, maintained by the community. 2013
Updated 28 days ago

write-ups-2014 — Python ★ 1,016 ⑂ 360
Wiki-like CTF write-ups repository, maintained by the community. 2014
Updated on Sep 27

People 3 >
- dhanvi — Tummala Dhanvi (c0mrad3)
- fariskhi — Fariskhi Vidyan
- YASME-Tim — polym

xx.blogspot.com/2013/08/bootcamp-project-ctf-primer.html

LOCKBOXX
A CONSTANTLY EVOLVING SECURITY BLOG

MONDAY, AUGUST 19, 2013

Bootcamp Project: CTF Primer

So you want to play in a digital Capture The Flag (CTF)??

The first step is always finding and registering in the CTF you want to play in. After that, this guide is here to help you prepare for your first CTF, aka expect the unexpected.

First, look for local CTFs or register for some Online CTFs.

Think CTFs are over your head? Think again:

If You Can Open The Terminal, You Can Capture T

```
~$ grep ^ctf /etc/group | formatgroup.sh
ad0nis
dth0m
j3remy
jwgoerlich
konradv2
mrjcomputing
niteshad
phgg
sukotto_san
ybat313
zandi
ztango
[and many more!  <3 you, @mlsec!]
~$
```

Breaking the CTFs down we have three styles to look at: Jeopardy, Offensive-Scenario, Offense-Defense.

Offensive-Scenario: Typically involves either attack scenarios or live systems to attack. These types of CTFs will emulate penetration testing over other forms of hacking.

ABOUT ME

ACTION DAN

I'm a fun loving 25 year old male, with an incredible sense of adventure. Technology is my passion, but I am an extremely diverse person.

VIEW MY COMPLETE

OpSec basics: don't use public / indexed services

https://web.archive.org/web/20150503024603/http://pastebin.com/p1nZwjnE

INTERNET ARCHIVE Wayback Machine

http://pastebin.com/p1nZwjnE

create new paste | trending pastes | 1 captures 3 May 15 - 3 May 15

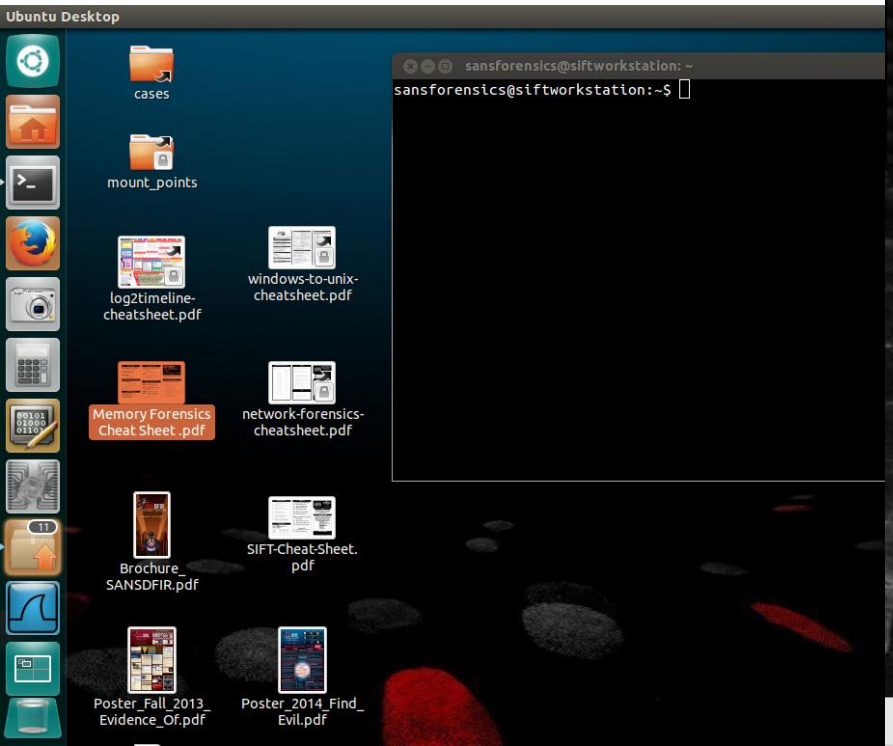
We got the flag

BY: A GUEST ON MAY 1ST, 2015 | SYNTAX: PYTHON | SIZE: 1.63 KB | VIEWS: 213 | EXPIRES: NEVER

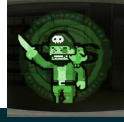
DOWNLOAD | RAW | EMBED | REPORT ABUSE | PRINT

```
1.  offset =
    "01111101001000101000000111101001001011111110010011100111010011000010101101110110100001101011
2.
3.  string  = "From a seed a mighty trunk may grow.\n"
4.
5.  #string_entered =
    "0100011001110010011011110110110100100000011000010010000001110011011001010110010101100100001000
6.  tmp = len(offset)
7.  result = [None] * tmp
8.
9.
10. '''finalString = []
11. for i in range(0, len(result) >> 3):
12.         tmpString = 0
13.         for j in range(0,8):
14.                 tmpString = (j + result[8*i] - 2 * tmpString) +48
```

Have your tools ready and updated!



Pure Funky Magic

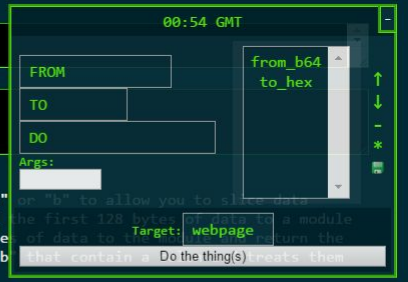


Input: Hide input | Choose File | No file chosen | Clear input

```
QUhBSEFIQQ==
```

Output: Move to input

```
00000000: 41 48 41 48 41 48 41                          AHAHAHA
```



Module descriptions

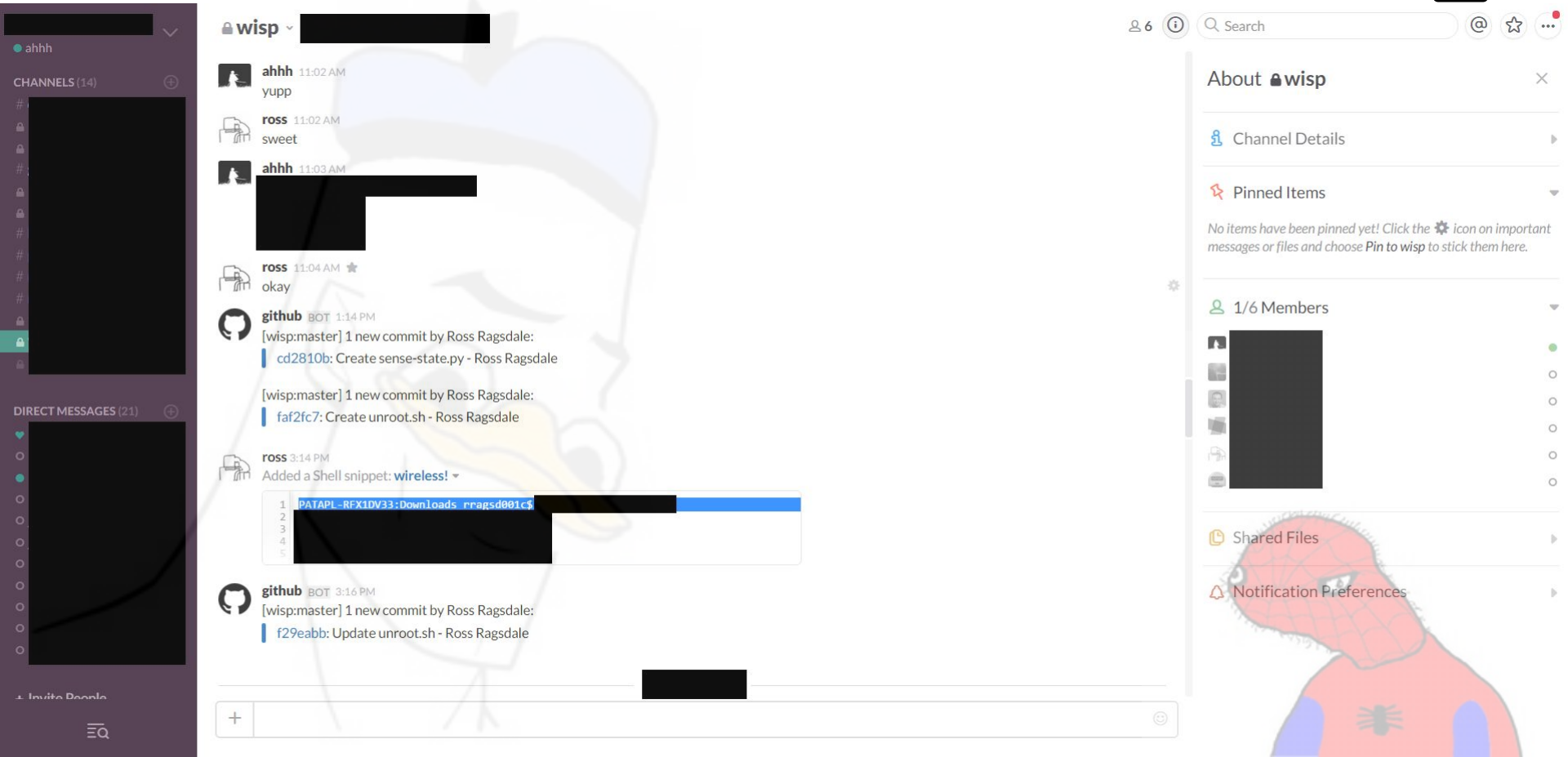
Note: Slicing is now enabled to trim data before and/or after processing by a specific module. This feature is based on the Python slicing notation preceded by an "a" or "b" to allow you to slice data before ("b") you send data to a PFM module or after ("a") it has been processed. For example, adding "b0:128 a64:-32" as module arguments will cause PFM to only send the first 128 bytes of data to a module and return only the 64th through whatever byte is 32 bytes from the end of the processed data for that module. Similarly, "b18: a:256" will suppress the first 18 bytes of data to the module and return the first 256 bytes of processed data. Stepping and other advanced slicing features are not supported however. The script checks for arguments beginning with an "a" or "b" that contain a ":" and treats them as slicing arguments. To override this for a specific argument (like a zip password that happens to match that pattern) prepend it with "--".

from_b64: Base64 decodes the input.
from_bplist: Converts a binary plist to XML (default) or JSON format.
- optional arg(s): "json" for JSON format.
from_bz2: Inflates bz2 compressed data.
from_deflate: Inflates a deflate compression stream.
from_gzip: Inflates a gzip stream.
from_hex: Converts a hex dump to raw data. Supported formats include xxd, Wireshark, hexdump and plain hex with no spacing.
from_unix_epoch: Converts unix epoch timestamp to human formatted time (GMT).
from_url: URL decodes the input (including encoded spaces).
from_wide: Converts two-byte wide characters (UTF-16 unicode) to one-byte characters.
from_zip: Expands a zip file.
- optional arg(s): Zip archive password
from_zlib: Inflates a zlib stream.

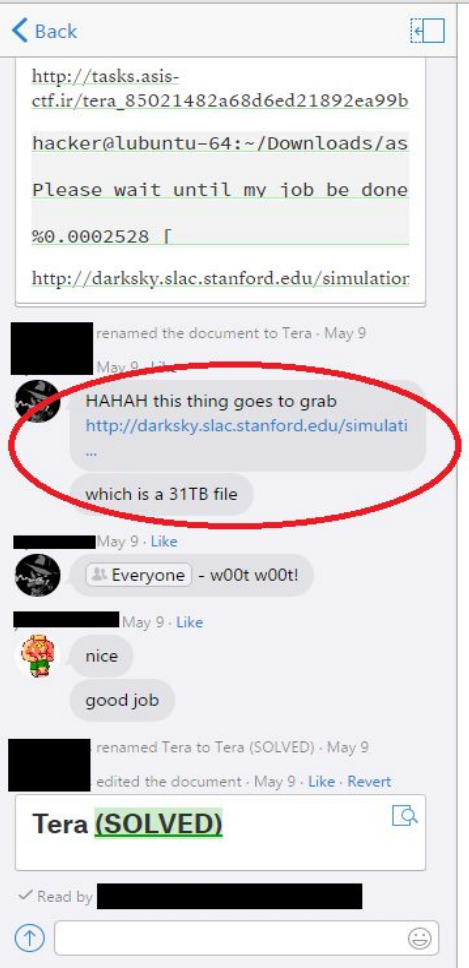
to_b64: Base64 encodes input.
to_bz2: Bz2 compresses data.
to_deflate: Creates a deflate compression stream.
to_gzip: Creates a gzip stream.
to_hex: Provides a hexdump of the input.
- optional arg(s): "stream" will convert to hex stream
to_url: URL encodes input (including spaces).
to_zip: Zip compresses input data.
to_zlib: Creates a zlib stream.

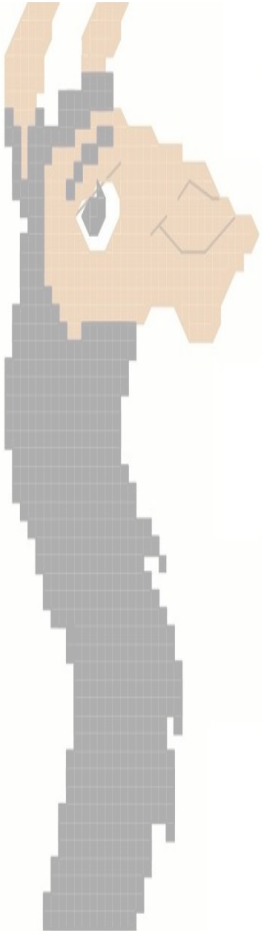
do_byte_spread: Quick byte frequency analysis of input. Also performs a Shannon entropy check - the closer to 1.0 the more random the data (possibly indicating encryption).
do_pkt_payloads: Extracts data payloads from all TCP, UDP & ICMP packets in a pcap. No ordering or defrag.
do_rc4: Applies RC4 crypto key to input.
- required arg(s): A key in hex format (no spaces or leading 0x) is required.
do_strip_HTTP_header: Strips the HTTP header from a dumped TCP stream.
do_xor: Performs an XOR against the input using a supplied key.
- required arg(s): A single or multi-byte XOR key in plain hex (no spaces, no leading 0x) is required.

Slack / Github / Google Docs integration for chat-ops



Using Quip for file sharing, interactive docs, and chat-ops





ASIS{3149ad5d3629581b17279cc889222b93}

```
import requests
import struct

url = "http://darksky.slac.stanford.edu/simulations/ds14_a/ds14_a_1.0000"

def chunks(l, n):
    for i in xrange(0, len(l), n):
        yield l[i:i+n]

def xor(data, key):
    return bytearray(a^b for a, b in zip(*map(bytearray, [data, key])))

with open('tera.elf', 'rb') as f:
    data = f.read()

# Extracted with IDA :)
memory_locations = data[0x1480:0x1480+0x130]
# Extracted with IDA :)
xor_key = data[0x1680:0x1680+0x98]

# Get all memory locations
mem_chunks = [struct.unpack("<Q", x)[0] for x in chunks(memory_locations, 8)]

xor_chunks = "".join([chr(struct.unpack("<L", x)[0]) for x in chunks(xor_key, 4)])

encrypted_string = ""

for chunk in mem_chunks:
    print "Fetched byte at %08x..." % chunk
    res = requests.get(url, headers={"range": "bytes=%d-%d" % (chunk, chunk)})

    encrypted_string += res.content

print xor(encrypted_string, xor_chunks)
```

Etherpad for a private interactive doc application

In lot of char fields, there is a useless size. List here char where we could remove the size (business logic)

- Remove size everywhere , do not updtate it.
  - Keep only size in bar code (ean13) field in product form
  - QDP: are you sure?

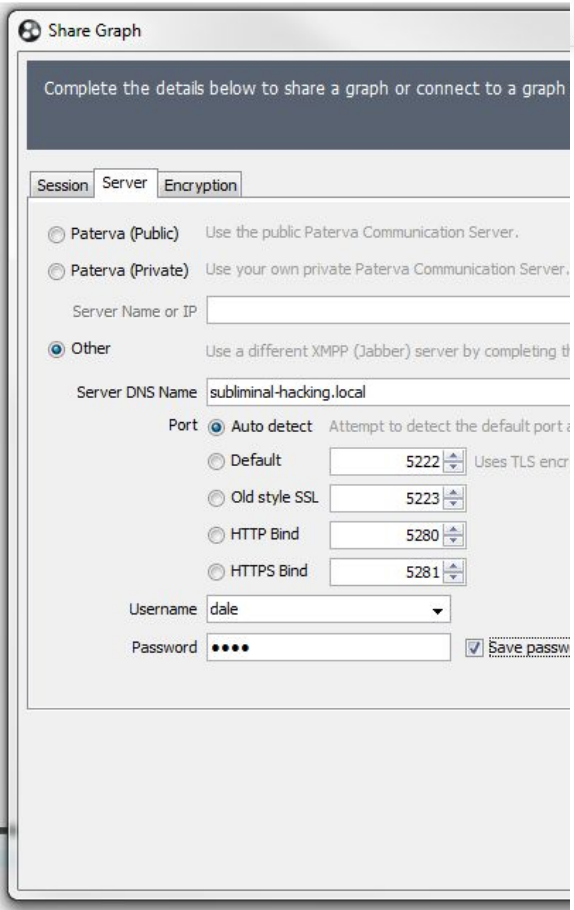
Specification:

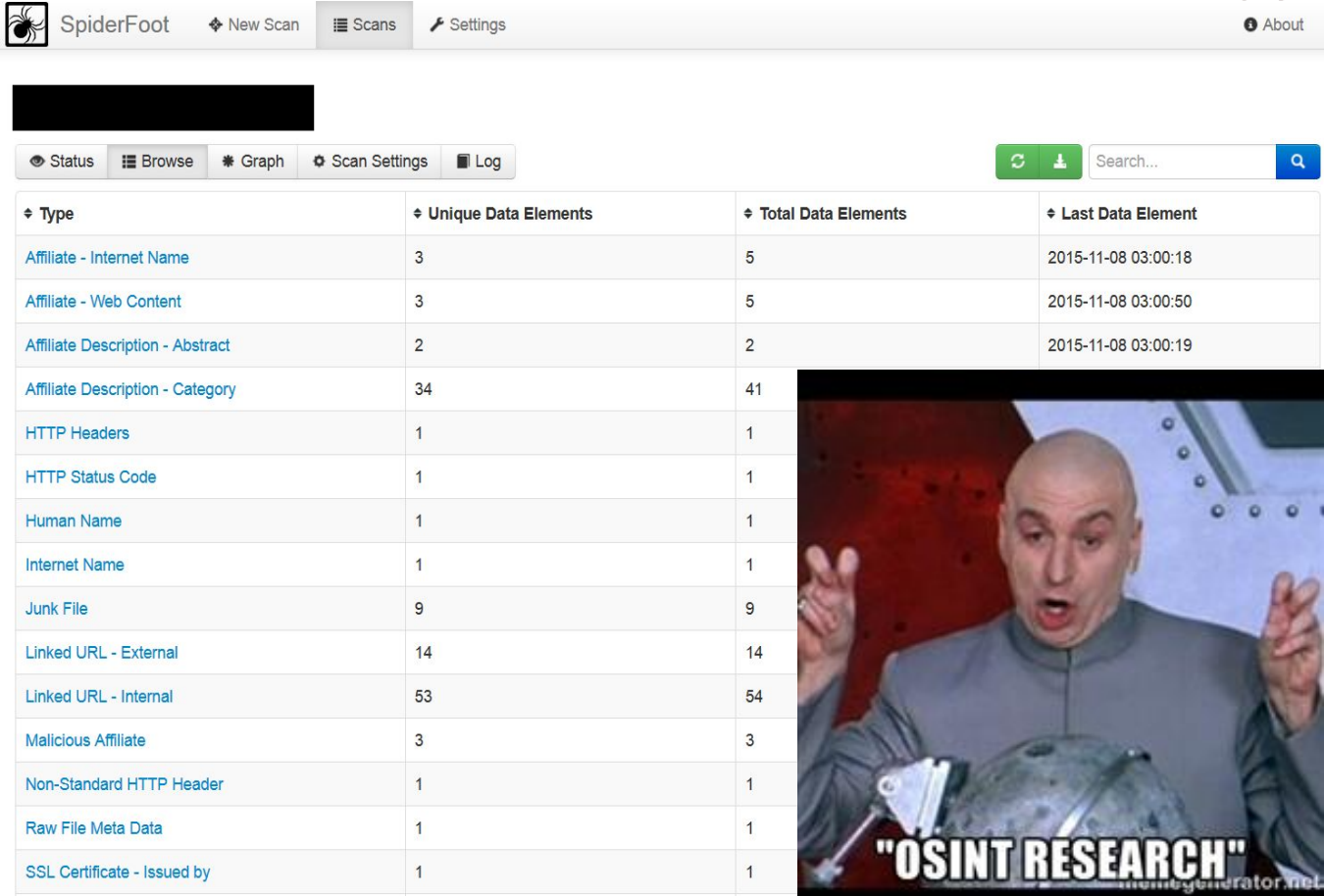
- ~~Remove size attribute form all modules, keeps with special cases~~
  - ~~Special cases like, (ean13, journal code*,...)~~
  - ~~**Note:** Special cases are manged during the development of task.~~
- Report: ~~Manage function 'ellipsis(size, value)' that crop a <value> to display only <size-3>~~
  - ~~return value + '...'~~
    - **_ellipsis(char, size=100, truncation_str='...')** , Feature already their within report to no require to create new function.
  - check all reports to see if, with normal values, it's still ok otherwise use the **_ellipsis()** method
  - *: journal code has been arbitrary set to stay 3 letters long because otherwise it's the mess in reports (e.g: general ledger..), in move names... and it's usually enough.
    - *Here use function:*ellipsis(size, value)'.

Branch:

- Addons:

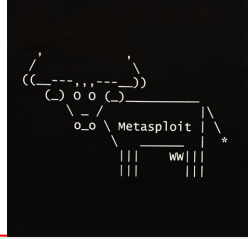
OSINT cooperation with Maltego or Spiderfoot





Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Internet Name	3	5	2015-11-08 03:00:18
Affiliate - Web Content	3	5	2015-11-08 03:00:50
Affiliate Description - Abstract	2	2	2015-11-08 03:00:19
Affiliate Description - Category	34	41	
HTTP Headers	1	1	
HTTP Status Code	1	1	
Human Name	1	1	
Internet Name	1	1	
Junk File	9	9	
Linked URL - External	14	14	
Linked URL - Internal	53	54	
Malicious Affiliate	3	3	
Non-Standard HTTP Header	1	1	
Raw File Meta Data	1	1	
SSL Certificate - Issued by	1	1	

Metasploit master server and connecting clients



Player

Applications Places Terminal Sat 03:11

Terminal

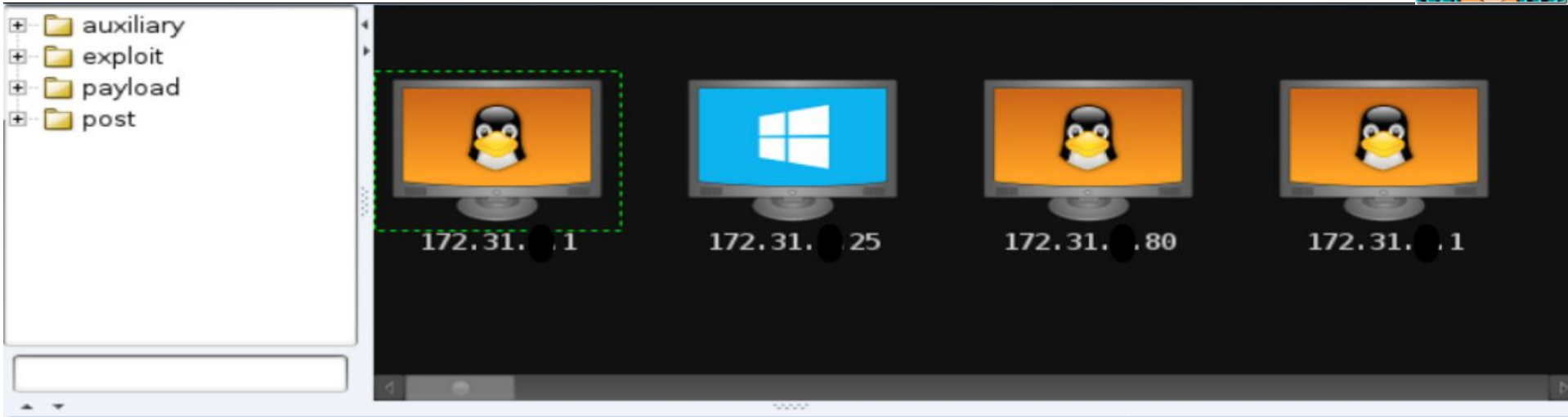
File Edit View Search Terminal Help

```
Easy phishing: Set up email templates, landing pages and listeners
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.4-2015092301                   ]
+ -- --=[ 1485 exploits - 857 auxiliary - 250 post        ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > db_connect msf:[redacted]@[redacted].220/msf
[*] Rebuilding the module cache in the background...
msf > db_status
[*] postgresql connected to msf
msf > hosts

Hosts
=====

address      mac  name           os_name      os_flavor  os_sp  purpose   info  comm
ents
-------      ---  ----           -------      ---------  -----  -------   ----  ----
---
[redacted].43                                                  firewall
[redacted].97      WIN-[redacted]  Windows 2012               client

msf >
```

Command Prompt - ssh -i seasoned_security.pem ubuntu@[redacted]

```
[*] Time: 2015-08-02 06:26:52 UTC Note: host=[redacted].43 type=host.nat.server data={:info=>"This device is ac
[*] Time: 2015-08-02 06:26:52 UTC Note: host=[redacted].97 type=host.nat.client data={:info=>"This device is tr
msf > chosts
[-] Unknown command: chosts.
msf > notes
[*] Time: 2015-08-02 06:26:51 UTC Note: host=[redacted].97 type=host.os.session_fingerprint data={:name=>"WIN-[
[*] Time: 2015-08-02 06:26:52 UTC Note: host=[redacted].43 type=host.nat.server data={:info=>"This device is ac
[*] Time: 2015-08-02 06:26:52 UTC Note: host=[redacted].97 type=host.nat.client data={:info=>"This device is tr
msf > hosts

Hosts
=====

address      mac  name           os_name      os_flavor  os_sp  purpose   info  comments
-------      ---  ----           -------      ---------  -----  -------   ----  --------
[redacted].43                                                  firewall
[redacted].97      WIN-[redacted]  Windows 2012               client

msf > curl icanhazip.com
[*] exec: curl icanhazip.com

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    15  100    15    0     0    175      0 --:--:-- --:--:-- --:--:--   176
[redacted].220
msf > db_status
[*] postgresql connected to msf
msf >
```

Cobalt Strike / Armitage Team-servers



Event Log X | Console X | nmap X

Scan X | Scan X | Scan X | Services X | Services X | Services X | Services X

host	name	port	proto	info
172.31.[redacted].1	ssh	22	tcp	SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze5
172.31.[redacted].1	telnet	23	tcp	Welcome to VyOS\x0a\x0dvyos login:
172.31.[redacted].1	mysql	3306	tcp	5.5.41-0ubuntu0.12.04.1
172.31.[redacted].1	ms-wbt-server	3389	tcp	Microsoft Terminal Service

Everyone loves cracking hashes - have them on demand

```
sudo echo "/usr/local/cuda/lib" | sudo tee -a
/etc/ld.so.conf.d/cuda.conf
sudo ldconfig
cd /usr/local/cuda/samples/1_Utilities/deviceQuery
sudo make
sudo ./deviceQuery
cd ~
wget http://us.download.nvidia.com/XFree86/Linux-x86_64/340.32/NVIDIA-
Linux-x86_64-340.32.run
chmod +x NVIDIA-Linux-x86_64-340.32.run
sudo ./NVIDIA-Linux-x86_64-340.32.run
wget http://hashcat.net/files/cudaHashcat-1.21.7z
7za x cudaHashcat-1.21.7z
```

Finally you should be able to bench mark it using:

```
./cudaHashcat64.bin -b
```

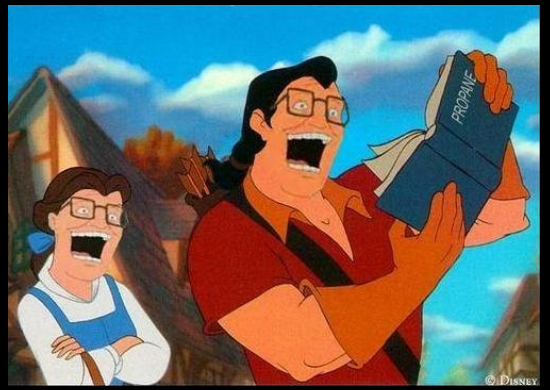
You will probably want some wordlists for your cracking, I've assembled a collection of wordlist collections here, with a little bit on creating customized wordlists.

And best of all, it works :) Here's cracking some MD5(Unix) passwords, and here's to cracking more!

```
./cudaHashcat64.bin -m 500 -a 0 ~/hashes/md5.unix.txt
~/wordlists/rockyou.txt

Session.Name...: cudaHashcat
Status.........: Exhausted
Input.Mode.....: File (/home/ubuntu/wordlists/rockyou.txt)
Hash.Target....: File (/home/ubuntu/hashes/md5.unix.txt)
Hash.Type......: md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
Time.Started...: Sat Aug 16 01:59:39 2014 (3 mins, 26 secs)
Time.Estimated.: 0 secs
Speed.GPU.#1...: 75697 H/s
Recovered......: 4/9 (44.44%) Digests, 4/9 (44.44%) Salts
Progress.......: 129089664/129089664 (100.00%)
Skipped........: 55538172/129089664 (43.02%)
Rejected.......: 9/129089664 (0.00%)
HWMon.GPU.#1...: 0% Util, 64c Temp, -1% Fan
```

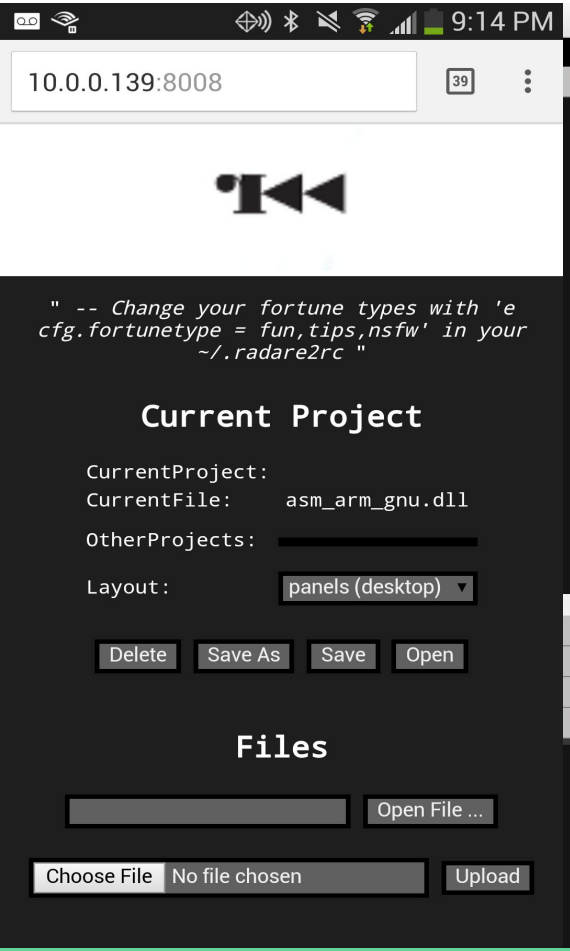
Command Prompt - ssh -i seasoned_security.pem ubuntu@[redacted]

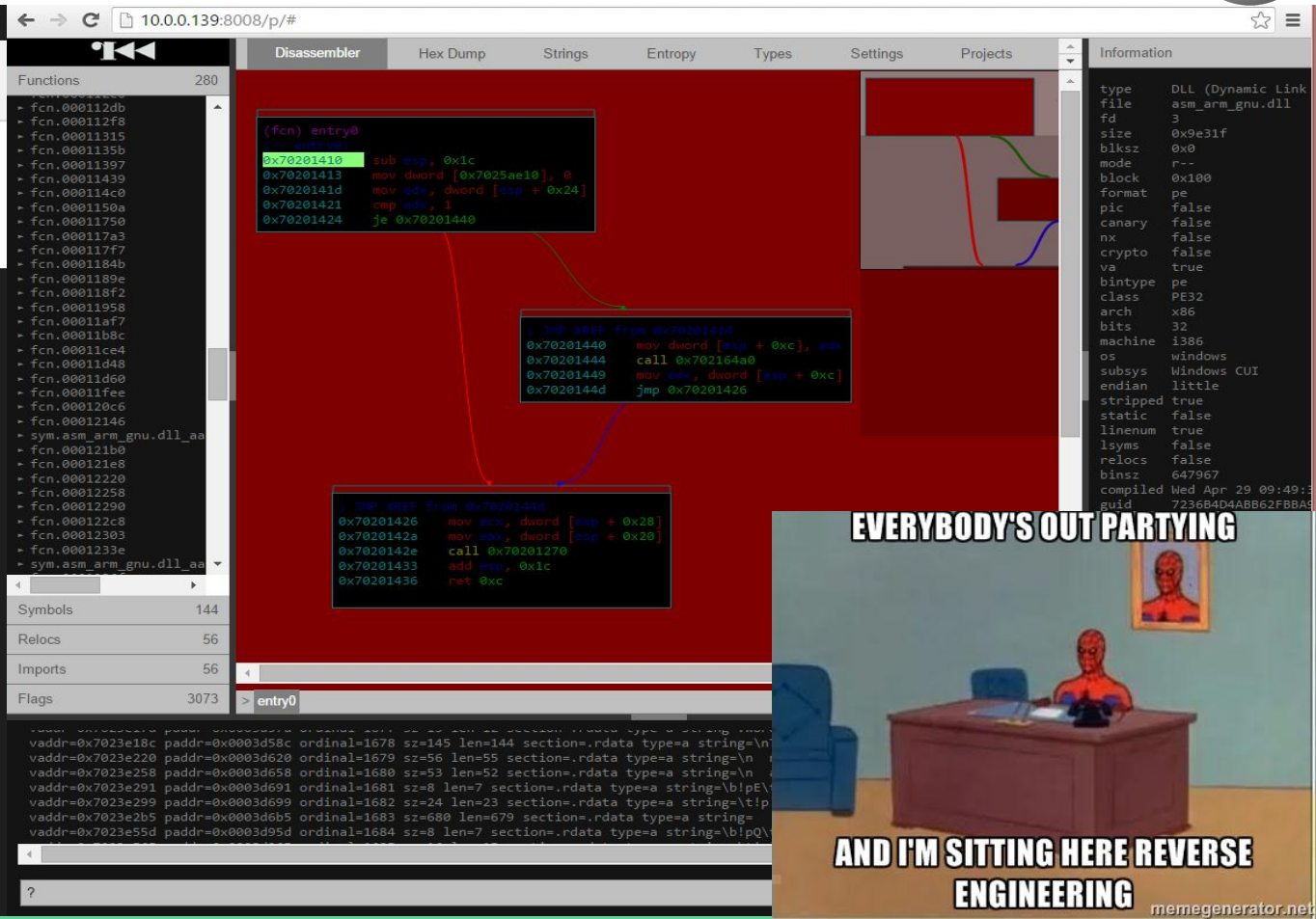
```
Speed.GPU.#1.:   683.6 MH/s

Hashtype: SHA256
Workload: 1024 loops, 256 accel

Speed.GPU.#1.:   290.5 MH/s

Hashtype: SHA384
Workload: 256 loops, 256 accel

Speed.GPU.#1.: 71960.0 kH/s

Hashtype: SHA512
Workload: 256 loops, 256 accel

Speed.GPU.#1.: 71907.4 kH/s

Hashtype: SHA-3(Keccak)
Workload: 128 loops, 256 accel

Speed.GPU.#1.: 58364.7 kH/s

Hashtype: SipHash
Workload: 1024 loops, 256 accel

Speed.GPU.#1.:  3105.0 MH/s

Hashtype: RipeMD160
Workload: 1024 loops, 256 accel

Speed.GPU.#1.:   481.6 MH/s

Hashtype: Whirlpool
Workload: 512 loops, 32 accel

Speed.GPU.#1.: 43914.4 kH/s
```



Fixed Disks
Fixed Disk 0
Disk IO

Cooperative disassembly with a radare2 webserver





CloudShark.com for pcap collaboration in browser

mDNSResponder - 11-5-15 at 7.32 PM.pcapng 42.7 kb · 241 packets · more info

Start typing a Display Filter | Apply | Clear | Analysis Tools | Graphs | Download | Profile

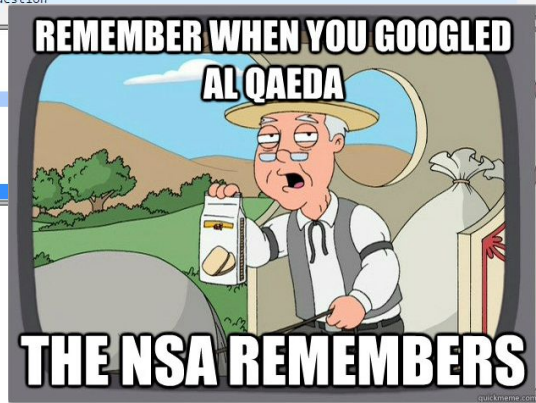
No.	Time	Source	Destination	Protocol	Length	Info
110	30.850819000	0.0.0.53	75.75.75.75	DNS	106	Standard query 0x8574 A user-att-99-38-152-0.e3191.dscc.akamaiedge.net
111	30.850821000	0.0.0.53	75.75.75.75	DNS	128	Standard query 0x9965 AAAA user-att-99-38-152-0.e3191.dscc.akamaiedge.net.0.1.cn.akamaiedge.net
112		75.75.75.75	0.0.0.53	DNS	224	Standard query response 0x9e62 A 17.110.227.83 A 17.110.229.80 A 17.110.224.215 A 17.110.229.140 A 17.110.228.17 A 17.110.224.20 A 17.110.229.91 A 17.110.226.73
113		75.75.75.75	0.0.0.53	DNS	122	Standard query response 0x8574 A 23.198.170.139
114		75.75.75.75	0.0.0.53	DNS	184	Standard query response 0x9965 AAAA 2600:1406:34:282::c77 AAAA 2600:1406:34:287::c77
115	31.646934000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR c.8.b.a.2.d.e.f.f.f.b.3.6.3.6.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
116	31.647082000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR c.8.b.a.2.d.e.f.f.f.b.3.6.3.6.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
117	31.647208000	10.0.0.82	0.0.0.53	MDNS	152	Standard query 0x0000 PTR c.8.b.a.2.d.e.f.f.f.b.3.6.3.6.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
118	31.647332000	::35	ff02::fb	MDNS	152	Standard query 0x0000 PTR c.8.b.a.2.d.e.f.f.f.b.3.6.3.6.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
119	31.647371000	fe80::1610:9fff:feeb:d2dc	::35	MDNS	152	Standard query 0x0000 PTR c.8.b.a.2.d.e.f.f.f.b.3.6.3.6.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
120	32.628823000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
121	32.628824000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
122	32.628825000	::35	ff02::fb	MDNS	152	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
123	32.628825000	10.0.0.82	0.0.0.53	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
124	32.628825000	fe80::1610:9fff:feeb:d2dc	::35	MDNS	152	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
125	33.628157000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
126	33.628158000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
127	33.628158000	::35	ff02::fb	MDNS	152	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
128	33.628159000	10.0.0.82	0.0.0.53	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
129	33.628159000	fe80::1610:9fff:feeb:d2dc	::35	MDNS	152	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
130	36.625722000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question
131	36.625723000	0.0.0.53	224.0.0.251	MDNS	132	Standard query 0x0000 PTR b.f.1.3.4.b.c.a.b.0.3.3.3.0.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" question

Interesting DNS Responses here

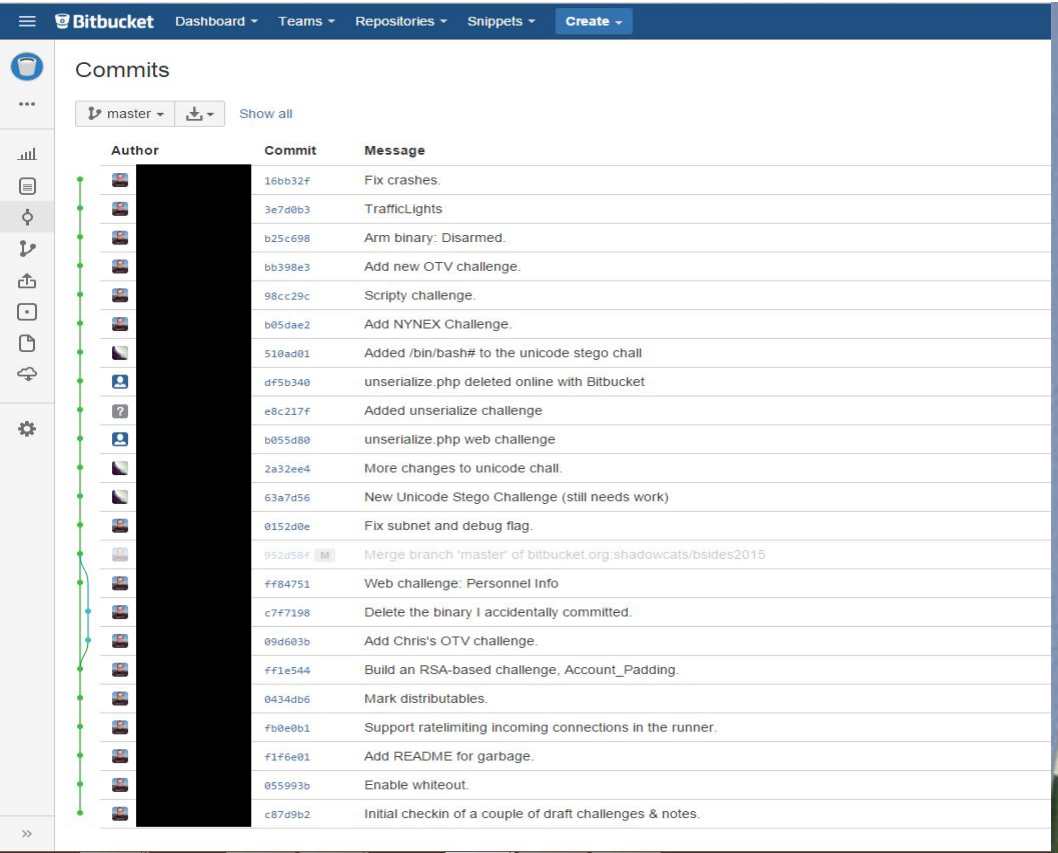
Group annotation of packets

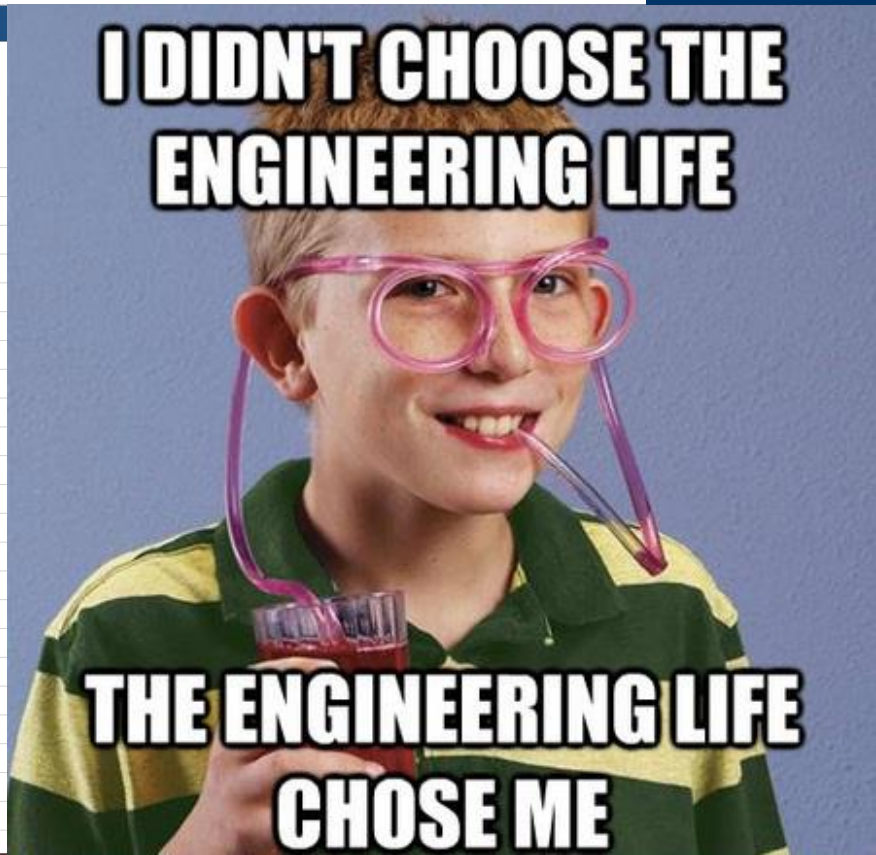
```
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  Answers
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.227.83
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.229.80
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.224.215
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.229.140
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.228.17
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.224.20
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.229.91
    us-courier.push-apple.com.akadns.net: type A, class IN, addr 17.110.226.73
```

```
0000  00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  ..............E.
0010  00 d2 90 d1 40 00 40 11 12 7f 4b 4b 4b 4b 00 00  ....@.@...KKKK..
0020  00 35 00 35 90 d1 00 be 00 00 9e 62 81 80 00 01  .5.5.......b....
0030  00 08 00 00 00 00 0a 75 73 2d 63 6f 75 72 69 65  .......us-courie
0040  72 0a 70 75 73 68 2d 61 70 70 6c 65 03 63 6f 6d  r.push-apple.com
0050  06 61 6b 61 64 6e 73 03 6e 65 74 00 00 01 00 01  .akadns.net.....
0060  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e3 53  .........8...n.S
0070  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e5 50  .........8...n.P
0080  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e0 d7  .........8...n..
0090  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e5 8c  .........8...n..
00a0  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e4 11  .........8...n..
00b0  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e0 14  .........8...n..
00c0  c0 0c 00 01 00 01 00 00 00 38 00 04 11 6e e5 5b  .........8...n.[
```

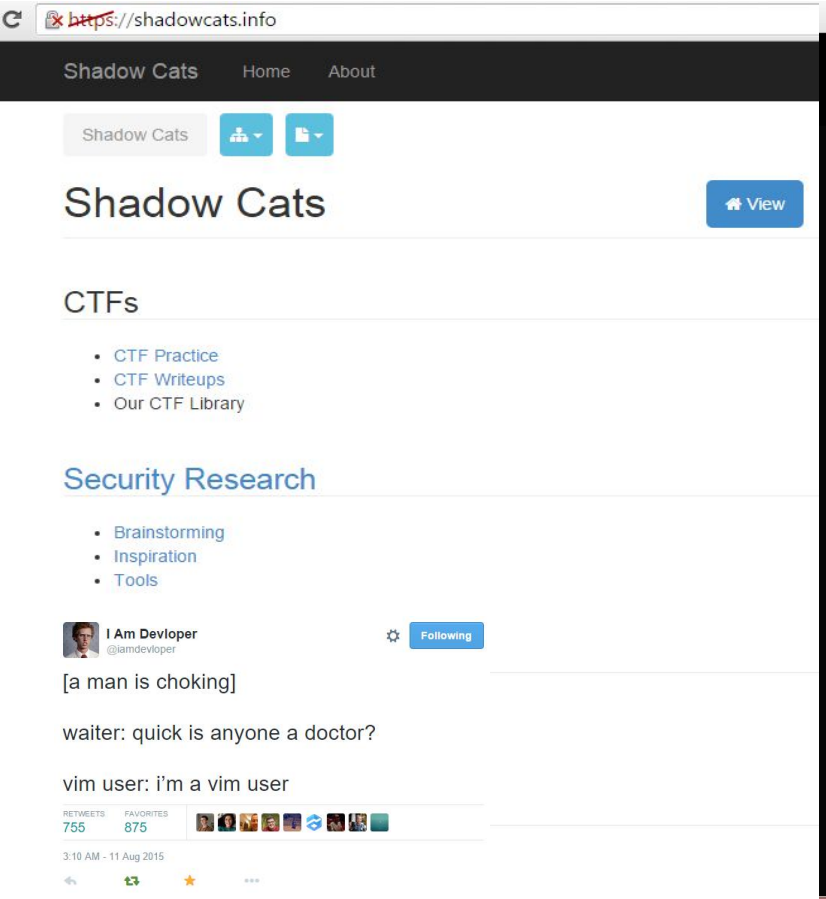


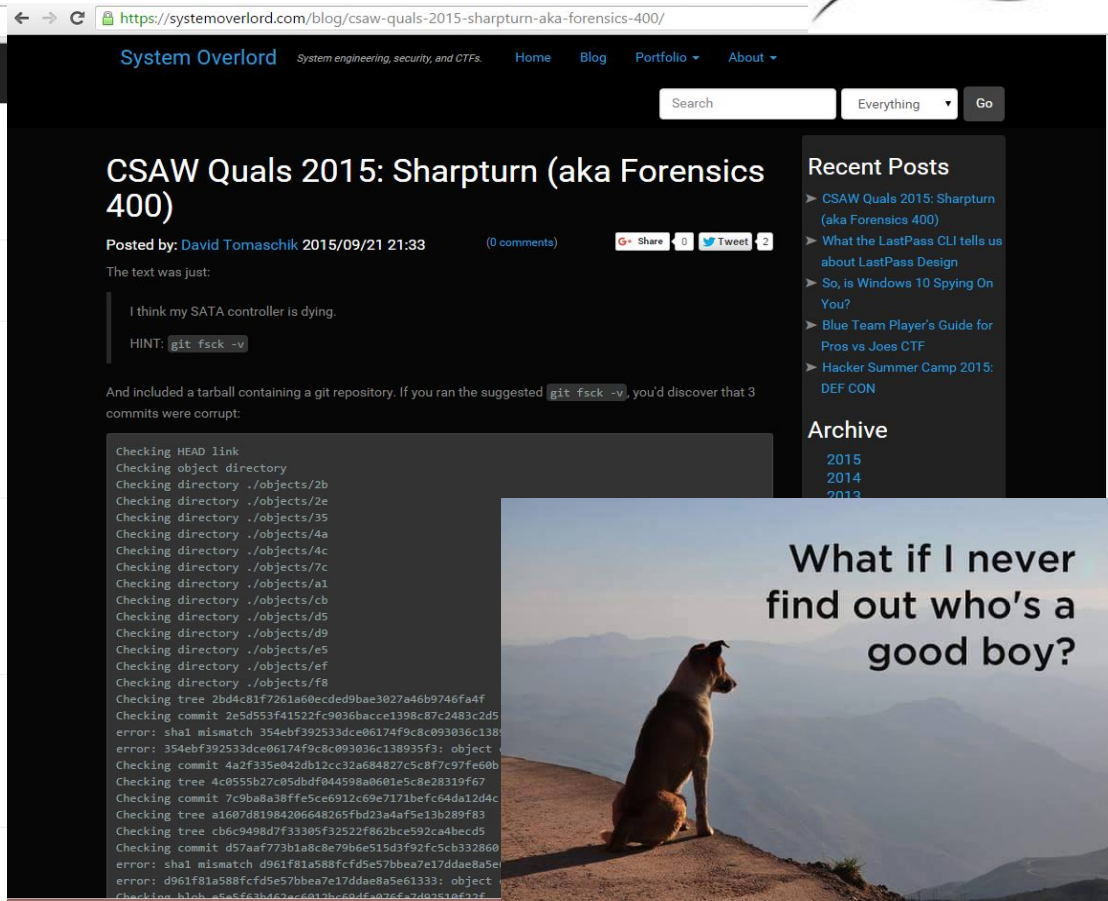
Private BitBucket or GitHub to work on exploit or challenge code together





Start a wiki or blog for notes and lessons learned





¿¿ Questions ??