CS 155: Real-World Security

April 14, 2016

Alex Stamos
CSO, Facebook

Why are you here?

Agenda

We are going to discuss:

- How bugs are found
- How defense works in the real world

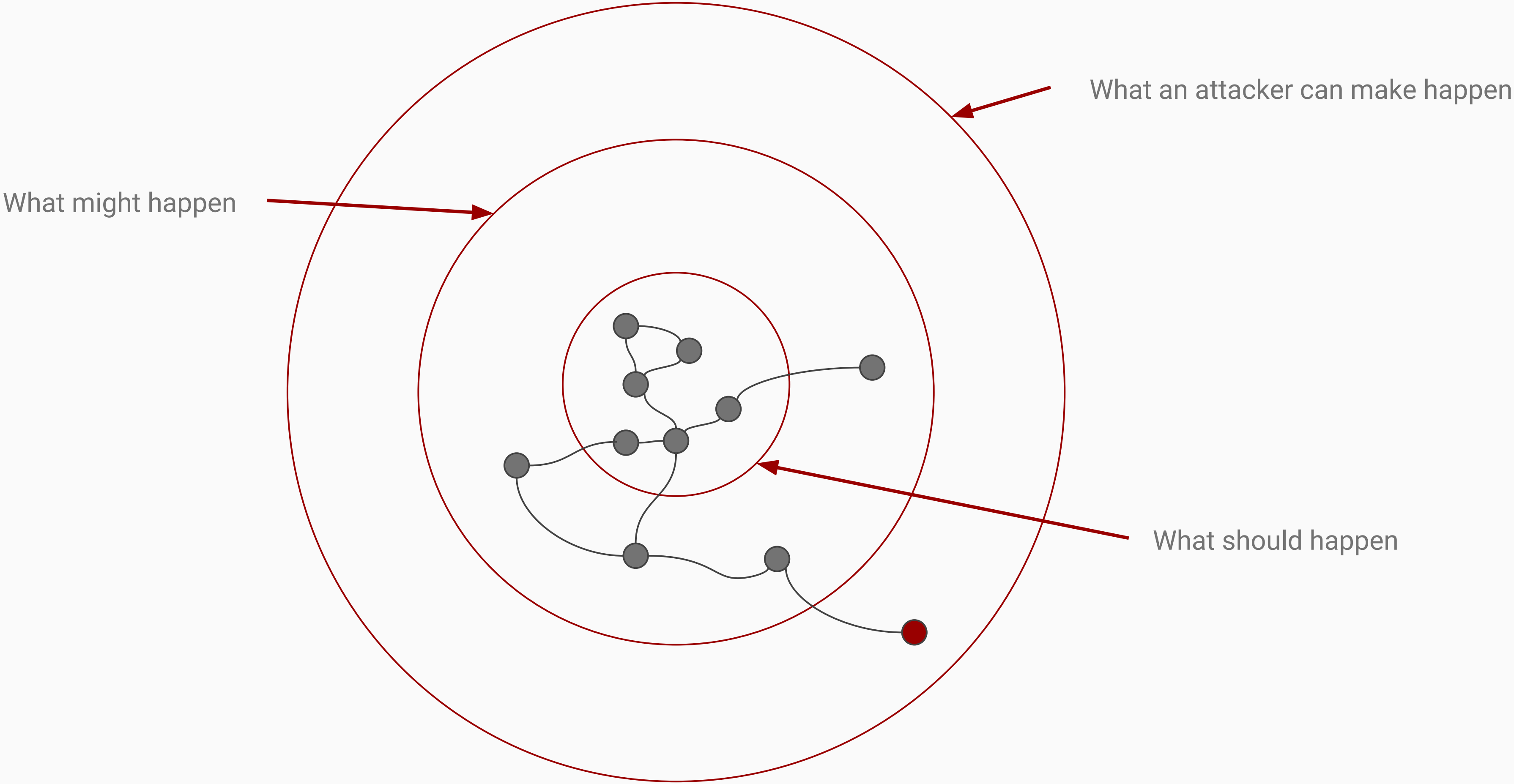
We will walk through some:

- Real bugs
- Real impacts

Then we will discuss:

- Interesting problems for you to solve
- Five basic tips for career success

How are bugs found?



What an attacker can make happen

What might happen

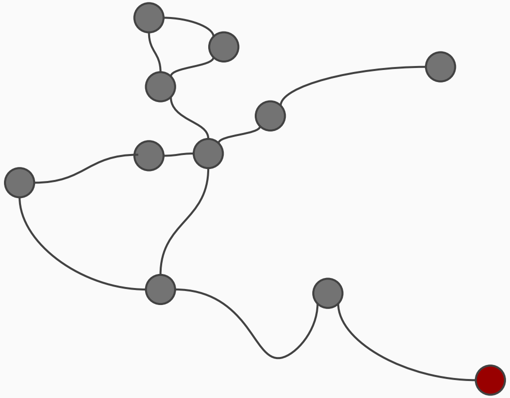
What should happen

Vulnerability Discovery is the art of...

- Pushing software into exploitable states
- Predicting the kinds of mistakes engineers will make and QA/security teams will miss
- Making the impossible possible

Fuzzing

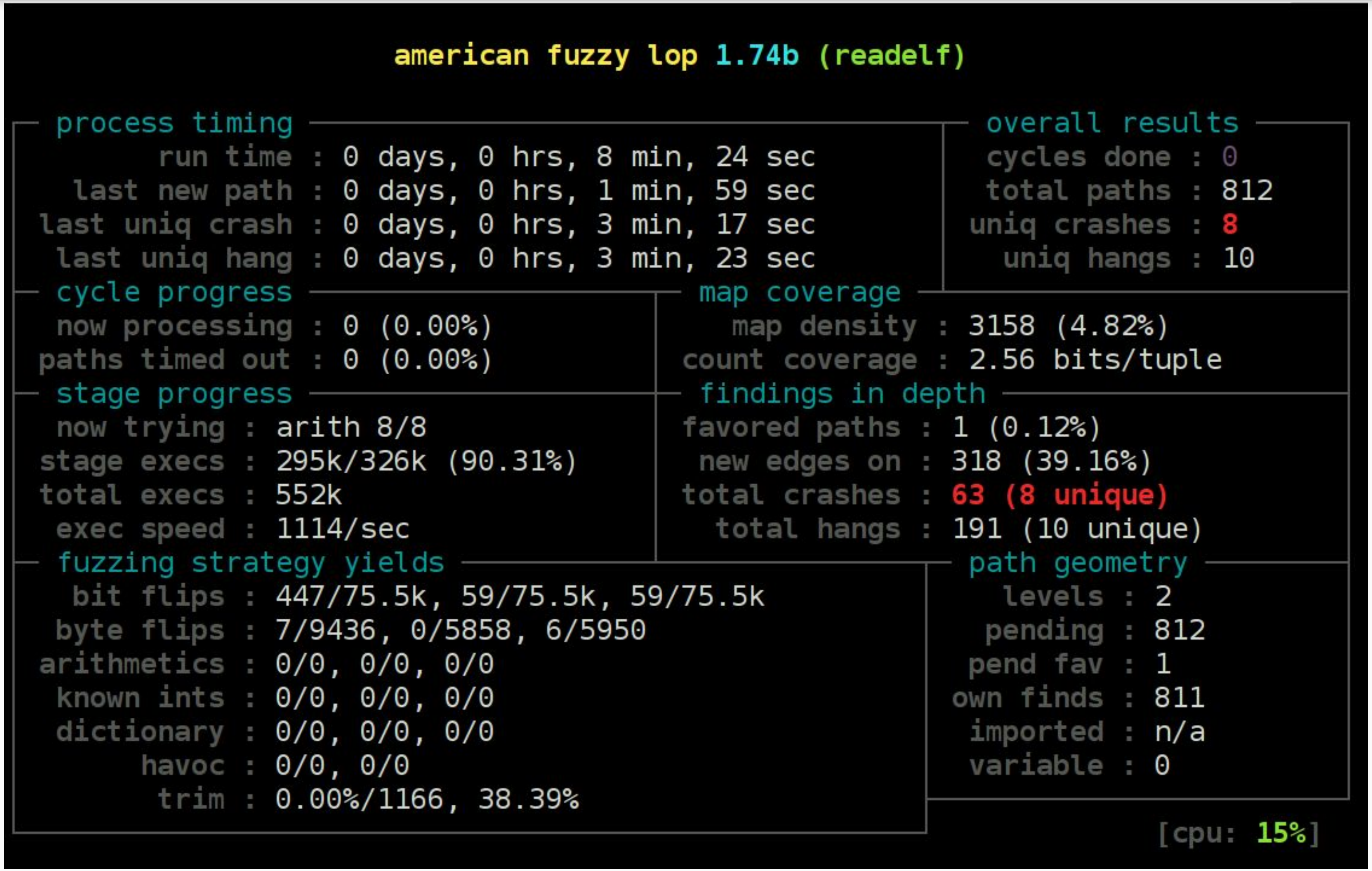
Using automation to mutate input into a system and look for exploitable states



Enhanced by:

- Intelligently unpacking, mutating, and re-packing formats
- Instrumenting the binary to accelerate input and look for caught exceptions
- Studying control-flow and intentionally hitting corner cases

Fuzzing



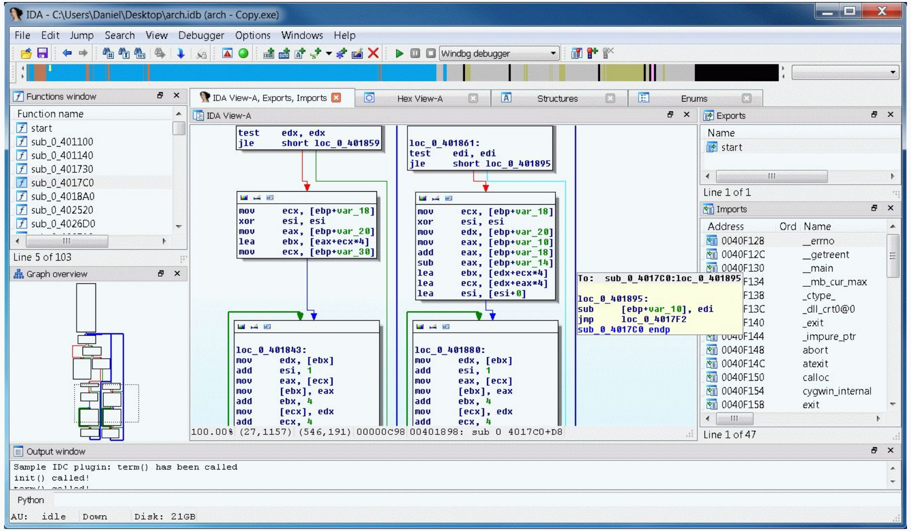
```
                    american fuzzy lop 1.74b (readelf)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 8 min, 24 sec      │  cycles done : 0      │
│   last new path : 0 days, 0 hrs, 1 min, 59 sec      │  total paths : 812    │
│ last uniq crash : 0 days, 0 hrs, 3 min, 17 sec      │ uniq crashes : 8      │
│  last uniq hang : 0 days, 0 hrs, 3 min, 23 sec      │   uniq hangs : 10     │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 3158 (4.82%)         │
│ paths timed out : 0 (0.00%)         │ count coverage : 2.56 bits/tuple      │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : arith 8/8             │ favored paths : 1 (0.12%)              │
│ stage execs : 295k/326k (90.31%)    │  new edges on : 318 (39.16%)           │
│ total execs : 552k                  │ total crashes : 63 (8 unique)          │
│  exec speed : 1114/sec              │   total hangs : 191 (10 unique)        │
├─ fuzzing strategy yields ───────────┴───────────────┬─ path geometry ────────┤
│   bit flips : 447/75.5k, 59/75.5k, 59/75.5k         │    levels : 2          │
│  byte flips : 7/9436, 0/5858, 6/5950                │   pending : 812        │
│ arithmetics : 0/0, 0/0, 0/0                         │  pend fav : 1          │
│  known ints : 0/0, 0/0, 0/0                         │ own finds : 811        │
│  dictionary : 0/0, 0/0, 0/0                         │  imported : n/a        │
│       havoc : 0/0, 0/0                              │  variable : 0          │
│        trim : 0.00%/1166, 38.39%                    ├────────────────────────┘
└─────────────────────────────────────────────────────┘             [cpu: 15%]
```

http://lcamtuf.coredump.cx/afl/

Reverse Engineering

Reverse engineering allows the researcher to:

- Find exploitable states and work backward
- Look for common antipatterns
- Understand and bypass sanity checks and protections


IDA - C:\Users\Daniel\Desktop\arch.idb (arch - Copy.exe)
File Edit Jump Search View Debugger Options Windows Help
Windbg debugger
Functions window
IDA View-A, Exports, Imports
Hex View-A
Structures
Enums
IDA View-A
Exports
Imports
Graph overview
Output window
Python

Function name
- start
- sub_0_401100
- sub_0_401140
- sub_0_401730
- sub_0_4017C0
- sub_0_4018A0
- sub_0_402520
- sub_0_4026D0

Line 5 of 103

```
test    edx, edx
jle     short loc_0_401859

mov     ecx, [ebp+var_18]
xor     esi, esi
mov     eax, [ebp+var_20]
lea     ebx, [eax+ecx*4]
mov     ecx, [ebp+var_30]

loc_0_401843:
mov     edx, [ebx]
add     esi, 1
mov     eax, [ecx]
mov     [ebx], eax
add     ebx, 4
mov     [ecx], edx
add     ecx, 4
```

```
loc_0_401861:
test    edi, edi
jle     short loc_0_401895

mov     ecx, [ebp+var_18]
xor     esi, esi
mov     edx, [ebp+var_20]
mov     eax, [ebp+var_10]
add     eax, [ebp+var_18]
sub     eax, [ebp+var_14]
lea     ebx, [edx+ecx*4]
lea     ecx, [edx+eax*4]
lea     esi, [esi+0]

loc_0_401880:
mov     edx, [ebx]
add     esi, 1
mov     eax, [ecx]
mov     [ebx], eax
add     ebx, 4
mov     [ecx], edx
add     ecx, 4
```

```
To:  sub_0_4017C0:loc_0_401895

loc_0_401895:
sub     [ebp+var_10], edi
jmp     loc_0_4017F2
sub_0_4017C0 endp
```

```
100.00% (27,1157) (546,191) 00000C98 00401898: sub 0 4017C0+D8
```

Name
- start

Line 1 of 1

Address	Ord	Name
0040F128		_errno
0040F12C		_getreent
0040F130		_main
F134		_mb_cur_max
F138		_ctype_
F13C		_dll_crt0@0
F140		_exit
0040F144		_impure_ptr
0040F148		abort
0040F14C		atexit
0040F150		calloc
0040F154		cygwin_internal
0040F158		exit

Line 1 of 47

```
Sample IDC plugin: term() has been called
init() called!
term() called!
```

```
AU:  idle   Down    Disk: 21GB
```

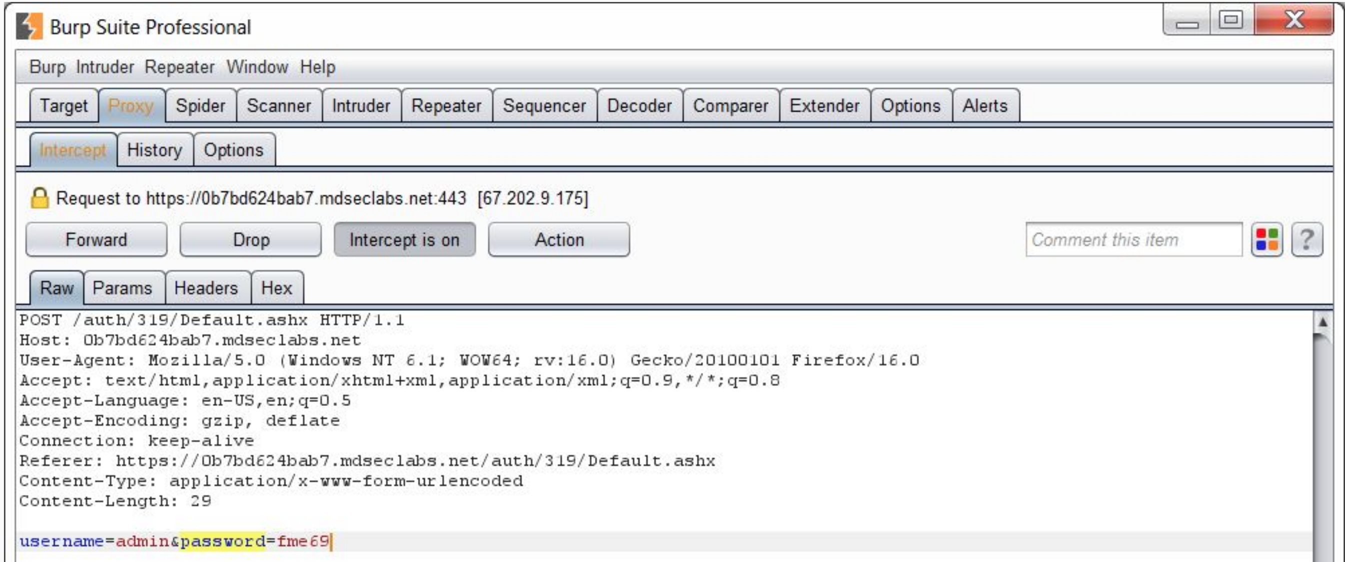
Includes:

- Debugging
- Disassembly
- Binary diffing
- Decompilation

Manual Manipulation

- Many interesting flaws boil down to asking the software to do something
- Due to:
  - Confused deputy problems
  - Missing access control checks
  - Lack of data consistency checks

- Often using tools to intercept and manipulate inputs



Burp Suite Professional
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Intercept History Options
Forward
Drop
Intercept is on
Action
Comment this item
?
Raw Params Headers Hex

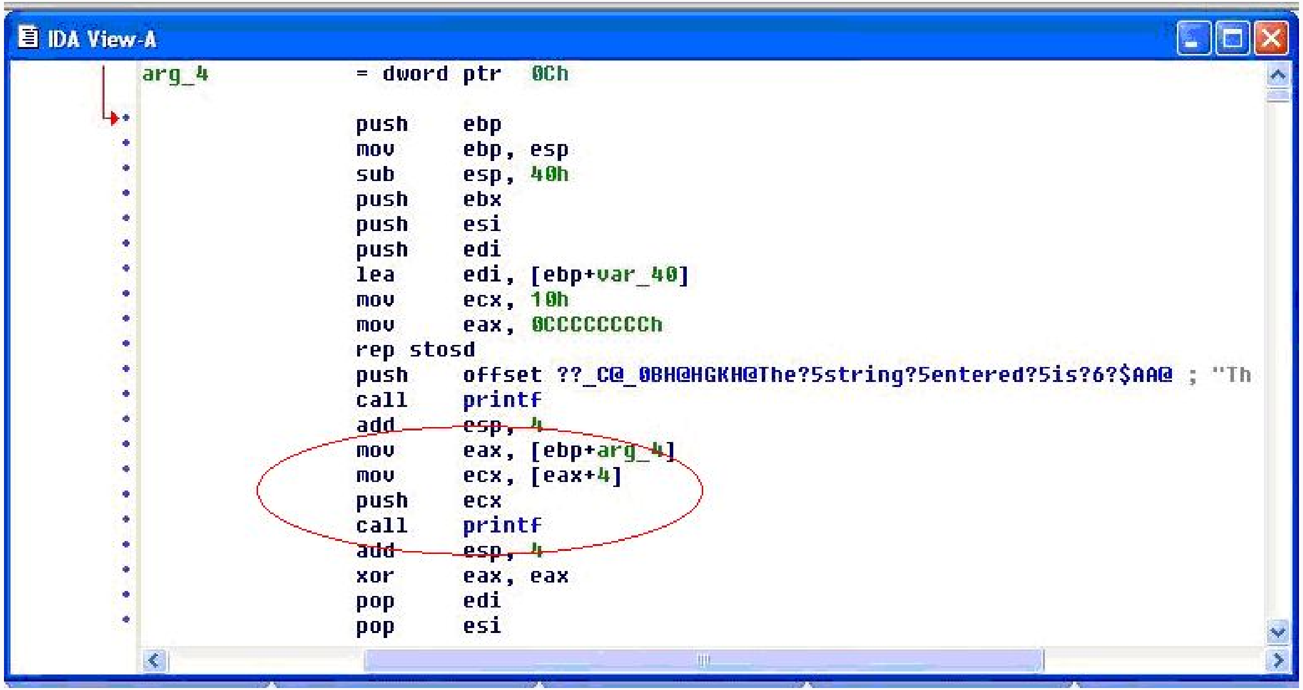
Request to https://0b7bd624bab7.mdseclabs.net:443 [67.202.9.175]

```
POST /auth/319/Default.ashx HTTP/1.1
Host: 0b7bd624bab7.mdseclabs.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/20100101 Firefox/16.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: https://0b7bd624bab7.mdseclabs.net/auth/319/Default.ashx
Content-Type: application/x-www-form-urlencoded
Content-Length: 29

username=admin&password=fme69
```

Pulling it Together

Professional bug hunters often pull many techniques together:

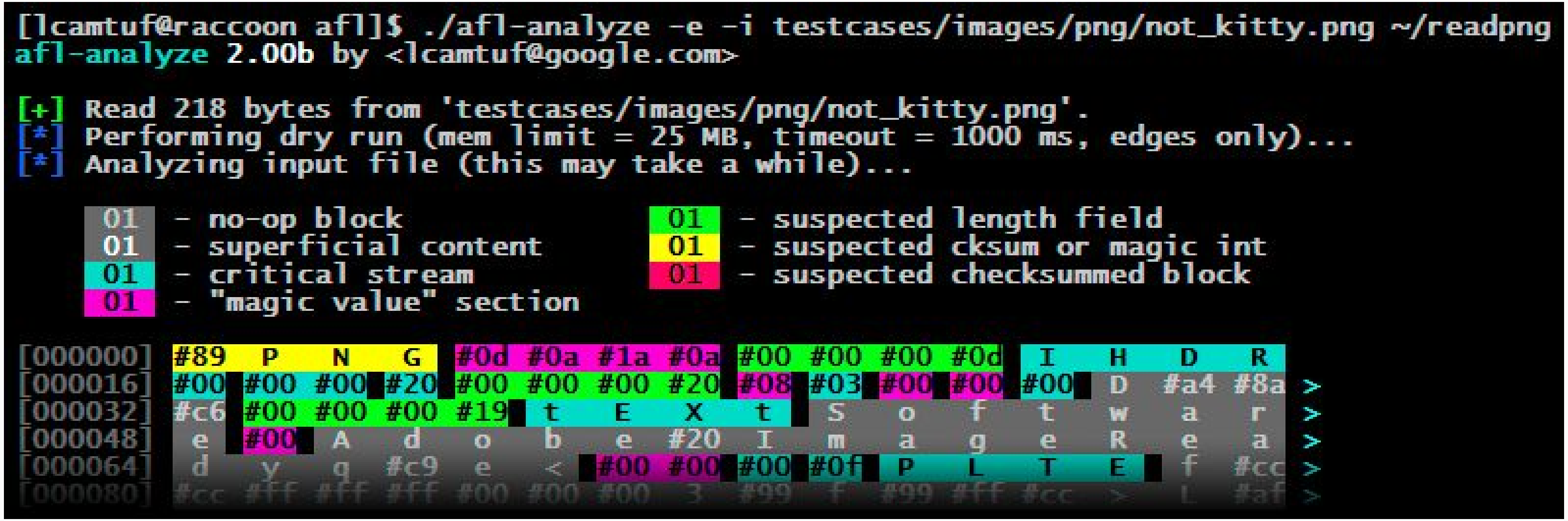
1. Disassemble a binary to discover:



```
arg_4           = dword ptr  0Ch

                push    ebp
                mov     ebp, esp
                sub     esp, 40h
                push    ebx
                push    esi
                push    edi
                lea     edi, [ebp+var_40]
                mov     ecx, 10h
                mov     eax, 0CCCCCCCCh
                rep stosd
                push    offset ??_C@_0BH@HGKH@The?5string?5entered?5is?6?$AA@ ; "Th
                call    printf
                add     esp, 4
                mov     eax, [ebp+arg_4]
                mov     ecx, [eax+4]
                push    ecx
                call    printf
                add     esp, 4
                xor     eax, eax
                pop     edi
                pop     esi
```

Pulling it Together

2. Use format-aware fuzzing to try to find entry points that lead to format string

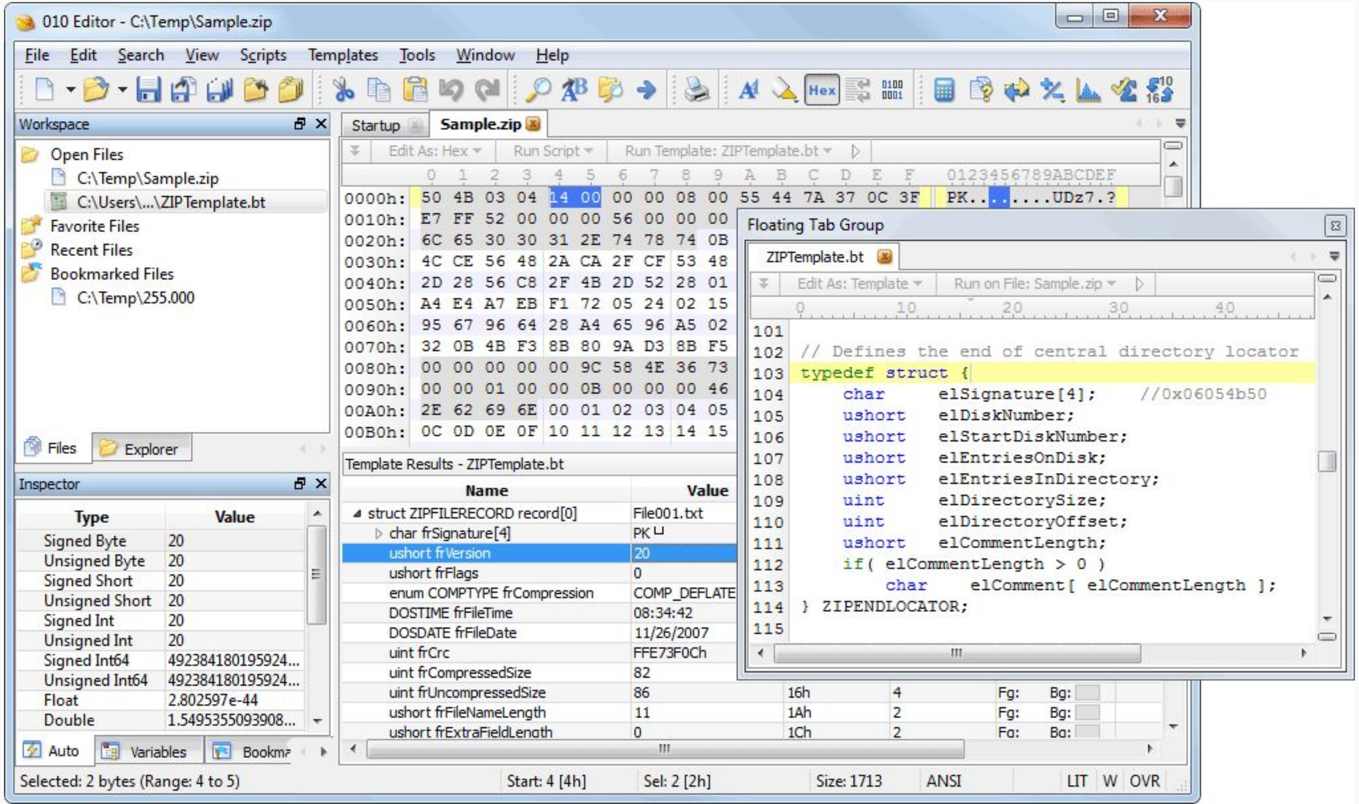


```
[lcamtuf@raccoon afl]$ ./afl-analyze -e -i testcases/images/png/not_kitty.png ~/readpng
afl-analyze 2.00b by <lcamtuf@google.com>

[+] Read 218 bytes from 'testcases/images/png/not_kitty.png'.
[*] Performing dry run (mem limit = 25 MB, timeout = 1000 ms, edges only)...
[*] Analyzing input file (this may take a while)...

    01  - no-op block                  01  - suspected length field
    01  - superficial content          01  - suspected cksum or magic int
    01  - critical stream              01  - suspected checksummed block
    01  - "magic value" section

[000000] #89  P   N   G  #0d #0a #1a #0a #00 #00 #00 #0d  I   H   D   R
[000016] #00 #00 #00 #20 #00 #00 #00 #20 #08 #03 #00 #00 #00  D  #a4 #8a >
[000032] #c6 #00 #00 #00 #19  t   E   X   t       S   o   f   t   w   a   r >
[000048]  e  #00  A   d   o   b   e  #20  I   m   a   g   e   R   e   a >
[000064]  d   y   q  #c9  e   <  #00 #00 #00 #0f  P   L   T   E   f  #cc >
[000080] #cc #ff #ff #ff #00 #00 #00  3  #99  f  #99 #ff #cc  >   L  #af >
```

https://lcamtuf.blogspot.com/2016/02/say-hello-to-afl-analyze.html

Pulling it Together

3. Researcher carefully modifies crash-creating documents by the fuzzer to obtain execution



```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 50 4B 03 04 14 00 00 00 08 00 55 44 7A 37 0C 3F  PK........UDz7.?
0010h: E7 FF 52 00 00 00 56 00 00 00
0020h: 6C 65 30 30 31 2E 74 78 74 0B
0030h: 4C CE 56 48 2A CA 2F CF 53 48
0040h: 2D 28 56 C8 2F 4B 2D 52 28 01
0050h: A4 E4 A7 EB F1 72 05 24 02 15
0060h: 95 67 96 64 28 A4 65 96 A5 02
0070h: 32 0B 4B F3 8B 80 9A D3 8B F5
0080h: 00 00 00 00 00 9C 58 4E 36 73
0090h: 00 00 01 00 00 0B 00 00 00 46
00A0h: 2E 62 69 6E 00 01 02 03 04 05
00B0h: 0C 0D 0E 0F 10 11 12 13 14 15
```

```
// Defines the end of central directory locator
typedef struct {
    char      elSignature[4];     //0x06054b50
    ushort    elDiskNumber;
    ushort    elStartDiskNumber;
    ushort    elEntriesOnDisk;
    ushort    elEntriesInDirectory;
    uint      elDirectorySize;
    uint      elDirectoryOffset;
    ushort    elCommentLength;
    if( elCommentLength > 0 )
        char    elComment[ elCommentLength ];
} ZIPENDLOCATOR;
```

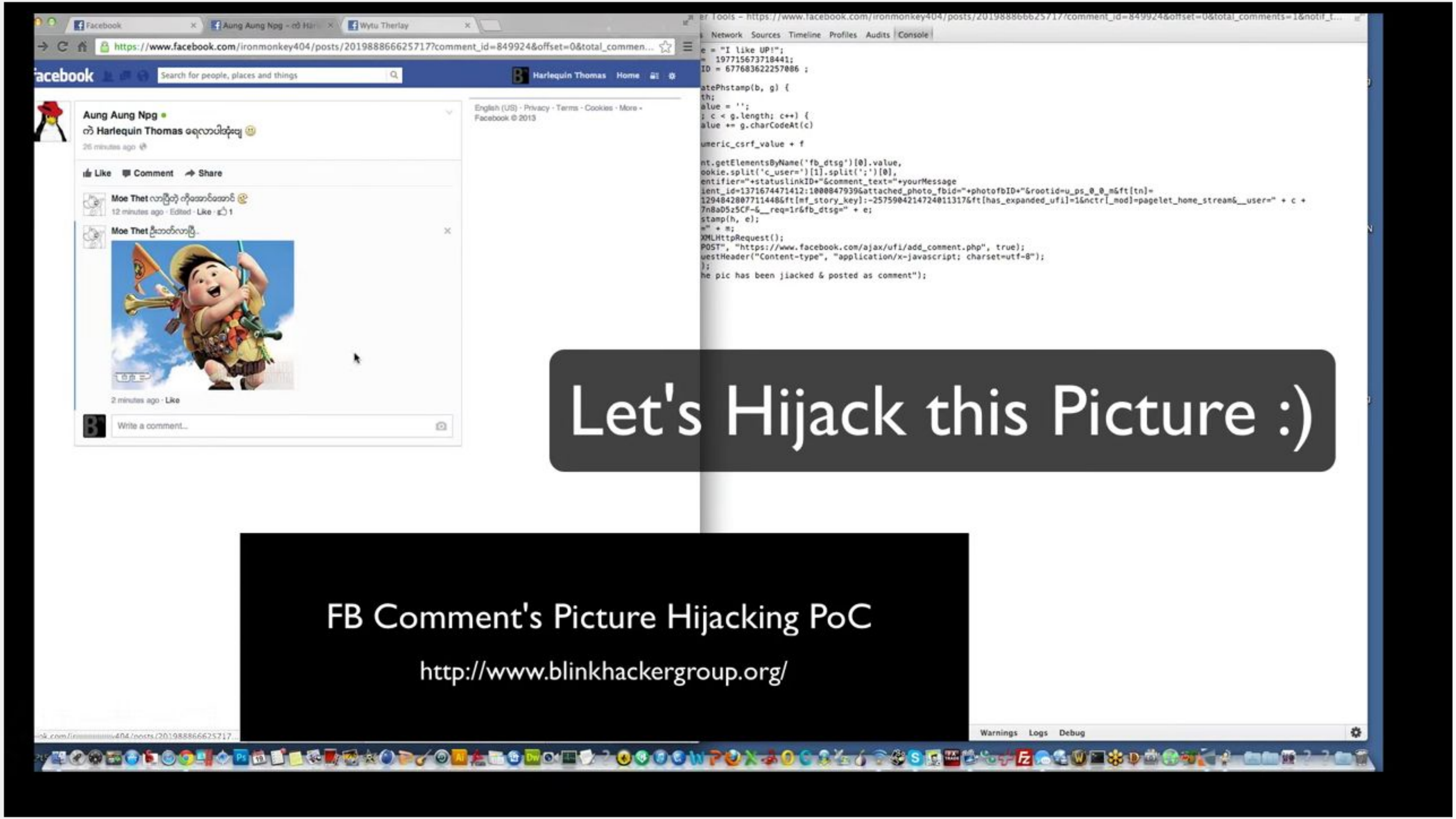
Type	Value
Signed Byte	20
Unsigned Byte	20
Signed Short	20
Unsigned Short	20
Signed Int	20
Unsigned Int	20
Signed Int64	492384180195924...
Unsigned Int64	492384180195924...
Float	2.802597e-44
Double	1.5495355093908...

Template Results - ZIPTemplate.bt

Name	Value				
struct ZIPFILERECORD record[0]	File001.txt				
char frSignature[4]	PK ⊔				
ushort frVersion	20				
ushort frFlags	0				
enum COMPTYPE frCompression	COMP_DEFLATE				
DOSTIME frFileTime	08:34:42				
DOSDATE frFileDate	11/26/2007				
uint frCrc	FFE73F0Ch				
uint frCompressedSize	82				
uint frUncompressedSize	86	16h	4	Fg:	Bg:
ushort frFileNameLength	11	1Ah	2	Fg:	Bg:
ushort frExtraFieldLength	0	1Ch	2	Fg:	Bg:

Selected: 2 bytes (Range: 4 to 5) Start: 4 [4h] Sel: 2 [2h] Size: 1713 ANSI LIT W OVR

Real World Bugs



Facebook | Aung Aung Npg - ကဲ Har | Wytu Therlay

https://www.facebook.com/ironmonkey404/posts/201988866625717?comment_id=849924&offset=0&total_commen...

Aung Aung Npg

ကဲ Harlequin Thomas ရေလာပါအုံးဗျ

26 minutes ago

Moe Thet လာပြီတဲ့ ကိုအောင်အောင်

12 minutes ago · Edited · Like · 1

Moe Thet ဦးဘတ်လာပြီ..

2 minutes ago · Like

English (US) · Privacy · Terms · Cookies · More ·
Facebook © 2013

er Tools - https://www.facebook.com/ironmonkey404/posts/201988866625717?comment_id=849924&offset=0&total_comments=1&notif_t...

```
e = "I like UP!";
=  197715673718441;
ID = 677683622257086 ;

atePhstamp(b, g) {
th;
alue = '';
; c < g.length; c++) {
alue += g.charCodeAt(c)

umeric_csrf_value + f

nt.getElementsByName('fb_dtsg')[0].value,
ookie.split('c_user=')[1].split(';')[0],
entifier="+statuslinkID+"&comment_text="+yourMessage
ient_id=1371674471412:1000847939&attached_photo_fbid="+photofbID+"&rootid=u_ps_0_0_m&ft[tn]=
1294842807711448&ft[mf_story_key]:-2575904214724011317&ft[has_expanded_ufi]=1&nctr[_mod]=pagelet_home_stream&__user=" + c +
7n8aD5z5CF-&__req=1r&fb_dtsg=" + e;
stamp(h, e);
=" + m;
XMLHttpRequest();
POST", "https://www.facebook.com/ajax/ufi/add_comment.php", true);
uestHeader("Content-type", "application/x-javascript; charset=utf-8");
);
he pic has been jiacked & posted as comment");
```

Let's Hijack this Picture :)

FB Comment's Picture Hijacking PoC

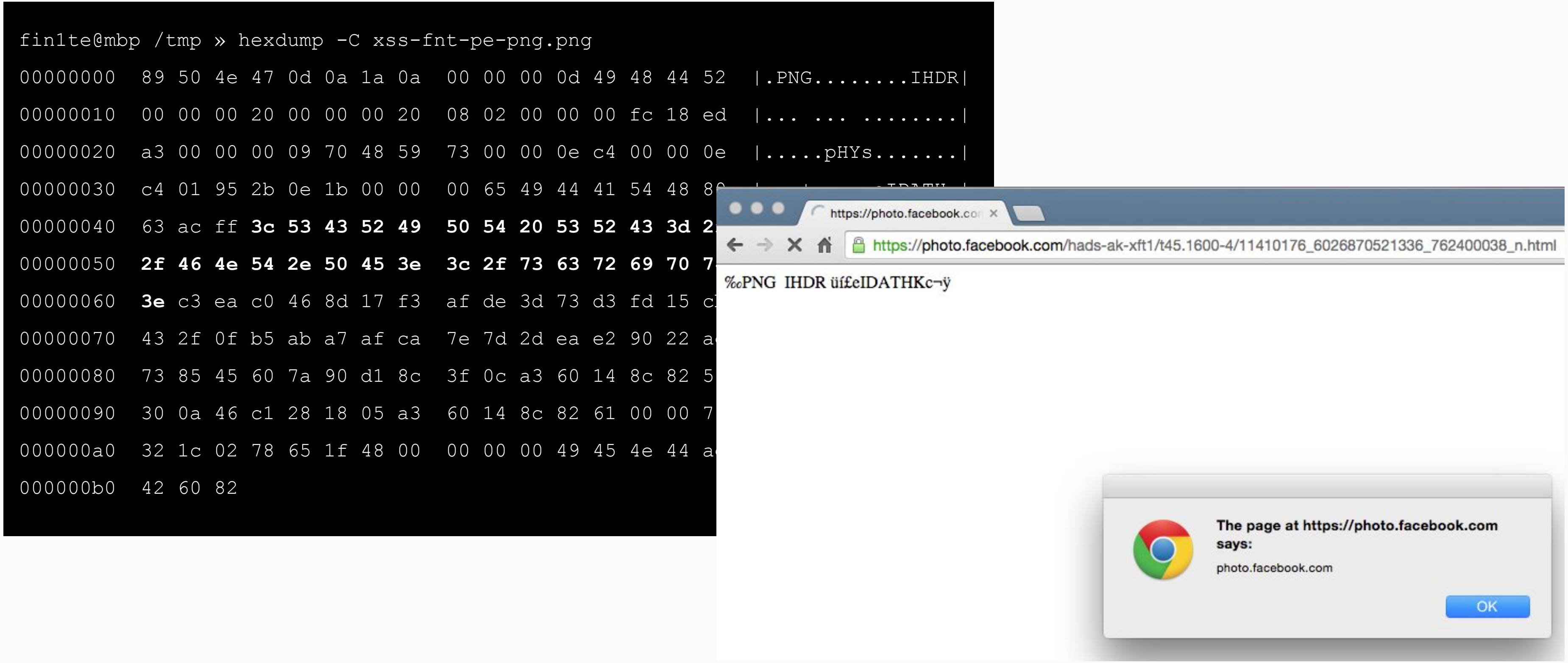
http://www.blinkhackergroup.org/

Facebook Picture Sharing on Comment Exploit

Apple’s TLS Code

```
hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
hashOut.length = SSL_SHA1_DIGEST_LEN;
if ((err = SSLFreeBuffer(&hashCtx)) != 0)
 goto fail;
if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
 goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
 goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
 goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
 goto fail;
 goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
 goto fail;
err = sslRawVerify(...);
```

Embedding Script in Images



```
fin1te@mbp /tmp » hexdump -C xss-fnt-pe-png.png
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG........IHDR|
00000010  00 00 00 20 00 00 00 20  08 02 00 00 00 fc 18 ed  |... ... ........|
00000020  a3 00 00 00 09 70 48 59  73 00 00 0e c4 00 00 0e  |.....pHYs.......|
00000030  c4 01 95 2b 0e 1b 00 00  00 65 49 44 41 54 48 80
00000040  63 ac ff 3c 53 43 52 49  50 54 20 53 52 43 3d 2
00000050  2f 46 4e 54 2e 50 45 3e  3c 2f 73 63 72 69 70 7
00000060  3e c3 ea c0 46 8d 17 f3  af de 3d 73 d3 fd 15 c
00000070  43 2f 0f b5 ab a7 af ca  7e 7d 2d ea e2 90 22 a
00000080  73 85 45 60 7a 90 d1 8c  3f 0c a3 60 14 8c 82 5
00000090  30 0a 46 c1 28 18 05 a3  60 14 8c 82 61 00 00 7
000000a0  32 1c 02 78 65 1f 48 00  00 00 00 49 45 4e 44 a
000000b0  42 60 82
```

https://photo.facebook.com/hads-ak-xft1/t45.1600-4/11410176_6026870521336_762400038_n.html

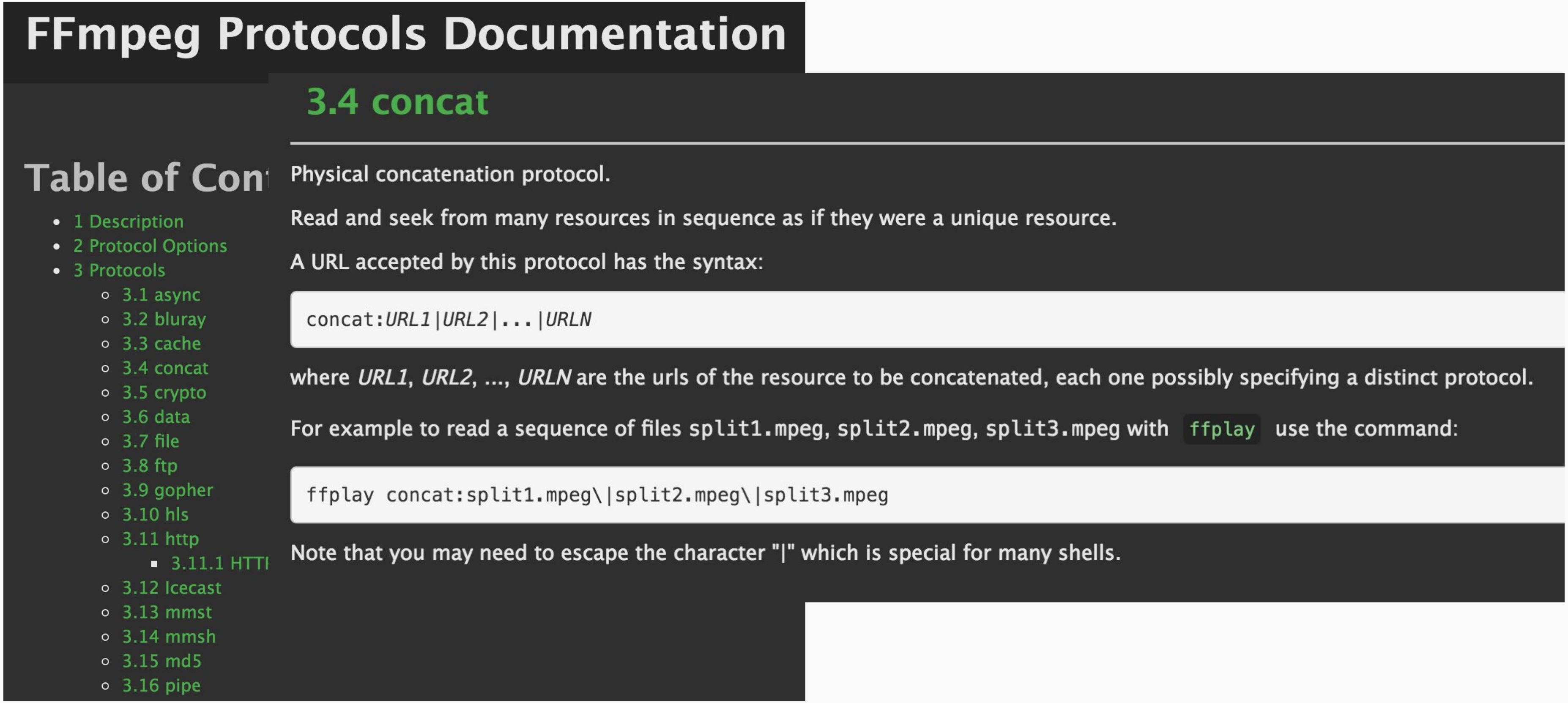
‰PNG IHDR üí£cIDATHKc¬ÿ

The page at https://photo.facebook.com says:

photo.facebook.com

https://whitton.io/articles/xss-on-facebook-via-png-content-types/

Bug or feature?



FFmpeg Protocols Documentation

Table of Cont

3.4 concat

Physical concatenation protocol.

Read and seek from many resources in sequence as if they were a unique resource.

A URL accepted by this protocol has the syntax:

```
concat:URL1|URL2|...|URLN
```

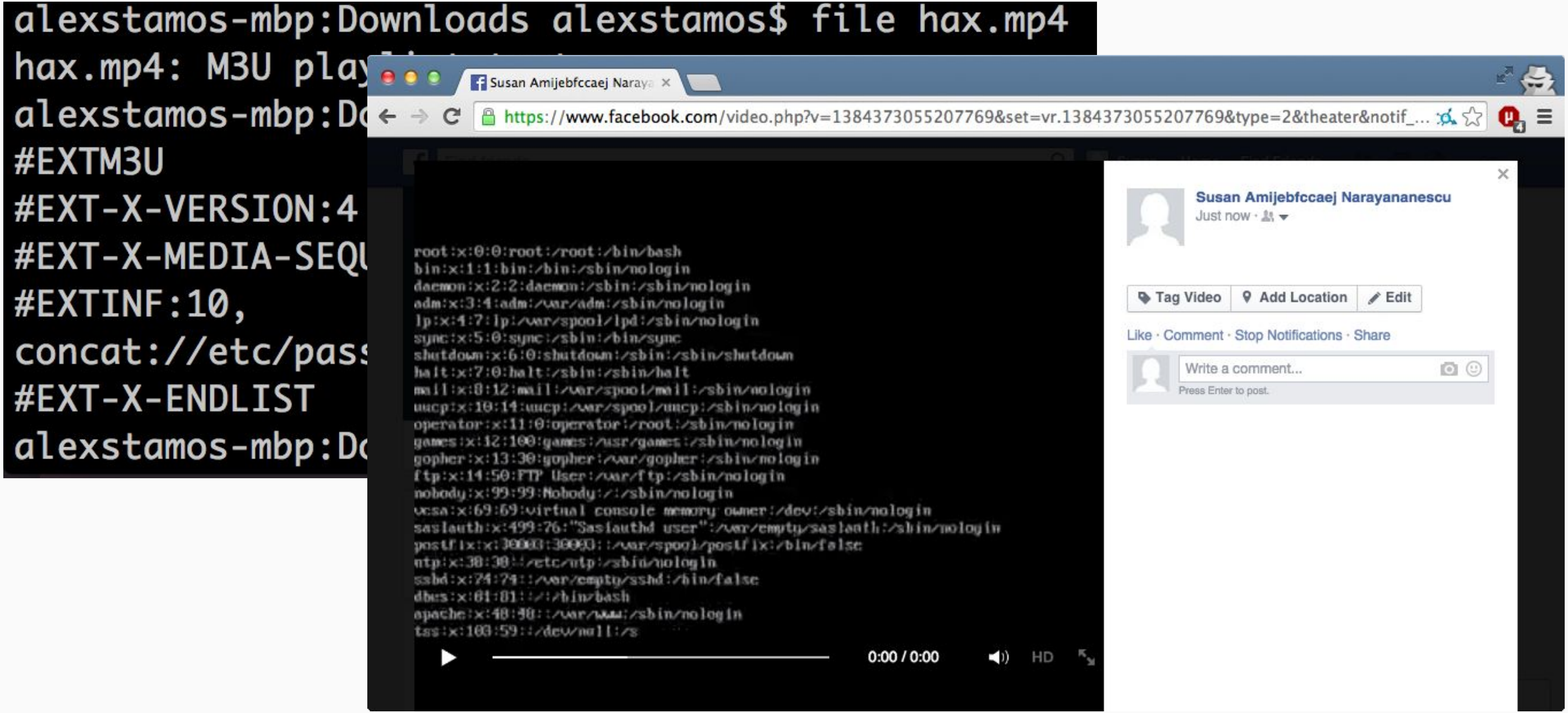
where *URL1*, *URL2*, ..., *URLN* are the urls of the resource to be concatenated, each one possibly specifying a distinct protocol.

For example to read a sequence of files `split1.mpeg`, `split2.mpeg`, `split3.mpeg` with `ffplay` use the command:

```
ffplay concat:split1.mpeg\|split2.mpeg\|split3.mpeg
```

Note that you may need to escape the character "|" which is special for many shells.

Bug or feature?



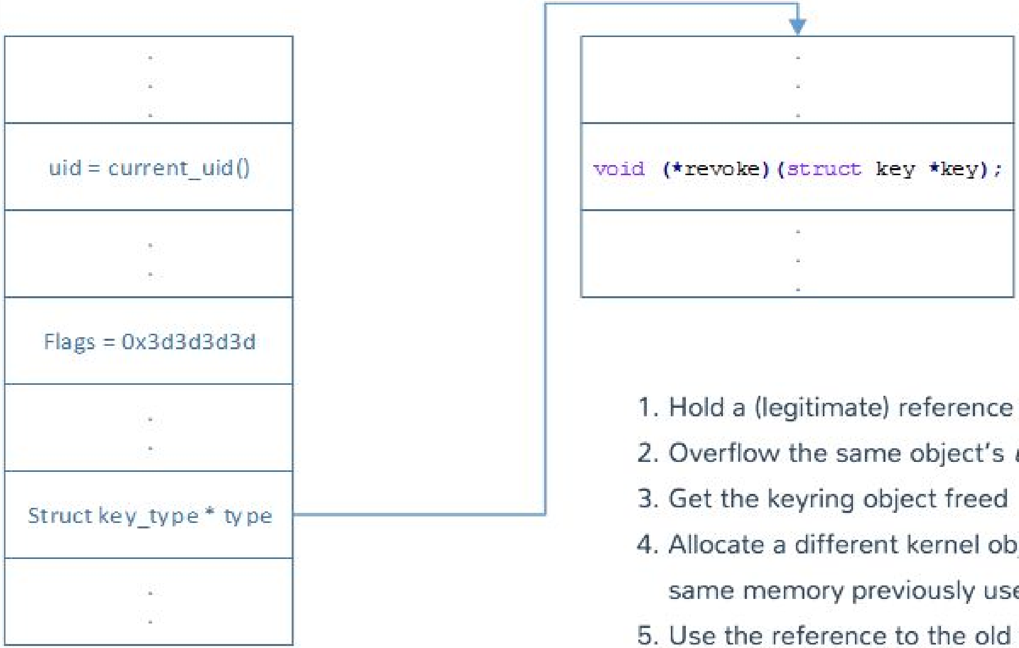
```
alexstamos-mbp:Downloads alexstamos$ file hax.mp4
hax.mp4: M3U play
alexstamos-mbp:Do
#EXTM3U
#EXT-X-VERSION:4
#EXT-X-MEDIA-SEQU
#EXTINF:10,
concat://etc/pas
#EXT-X-ENDLIST
alexstamos-mbp:Do
```

Susan Amijebfccaej Naraya ×
https://www.facebook.com/video.php?v=1384373055207769&set=vr.1384373055207769&type=2&theater&notif_...

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
saslauth:x:499:76:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
postfix:x:30000:30000::/var/spool/postfix:/bin/false
ntp:x:38:38::/etc/ntp:/sbin/nologin
sshd:x:74:74::/var/empty/sshd:/bin/false
dbus:x:81:81::/:/bin/bash
apache:x:48:48::/var/www:/sbin/nologin
tss:x:100:59::/dev/null:/s
```

Susan Amijebfccaej Narayananescu
Just now ·

Memory Management



uid = current_uid()

Flags = 0x3d3d3d3d

Struct key_type * type

```
void (*revoke)(struct key *key);
```

1. Hold a (legitimate) reference to a key object
2. Overflow the same object's *usage*
3. Get the keyring object freed
4. Allocate a different kernel object from user-space, with a user-controlled content, over the same memory previously used by the freed keyring object
5. Use the reference to the old key object and trigger code execution

http://perception-point.io/2016/01/14/analysis-and-exploitation-of-a-linux-kernel-vulnerability-cve-2016-0728/

Who Finds Bugs?

Who Looks for Bugs?

Defenders:

- Have benefit of source code, access to engineers
- Target 100% coverage, so broad-and-shallow testing is common
- Generally need automation to assist

Attackers:

- Have less information, not a huge problem with shipped code
- Only need a handful of flaws to chain them together
- Need to find and explore issues without alerting defenders

Researchers:

- Various motivations. Money? Fame?
- Lots of ethical reporting options via bug bounties
- Generally want to stay on right side of the law

Real World Defense

Real World Defense Should Focus on...

… **Real World Problems** (for securing people)

Three biggest problems for most people:

1. Compromised reused passwords
2. Phishing credentials
3. Common, n-day malware

… **Real World Attackers** (for securing enterprises)

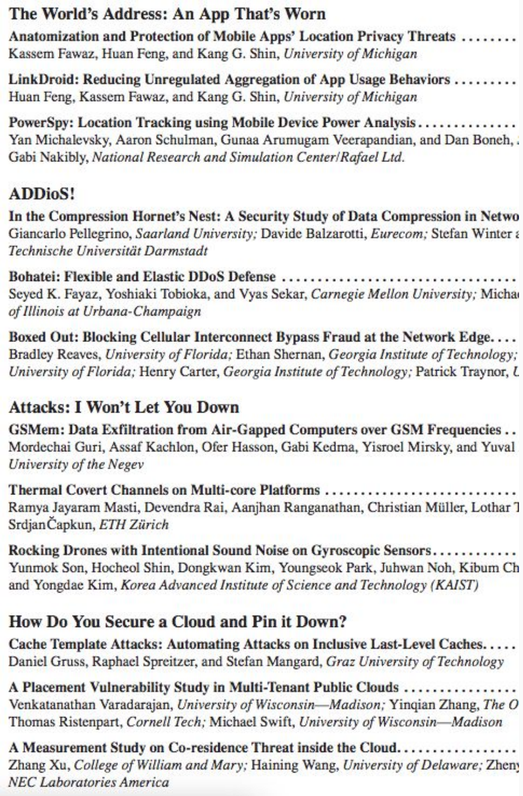
Namely, capabilities, tools, techniques and procedures for the intrusion kill chain:

1. Reconnaissance
2. Weaponization
3. Delivery
4. Exploitation
5. Installation
6. Command and Control
7. Actions on Objectives

http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf

Security research often misses the point

The incentives for private and academic research point the wrong way:



The World's Address: An App That's Worn

Anatomization and Protection of Mobile Apps' Location Privacy Threats . . . .
Kassem Fawaz, Huan Feng, and Kang G. Shin, *University of Michigan*

LinkDroid: Reducing Unregulated Aggregation of App Usage Behaviors . . . .
Huan Feng, Kassem Fawaz, and Kang G. Shin, *University of Michigan*

PowerSpy: Location Tracking using Mobile Device Power Analysis . . . .
Yan Michalevsky, Aaron Schulman, Gunaa Arumugam Veerapandian, and Dan Boneh,
Gabi Nakibly, *National Research and Simulation Center/Rafael Ltd.*

ADDioS!

In the Compression Hornet's Nest: A Security Study of Data Compression in Netwo
Giancarlo Pellegrino, *Saarland University;* Davide Balzarotti, *Eurecom;* Stefan Winter
Technische Universität Darmstadt

Bohatei: Flexible and Elastic DDoS Defense . . . .
Seyed K. Fayaz, Yoshiaki Tobioka, and Vyas Sekar, *Carnegie Mellon University;* Micha
of Illinois at Urbana-Champaign

Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. . . .
Bradley Reaves, *University of Florida;* Ethan Shernan, *Georgia Institute of Technology;*
University of Florida; Henry Carter, *Georgia Institute of Technology;* Patrick Traynor,

Attacks: I Won't Let You Down

GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies . .
Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval
University of the Negev

Thermal Covert Channels on Multi-core Platforms . . . .
Ramya Jayaram Masti, Devendra Rai, Aanjhan Ranganathan, Christian Müller, Lothar
SrdjanČapkun, *ETH Zürich*

Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors. . . . .
Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Ch
and Yongdae Kim, *Korea Advanced Institute of Science and Technology (KAIST)*

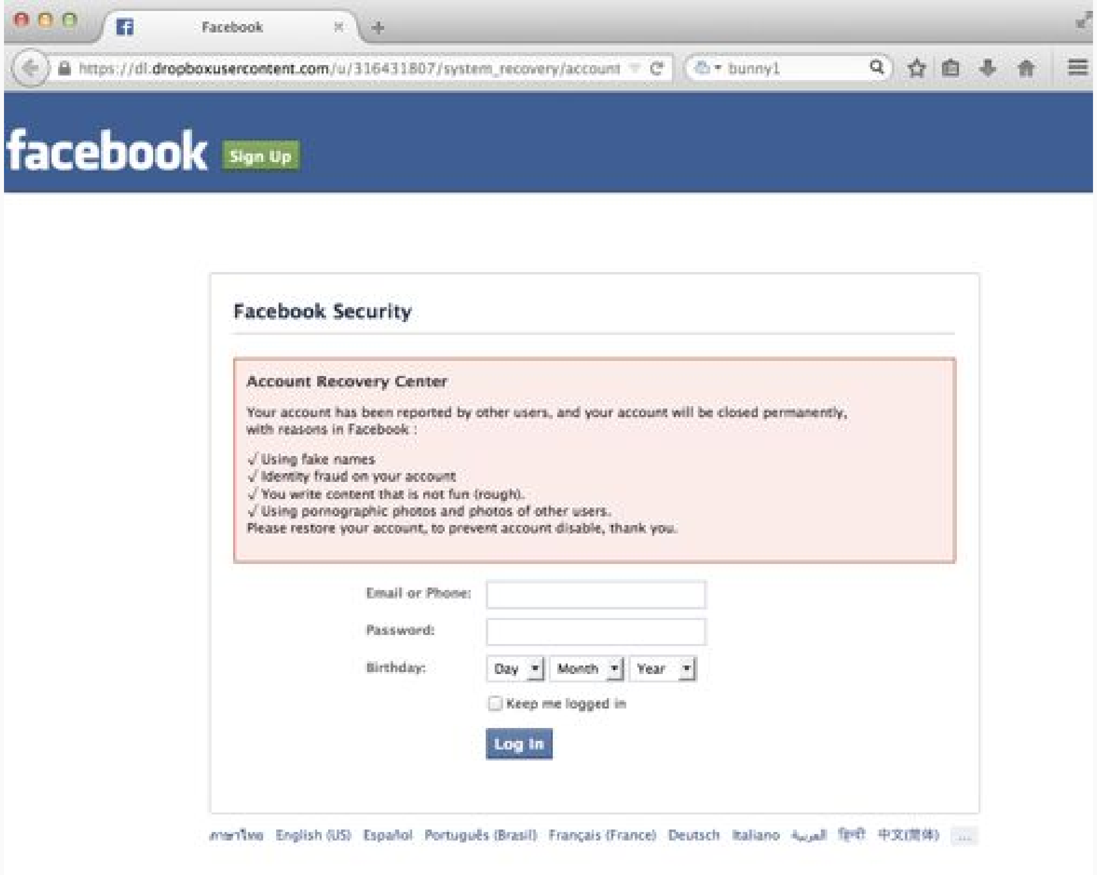
How Do You Secure a Cloud and Pin it Down?

Cache Template Attacks: Automating Attacks on Inclusive Last-Level Caches. . . . .
Daniel Gruss, Raphael Spreitzer, and Stefan Mangard, *Graz University of Technology*

A Placement Vulnerability Study in Multi-Tenant Public Clouds . . . .
Venkatanathan Varadarajan, *University of Wisconsin—Madison;* Yinqian Zhang, *The O*
Thomas Ristenpart, *Cornell Tech;* Michael Swift, *University of Wisconsin—Madison*

A Measurement Study on Co-residence Threat inside the Cloud. . . . .
Zhang Xu, *College of William and Mary;* Haining Wang, *University of Delaware;* Zhen
NEC Laboratories America

What causes the most problems for normal users?


Facebook
https://dl.dropboxusercontent.com/u/316431807/system_recovery/account
bunny1
facebook
Sign Up

Facebook Security

Account Recovery Center

Your account has been reported by other users, and your account will be closed permanently,
with reasons in Facebook :

√ Using fake names
√ Identity fraud on your account
√ You write content that is not fun (rough).
√ Using pornographic photos and photos of other users.
Please restore your account, to prevent account disable, thank you.

Email or Phone:

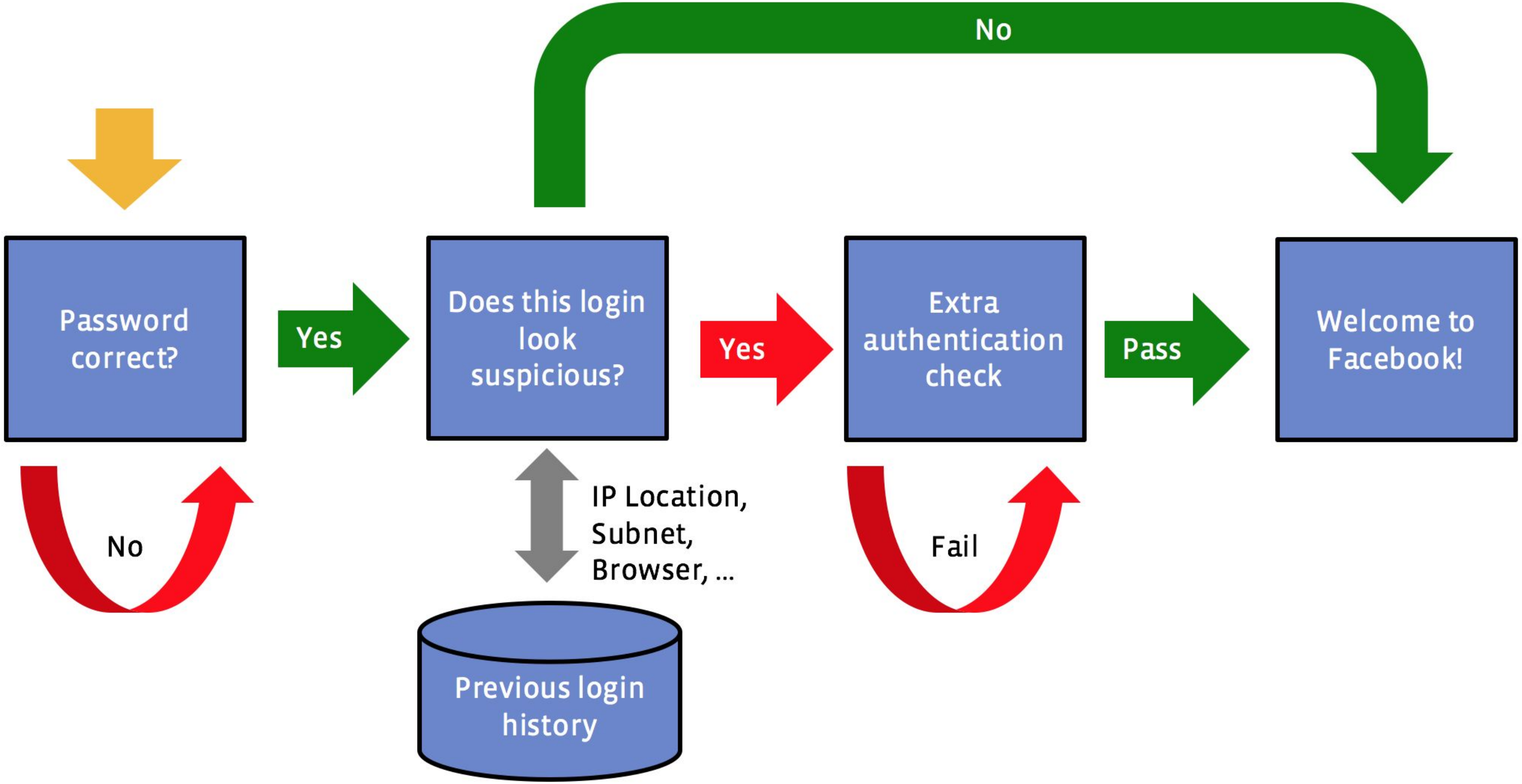
Password:

Birthday: Day Month Year

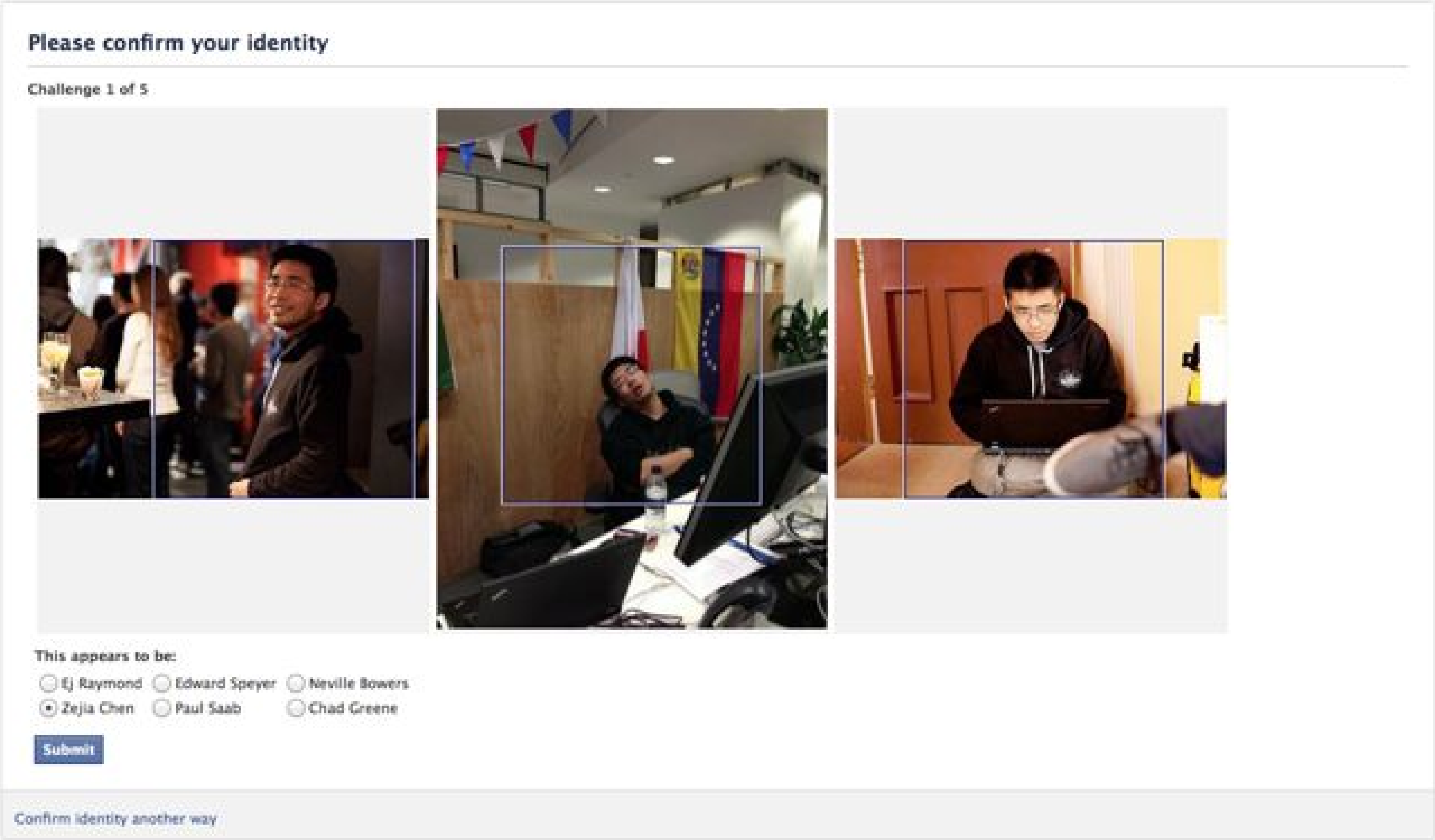
□ Keep me logged in

Log In
ภาษาไทย English (US) Español Português (Brasil) Français (France) Deutsch Italiano العربية हिन्दी 中文(简体) ...

Real World Defense - Logins


No
Password correct?
Yes
Does this login look suspicious?
Yes
Extra authentication check
Pass
Welcome to Facebook!
No
IP Location, Subnet, Browser, ...
Fail
Previous login history

Supplemental Authentication



Please confirm your identity

Challenge 1 of 5

This appears to be:

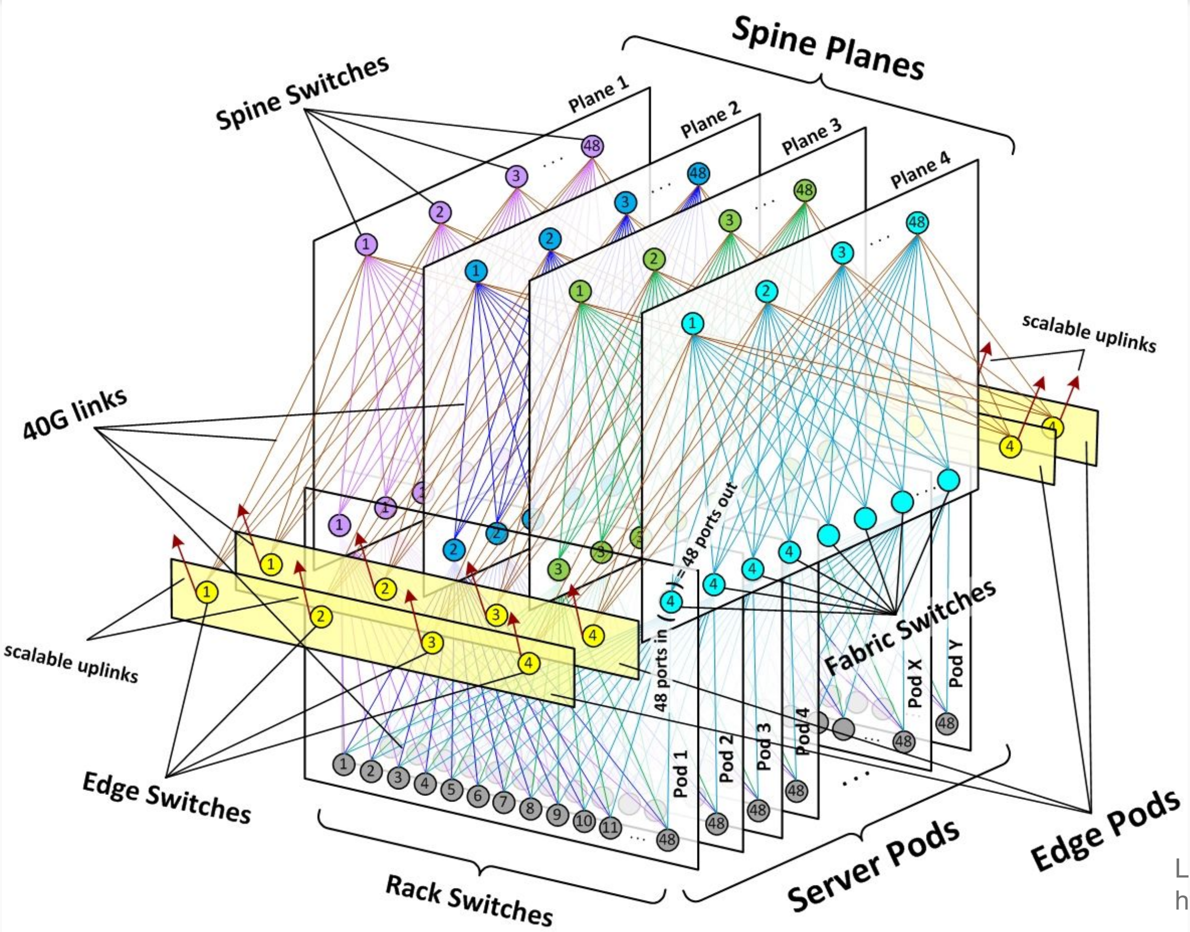
○ Ej Raymond ○ Edward Speyer ○ Neville Bowers

● Zejia Chen ○ Paul Saab ○ Chad Greene

Confirm identity another way

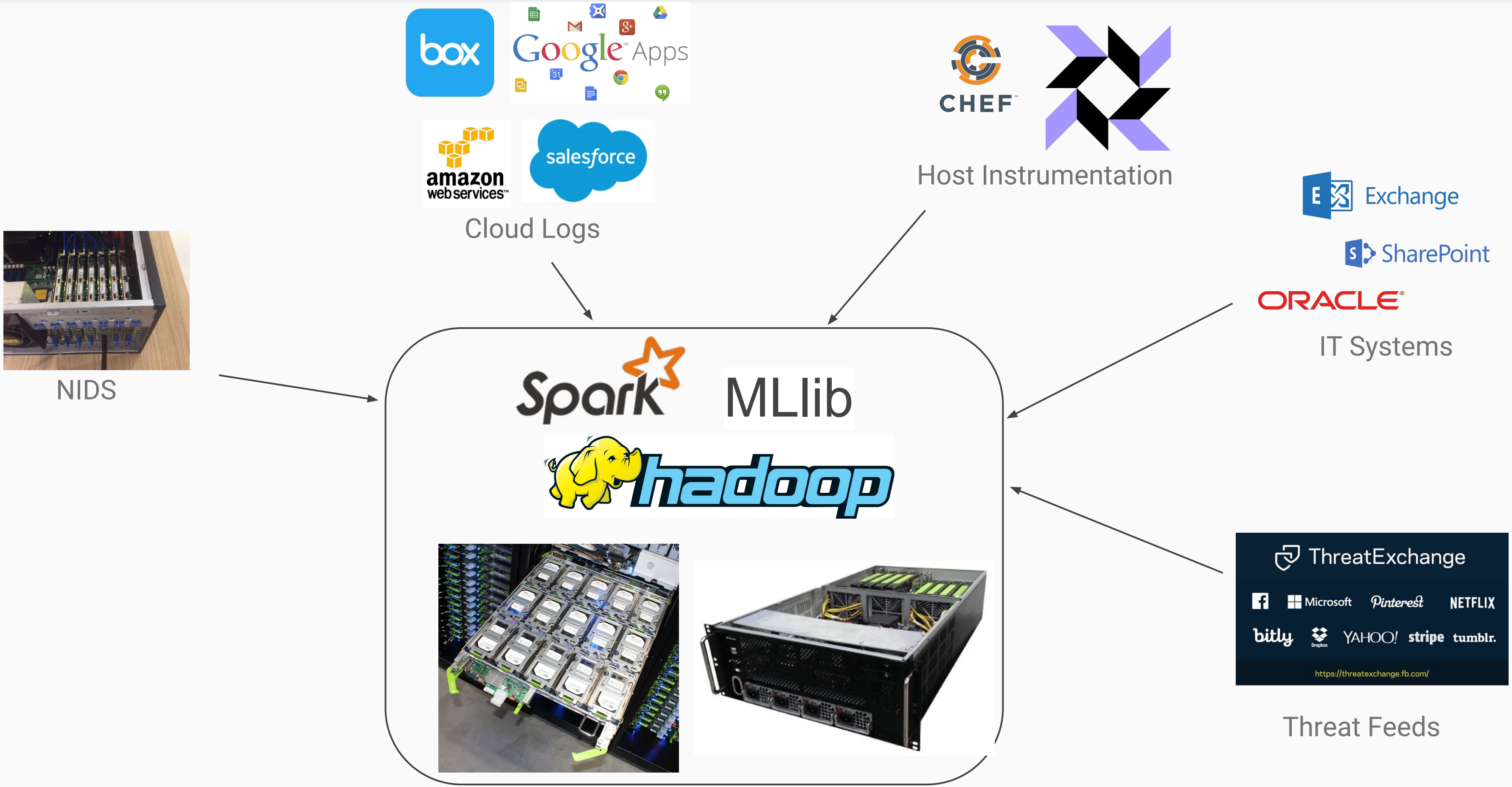
Open Real World Problems

Network Security at Scale



Learn More:
https://youtu.be/mLEawo6OzFM

Dumb sensors, smart (delayed) decisions



Cloud Logs

Host Instrumentation

NIDS

IT Systems

Threat Feeds

Careers in Security

What impact do you want to have on the world?

InfoSec might be the most impactful engineering discipline of the 21st century.

You can choose to:

- Protect those who cannot protect themselves
- Bring voice to those who have never had it
- Secure the technologies that billions depend upon
- Stop those who wish to use technology to control and oppress millions

Participating in this industry makes you a moral actor.

Shape your career around your ethical choices, not vice versa.

Six Tips for a Successful Career

1. Always put yourself in a position to learn and grow. Comfort == decay
2. Be part of the product, not the plumbing
3. Your point of maximum leverage comes right after you get a job offer
4. Understand the Cap Table for any private company
5. Always go into a meeting knowing what you want the outcome to be
6. It’s a small industry. Be nice

Thank you and good luck!

alex@stamos.org