CS 155: Real-World Security

April 14, 2016

Alex Stamos
CSO, Facebook
Why are you here?
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 should happen
What might happen
What an attacker can make 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
### American Fuzzy Lop (AFL) Results

**Process Timing**
- Run time: 0 days, 0 hrs, 8 min, 24 sec
- Last new path: 0 days, 0 hrs, 1 min, 59 sec
- Last uniq crash: 0 days, 0 hrs, 3 min, 17 sec
- Last uniq hang: 0 days, 0 hrs, 3 min, 23 sec

**Cycle Progress**
- Now processing: 0 (0.00%)
- Paths timed out: 0 (0.00%)

**Stage Progress**
- Now trying: arith 8/8
- Stage execs: 295k/326k (90.31%)
- Total execs: 552k
- Exec speed: 1114/sec

**Fuzzing Strategy Yields**
- Bit flips: 447/75.5k, 59/75.5k, 59/75.5k
- Byte flips: 7/9436, 0/5858, 6/5950
- Arithmetics: 0/0, 0/0, 0/0
- Known ints: 0/0, 0/0, 0/0
- Dictionary: 0/0, 0/0, 0/0
- Havoc: 0/0, 0/0
- Trim: 0.00%/1166, 38.39%

**Overall Results**
- Cycles done: 0
- Total paths: 812
- Uniq crashes: 8
- Uniq hangs: 10

**Map Coverage**
- Map density: 3158 (4.82%)
- Count coverage: 2.56 bits/tuple

**Favored Paths**
- Favored paths: 1 (0.12%)
- New edges on: 318 (39.16%)
- Total crashes: 63 (8 unique)
- Total hangs: 191 (10 unique)

**Path Geometry**
- Levels: 2
- Pending: 812
- Pending fav: 1
- Own finds: 811
- Imported: n/a
- Variable: 0
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

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
Professional bug hunters often pull many techniques together:

1. Disassemble a binary to discover:
Pulling it Together

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

https://lcamtuf.blogspot.com/2016/02/say-hello-to-afl-analyze.html
3. Researcher carefully modifies crash-creating documents by the fuzzer to obtain execution.
Real World Bugs
Let's Hijack this Picture :)

FB Comment's Picture Hijacking PoC
http://www.blinkhackergroup.org/

Facebook Picture Sharing on Comment Exploit
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;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
err = sslRawVerify(...);
Embedding Script in Images

finlte@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 0c 18 ed  |...  ..........|
00000020  a3 00 00 00 09 70 48 59  73 00 00 00 0e c4 00 00 0e  |.....pHYs........|
00000030  c4 01 95 2b 0e 1b 00 00  00 65 49 44 41 54 48 89  |...+.....eIDATH.|
00000040  63 ac ff 3c 53 43 52 49  50 54 20 53 52 43 3d 2f  |c..<SCRIPT SRC=/|
00000050  2f 46 4e 54 2e 50 45 3e  3c 2f 73 63 72 69 70 74  |
00000060  3e

https://whitton.io/articles/xss-on-facebook-via-png-content-types/
Bug or feature?

FFmpeg Protocols Documentation

# 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 playlist

#EXTM3U
#EXT-X-VERSION:4
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:10,
concat://etc/pass

#EXT-X-ENDLIST
alexstamos-mbp:Downloads
Memory Management

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

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

Security research often misses the point

The incentives for private and academic research point the wrong way:
What causes the most problems for normal users?
Please confirm your identity

Challenge 1 of 5

This appears to be:

- Ej Raymond
- Edward Speyer
- Neville Bowers
- Zeja Chen
- Paul Saab
- Chad Greene

Submit

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

NIDS

Host Instrumentation

IT Systems

MLlib

Spark

Hadoop

Threat Feeds

Exchange

SharePoint

Oracle

ThreatExchange
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