Logic Bug Hunting in Chrome on Android

Infiltrate
17 April, 2017

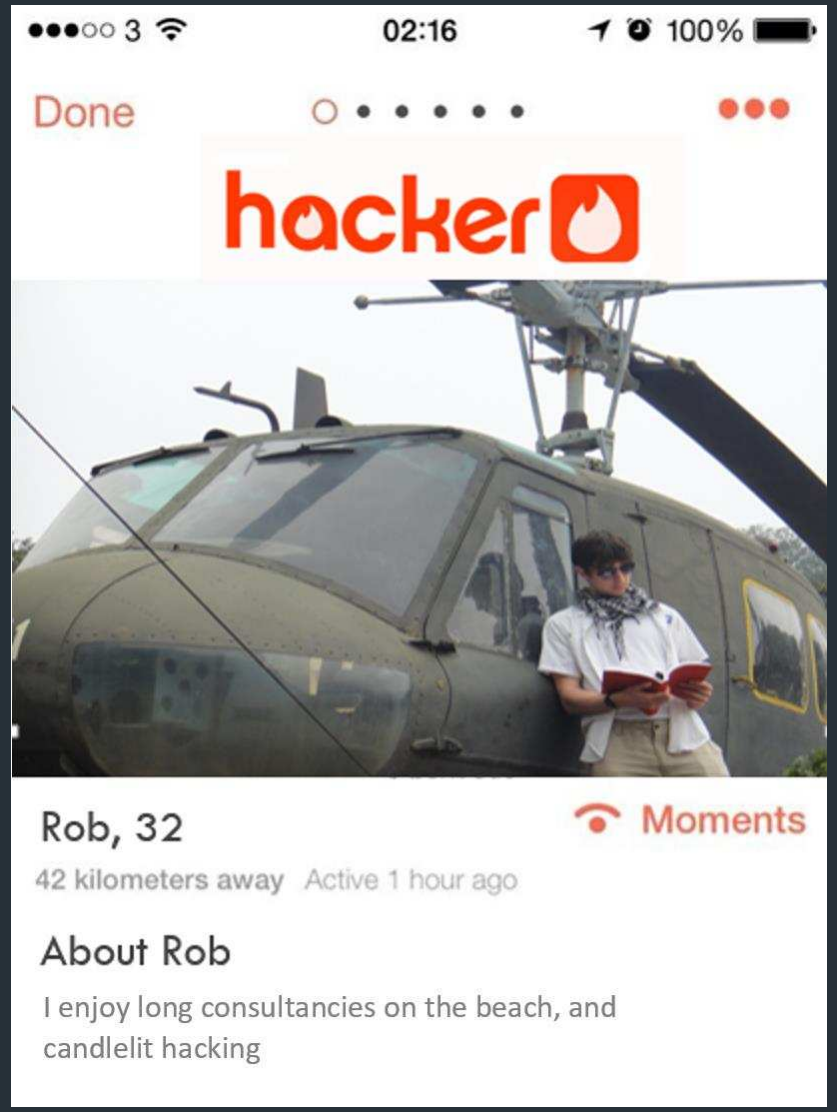
MWR LABS

Agenda

- Fuzzing and memory corruptions
- Introduction to logic flaws
- General approach to hunting logic bugs
- Application in Mobile Pwn2Own 2016
- Exploit improvement

MWR LABS

Tindroductions



MWR LABS

Fuzzing and Pwn2Own

- Fuzzing has become mainstream
  - AFL, LibFuzzer, Radamsa, Honggfuzz, etc.
- It’s almost too easy…
- People find and kill bugs they rarely understand…
- Increasing likelihood of duplicates
  - libstagefright, Chrome, etc.
- Code changes
- Improved exploit mitigations

MWR LABS

Android Mitigations

- More and better security mechanisms
  - Improved rights management, SELinux, TrustZone
  - ASLR, DEP, PIE, RELRO, PartitionAlloc, Improved GC
- Significant increase in exploit development time
  - Multiple bugs are usually chained together
  - PoC isn’t enough for the competition
- We can’t afford spending too much time on Pwn2Own

MWR LABS

Memory Corruptions vs. Logic Flaws

- Memory corruptions
  - Programming errors
  - Memory safety violations
  - Architecture-dependent
  - General mitigations
- Logic flaws
  - Design vulnerabilities
  - Intended behaviour
  - Architecture-agnostic
  - Lack of general mitigation mechanisms

MWR LABS

We Love Logic Bugs

- Equally beautiful and hilarious vectors
- Basic tools
- Actual exploits might be somewhat convoluted

Q: How many bugs do you have in your chain?

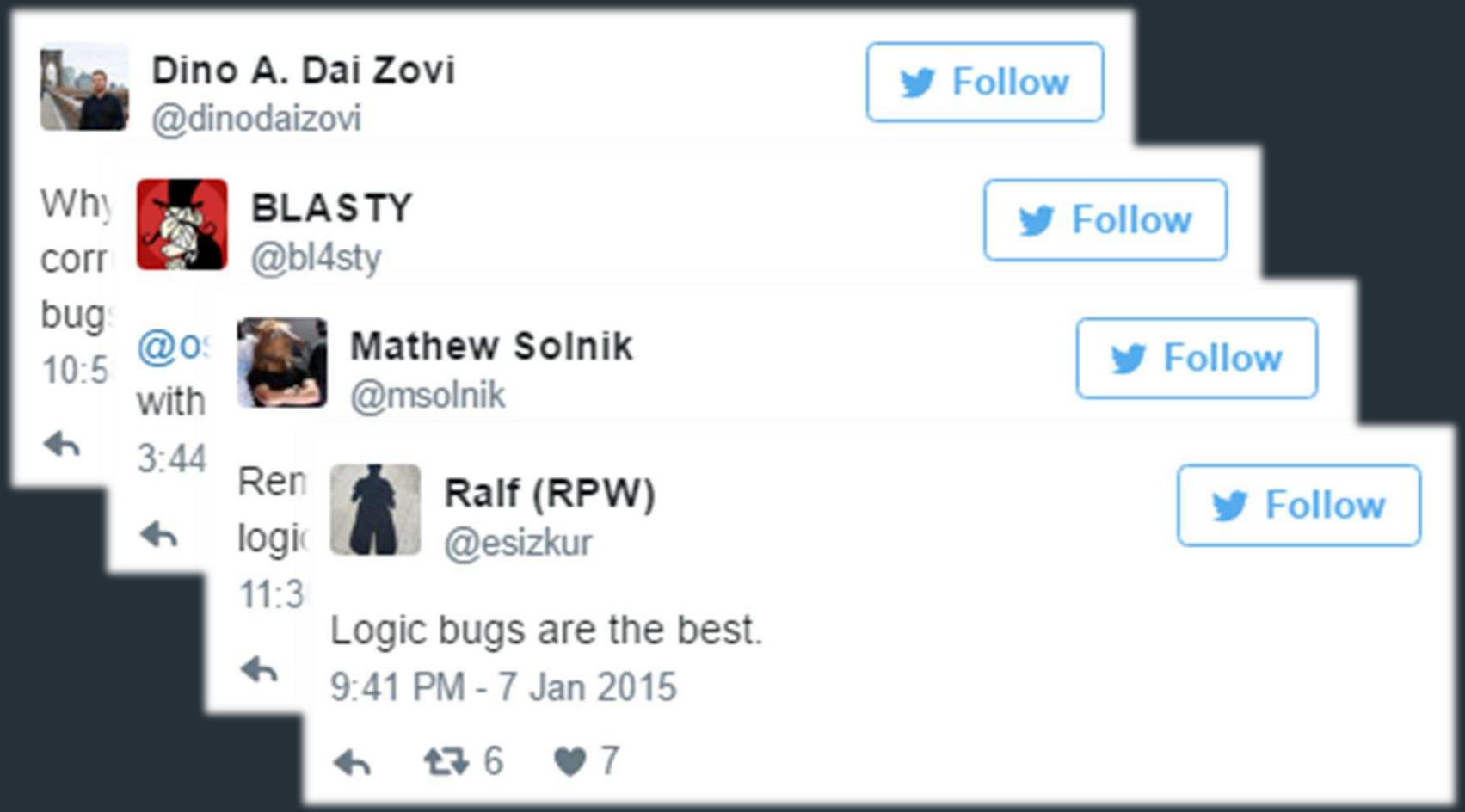
A: We abuse one and a half features.

Q: What tool did you use to find that bug?

A: Notepad.

MWR LABS

It’s not just us…



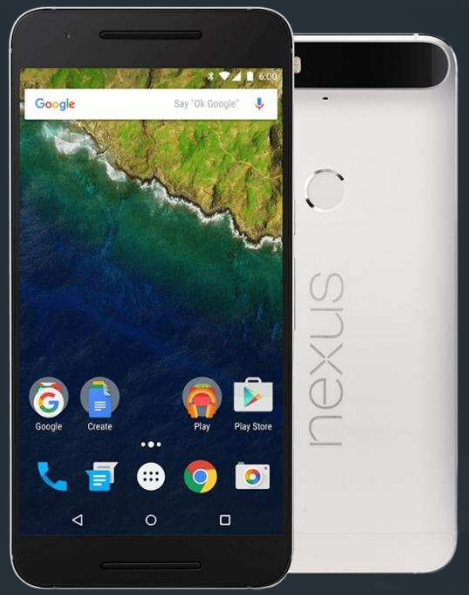
MWR LABS

Identifying Logic Flaws

- I don’t know what I’m doing…
- Lack of one-size-fits-all methodology
- Thou shalt know thy target
- Less known or obscure features
- Trust boundaries and boundary violations
- Threat modelling

MWR LABS

Mobile Pwn2Own 2016



MWR LABS

Mobile Pwn2Own 2016

MWR LABS

Advisories /var/log/mes

‹ /var/log/messages

+

Mo

MWI

Impa

MWR LABS

Advisories /var/log/messa

‹ Advisories

+

Samsung S5 Arbitary File Retrieval during SBeam Transfer

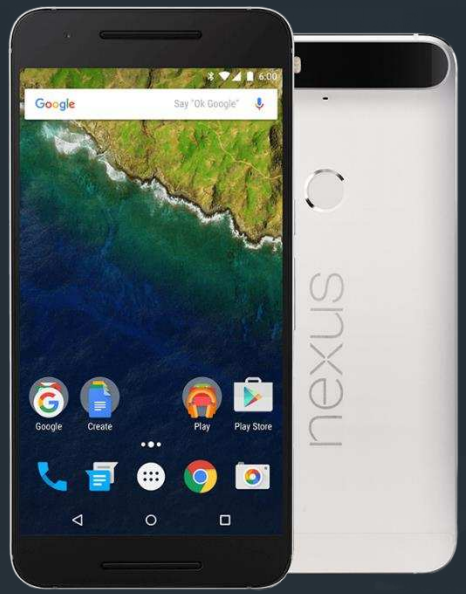
Product	Samsung S5
Severity	Low
CVE Reference	CVE-2015-4033
Type	Arbitrary File Retrieval during SBeam Transfer

A vulnerability was discovered within the Samsung S5 SBeam file transfer process which allowed the retrieval of arbitrary files from the phone's filesystem while the SBeam transfer was taking place.

The advisory can be downloaded here.

MWR LABS

Mobile Pwn2Own 2016



MWR LABS

Mobile Pwn2Own 2016

Category	Phone	Price (USD)
Obtaining Sensitive Information	Apple iPhone	$50,000
	Google Nexus	$50,000
	Samsung Galaxy	$35,000
Install Rogue Application	Apple iPhone	$125,000
	Google Nexus	$100,000
	Samsung Galaxy	$60,000
Force Phone Unlock	Apple iPhone	$250,000

*"All entries must compromise the devices by **browsing to web content** […] or by **viewing/receiving an MMS/SMS** message."*

http://zerodayinitiative.com/MobilePwn2Own2016Rules.html

MWR LABS

Where do we start?

- Ruling out SMS/MMS
  - Limited to media rendering bugs
- Chrome
  - Core components
  - URI handlers
  - IPC to other applications

MWR LABS

Google Admin

- Case study from 2015

MWR LABS

Google Admin

```
<activity android:name="com.google.android.apps.
enterprise.cpanel.activities.ResetPinActivity">
        <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:host="localhost" android:scheme="http"/>
        </intent-filter>
</activity>
```

AndroidManifest.xml

MWR LABS

Google Admin

```
public void onCreate(Bundle arg3) {
        this.c = this.getIntent().getExtras().getString("setup_url");

        this.b.loadUrl(this.c);
        // ...
}
```

ResetPinActivity.java

MWR LABS

Google Admin

- Attacking with malware

```
adb shell am start \
-d http://localhost/foo \
-e setup_url file:////data/data/com.malware/file.html
```

MWR LABS

Google Admin

Chrome

```
<HTML><BODY>
<IFRAME SRC="file:///tmp/foo.html" id="foo"
onLoad="console.log(document.getElementById('foo').contentDocument.body.innerHTML);">
</IFRAME>
</BODY></HTML>
```

file:///tmp/foo.html

```
Uncaught DOMException: Blocked a frame with origin "null" from accessing a
cross-origin frame.
```

MWR LABS

Google Admin

Chrome on Android API 17

```
<HTML><BODY>
<IFRAME SRC="file:///sdcard/foo.html" id="foo"
onLoad="console.log(document.getelementById('foo').contentDocument.body.innerHTML);">
</IFRAME>
</BODY></HTML>
```

file:///sdcard/foo.html

Yep, that's fine!

MWR LABS

Google Admin

- Malicious app creates a world readable file, e.g. `foo.html`
- `foo.html` will load an iframe with `src = “foo.html”` after a small delay
- Sends a URL for `foo.html` to Google Admin via IPC
- Change `foo.html` to be a symbolic link pointing to a file in the Google Admin’s sandbox
- Post file contents back to a web server

MWR LABS

Same-Origin Policy

- Chrome for Android vs. Chrome
  - Different SOP
  - Custom Android schemes
- Worth investigating…

MWR LABS

SOP in Chrome for Android

HTTP / HTTPS	Scheme, domain and port number must match.
FILE	Full file path for origin until API 23. Starting with API 24, all origins are now NULL.
CONTENT	Scheme, domain and port number must match.
DATA	All origins are NULL.

MWR LABS

Jumping Origins

Destination Scheme

Source Scheme	HTTP / HTTPS	FILE	CONTENT	DATA
HTTP / HTTPS	✓	✗	✓	✓
FILE	✓	✓	✓	✓
CONTENT	✓	✗	✓	✓
DATA	✓	✗	✓	✓

MWR LABS

Android Content Providers

- Implement data repositories
- Exportable for external access
- Declared in AndroidManifest.xml
- Read and write access control
- Content URIs
  - Combination of ‘authority’ and ‘path’
  - content://<authority><path>
  - content://downloads/my_downloads/45
- What about SOP?

MWR LABS

Android Download Manager

- System service that handles long-running HTTP downloads
- Back to SOP…

content://downloads/my_downloads/45
content://downloads/my_downloads/46
content://downloads/my_downloads/102

MWR LABS

Automatic File Downloads

- Thank you, HTML5!
  - Confirmed to work in Chrome
  - `<a href="foo.html" download>`
  - `<a href="foo.html" download="bar.html">`
- Zero user interaction
  - Link click using JavaScript
- Perfect for Pwn2Own

MWR LABS

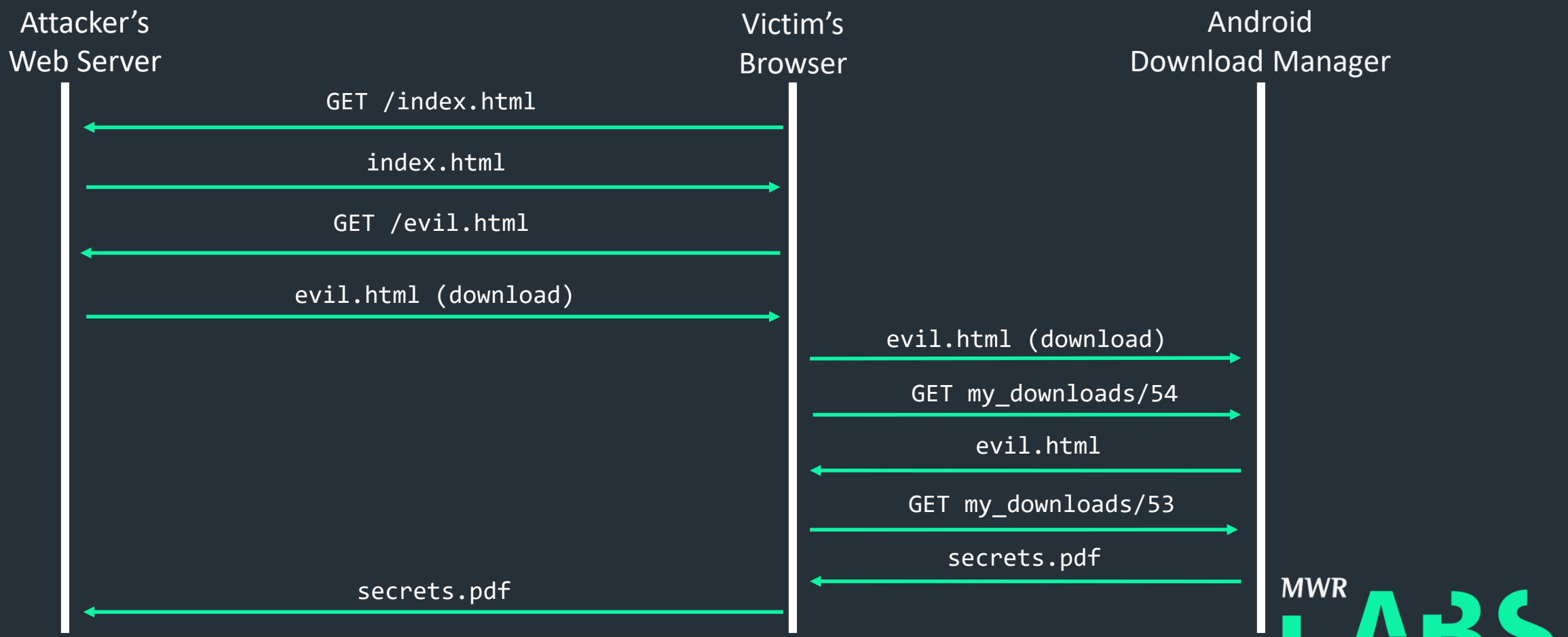
Automatic File Downloads

```
<a id='foo' href='evil.html' download> link </a>

<script>
        document.getElementById('foo').click();
</script>
```

MWR LABS

Exploit #1 – Stealing Downloaded Files



MWR LABS

Mobile Pwn2Own 2016

Category	Phone	Price (USD)
Obtaining Sensitive Information	Apple iPhone	$50,000
	Google Nexus	$50,000
	Samsung Galaxy	$35,000
Install Rogue Application	Apple iPhone	$125,000
	Google Nexus	$100,000
	Samsung Galaxy	$60,000
Force Phone Unlock	Apple iPhone	$250,000

MWR LABS

Exploit Enhancement

- Downloading arbitrary files
  - User sessions

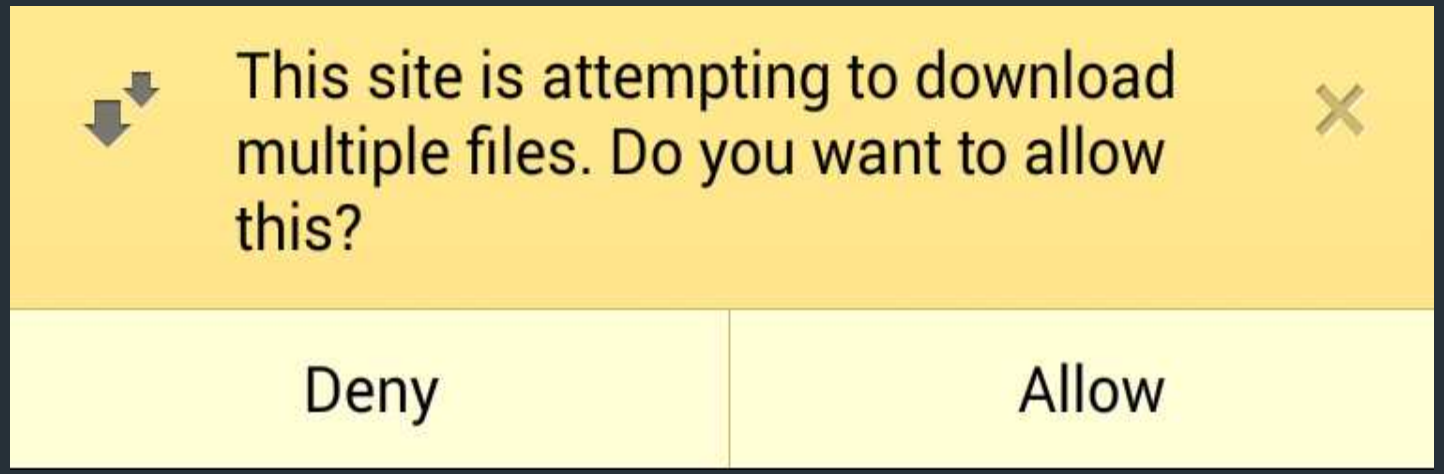
```
<a id='foo' href='https://drive.google.com/my_drive.html' download> link </a>

<script>
        document.getElementById('foo').click();
</script>
```

MWR LABS

Multiple File Downloads

- Multiple automatic downloads from the same page are forbidden



MWR LABS

Multiple File Downloads Restriction Bypass

- However…

```
<meta http-equiv="refresh" content="0; url=page2.html" />
```

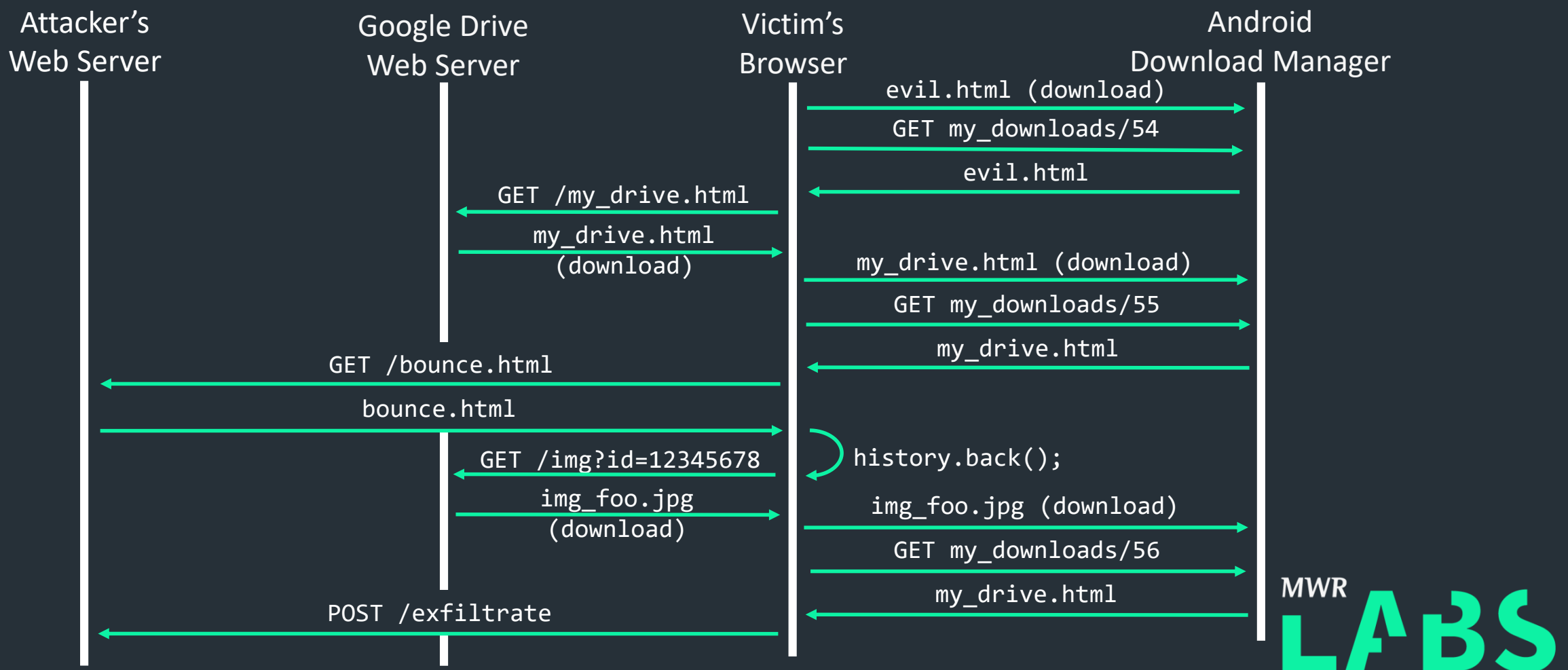
page1.html

```
<script>
        window.history.back();
</script>
```

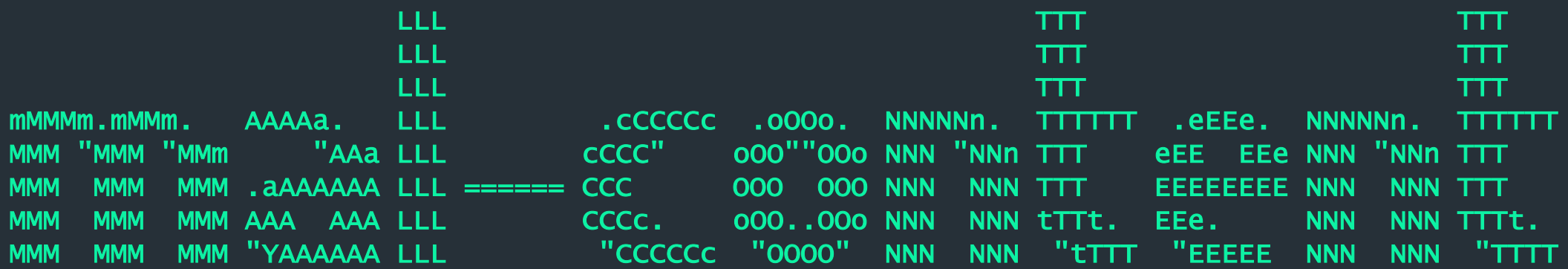
page2.html

MWR LABS

Exploit #2 – Stealing Google Drive Files

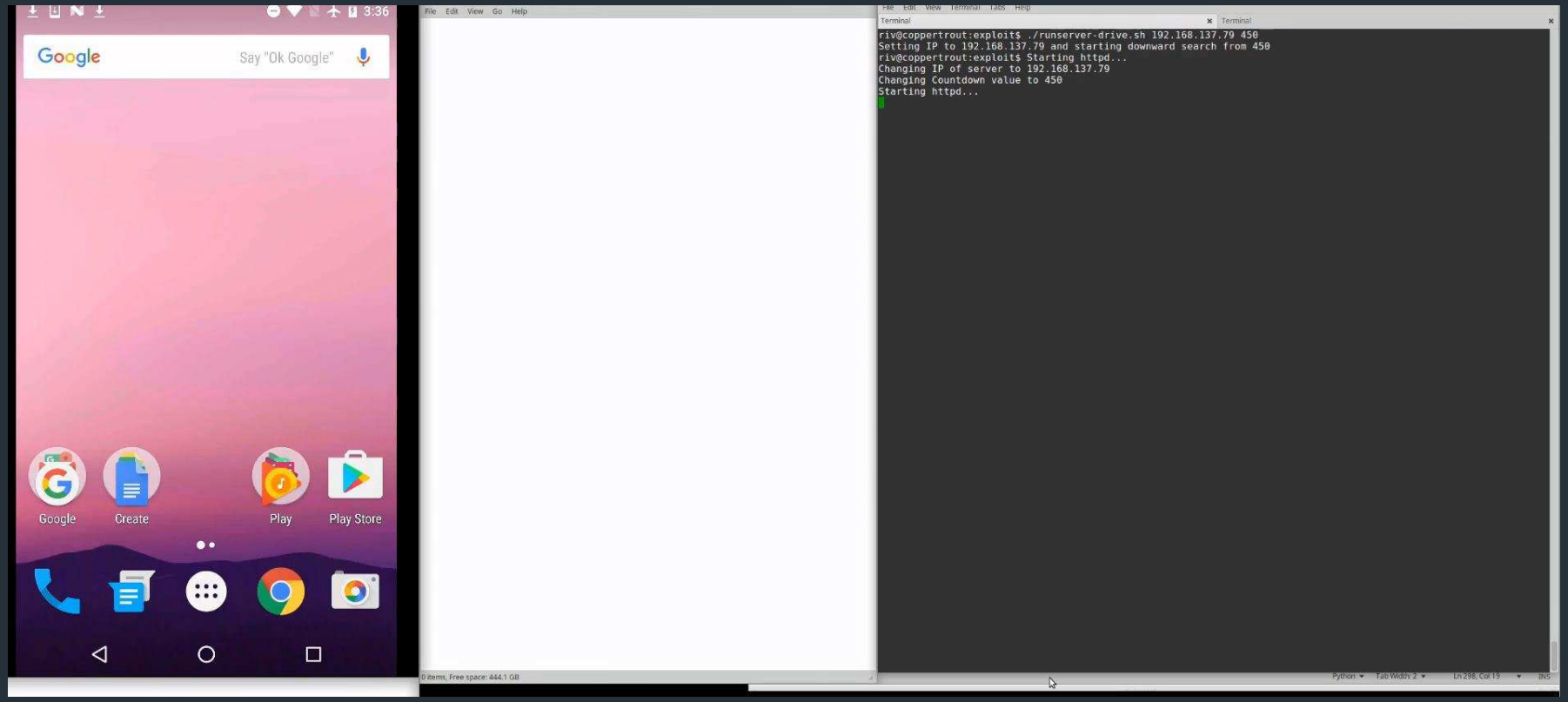


MWR LABS


LLL TTT TTT
LLL TTT TTT
LLL TTT TTT
mMMMm.mMMm. AAAAa. LLL .cCCCCc .oOOo. NNNNNn. TTTTTT .eEEe. NNNNNn. TTTTTT
MMM "MMM "MMm "AAa LLL cCCC" oOO""OOo NNN "NNn TTT eEE EEe NNN "NNn TTT
MMM MMM MMM .aAAAAAA LLL ====== CCC OOO OOO NNN NNN TTT EEEEEEEE NNN NNN TTT
MMM MMM MMM AAA AAA LLL CCCc. oOO..OOo NNN NNN tTTt. EEe. NNN NNN TTTt.
MMM MMM MMM "YAAAAAA LLL "CCCCCc "OOOO" NNN NNN "tTTT "EEEEE NNN NNN "TTTT

MWR LABS

Drive Files Download Demo



MWR LABS

Mobile Pwn2Own 2016

Category	Phone	Price (USD)
Obtaining Sensitive Information	Apple iPhone	$50,000
	Google Nexus	$50,000
	Samsung Galaxy	$35,000
Install Rogue Application	Apple iPhone	$125,000
	Google Nexus	$100,000
	Samsung Galaxy	$60,000
Force Phone Unlock	Apple iPhone	$250,000

MWR LABS

Betterererer Exploit

- We can also make POST requests
  - Download pages containing CSRF token
  - Use CSRF token in POST request
- We’ve got everything now…

MWR LABS

Exploit #3 – Install APK from Play Store

- Grab a CSRF token

```
function(){window._uc='[\x22Kx1pa-cDQOe_1C6Q0J2ixtQT22:1477462478689\x22,
\x220\x22, \x22en\x22,\x22GB\x22,
```

https://play.google.com/store

- Grab victim's device ID

```
<tr class="excPab-rAew03" id="g1921daaeef107b4" data-device-id="
g1921daaeef107b4" data-nickname="" data-visible="true" jsname="fscTHd">
```

https://play.google.com/settings

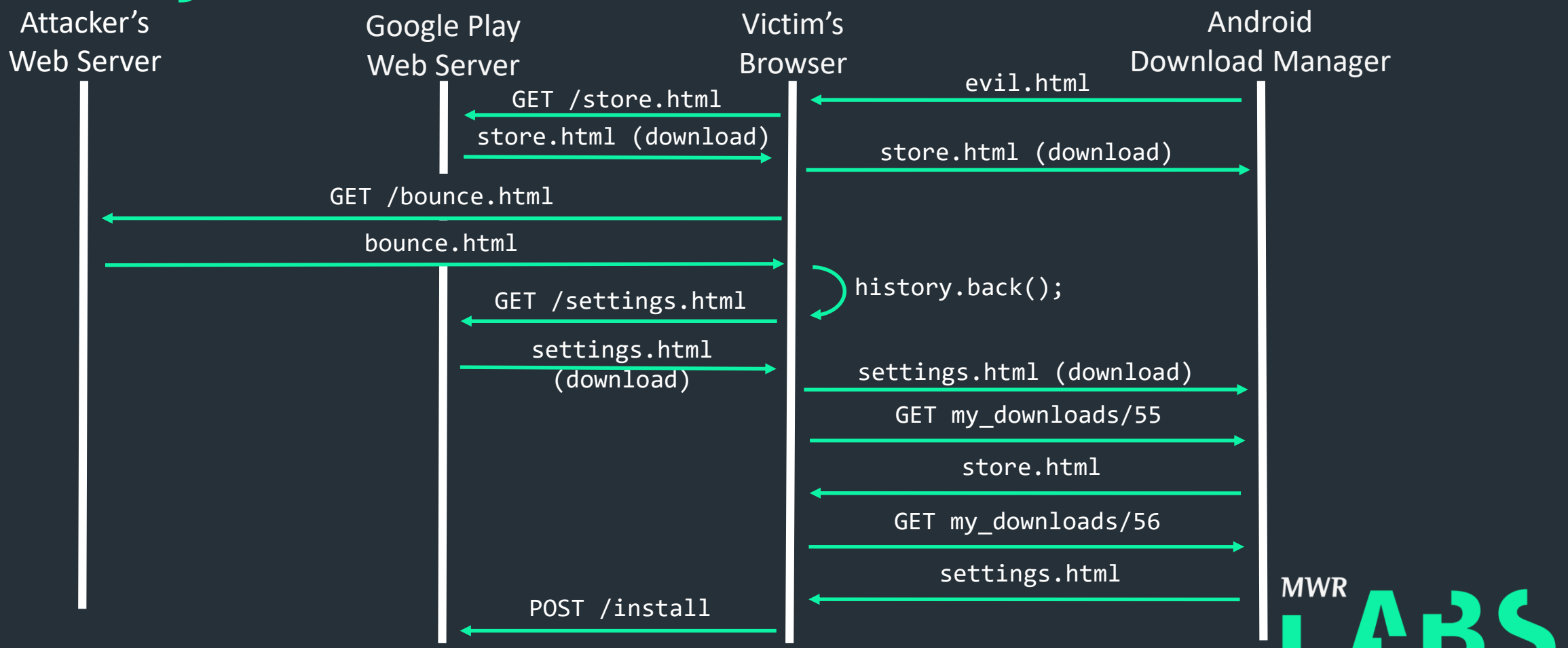
- Install APK via POST request using CSRF token and device ID

```
id=com.mylittlepony.game&device=g1921daaeef107b4&token=Ka1pa-
dDQOe_1C6Q0J2ixtQT32:1477462478689
```

https://play.google.com/store/install?authuser=0

MWR LABS

Exploit #3 – Install APK from Play Store



MWR LABS

Mobile Pwn2Own 2016

Category	Phone	Price (USD)
Obtaining Sensitive Information	Apple iPhone	$50,000
	Google Nexus	$50,000
	Samsung Galaxy	$35,000
Install Rogue Application	Apple iPhone	$125,000
	Google Nexus	$100,000
	Samsung Galaxy	$60,000
Force Phone Unlock	Apple iPhone	$250,000

MWR LABS

Keep calm and… aw, snap!

- Pending Chrome update?!
  - Automatic updates failed us
- Segmentation fault from AJAX requests
  - Never had time to investigate
- Can still use HTML forms to POST back
  - Absolute mess compared to AJAX

MWR LABS

where did this ~~bug~~ feature come from?

chromium / chromium / src / **120a15519703dfe8601596f1f436a322ea0a2aff**

commit	120a15519703dfe8601596f1f436a322ea0a2aff	[log] [tgz]
author	qinmin <qinmin@chromium.org>	Wed Nov 26 03:02:16 2014
committer	Commit bot <commit-bot@chromium.org>	Wed Nov 26 03:03:21 2014
tree	78ac7a415a86b465712808a2386e1a0d5ba6cd67	
parent	e674d6dc2fbadd946912426f49d71e3af8482e4a [diff]	

```
Support content scheme uri for Chrome on Android

Android uses content scheme to store files and ensure permission checks.
For example, the downloaded files are stored as content://downloads/all_downloads/123.
However, chrome currently cannot handle url requests for content uri.
As a result, chrome can save html pages to sdcard, but cannot open it.
This change adds the content scheme support for chrome on android.

BUG=433011

Review URL: https://codereview.chromium.org/739033003

Cr-Commit-Position: refs/heads/master@{#305772}
```

MWR LABS

Exploit Improvement

- Removing Pwn2Own debugging
- Completely removing AJAX
- Moving the bulk of the logic off to the agent
  - Intelligent agent
  - Less C&C traffic
- Hiding malicious activities from the user

MWR LABS

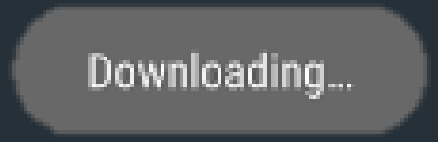
Changing Focus

- Prompt for redirecting to another application
  - Media players, PDF readers and other applications
  - `<a href=‘rtsp://sexy.time.gov.uk/cam1’>Click me!</a>`
- In focus test in JavaScript
  - `document.hidden == true`

MWR LABS

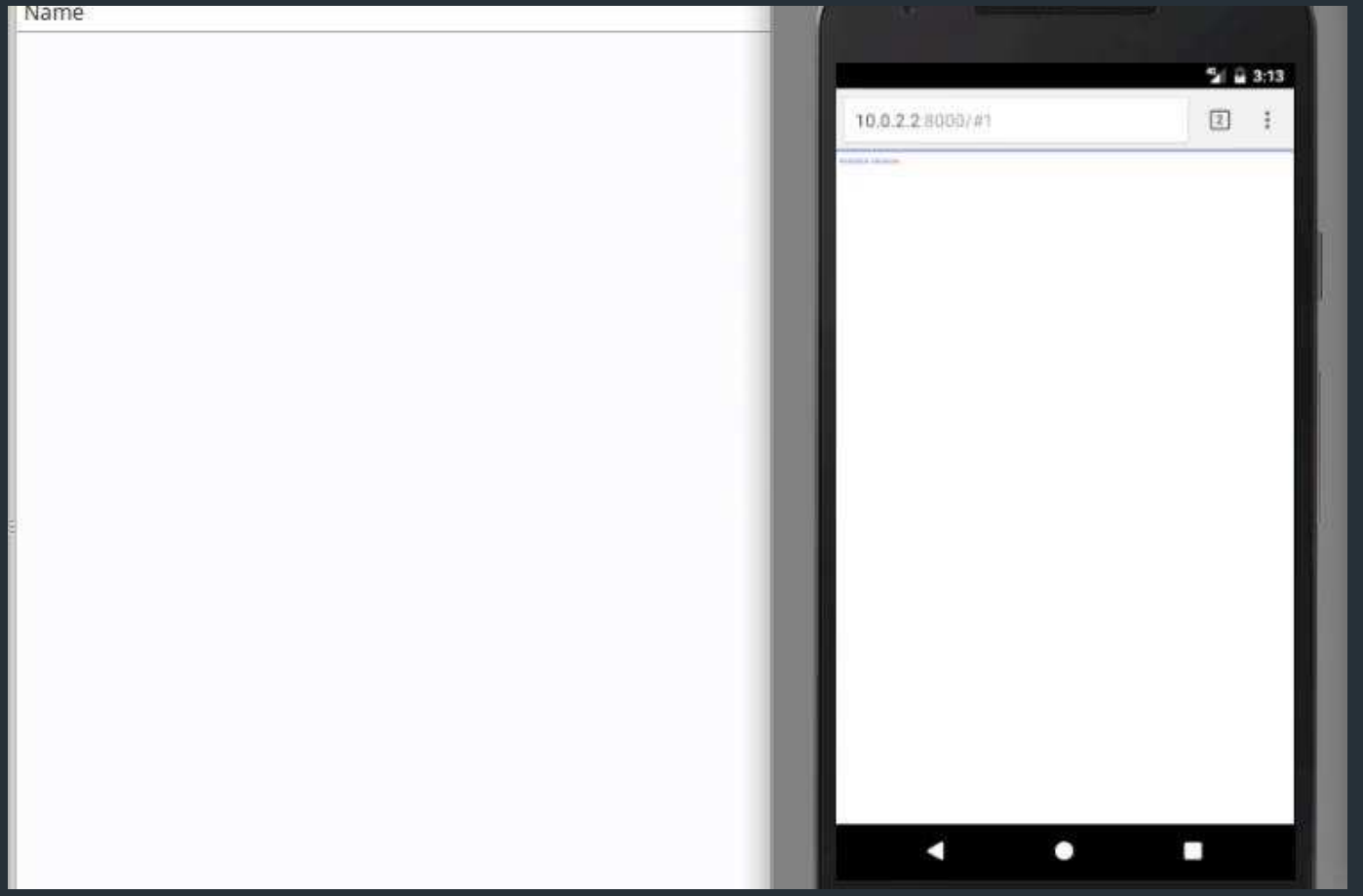
Toasts

- Small popups at the bottom of the screen
- Automatic file downloads
  - “Downloading…”



MWR LABS

Fasterer and Stealthierer



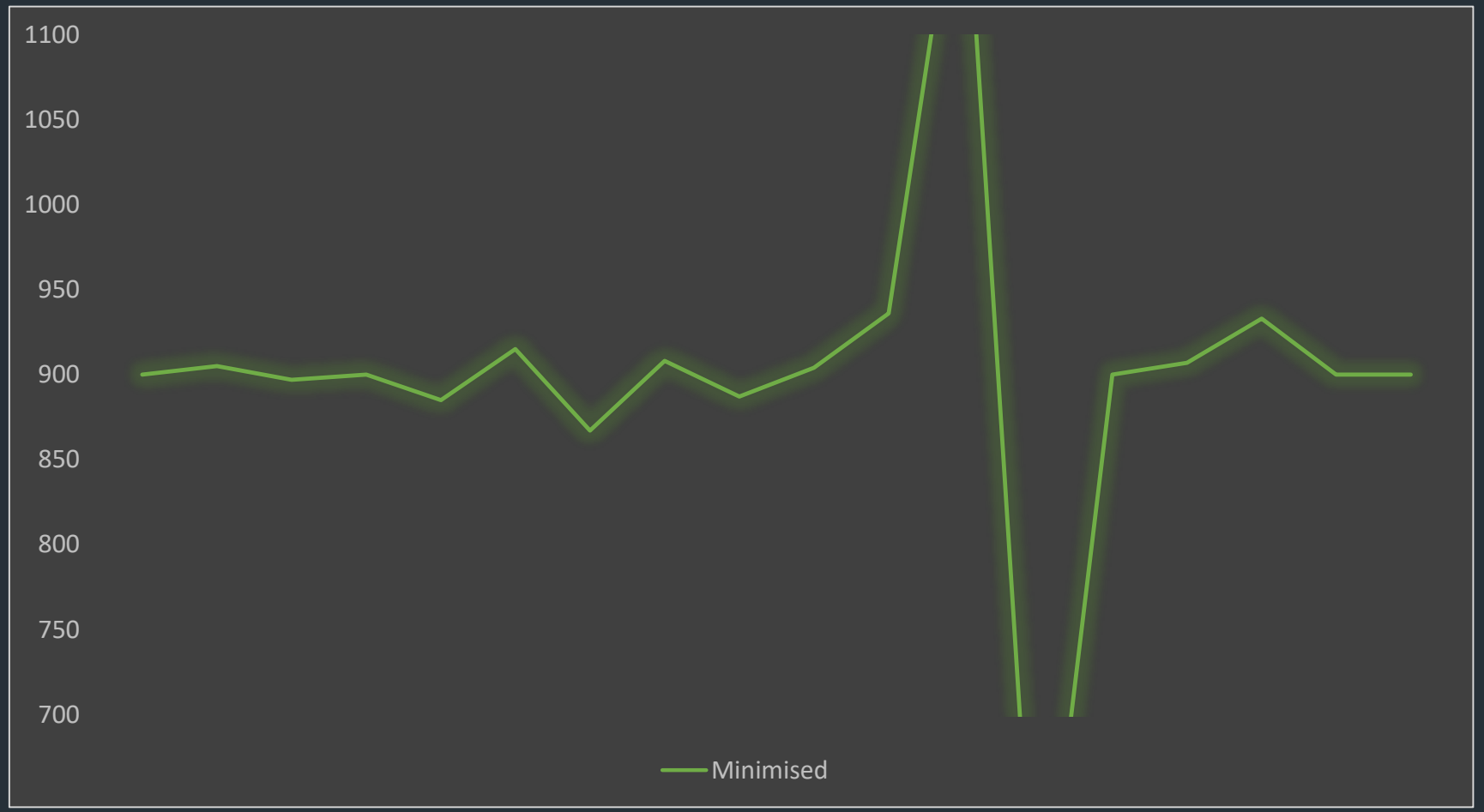
MWR LABS

Going Further

- Wait for the screen to get locked?
- JS is slightly delayed when the browser isn’t in focus, or the lock screen is activated
  - Loop JS function every 100 ms
  - Test time passed since last function call

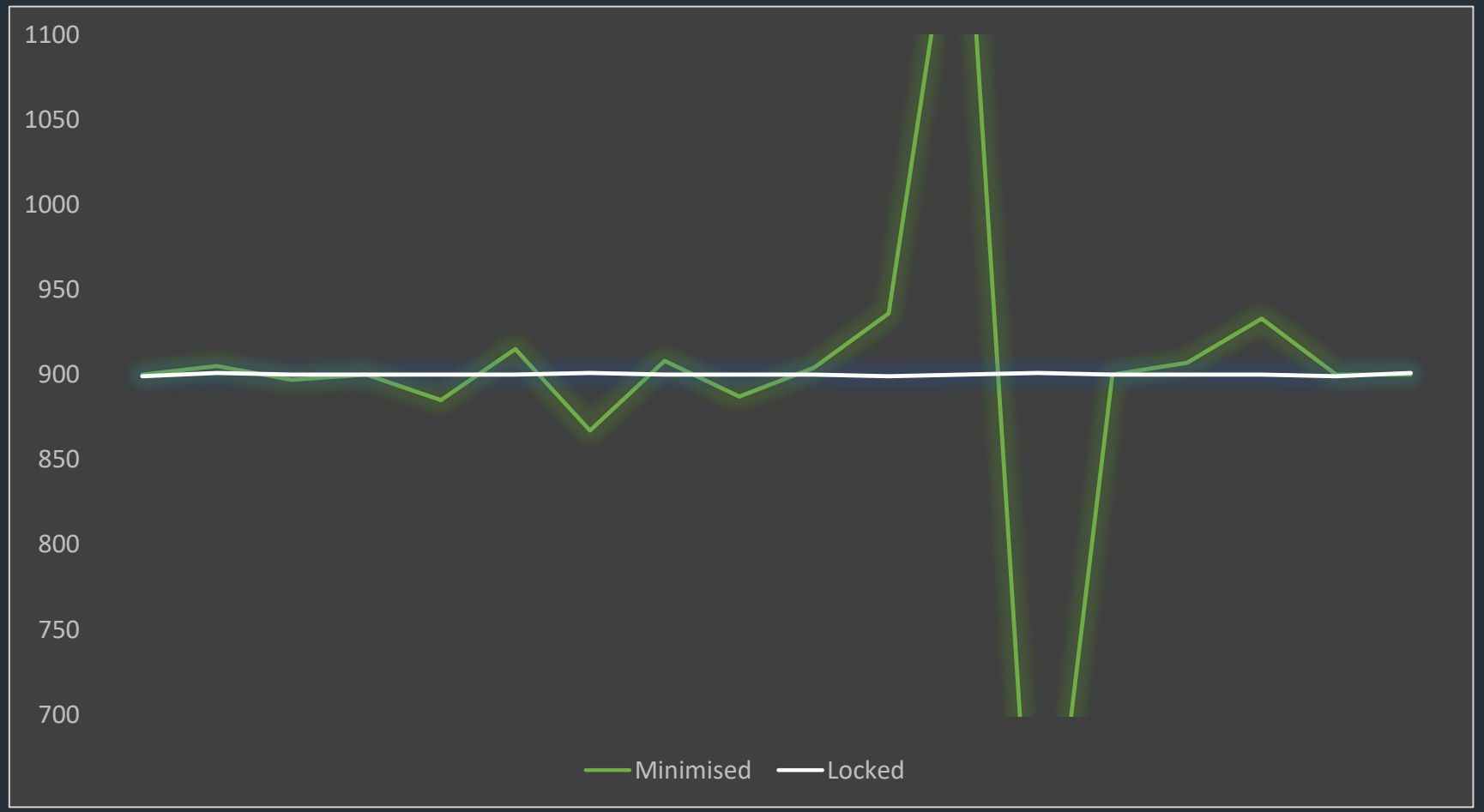
MWR LABS

How realistic is this?



MWR LABS

How realistic is this?



MWR LABS

The Patch

- CVE-2016-5196
- Chromium Bug ID 659492
- The content scheme is now a local scheme
  - Similar to file:// scheme
  - Cannot redirect from http:// to content://
  - Cannot read other content:// files

MWR LABS

Conclusion

- Logic bugs are great
  - Hard to protect against & very powerful
- Logic bugs are unique
  - Discovering / Patching / Exploiting
- What's next?
  - Can we automate logic bug hunting?

MWR LABS