
IRONSQUIRREL
DESIGN: KONYA LÁSZLÓ

# How to protect your browser 0-day

Codenamed #IRONSQUIRREL

~~TS//SI//FVEY~~

FOUO//SI//FVEY

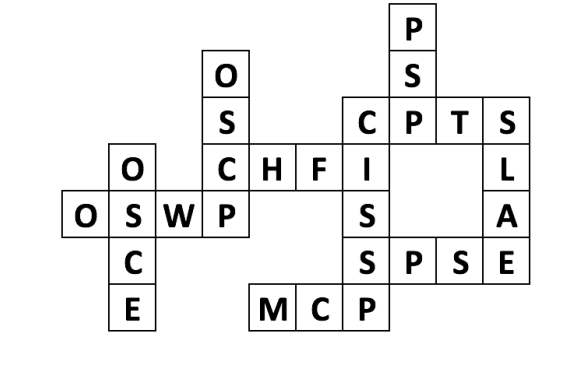
Zoltan Balazs – MRG Effitas

2017 November

DEEPSEC

DEEPSEC
IN-DEPTH SECURITY

# Whoami?


Zombie Browser Toolkit

https://github.com/Z6543/ZombieBrowserPack

HWFW Bypass tool

- Idea later(?) implemented by nation state attackers in Duqu 2

https://github.com/MRGEffitas/hwfwbypass

Malware Analysis Sandbox Tester tool

https://github.com/MRGEffitas/Sandbox_tester

Played with crappy IoT devices

https://jumpespjump.blogspot.hu/2015/09/how-i-hacked-my-ip-camera-and-found.html

https://jumpespjump.blogspot.hu/2015/08/how-to-secure-your-home-against.html

Table of contents

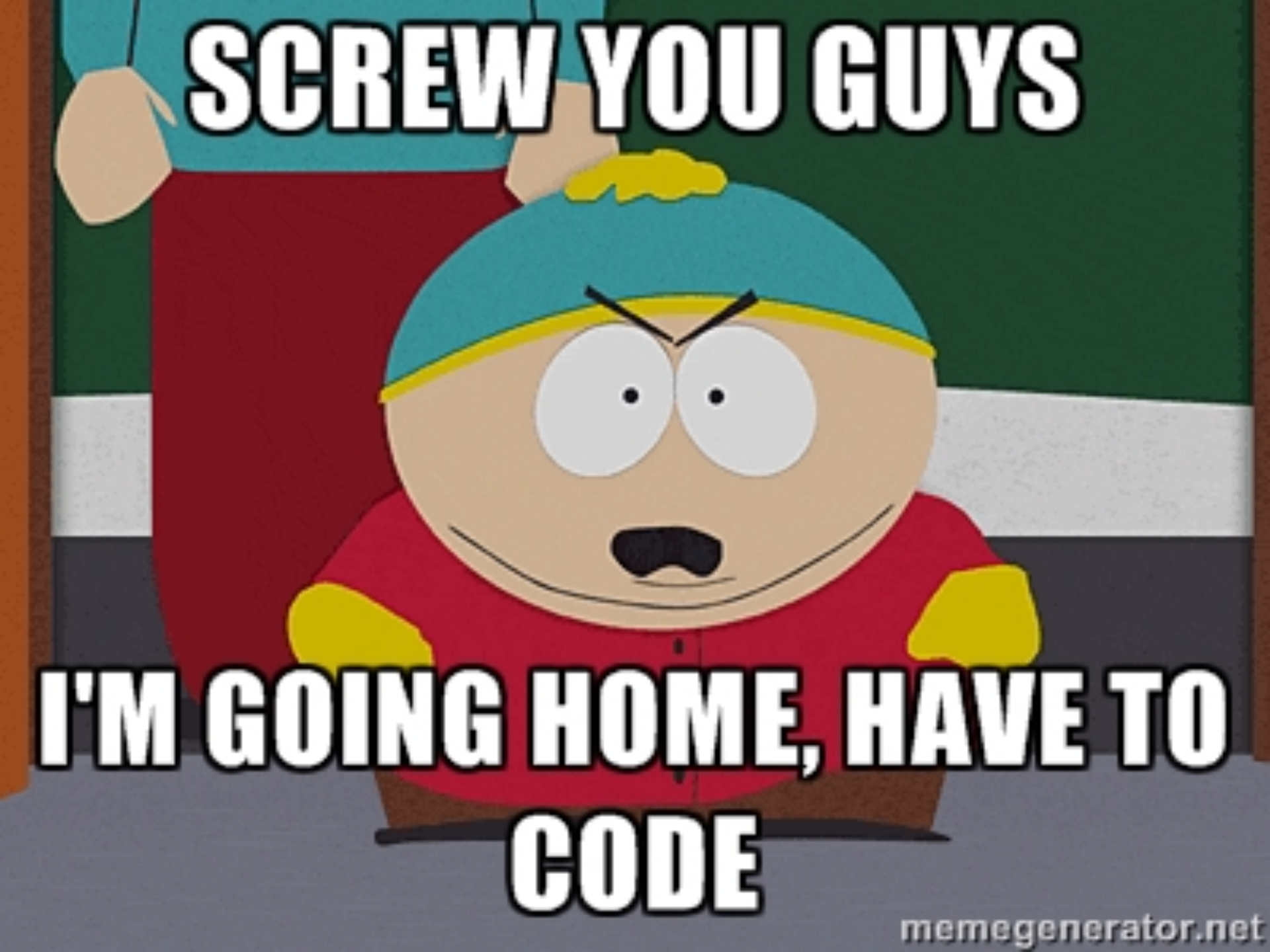
Win Hacker Pschorr

Find Cyber on the slides

How did it all begin?

I had this “discussion” with nextgen/breach-detection vendors that their network appliance can be bypassed in a way that they can’t even see an exploit happened or malware was delivered

They told me it is impossible


SCREW YOU GUYS
I'M GOING HOME, HAVE TO CODE
memegenerator.net

# Why should you listen to this talk?

Exploit brokers and law enforcement

- Effective way to prevent the 0-day exploit code being leaked

Pentesters/red team members

- Bypass perimeter defenses, some host IDS

Blue team members, forensics investigators, exploit kit researchers

- How current defenses can be bypassed via #IRONSQUIRREL browser exploit delivery

Rest of you

- Learning about elliptic curve cryptography is always fun

Introduction to

Exploit kits, targeted attacks with 0-dayz

DH key agreement

ECDH key agreement

Encrypted browser exploit delivery

My idea implemented by the bad guys

## Browser exploits, exploit kits

"An **exploit kit** is a software kit designed to run on web servers, with the purpose of identifying software vulnerabilities in client machines communicating with it, and discovering and **exploiting** vulnerabilities to upload and execute malicious code on the client."

https://en.wikipedia.org/wiki/Exploit_kit

# Lost 0-day exploit => $$$--

Targeting of Ahmed Mansoor with iOS Safari 0-day exploit

- http://www.5z8.info/malicious-cookie_z2m5jd_mydick
- iOS 0-day exploit
  - 100 000 USD – 1 500 000 USD
- Mansoor still in prison ☹

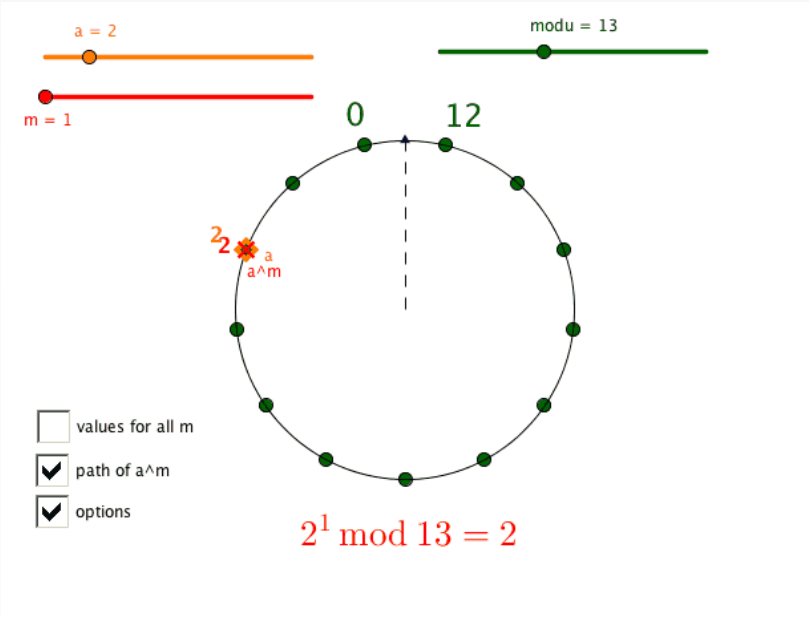
Tor browser 0-day exploit used by law enforcement on pedophile site

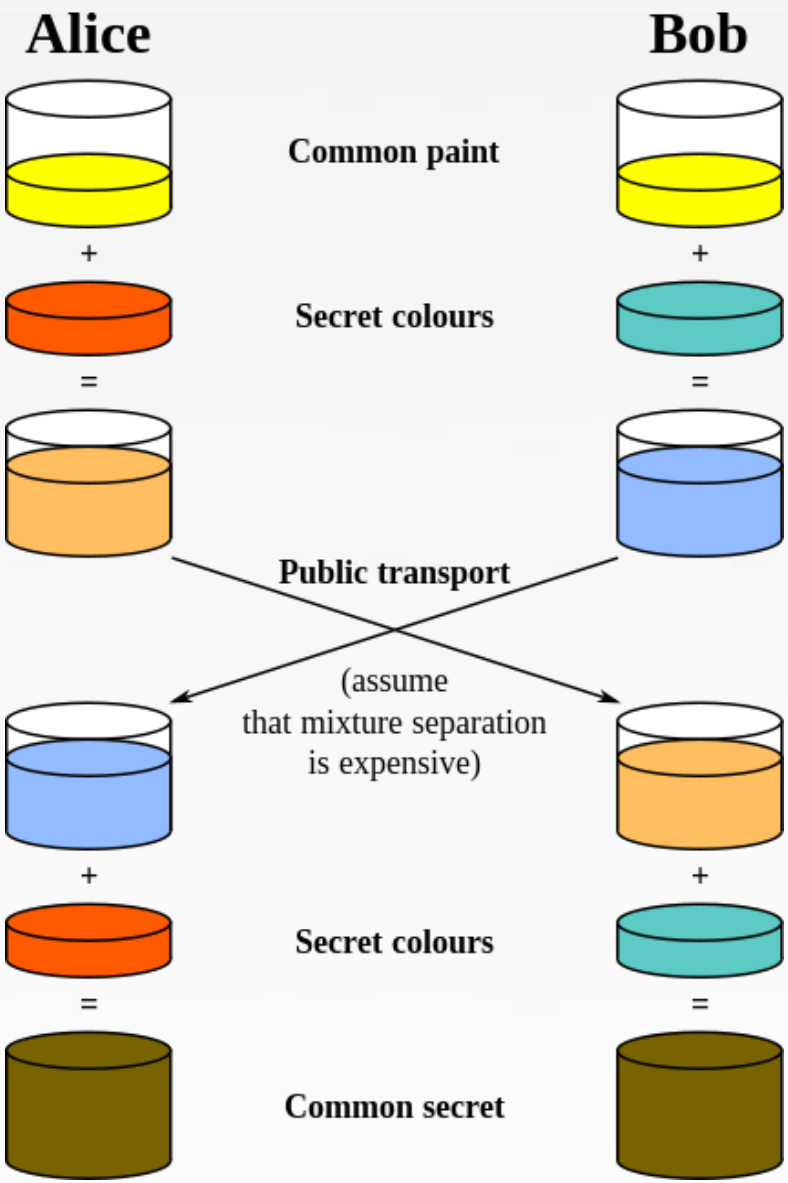
- http://www.5z8.info/twitterhack_u3o2ex_this-page-will-steal-all-of-your-personal-data
- Tor Browser 0-day : 30 000 USD

https://www.zerodium.com/program.html

Both exploit leaked, burnt

# Diffie-Hellman key agreement - 1976



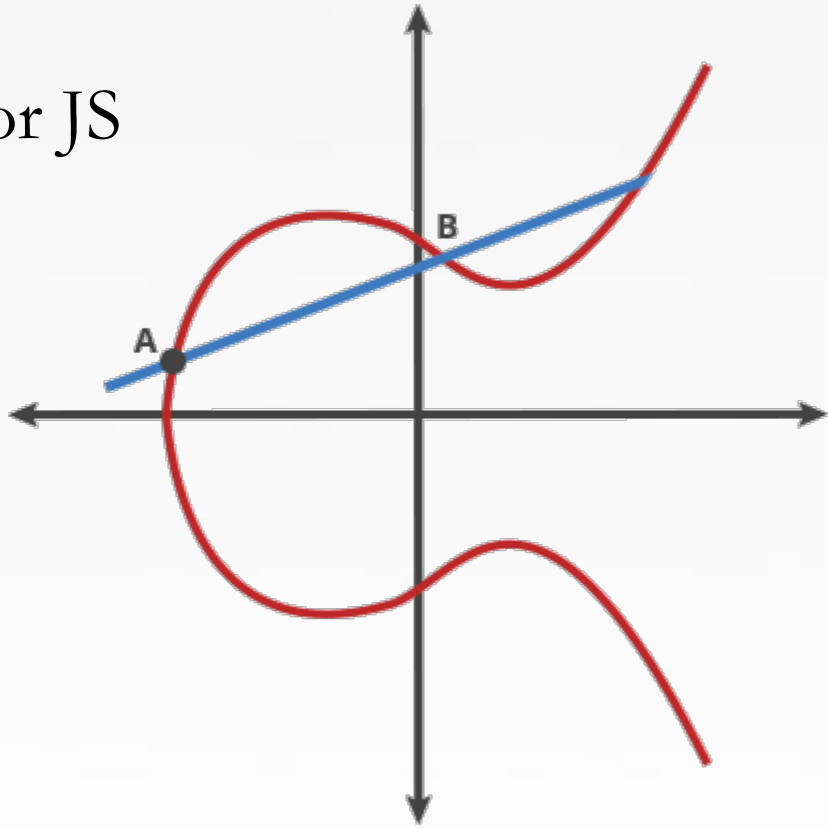

http://mathhombre.blogspot.hu/2014_05_01_archive.html
https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

**E**lliptic **C**urve based **D**iffie-**H**ellman (ECDH) key agreement

ECDH key agreement 5-10 times faster on same CPU
[citation needed]

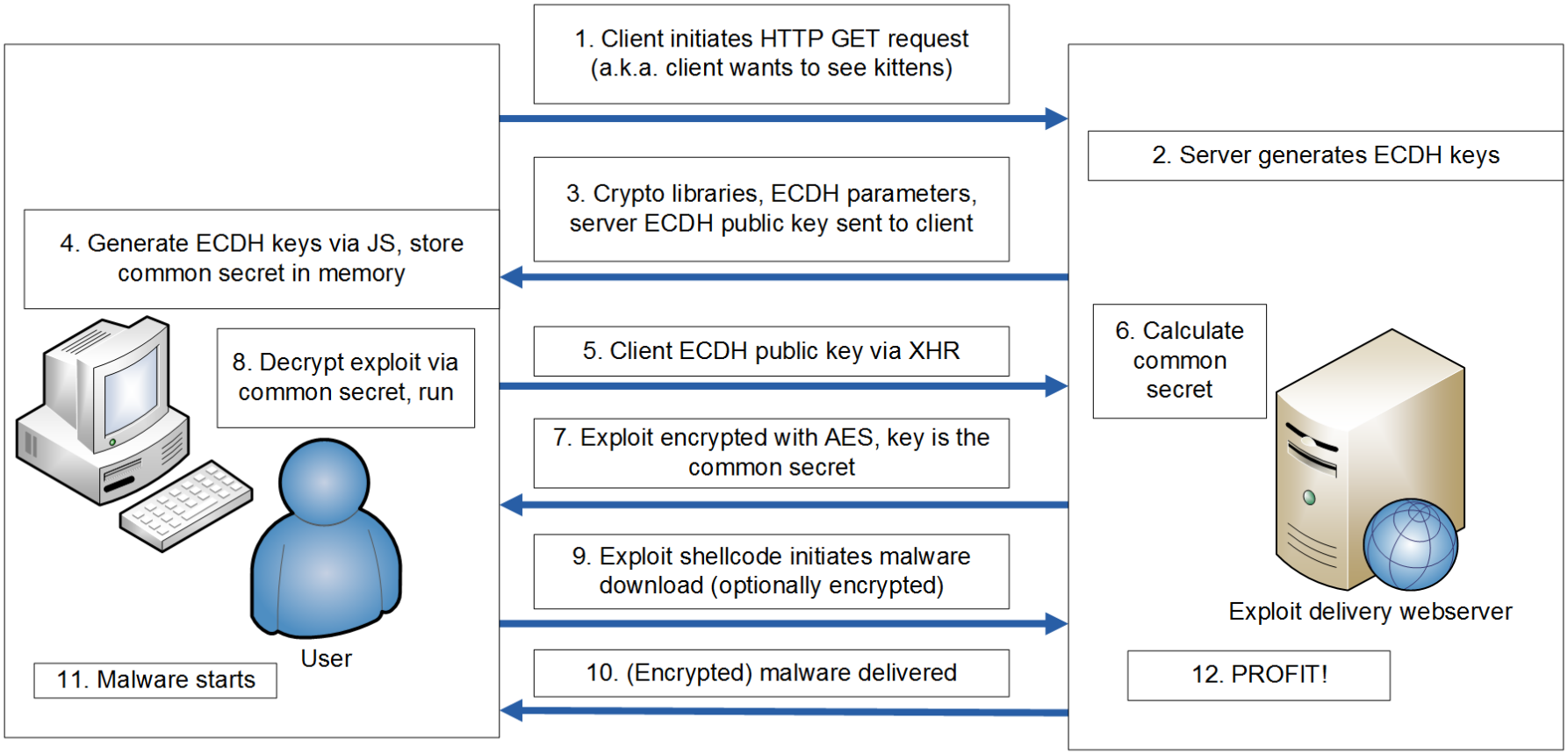
DH key agreement is too slow for JS

It is like you know the start and end position of the billiard ball on the table, but god knows the way it took to get there




http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/1/

# #IRONSQUIRREL



# Demo with test JavaScript in Chrome

chrome.mrg-effitas.com:2345/final.html

**Some random text**

Elements Console Sources Network Timeline Profiles

top Preserve log

```
Navigated to http://chrome.mrg-effitas.com:2345/
▶ The devicemotion event is deprecated on insecure origins, and support will be removed in the future. You should consider switching your application to a secure origin, such as HTTPS. See https://goo.gl/rStTGz for more details.   sjcl.js:1
GET http://chrome.mrg-effitas.com:2345/favicon.ico 404 (Not Found)   favicon.ico:1
Navigated to http://chrome.mrg-effitas.com:2345/final.html
> console.log(sjcl.decrypt(sessionStorage.SK_base64,'{"v":1,"adata":"","ks":256,"ct":"
afERIOOC03[illegible]T31qqSwNFsO12N4Mqe0O8E/ZHw5biKGFOqSiCq9/YB11hnJhoTJFBBjFKyW8liBO11dc8cGE
","ts":96,"mode":"gcm","cipher":"aes","iter":100000,"iv":"jzgH/Lf9jLbM6mlV","salt":"
q06Z7+/9CWw="}'));
alert('Malicious Javascript code is running here');   VM149:1
<· undefined
>
```

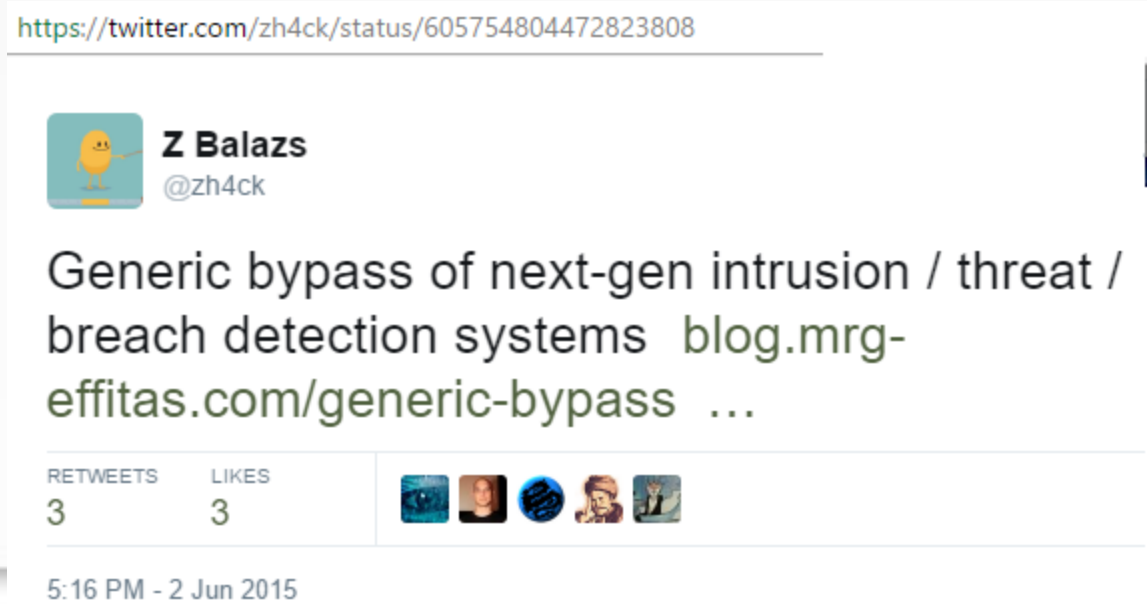
Console

# Implementation details

## Original Node.JS POC – 2 June, 2015

## New Ruby POC compatible and tested with

- Edge
- IE11 (older IE just sucks, can't crypto)
- Firefox (Tor Browser)
- Chrome
- Opera
- Mobile Safari
- Mobile Chrome
- Android built-in browser



# DH implemented in exploit kits

## FireEye analysis – Angler exploit kit

- "First" in-the-wild DH encrypted exploit
- Only shellcode was protected by encryption

https://www.fireeye.com/blog/threat-research/2015/08/cve-2015-2419_inte.html

### CVE-2015-2419 – Internet Explorer Double-Free in Angler EK

August 10, 2015 | by Sudeep Singh, Dan Caselden

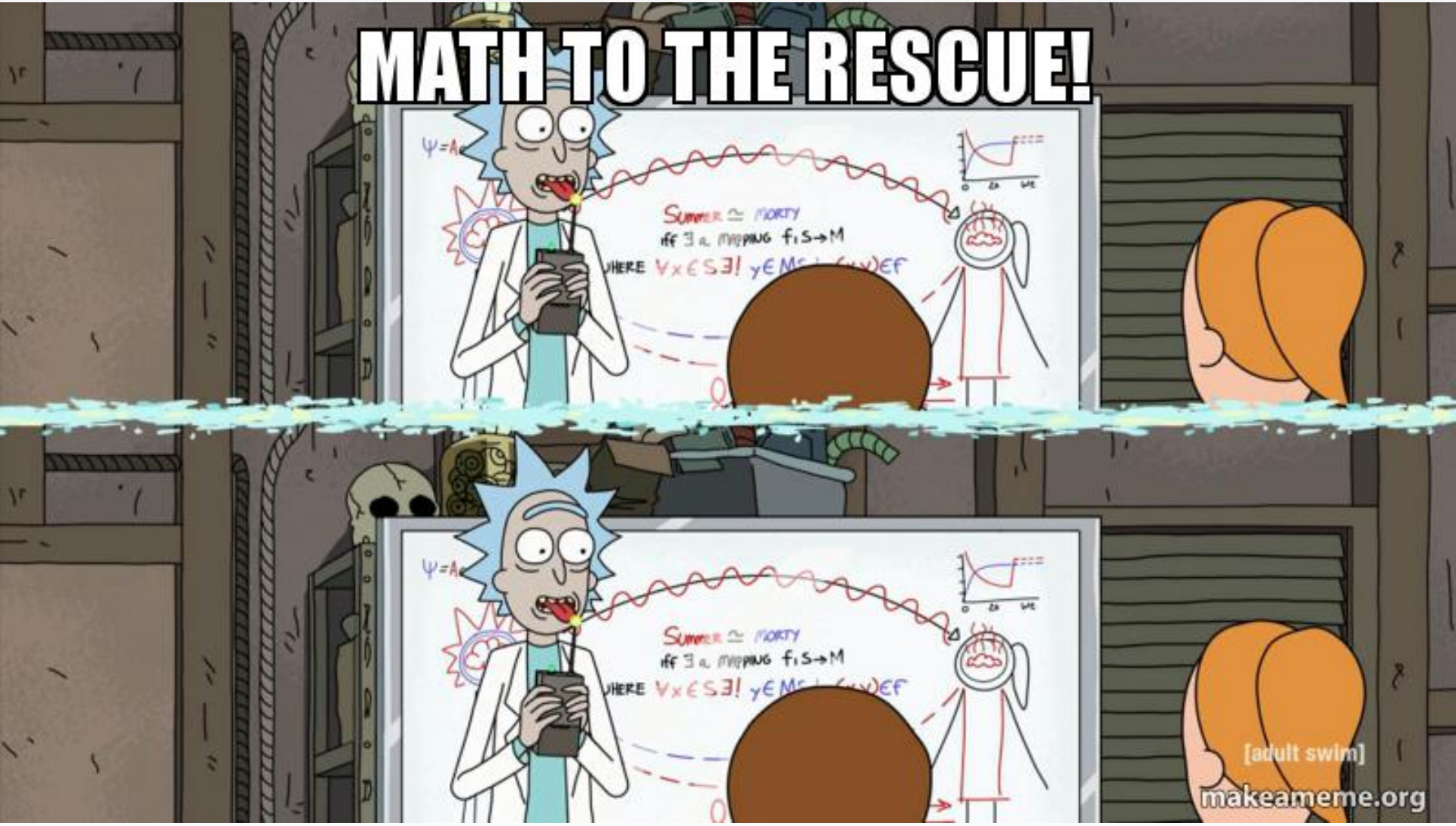
The Angler Exploit Kit (EK) recently added support for an Internet Explorer (IE) vulnerability (CVE-2015-2419) that was patched in July 2015. Quickly exploiting recently patched vulnerabilities is standard for Angler EK authors, but the target has been Adobe Flash Player since the second half of 2014. The exploitation of CVE-2015-2419 marks the second departure from Flash exploits for Angler (the first being the inclusion of CVE-2015-1671 in Silverlight). This may be the result of Adobe's recent exploit mitigations in Flash Player that prevent attackers from using Vector (and similar) objects to develop their control over corrupted Flash processes. To date, Angler will deliver Flash, IE, and/or Silverlight exploits depending upon the target's environment.

Angler also added a new obfuscation to its IE exploit. The landing page fetches a stub of keys and data necessary to run the exploit from the server each time it executes. The stub of information is only sent to victims that broadcast vulnerable browsers, and is protected with XTEA over a homebrew Diffie-Hellman.

### IE Exploit Delivery Protection using Diffie-Hellman Key Exchange

"You might think this is coincidental, but I assure you it is not …"

https://www.youtube.com/watch?v=XeDqGwQkDk8


MATH TO THE RESCUE!
[adult swim]
makeameme.org

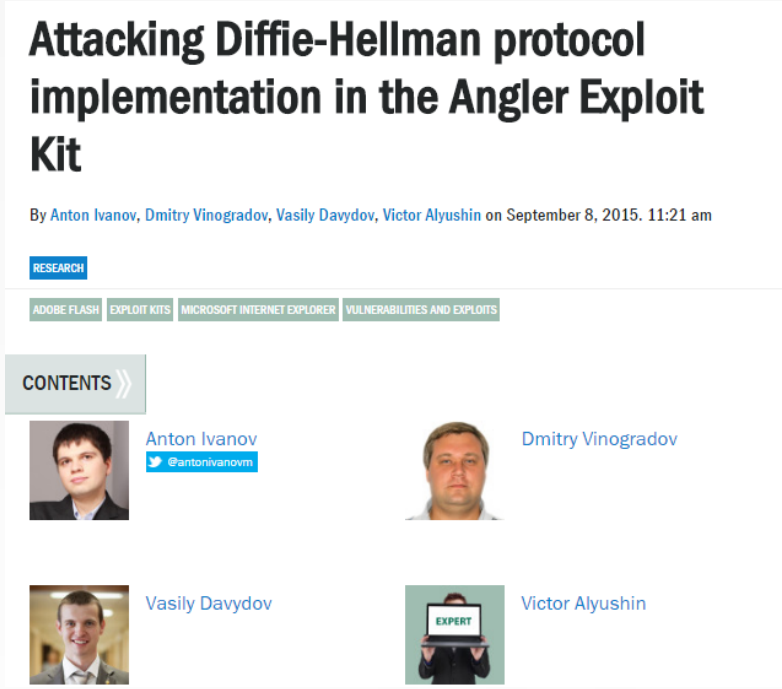
# DH implemented in exploit kits

"Several days ago analysts found the usage of the Diffie-Hellman cryptographic protocol in the Angler Exploit Kit, … that is the first known case of its usage in an exploit kit."

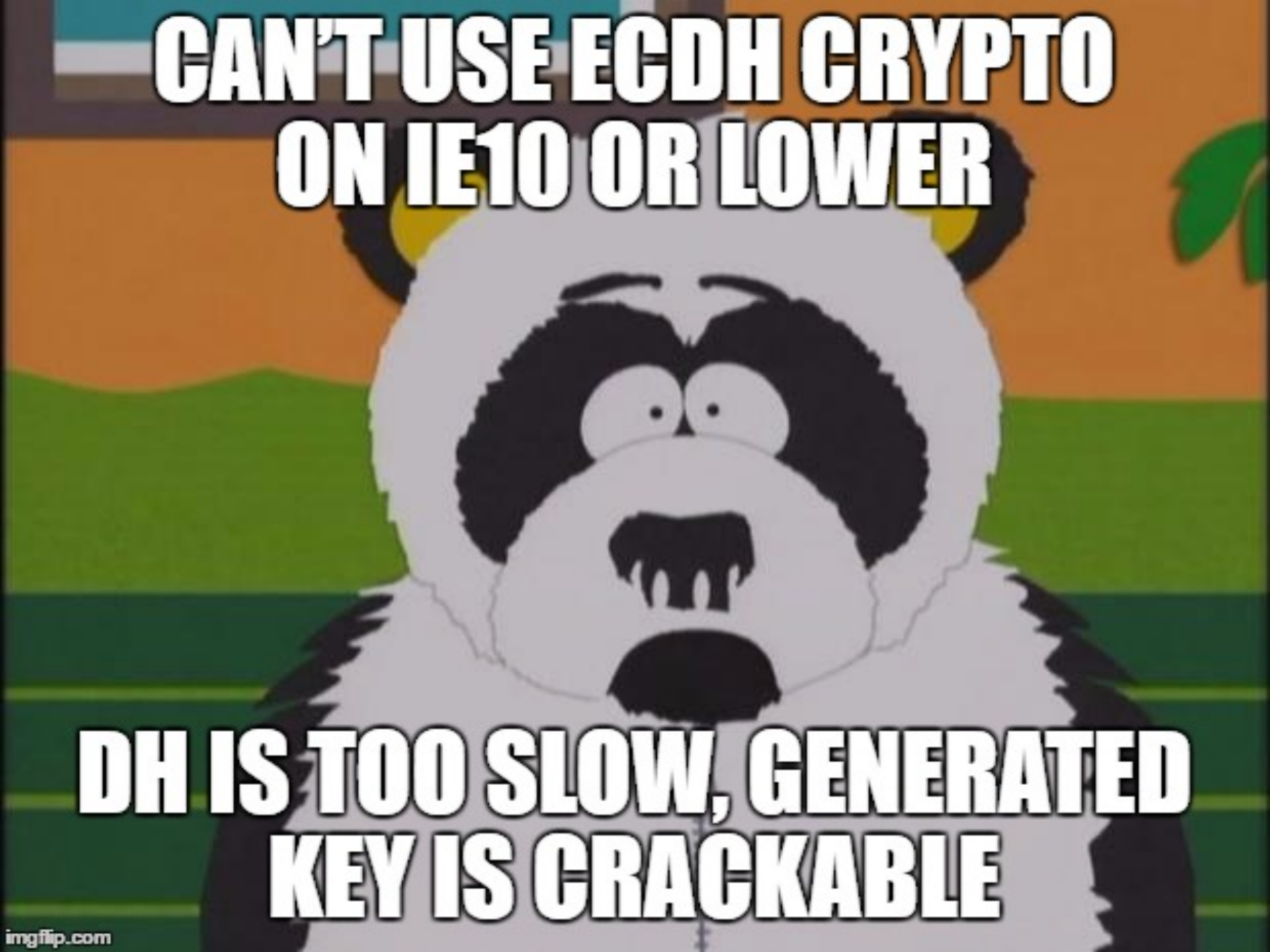
## Weakness demonstrations

- Use of DH instead of ECDH
- Short keys suspected to be factorized

2017 May: Astrum/Stegano exploit kit back with DH exploit delivery




https://securelist.com/blog/research/72097/attacking-diffie-hellman-protocol-implementation-in-the-angler-exploit-kit/


CAN'T USE ECDH CRYPTO ON IE10 OR LOWER
DH IS TOO SLOW, GENERATED KEY IS CRACKABLE
imgflip.com

## Attacker model

Who is my attacker?

- The reverse engineer (RE), who tries to reverse the precious 0-day exploit
- The nextgen/breach-detection system

What is the capability of the attacker?

- See next slides

# RE can record (and replay) network traffic

ECDH.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

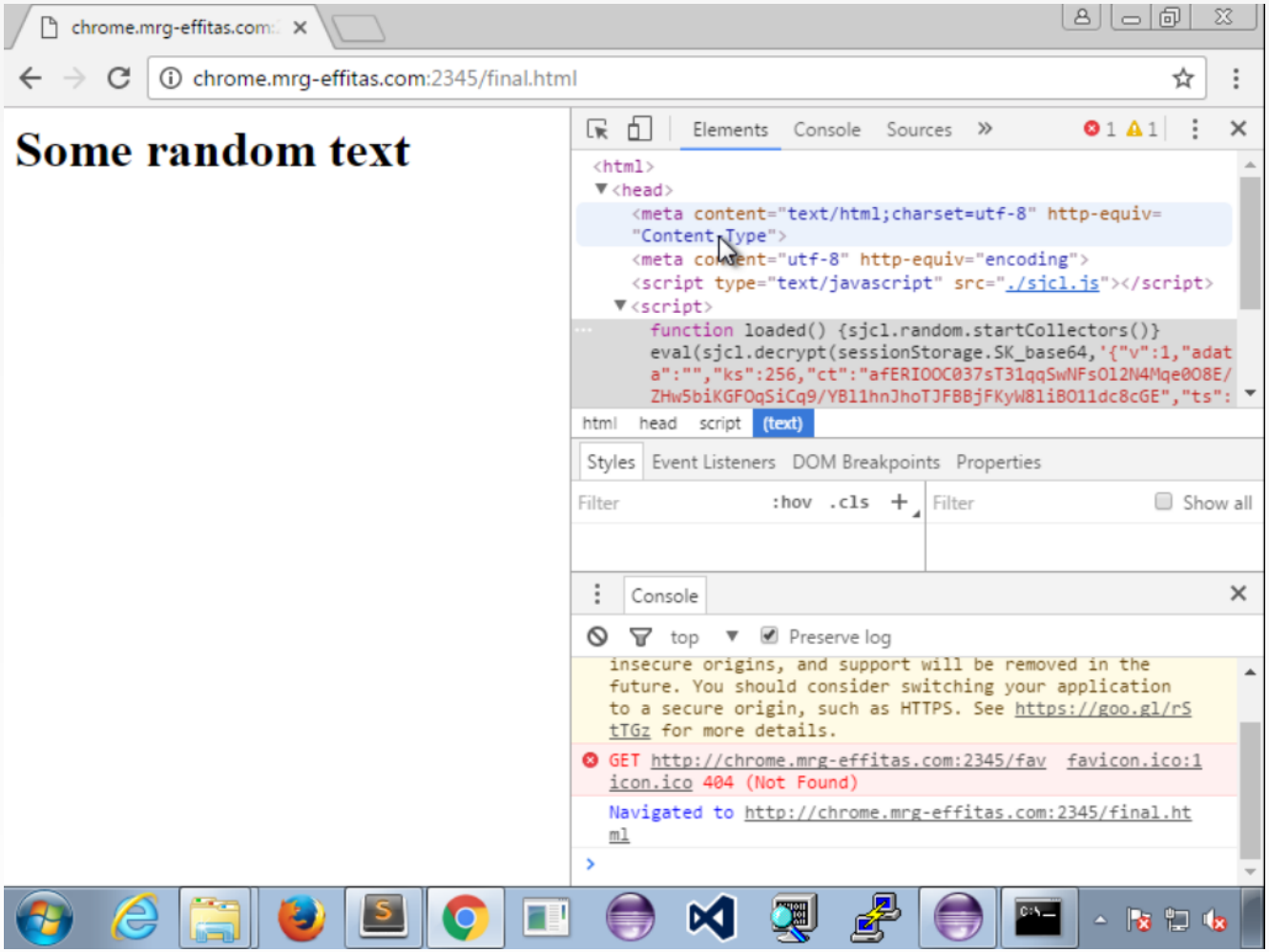
Apply a display filter ... <Ctrl-/> Expression...

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 501 | 2015-06-02 11:40:08.132061 | 192.168.56.101 | 192.168.56.1 | HTTP | 187 | HTTP/1.1 2 |
| 502 | 2015-06-02 11:40:08.190202 | 192.168.56.1 | 192.168.56.101 | TCP | 54 | 48923 → 80 |
| 503 | 2015-06-02 11:40:08.633239 | 192.168.56.1 | 192.168.56.101 | HTTP | 477 | GET /?bob_ |
| 504 | 2015-06-02 11:40:08.636336 | 192.168.56.101 | 192.168.56.1 | HTTP | 295 | HTTP/1.1 2 |
| 505 | 2015-06-02 11:40:08.683890 | 192.168.56.1 | 192.168.56.101 | TCP | 54 | 48921 → 80 |
| 506 | 2015-06-02 11:40:10.504095 | 192.168.56.1 | 192.168.56.101 | TCP | 54 | 48923 → 80 |
| 507 | 2015-06-02 11:40:10.504746 | 192.168.56.101 | 192.168.56.1 | TCP | 54 | 8080 → 489 |
| 508 | 2015-06-02 11:40:10.504994 | 192.168.56.1 | 192.168.56.101 | TCP | 54 | 48921 → 80 |
| 509 | 2015-06-02 11:40:10.505436 | 192.168.56.101 | 192.168.56.1 | TCP | 54 | 8081 → 489 |
| 510 | 2015-06-02 11:40:10.507462 | 192.168.56.101 | 192.168.56.1 | TCP | 54 | 8080 → 489 |

```
> Frame 503: 477 bytes on wire (3816 bits), 477 bytes captured (3816 bits) on interface 0
> Ethernet II, Src: CadmusCo_00:1c:6e (08:00:27:00:1c:6e), Dst: CadmusCo_01:2b:4a (08:00:27:01:2b:4a)
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.101
> Transmission Control Protocol, Src Port: 48921 (48921), Dst Port: 8081 (8081), Seq: 281, Ack: 459, Len: 423
v Hypertext Transfer Protocol
  > GET /?bob_pub=048be73feaeb9eaa2fa748d350e880d8465ac0337cbe3550ab2328d961d8e2116a855b3619bb49fb80128a0df6176a755361b6e9a85affa7c3
    Accept: */*\r\n
```

```
0000  08 00 27 01 2b 4a 08 00  27 00 1c 6e 08 00 45 00   ..'.+J.. '..n..E.
0010  01 cf 27 70 40 00 80 06  e0 01 c0 a8 38 01 c0 a8   ..'p@... ....8...
0020  38 65 bf 19 1f 91 59 bf  bc 7d f7 67 38 9a 50 18   8e....Y. .}.g8.P.
0030  00 fe e0 77 00 00 47 45  54 20 2f 3f 62 6f 62 5f   ...w..GE T /?bob_
0040  70 75 62 3d 30 34 38 62  65 37 33 66 65 61 65 62   pub=048b e73feaeb
0050  39 65 61 61 32 66 61 37  34 38 64 33 35 30 65 38   9eaa2fa7 48d350e8
0060  38 30 64 38 34 36 35 61  63 30 33 33 37 63 62 65   80d8465a c0337cbe
0070  33 35 35 30 61 62 32 33  32 38 64 39 36 31 64 38   3550ab23 28d961d8
0080  65 32 31 31 36 61 38 35  35 62 33 36 31 39 62 62   e2116a85 5b3619bb
0090  34 39 66 62 38 30 31 32  38 61 30 64 66 36 31 37   49fb8012 8a0df617
00a0  36 61 37 35 35 33 36 31  62 36 65 39 61 38 35 61   6a755361 b6e9a85a
00b0  66 66 61 37 63 33 32 63  30 34 31 35 61 33 65 37   ffa7c32c 0415a3e7
```

# RE can debug in browser – JavaScript level
## Has access to DOM in browser



# RE can debug the browser – Assembly level

## This is not always trivial – e.g. if you can't jailbreak iOS

Pid 3764 - WinDbg:6.12.0002.633 X86

File Edit View Debug Window Help

Command - Pid 3764 - WinDbg:6.12.0002.633 X86

```
[+] Command used:
!py mona.py find -type asc -s var MyEgG -unicode

---------- Mona command started on 2017-03-04 12:12:46 (v2.
[+] Processing arguments and criteria
    - Pointer access level : *
    - Expanded ascii pattern to unicode, switched search mo
    - Treating search pattern as bin
[+] Searching from 0x00000000 to 0x7fffffff
[+] Preparing output file 'find.txt'
    - (Re)setting logfile find.txt
[+] Generating module info table, hang on...
    - Processing modules
    - Done. Let's rock 'n roll.
    Found a total of 0 pointers

[+] This mona.py action took 0:00:09.672000
```

0:018>

Calls - Pid 3764 - WinDbg:6.12.0002.633 X86

Raw args Func info Source Addrs Headings Nonvolatile regs Frame nums

Source args

More Less

```
MSHTML!CAryWindowTbl::SafeIterateProxies
MSHTML!COmWindowProxy::SwitchMarkup+0x710
MSHTML!CDocument::open+0x538
MSHTML!CDocument::WriteInternal+0x4bb
MSHTML!CDocument::Var_write+0x48
MSHTML!CFastDOM::CDocument::Trampoline_write+0x91
jscript9!Js::JavascriptExternalFunction::ExternalFunctionTh
jscript9!Js::InterpreterStackFrame::Process+0x1940
jscript9!Js::InterpreterStackFrame::InterpreterThunk<1>+0x1
WARNING: Frame IP not in any known module. Following frames
0x3bd0fe9
jscript9!Js::JavascriptFunction::CallFunction<1>+0x91
```

Disassembly - Pid 3764 - WinDbg:6.12.0002.633 X86

Offset: @$scopeip Previous Next

```
61cd9e23 61              popad
61cd9e24 d059d3          rcr     byte ptr [ecx-2Dh],1
61cd9e27 61              popad
MSHTML!CDOMEvent::`vftable':
61cd9e28 90              nop
61cd9e29 90              nop
61cd9e2a 90              nop
61cd9e2b 90              nop
61cd9e2c 90              nop
MSHTML!CAryWindowTbl::SafeIterateProxies:
61cd9e2d 8bff            mov     edi,edi
61cd9e2f 55              push    ebp
61cd9e30 8bec            mov     ebp,esp
61cd9e32 83ec14          sub     esp,14h
61cd9e35 53              push    ebx
61cd9e36 56              push    esi
61cd9e37 57              push    edi
61cd9e38 33c0            xor     eax,eax
61cd9e3a 8d7dec          lea     edi,[ebp-14h]
```

Registers - Pid 3764 - WinDbg:6.12.0002.633 X86

Customize...

| Reg | Value |
|---|---|
| gs | 0 |
| fs | 3b |
| es | 23 |
| ds | 23 |
| edi | 549b318 |
| esi | 27e5a30 |
| ebx | 3b4d888 |
| edx | 592840 |
| ecx | 3a989c8 |
| eax | 3aa35f8 |
| ebp | 549b3b0 |
| eip | 61cd9e2d |

Ln 0, Col 0 | Sys 0:<Local> | Proc 000:eb4 | Thrd 018:774 | ASM | OVR | CAPS | NUM

# Network forensics

When checking IRONSQUIRREL network traffic, you see

- Bunch of crypto libraries
- Public key exchange
- Encrypted blobs
  - Without the shared key, you can't do much
    - Unless you have a kick-ass quantum computer
    - Attackers: just use quantum resistant key exchange

Debugging in browser is possible – but I will recommend some tricks to make this harder

Why is this different, new?

Protecting the browser exploit code was so far obfuscation only

- It was encryption with keys known to the attacker
- Now, it is encryption with keys not know to the attacker

Why is this different then SSL/TLS ?

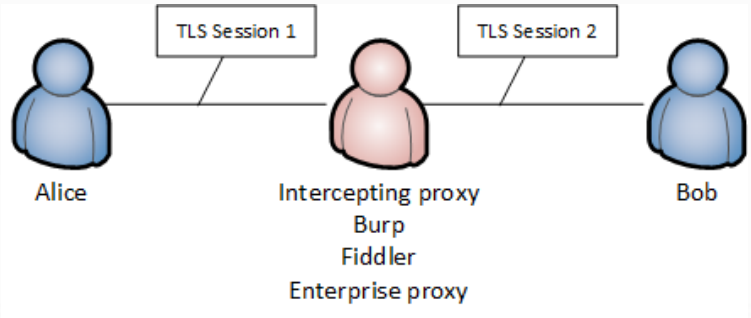
How does this affect exploit replay?

# IRONSQUIRREL exploit delivery VS exploit kits using SSL/TLS

If you control the client (the analysis machine), TLS MiTM is trivial

Deep Packet Inspection

- TLS MiTM at enterprises
- TLS MiTM with intercept proxies like Burp or Fiddler at home or your lab



## Traditional browser exploits forensics

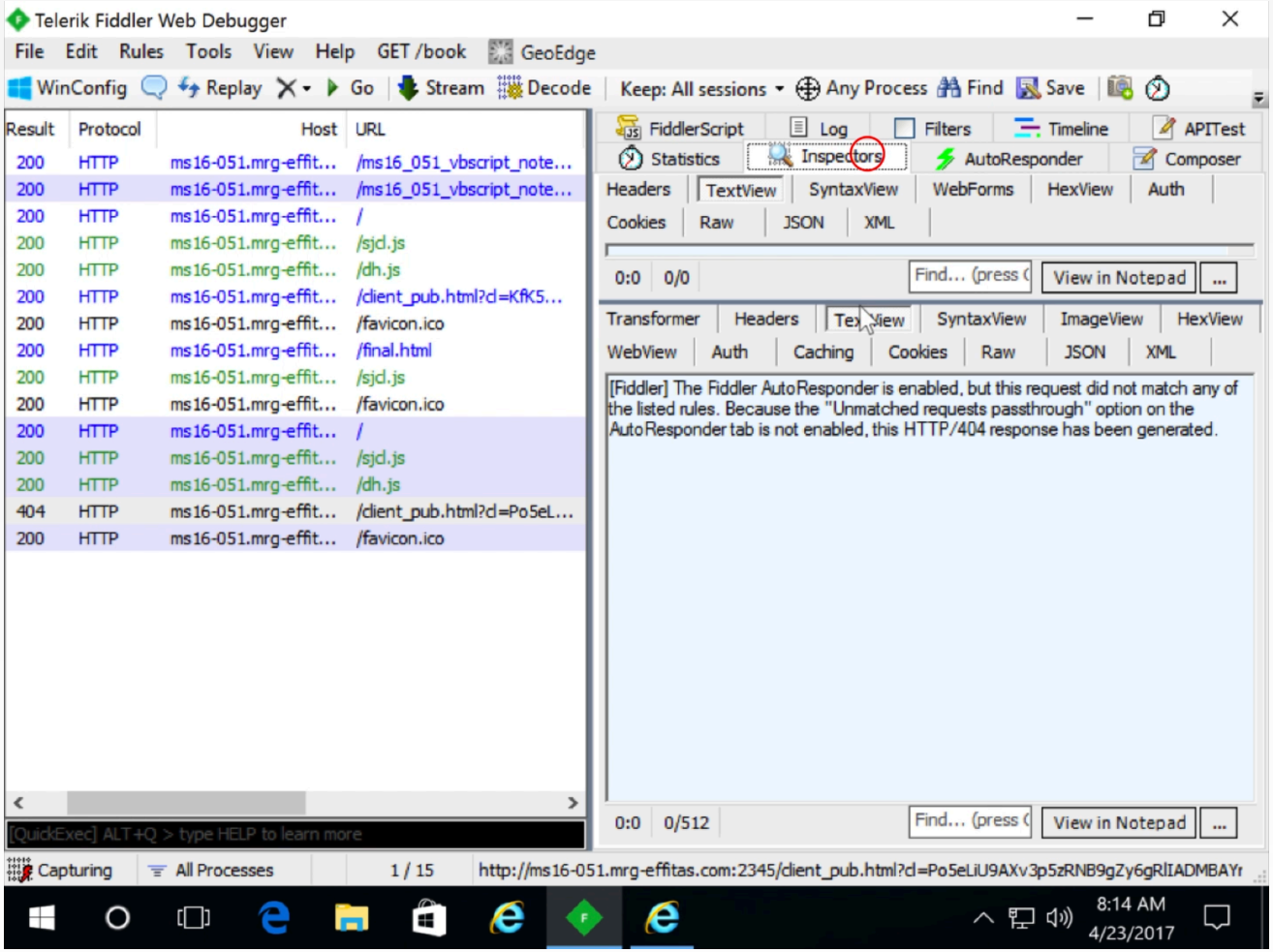
Reproducible exploit replay with Fiddler or similar

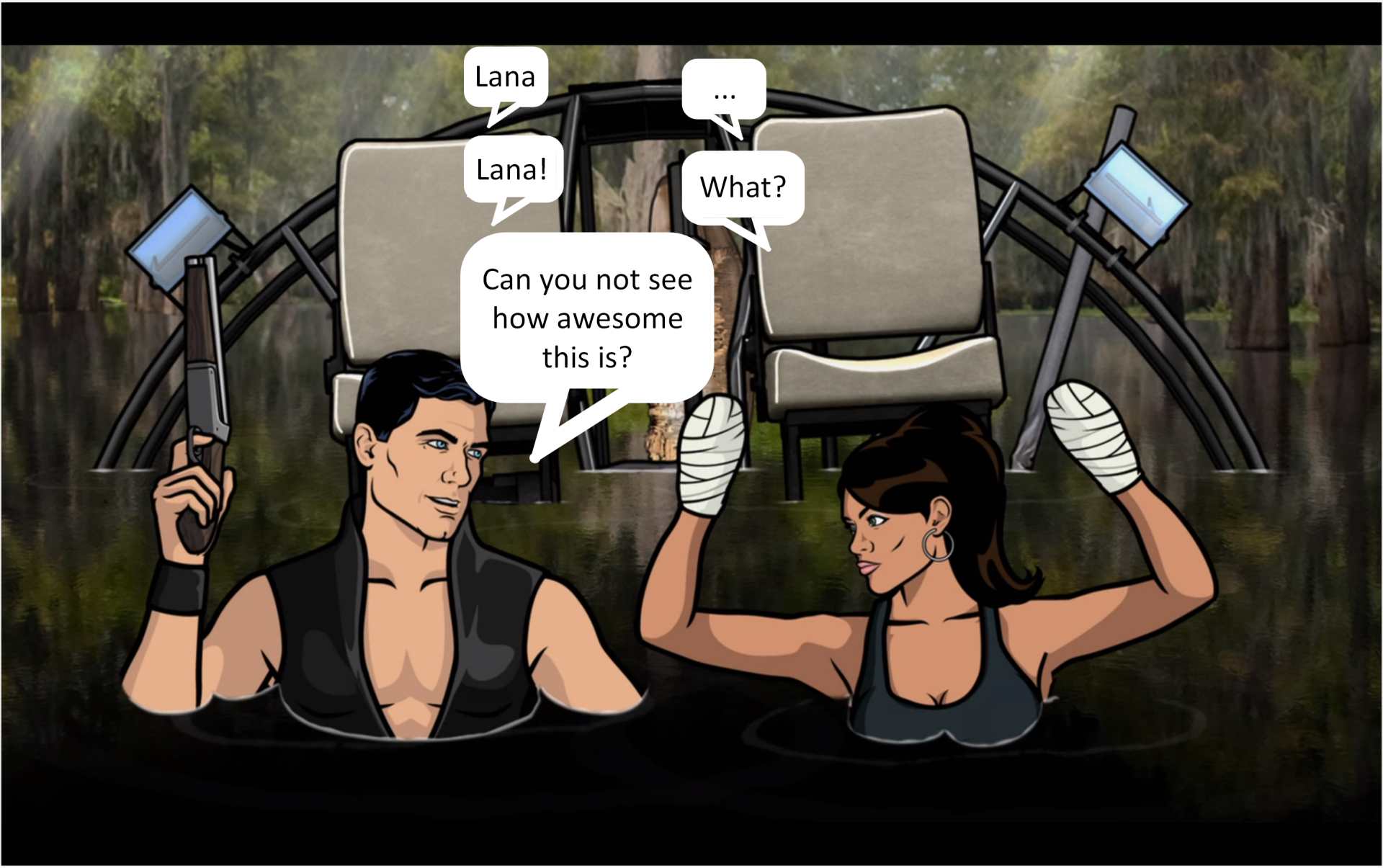
SSL/TLS exploit delivery can be replayed if MiTM is possible

IRONSQUIRREL exploit delivery cannot be replayed

- The client will generate different public/private key
- Client will send different public key to replay server
- Replay server either sends the encrypted data with the old key, or can't generate new ECDH key thus fails to replay

# Exploit replay with and without IRONSQUIRREL




Lana
Lana!
Can you not see how awesome this is?
...
What?

# Astrum EK replay broken

| Result | Protocol | Host | URL | Comments | Body | Content-Type |
|---|---|---|---|---|---|---|
| 200 | HTTP | define.predatorhuntingusa.com | /s_u1w_/gl089yt3p-eh-zby1dru3h_0sz8h-fpfy8evte5xm1/cwi06y | Astrum Exploit Kit | 4,066 | text/html;charset=UTF-8 |
| 200 | HTTP | define.predatorhuntingusa.com | /ky3ai7qw-ezr947i5ub9rsf9f0c1wl8xdbmyd7gtlnu50r325p_1yeo8?q=eyJnIjoieDg2IiwiYiI6IjUu... | Astrum Exploit Kit | 3,869 | text/html;charset=UTF-8 |
| 200 | HTTP | define.predatorhuntingusa.com | /aoheprazfjtlxrsjhoi/3755417082/i/gzohiswl916/698385664/mesp/79u7gd5_svuey | Astrum Exploit Kit | 19,914 | application/x-shockwave-flash |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=fl%20cr | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /oxamqjtprng/1629277540/qy75spcwe2dv12f5/2540393886/5kmghnc/q.gif?p=FUNYzJxoima... | Astrum Exploit Kit | 52,211 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=fl%20hd | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=f1 | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /xjoaoprjuzd/1904034186/rov1toas564mzej/2265635184/3r2tlid5ubsi/d9px.gif | Astrum Exploit Kit | 7,258 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /sxzpgabcvwmgr/2510716582/4gu5v4c_b016108c/9/1674700380/ncgczl4le_j5vbo.gif | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=sp1 | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=dt | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=sp2 | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=jsb | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=sc | Astrum Exploit Kit | 42 | image/gif |
| Result | Protocol | Host | URL | Comments | Body | Content-Type |
| 200 | HTTP | define.predatorhuntingusa.com | /s_u1w_/gl089yt3p-eh-zby1dru3h_0sz8h-fpfy8evte5xm1/cwi06y | Astrum Exploit Kit | 4,066 | text/html;charset=UTF-8 |
| 200 | HTTP | define.predatorhuntingusa.com | /ky3ai7qw-ezr947i5ub9rsf9f0c1wl8xdbmyd7gtlnu50r325p_1yeo8?q=eyJnIjoieDg2IiwiYiI6IjUu... | Astrum Exploit Kit | 3,869 | text/html;charset=UTF-8 |
| 200 | HTTP | define.predatorhuntingusa.com | /aoheprazfjtlxrsjhoi/3755417082/i/gzohiswl916/698385664/mesp/79u7gd5_svuey | Astrum Exploit Kit | 19,914 | application/x-shockwave-flash |
| 200 | HTTP | define.predatorhuntingusa.com | /ngrcpxr/930292396/0qgzems5hn8vxy1_ifqb/3238312534/6ih24qv3rf_he6.gif?a=fl%20cr | Astrum Exploit Kit | 42 | image/gif |
| 200 | HTTP | define.predatorhuntingusa.com | /oxamqjtprng/1629277540/qy75spcwe2dv12f5/2540393886/5kmghnc/q.gif?p=TwajBZHE6aB... | Astrum Exploit Kit | 52,211 | image/gif |
| 502 | HTTP | define.predatorhuntingusa.com | /mudizbyo/3594307092/iq12p_xcf9ij2v/541810414/b2xh7.gif?g=fl%20cr%20dec%20err | Failed Decrypt | 166 | text/html |

Fun fact: even if exploit is not 0-day,
other threat groups can't steal your exploit
code

http://blog.trendmicro.com/trendlabs-security-intelligence/astrum-exploit-kit-abuses-diffie-hellman-key-exchange/

# IRONSQUIRREL exploit delivery VS Stegosploit

"Stegosploit creates a new way to **encode** "drive-by" browser exploits and deliver them through image files" … "image based exploit delivery - Steganography and Polyglots"

Stegosploit is good at hiding your exploit. But it is replayable, thus easy to analyse once recorded/identified

http://stegosploit.info/

It is possible to combine Stegosploit

with IRONSQUIRREL

# IRONSQUIRREL exploit delivery VS Heartbleed

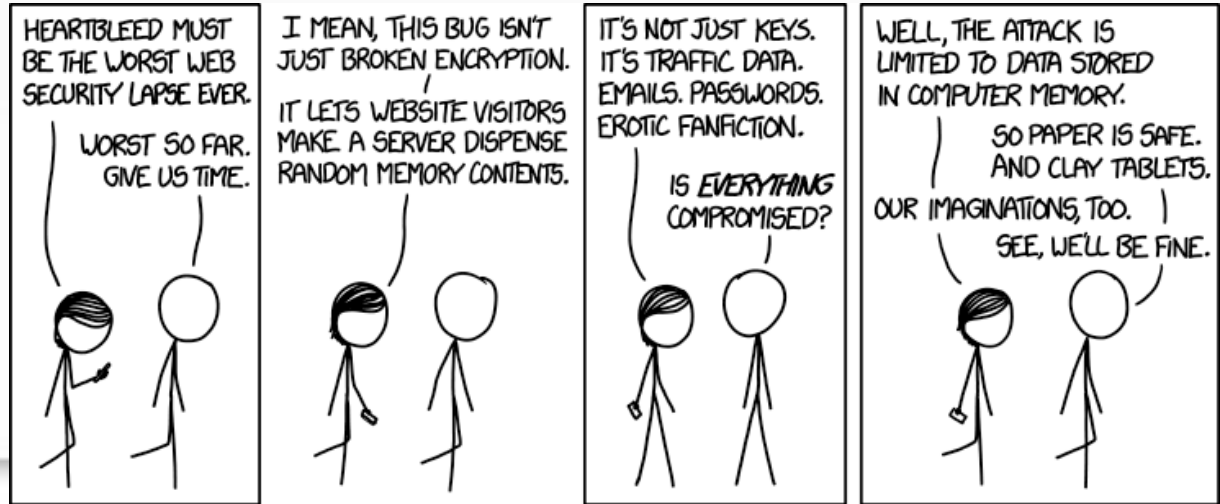
TLS Heartbeet can be sent either

- In clear-text before handshake finished
- Encrypted, after handshake

It is harder to create IDS signatures for the encrypted payload.

Heartbleed exploit uses encryption as part of the protocol.

IRONSQUIRREL exploit delivery uses encryption as an additional module to make reversing harder



Defense and offense

Prevention and detection on the network level

Analysis on the endpoint

How to make endpoint analysis (a lot) harder

# Anti-analysis improvements

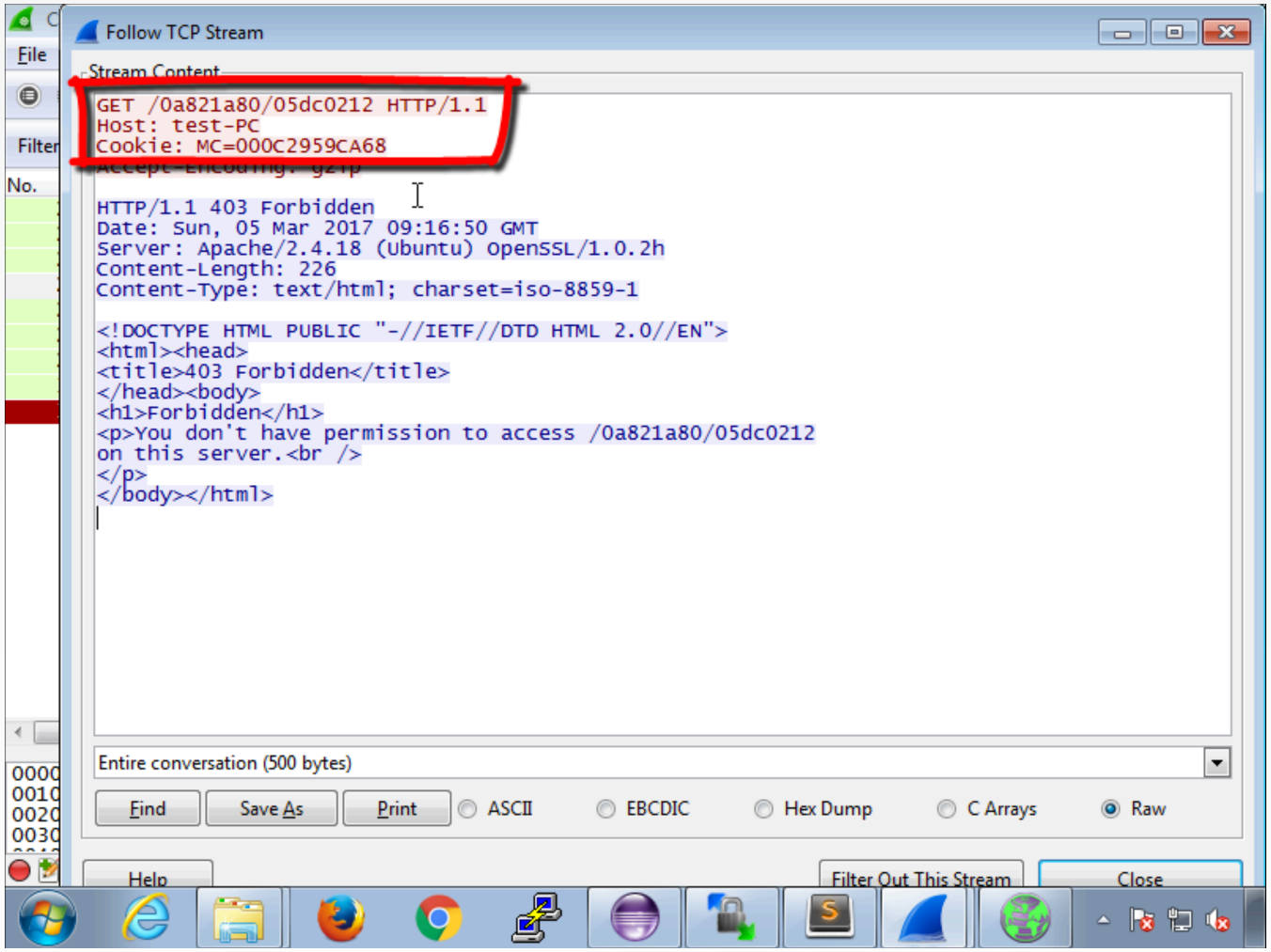
One-time URLs (URL is dead after one use) **Implemented**

- In Law Enforcement mode, use one-time URL per logged in user!

Time-limits to prevent manual debugging **Implemented**

Remove full DOM after exploit runs **Implemented**

# Case study – Tor browser exploit



Prevent the IRONSQUIRREL exploit attacks via network defenses

IRONSQUIRREL specific blocking/detection

- Detection of (EC)DH encrypted traffic
- Will lead to False Positives (FP)

Non IRONSQUIRREL specific blocking/detection

- Block uncategorized/new domains
- Domain white-listing

# Web ISOLATION

Web Isolation is

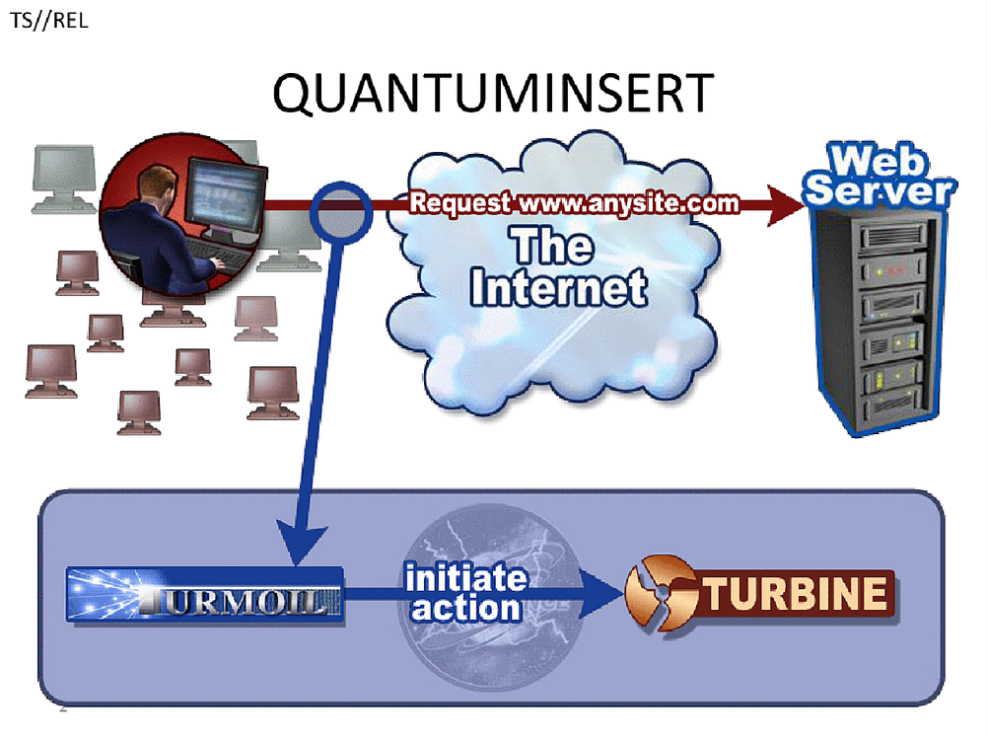
- Something like a proxy
- Code runs on a remote server
- Rendered data is forwarded to client browser
- Exploit code "runs" on remote server

- Tested, it blocked Firefox and IE exploits
  - If you have Chrome 0-day targeting Linux, let me know

# Delivery method improvements

To bypass uncategorized/new domain prevention/detection

- Use of watering hole
- Quantum insert techniques
  - Warning, might not be available in your attacker capability



## Analyze IRONSQUIRREL exploits on the endpoint

Log the shared key and/or client private key

"Fix" the random generator – generate same client private keys always

"Hook" the JS code to immediately return with the same client secret key

Remote debugging iOS Safari on OS X

Detailed JS execution Tracelog

- https://github.com/szimeus/evalyzer
  --> check out this great project!

# Evalyzer MS16-051 demo

C:\Program Files\Debugging Tools for Windows (x86)\ntsd.exe

```
*** ERROR: Symbol file could not be found.  Defaulted to export symbol
s for C:\Windows\system32\vm3dum_10.dll -
Couldn't resolve error at 'rto'
0:020> .foreach /s (exc "epr sbo sov gp ii av") {sxe ${exc}}
0:020> sxe -c ".logclose" -h epr
0:020> bu jscript9!Js::ScriptContext::IsInEvalMap ".echo EVAL(dyn)----
-;.printf \"%mu\", poi(esp+0x18);.echo;g"
0:020> bu MSHTML!CHTMLoad::Write ".echo DOCUMENT.WRITE(dyn)-----;.prin
tf \"%mu\",poi(esp+0x4);.echo;g"
0:020> g
EVAL(dyn)-----

var x = "Hello World! :)"

DOCUMENT.WRITE(dyn)-----

<script type="text/vbscript">
        Dim aw
        Dim plunge(32)
        Dim y(32)
        prefix = "%u4141%u4141"
        d = prefix & "%u0016%u4141%u4141%u4141%u4242%u4242"
```

# Anti-analysis improvements

Detect debug window (client-side protection ☹ )
https://github.com/zswang/jdetects


Proper fingerprinting of the target before exploit delivery


Code obfuscation – effective against MiTM *


Generate multiple DH private keys and check if it is the same

Implemented

* http://blog.trendmicro.com/trendlabs-security-intelligence/how-exploit-kit-operators-are-misusing-diffie-hellman-key-exchange/

# Anti-analysis improvements

Adding lot of junk code to DoS the analysis environment


Use eval equivalent functions like SetTimeout, new Function(), ... to bypass default Evalyzer

https://www.slideshare.net/x00mario/in-the-dom-no-one-will-hear-you-scream

# Conclusion of the RE attacker

Determined RE engineer can restore exploit from a memory dump

Determined attacker can put breakpoints on DEP related VirtualProtects or use Guard Pages, and reverse the vulnerability *

But it can delay the analysis/discovery of the exploit by days/weeks/months if the attacker implements my suggestions

* Windows only method

# Conclusion of the RE attacker

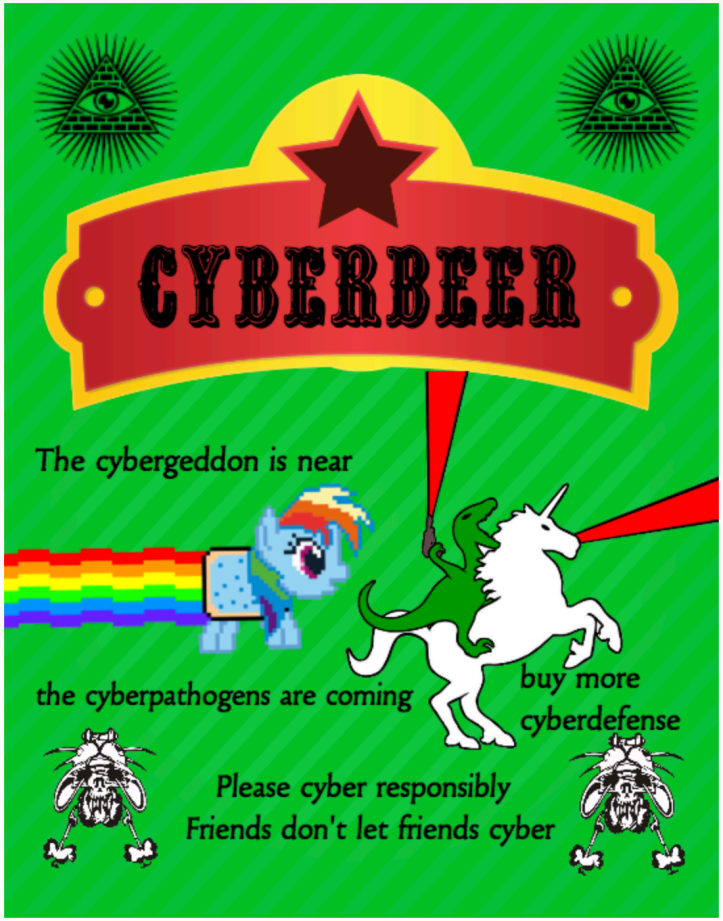

Determined RE engineer can restore exploit from a memory dump

Determined attacker can put breakpoints on DEP related VirtualProtects or use Guard Pages, and reverse the vulnerability *

But it can delay the analysis/discovery of the exploit by days/weeks/months if the attacker implements my suggestions

* Windows only method

# Hacker and Cyber Pschorr (limited edition!)




# Chain the IRONSQUIRREL exploit to malware execution

Encrypted malware payload delivery

Target aware malware payload

Implemented

- Gauss - https://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/kaspersky-lab-gauss.pdf
- Ebowla - https://github.com/Genetic-Malware/Ebowla

All the "anti" stuff

- Anti-debug
- Anti-memory forensics
- Anti-disassemble
- Anti-sandbox
- Anti-dump
- Anti-trace

KIM ZETTER SECURITY 08.14.12 9:00 AM

RESEARCHERS SEEK HELP CRACKING GAUSS MYSTERY PAYLOAD

```
43 00 3A 00  5C 00 44 00  6F 00 63 00  75 00 6D 00  C.:.\.D.o.c.u.m.
65 00 6E 00  74 00 73 00  20 00 61 00  6E 00 64 00  e.n.t.s. .a.n.d.
20 00 53 00  65 00 74 00  74 00 69 00  6E 00 67 00   .S.e.t.t.i.n.g.
73 00 5C 00  6A 00 6F 00  68 00 6E 00  5C 00 4C 00  s.\.j.o.h.n.\.L.
6F 00 63 00  61 00 6C 00  20 00 53 00  65 00 74 00  o.c.a.l. .S.e.t.
74 00 69 00  6E 00 67 00  73 00 5C 00  41 00 70 00  t.i.n.g.s.\.A.p.
70 00 6C 00  69 00 63 00  61 00 74 00  69 00 6F 00  p.l.i.c.a.t.i.o.
6E 00 20 00  44 00 61 00  74 00 61 00  5C 00 47 00  n. .D.a.t.a.\.G.
6F 00 6F 00  67 00 6C 00  65 00 5C 00  43 00 68 00  o.o.g.l.e.\.C.h.
72 00 6F 00  6D 00 65 00  5C 00 41 00  70 00 70 00  r.o.m.e.\.A.p.p.
6C 00 69 00  63 00 61 00  74 00 69 00  6F 00 6E 00  l.i.c.a.t.i.o.n.
7E 00 64 00  69 00 72 00  31 00 97 48  6C AA 22 5F  ~.d.i.r.1..Hl."_
E8 77 C0 35  CC 03 73 23  6D 51                     .w.5..s#mQ
```

A string pair from the Gauss malware. *Image courtesy of Kaspersky Lab*

# Current Metasploit integration level

Pre-alpha (a.k.a non-existent) version 0.0

- Run Metasploit with (fake) victim
- Extract HTML file (now the exploit is static)
- Put extracted HTML into exploit folder
- Run IRONSQUIRREL with the HTML file

Need help!

Is there a logo???

This is not a vulnerability

Logos are lame

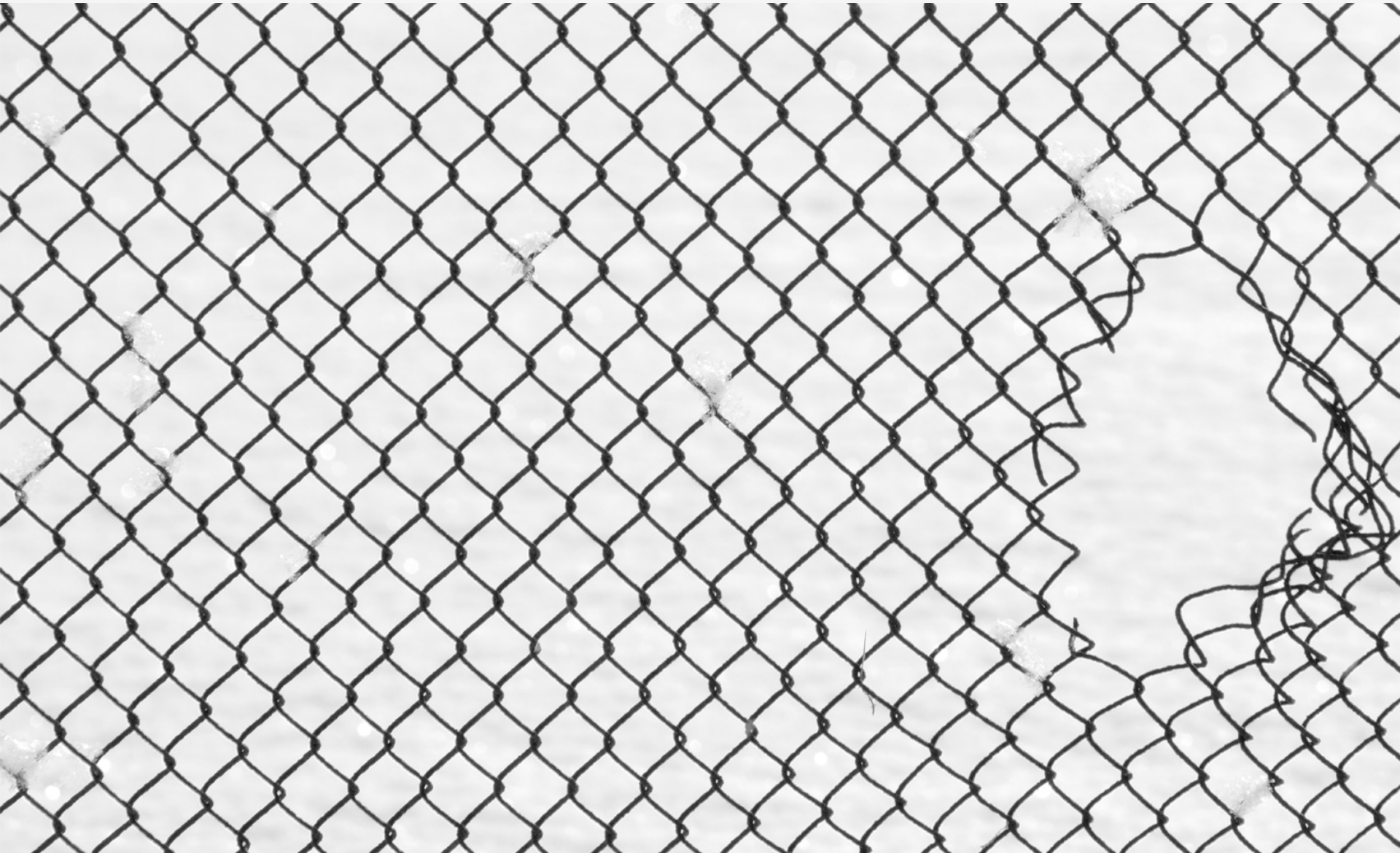
So the logical answer is that there is no logo

Hell yeah I made a logo ☺

Code publish

Perimeter security is dying

Mobile devices and encryption trends



# Conclusion

IRONSQUIRREL could have prevented the leak of the iOS Safari 0-day

IRONSQUIRREL could have prevented (or significantly delay) the leak of the Tor Browser 0-day

IRONSQUIRREL with one-time exploits can make RE a nightmare

IRONSQUIRREL does not deal with endpoint exploit protections (EMET)

OPSEC is important

# Ethical dilemmas

Why do I help the "bad" guys?

Who are the bad guys?

- Neither offense nor defense is bad by itself
  - I consider the FBI being the good guys if they are catching the pedophiles
- It is all about evolution
  - Have better defense or offense than the others to survive

I agree that the current laws are not prepared for law enforcement hacking of Tor users

What happens if we don't prepare our defenses against these attacks?

# Hack the planet!

https://github.com/MRGEffitas/Ironsquirrel

zoltan.balazs@mrg-effitas.com

https://hu.linkedin.com/in/zbalazs

Twitter – @zh4ck

www.slideshare.net/bz98

HACKERSULI !!!1!

Greetz to @CrySySLab, @SpamAndHex, @midnite_runr,@buherator, @sghctoma, @zmadarassy, @DavidSzili, @xoreipeip, @theevilbit, @molnar_g, Szimeus

https://JumpESPJump.blogspot.com


DEEPSEC
IN-DEPTH SECURITY