Malware Analysis & Removal Tool

Gruppo 2
(CaMail team)

Francesco Carotenuto
Angelo D'Amato
Antonio Eletto
Dario Scarpa

Attività

- *Analisi*
  - Osservazione attività anomale
  - Identificazione e terminazione dei processi maliziosi
  - Identificazione e rimozione del meccanismo di avvio automatico
  - Individuazione e analisi sommaria dei files coinvolti
  - Ricostruzione della logica del malware

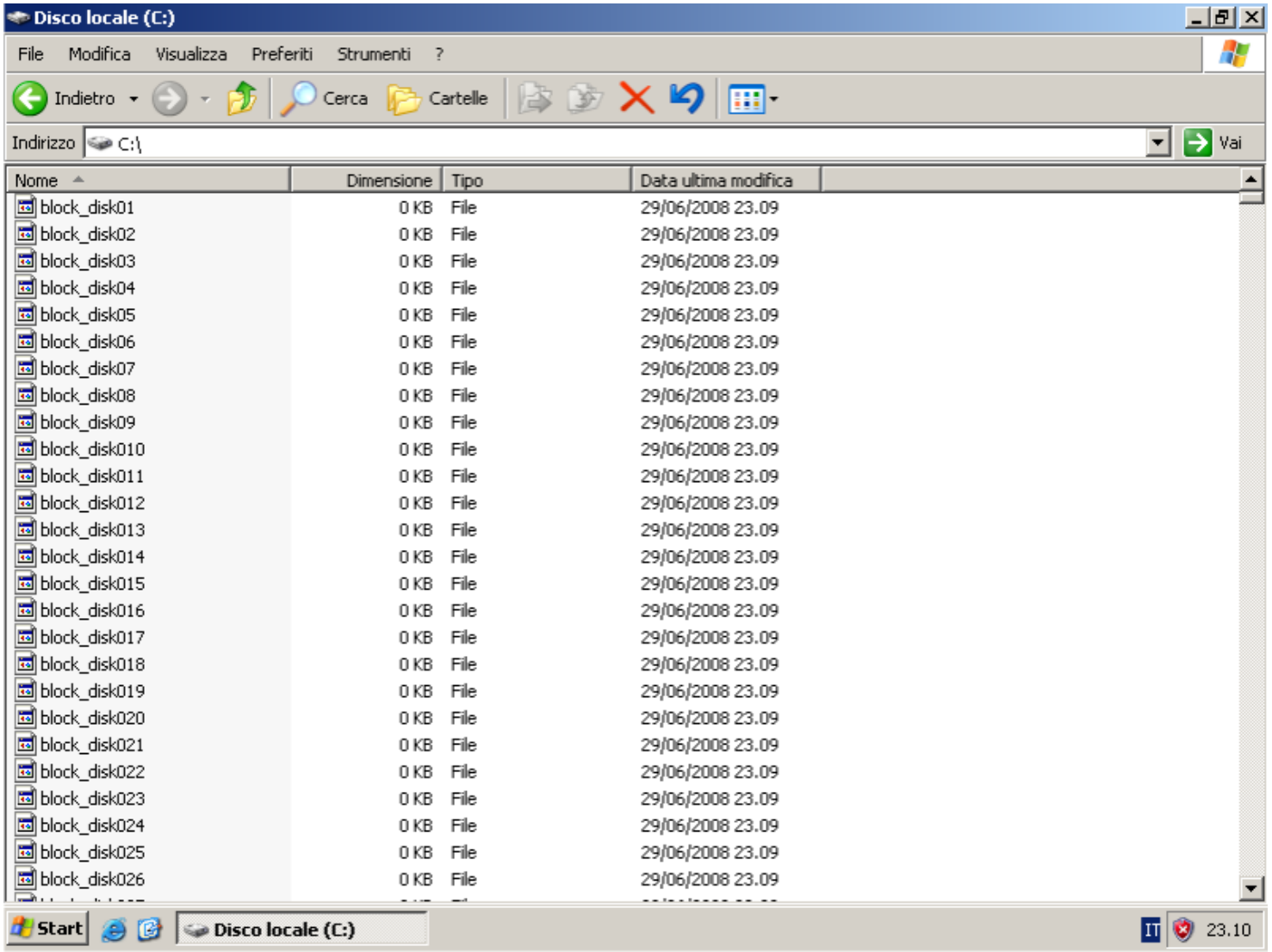
- *Rimozione*
  - Rimozione manuale e revert delle operazioni indesiderate effettuate dal malware
  - Sviluppo del removal tool
  - Reinstallazione del malware e test del removal tool

Osservazione attività anomale

- Attività su disco subito dopo il login
- Fake Popups
  - Un utente esperto può accorgersi che non si tratta di veri errori di sistema
    - Presenza sulla task bar
    - Icona di Visual Basic :)
    - Attention?

Osservazione attività anomale

- Cosa sarà successo su C:?



Osservazione attività anomale

- Il contenuto di C: sembra essere solo 10000 files di 0 Kb con nome block_disk_0#N#
- Ma il sistema è in esecuzione...
  - L'unità di sistema (C:) non può essere davvero corrotta in tal modo
  - Digitiamo "C:\windows" nella barra degli indirizzi e accediamo correttamente alla directory di sistema
    - Ne deduciamo che le directory in C:\ sono solo state nascoste
    - *attrib -h C:\windows*
      in un prompt dei comandi ripristina la visibilità della directory e ci conferma la deduzione
      - Successivamente determineremo con esattezza quali directory nasconde il malware

Osservazione attività anomale

- Poco dopo aver dato l'OK ai popup, il sistema diventa inutilizzabile...

- Di continuo:
  - apertura della cartella Documenti
  - avvio dell'Utilità di deframmentazione dischi
  - avvio di Internet Explorer

- Dopo qualche minuto, reboot del sistema!

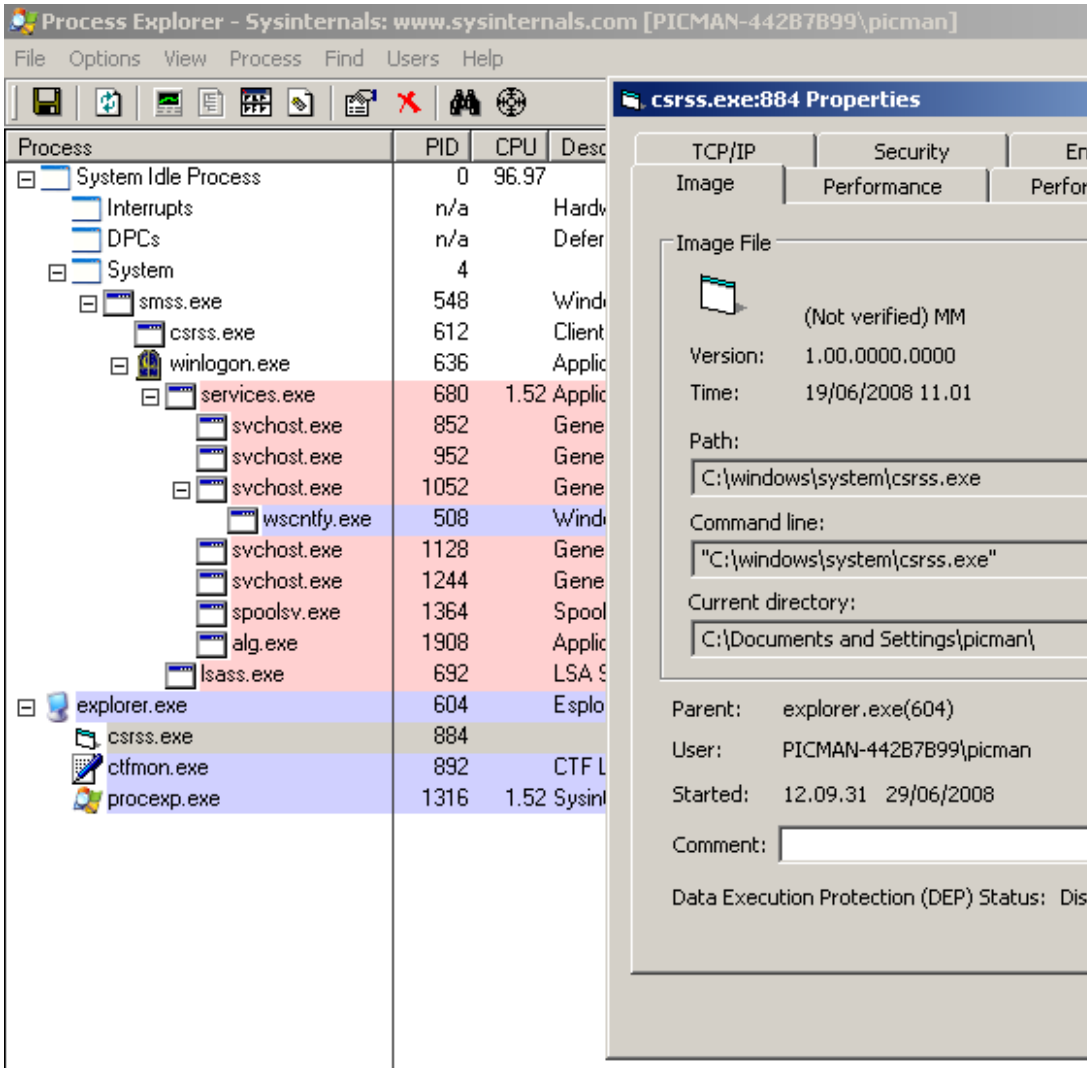
Live Analysis

Utilissimi in questa fase alcuni *Sysinternals Tools*

- Process Explorer
  - Un task manager evoluto
    - DLL e handles, killing, proprietà
- Process Monitor
    - Monitoraggio attività dei processi
      - Accessi al file system e al registro di Windows
- Autoruns
  - Individuazione applicazioni avviate al boot
    - ...in tutti i numerosi modi previsti dall'OS

Live Analysis

- Osserviamo che i “fake popup” sono bloccanti
- Non dando l'OK il malware non passa alla fase di avvio applicazioni/reboot che rende inutilizzabile il sistema
- Possiamo quindi individuare immediatamente e semplicemente il processo malevolo con Process Explorer

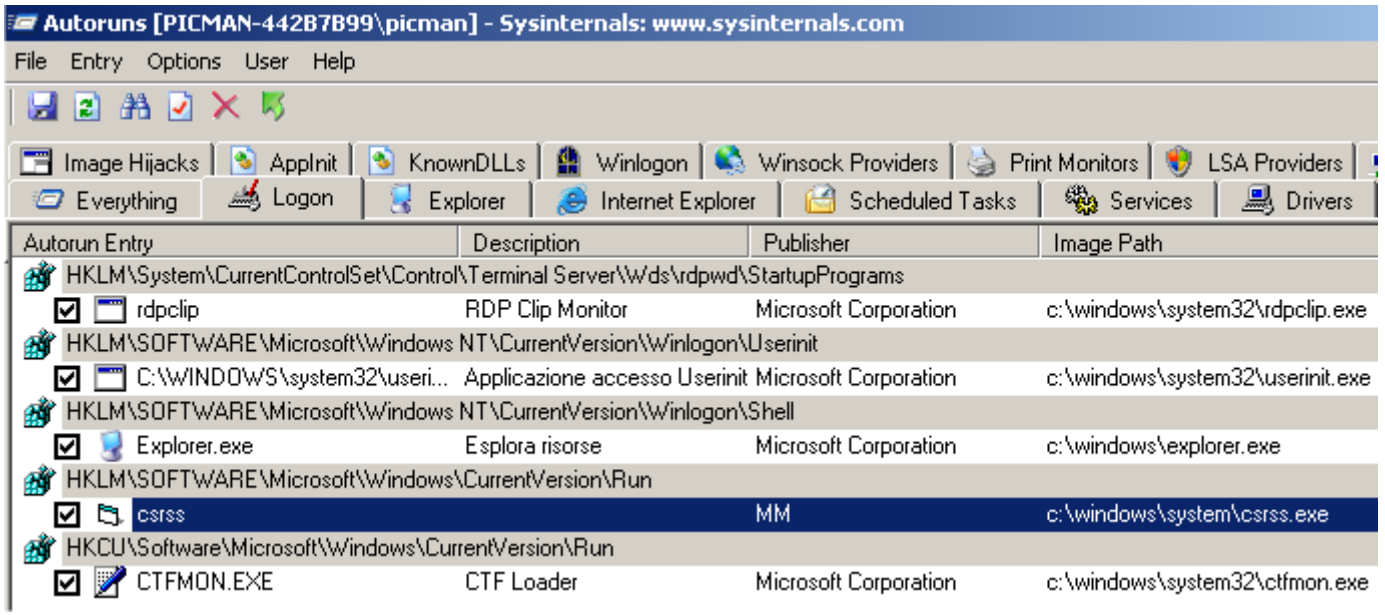


Live Analysis

- Identifichiamo immediatamente il processo csrss.exe,
  - omonimo di un processo di sistema legittimo
    - riconoscibile dall'icona e dalla posizione nell'albero dei processi.
- Annotiamo il path dell'eseguibile incriminato
  - C:\windows\system\csrss.exe
- Terminiamo il processo con Process Explorer
  - Il popup sparisce
  - Il sistema continua a funzionare correttamente
    - Abbiamo individuato la componente principale del malware :)

Live Analysis

- Utilizziamo Autoruns per capire con che meccanismo C:\windows\system\csrss.exe venga fatto partire in automatico al boot
- E' stata utilizzata un'entry nella chiave HKLM\Software\Microsoft\Windows\CurrentVersion\Run



- Rimuoviamo l'entry dal registro e verifichiamo con un reboot che non ci siano meccanismi d'avvio “di backup”

Live Analysis

- Cerchiamo di ottenere maggiori informazioni sulle attività svolte dal malware
- Utilizziamo Process Monitor per monitorare l'attività su disco e gli accessi al registro

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Sequ...	Time of Day	Process Name	PID	Operation	Path
6512	13.09.33,0121539	csrss.exe	388	ReadFile	C:\WINDOWS\system32\msvbvm60.dll
6513	13.09.33,1167695	csrss.exe	388	ReadFile	C:\WINDOWS\system32\msvbvm60.dll
6530	13.09.33,2752530	csrss.exe	388	ReadFile	C:\WINDOWS\system32\msvbvm60.dll
6718	13.09.33,5403658	csrss.exe	388	CreateFile	C:\WINDOWS
6719	13.09.33,5404817	csrss.exe	388	SetBasicInformationFile	C:\WINDOWS
6720	13.09.33,5409228	csrss.exe	388	CloseFile	C:\WINDOWS
6722	13.09.33,5412382	csrss.exe	388	CreateFile	C:\Programmi
6723	13.09.33,5413248	csrss.exe	388	SetBasicInformationFile	C:\Programmi
6724	13.09.33,5416352	csrss.exe	388	CloseFile	C:\Programmi
6726	13.09.33,5419176	csrss.exe	388	CreateFile	C:\Documents and Settings
6727	13.09.33,5420129	csrss.exe	388	SetBasicInformationFile	C:\DOCUMENTS AND SETTINGS
6728	13.09.33,5423646	csrss.exe	388	CloseFile	C:\DOCUMENTS AND SETTINGS

Live Analysis

- Otteniamo dal monitoraggio alcune conferme:
  - Una scrittura nel registro, della chiave già individuata con Autoruns
  - Delle operazioni SetBasicInformationFile che alterano l'attributo di visibilità di alcune directory in C:
  - La creazione di 10000 files di 0 Kb in C:\

- ... e alcune nuove informazioni:
  - L'eseguibile carica da disco alcune DLL sospette
  - Vengono creati 10000 files di 0 Kb anche in C:\windows\temp

Ricapitolando

- Percorso di installazione malware: C:\windows\system
- Eseguibile del malware: csrss.exe
  - DLL: PopupProject.dll, createDLL.dll, createDLL2Project.dll, runProject.dll, RebootProject.dll
- Meccanismi di replica: nessuno
- Operazioni distruttive effettuate: nessuna
- Operazioni reversibili effettuate:
  - Alterazione di una chiave del registro
  - Creazione 10000 files in c:\windows\temp
  - Creazione 10000 files in c:\
  - Attributo hidden su alcune directory in C:\
- Operazioni fastidiose effettuate:
  - Popups, avvio di applicazioni, reboot

Osservazioni

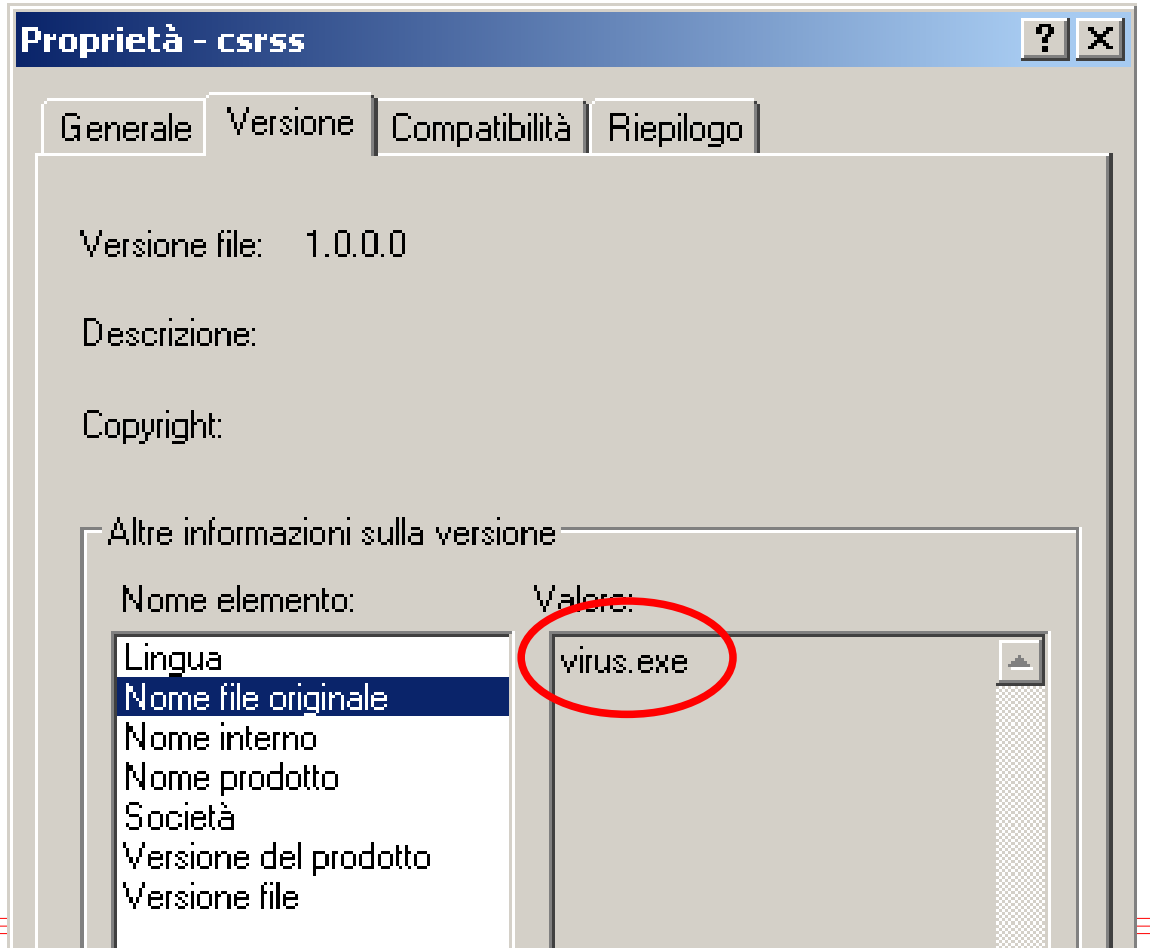
- Sulla macchina analizzata il Task Manager di Windows era inutilizzabile
- L'analisi ha rivelato che in effetti il file taskmgr.exe risultava essere un file di testo contenente il messaggio
  questo non è il task manager
  ahhhaaaahaa
  :)
  Ciao amici!
- Ripristinando l'eseguibile da un'installazione pulita di Windows e rieseguendo il malware non si verifica alcuna alterazione di taskmgr.exe.
- Ne deduciamo che l'alterazione era stata effettuata manualmente da chi ha manomesso la macchina
  - coerente con l'assenza di un meccanismo di replica...

Static Analysis

- Ci sarà sfuggito qualcosa?
  - In molti malware è presente un payload che viene eseguito solo in alcune circostanze
    - Es: a una specifica data/ora
- Questo genere di azioni non si può ovviamente individuare con una “Live Analysis” in cui le condizioni scatenanti potrebbero non verificarsi
- Per completare l'analisi di un software malevolo, dopo averlo identificato ed isolato, si deve ricorrere al reverse engineering.
- Si parla “binary analysis” o “static analysis”: si disassemblano gli eseguibili del malware e se ne ricostruisce la logica

Static Analysis: csrss.exe

- Prima di iniziare...
- Curiosiamo tra le “proprietà” di csrss.exe
  - Nome file originale: virus.exe
  - Società: MM
    - Firma degli autori? :)



Static Analysis: csrss.exe

```
loc_00401AB1: mov var_7C, 00401684h ;
    "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\csrss"
loc_00401AB8: mov var_84, 00000008h
loc_00401AC2: mov var_9C, 00401718h ; "C:\windows\system\csrss.exe"
loc_00401ACC: mov var_A4, 00000008h
loc_00401AD6: mov var_BC, 00401754h ; "REG_SZ"
...
loc_00401B70: push 00401764h ; "regwrite"
...
```

- La regwrite che installa il meccanismo di autostart di csrss.exe

Static Analysis: csrss.exe

```
loc_00401DB1: push 0040180Ch ; "C:\Windows"
loc_00401DB6: call [0040109Ch] ; SetAttr
loc_00401DBC: mov var_4, 0000000Ch
loc_00401DC3: push 00000002h
loc_00401DC5: push 00401828h ; "C:\Programmi"
loc_00401DCA: call [0040109Ch] ; SetAttr
loc_00401DD0: mov var_4, 0000000Dh
loc_00401DD7: push 00000002h
loc_00401DD9: push 00401848h ; "C:\Documents and Settings"
loc_00401DDE: call [0040109Ch] ; SetAttr
loc_00401DE4: mov var_4, 0000000Eh
loc_00401DEB: push 00000002h
loc_00401DED: push 00401880h ; "C:\Program Files"
loc_00401DF2: call [0040109Ch] ; SetAttr
```

- Le chiamate a SetAttr che settano “hidden” le directory

Static Analysis: PopupProject.dll

```
...
  loc_11001A61: mov var_7C, 11001820h ; "Fatal Error"
...
  loc_11001A76: mov var_6C, 110017ECh ; "Disk C:\  corrupt!"
  loc_11001A7D: mov var_74, edi
  loc_11001A80: call MSVBVM60.DLL.__vbaVarDup
  loc_11001A82: mov edi, [11001020h] ; MsgBox(arg_1, arg_2, arg_3, arg_4, arg_5)
...
  loc_11001AE8: mov var_7C, 11001864h ; "Attention"
  loc_11001AEF: mov var_84, 00000008h
  loc_11001AF9: call MSVBVM60.DLL.__vbaVarDup
  loc_11001AFB: lea edx, var_74
  loc_11001AFE: lea ecx, var_34
  loc_11001B01: mov var_6C, 11001840h ; "FORMAT C:"
...
```

- Riconosciamo ovviamente il testo dei finti popup d'errore

Static Analysis: RebootProject.dll

```
...
.text:110017A4              unicode 0, <select * from Win32_OperatingSystem where Pr>
.text:110017A4              unicode 0, <imary=true>,0
.text:11001812              align 4
.text:11001814 aF:
.text:11001814              unicode 0, <F>,0
.text:11001818 aWinmgmtsShutdo:                 ; DATA XREF: .text:11001ADC
.text:11001818              unicode 0, <winmgmts:{(Shutdown)}//./root/cimv2>,0
.text:11001860 aExecquery:                      ; DATA XREF: .text:11001B1A
.text:11001860              unicode 0, <ExecQuery>,0
.text:11001874 aReboot:                         ; DATA XREF: .text:11001BA0
.text:11001874              unicode 0, <Reboot>,0
...
```

- Reboot del sistema tramite Windows Management Instrumentation (WMI)

Static Analysis: runProject.dll

```
...
  loc_11001A0E: push 11001804h ; "Wscript.Shell"
...
  loc_11001A1F: call [11001054h] ; arg_1 = CreateObject(arg_2, arg_3)
...
  loc_11001A4B: mov eax, 11001824h ; "iexplore"
  loc_11001A50: push 00000001h
  loc_11001A52: push 11001838h ; "run"
...
  loc_11001ACB: mov eax, 1100185Ch ; "DFRG.MSC"
  loc_11001AD0: push 00000001h
  loc_11001AD2: push 11001838h ; "run"
...
```

- Avvio operazioni “fastidiose” (apertura Documenti, browser, defrag)

Static Analysis: createDLL.dll

```
...
.text:110017C0                  unicode 0, <Scripting.filesystemobject>,0
.text:110017F6                  align 4
.text:110017F8                  dd 1Ch
.text:110017FC aCBlock_disk0:                       ; DATA XREF: .text:11001A6E
.text:110017FC                  unicode 0, <C:\block_disk0>,0
.text:1100181A                  align 4
.text:1100181C a__vbastrmove    db '__vbaStrMove',0
.text:11001829                  align 10h
.text:11001830 aCreatetextfile:                     ; DATA XREF: .text:11001ABC
.text:11001830                  unicode 0, <CreateTextFile>,0
...
```

- Creazione dei files block_disk0[1-10000] in C:\

Static Analysis: createDLL2Project.dll

```
...
.text:110017A0                  unicode 0, <Scripting.filesystemobject>,0
.text:110017D6                  align 4
.text:110017D8                  unicode 0, <(>,0
.text:110017DC aCWindowsTempFi:                         ; DATA XREF: .text:11001A4E
.text:110017DC                  unicode 0, <C:\Windows\Temp\file>,0
.text:11001806                  align 4
.text:11001808 a__vbastrmove    db '__vbaStrMove',0
.text:11001815                  align 4
.text:11001818 aCreatetextfile:                         ; DATA XREF: .text:11001A9C
.text:11001818                  unicode 0, <CreateTextFile>,0
...
```

- Creazione dei files file[1-10000] in C:\Windows\Temp

Static Analysis: visione d'insieme

```
csrss.exe {
        regWritePerAutostart();
        createDLL2Project.start(); //  files spazzatura in c:\windows\temp
        createDLL.start();             //  files spazzatura in c:\
        PopupProject.showFirstBlocking();        // popup C: corrupt
        PopupProject.showSecondBlocking();     // popup FORMAT C:
        setDirectoriesHidden();
        do {
            runProject.startRandom();    // iexplore, Documenti, defrag ...
            sleep(random(5000));
        } while (executionTime() < maxTime);
        RebootProject.rebootNow();
}
```

Conclusioni analisi

- Non è stata trovata traccia di meccanismi di infezione e/o replica
  - Per questo in effetti parliamo genericamente di malware e non di virus
- La (sommaria) analisi dei binari ha confermato quanto osservato precedentemente, e ha fornito maggiori informazioni sul ruolo delle DLL
  - Sappiamo cosa fa ogni file
    - ...se non ci è sfuggito nulla

- Passiamo alla rimozione

Rimozione

- Per ripulire il sistema, in base a quanto concluso, occorre rimuovere i files del malware ed invertire le operazioni non distruttive da essi effettuate

- Per la precisione, il Removal Tool deve:
  - terminare il processo maligno (csrss.exe)
  - rimuovere dal registro il valore per l'autostart
  - cancellare i files del malware (exe e dll)
  - cancellare i files di 0 kb creati in C:\ e in C:\windows\temp
  - ripristinare la visibilità delle directories nascoste dal malware

Rimozione: tool development

- Per la realizzazione del Removal Tool abbiamo deciso di utilizzare VBS (Visual Basic Script)
- Al prezzo di una sintassi orrenda, otteniamo
  - Semplice accesso al file system e al registro
  - Sviluppo rapido
  - Indipendenza da compilatori/toolchains: è un linguaggio di scripting interpretato da una componente di Windows installata di default (il WSH, Windows Scripting Host)
- E poi, un malware scritto in Visual Basic non si merita un Removal Tool in C/C++ :)

Removal Tool: step by step

- Diamo uno sguardo passo passo al codice del removal tool
  - *Terminazione del processo*

```
''' termina processo del malware ''''''''''''''''''''''''''''
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\.\root\cimv2")

Set colProcess = objWMIService.ExecQuery _
("Select * from Win32_Process Where Name = 'csrss.exe' and
  CommandLine != null" )
     ' Nota:
     '   "CommandLine != null" esclude il "crsss.exe" legittimo,
     '   di sistema, che ha CommandLine == null

For Each objProcess in colProcess
     objProcess.Terminate()
     WSCript.Echo "Processo maligno terminato! Segue pulizia del
  sistema..."
Next
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Removal Tool: step by step

- *Rimozione meccanismo di autostart*

```
''' rimuove dal registro l'autostart del malware ''''''''''''''''
const HKEY_LOCAL_MACHINE = &H80000002

Set oReg=GetObject("winmgmts" _
& ":{impersonationLevel=impersonate}!\\.\root\default:StdRegProv")

strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strStringValueName = "csrss"
oReg.DeleteValue HKEY_LOCAL_MACHINE,strKeyPath,strStringValueName
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Removal Tool: step by step

- *Rimozione files del malware (exe e dll)*

```
''' rimuove files del malware ''''''''''''''''''''''''''''''''''
Set fso=CreateObject("Scripting.FileSystemObject")
WinDir = fso.GetSpecialFolder(0)   ' %systemroot%
Drive = Left(WinDir,3)             ' "c:\"
MalwarePath = WinDir & "\system\"

Dim mwFiles(6)
mwFiles(0) = "csrss.exe"
mwFiles(1) = "PopupProject.dll"
mwFiles(2) = "createDLL.dll"
mwFiles(3) = "createDLL2Project.dll"
mwFiles(4) = "runProject.dll"
mwFiles(5) = "RebootProject.dll"

For Each mwFile In mwFiles
     fso.DeleteFile(MalwarePath & mwFile)
Next
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Removal Tool: step by step

- *Rimozione files vuoti creati dal malware*

```
''' rimuove files spazzatura in c:\windows\temp ''''''''''''''''''
fso.DeleteFile(windir & "\temp\file*")
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'''rimuove files spazzatura in c:\ ''''''''''''''''''''''''''''''''
fso.DeleteFile(Drive & "block_disk*")
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Removal Tool: step by step

- *Ripristino visibilità directories*

```
''''ripristina visibilità  directories settate "hidden" ''''''''''
Dim dirs(4)
dirs(0) = WinDir
dirs(1) = Drive & "Documents And Settings"
dirs(2) = Drive & "Program Files"
dirs(3) = Drive & "Programmi"

For Each dir In dirs
     fso.GetFolder(dir).Attributes = 0
Next
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Removal Tool: step by step

- *Missione compiuta :)*

```
'''' Ok! '''''''''''''''''''''''''''''''''''''''''''''''''''''
WScript.Echo "Malware rimosso!"
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
WScript.Quit
```

Grazie dell'attenzione!