ERNW
Living Security.

Reversing Malware for Business Purposes

Michael Thumann, mthumann@ernw.de

#whoami

ERNW Living Security.

- **Head of Research & Chief Security Officer, ERNW GmbH**

- **Talks und Publications:**
  - “Application Trustworthiness“, Daycon, Dayton 2008
  - “Reversing – A structured approach“, Troopers, München 2008
  - “Hacking Second Life”, Metaverse08, Karlsruhe 2008
  - “Hacking Second Life”, Hack-in-the-Box, Dubai 2008
  - “Reversing – A structured approach”, RSA Conference, San Francisco 2008
  - “Hacking SecondLife”, Blackhat Europe, Amsterdam 2008
  - “Hacking the Cisco NAC Framework”, Sector, Toronto 2007
  - “Hacking SecondLife”, Daycon, Dayton 2007
  - “Hacking Cisco NAC”, Hack-in-the-Box, Kuala Lumpur, 2007
  - “NAC@ACK”, Blackhat-USA, Las Vegas, 2007
  - “NAC@ACK”, Blackhat-Europe, Amsterdam, 2007
  - “Mehr IT-Sicherheit durch PenTests”, Vieweg Verlag 2005

- **Main Tasks:**
  - Reverse Engineering
  - Security Research
  - Penentrationstests
  - Code Audits

Agenda

ERNW
Living Security.

1. **Introduction**
2. **Online Sandbox Systems**
3. **Running your own sandbox**
4. **Reverse Engineering**
5. **Recommendations**
6. **Summary**

ERNW
Living Security.

Introduction

Introduction

ERNW Living Security.

- **Malware is still one of the biggest issues 2009**
- **We're facing automated worms, e.g. Conficker**
- **There're targeted attacks with customized malware**
- **There're advanced stealth techniques used when creating malware**
- **We see an upcoming need for large enterprises to implement processes to deal with malware incidents**
- **This talk covers the most important approaches for analyzing malware in a business context**

ERNW
Living Security.

Online Sandbox Systems

Threat Expert

ERNW Living Security.

- **Free Online Sandbox**

- **Designed to analyze behavior of**
  - Viruses
  - Worms
  - Trojans
  - …

  } **Malware**

- **Results can be either be public visible or kept private**
- **Multiple analyzing possible**

Threat Expert - Example

ERNW Living Security.

Submission Summary:

- Submission details:
  - Submission received: 15 March 2009, 20:09:56
  - Processing time: 7 min 22 sec
  - Submitted sample:
    - File MD5: 0xC11FCC291843C1E3629D12FA01BDD8C8
    - File SHA-1: 0x7353AB44631C7C8BB5CD994B480B56BB969E1FF5
    - Filesize: 909.312 bytes
    - Alias:
      - Trojan Horse [Symantec]
      - Generic.dx [McAfee]
      - Trojan.Obfuscated [Ikarus]
- Summary of the findings:

What's been found	Severity Level
Contains characteristics of an identified security risk.	

Technical Details:

Possible Security Risk

- **Attention!** The following threat category was identified:

Threat Category	Description
	A malicious trojan horse or bot that may represent security risk for the compromised system and/or its network environment

Threat Expert - Information

- **Offered Information:**
  - Hash Values (SHA1 & MD5)
  - Aliases
  - *Security Level*
  - Category
  - File System Operation
  - Memory Operation
  - Origin Country

CWSandbox

- **Free online version of the commercial product**
- **Designed for W32 env. only**
- **Ran by University of Mannheim (GER)**
- **No Re-analyzing of already submitted samples**

CWSandbox - Example

ERNW Living Security.

XML (plain) - TXT (plain) - HTML (plain) - back to sample - PCAP - CAB

CWSandbox Sunbelt MALWARE ANALYSIS REPORT

Scan Summary | File Changes | Registry Changes | Network Activity | Technical Details

Submission Details

Date	16.03.2009 02:44:35
Sandbox Version	2.1.12
File Name	c:\Yaha.E.exe
Submitting Email	
Comment	

Summary Findings

Total Number of Processes	1
Termination Reason	NormalTermination
Start Time	00:00.547
Stop Time	00:01.391
Start Reason	AnalysisTarget

Analysis HighLights

Spawned Processes	Found 0 Processes. (View Activity by Process)
Filesystem Changes	View File Changes
Registry Changes	View Registry Changes
Network Activity	View Network Activity

CWSandbox - Information

- **Offered Information:**
  - Hash Value
  - File System Operation
    - File/Folder Operations
    - DLL-Handling
  - Registry Operation
  - Network Acitivity
  - Process Management
    - Child processes
    - IPC

Information more detailed but less conditioned than ThreadExpert

ERNW
Living Security.

Running your own Sandbox

Why a private sandbox?

ERNW Living Security.

- **Because it's yours!**
  - Define settings and environment as you need it
  - E.g. runtime
- **Independent from external services**
- **Licensing terms**
- **Information disclosure**

Zero wine

ERNW Living Security.

- **WINE based Sandbox system**
- **OpenSource (GPL v2)**
- **Records API calls**
- **Uses WINEDEBUG env. variable**

- **Available as QEMU virtual machine image**
  - Debian based
  - HTTP Server to upload malware and to review analysis results

- **Report isn’t soo useful**

Truman Sandbox

ERNW Living Security.

- **PoC how to build a private sandbox**
- **Extended approach**

"Truman is conceptually a very simple system (once it's set up properly, that is)"

- **Consists of Server (Linux) & Client component (Windows)**

- **Current Version: 0.1 …, so it's tricky to install**
- **Report is just diffing**

Why building an individual Sandbox?

ERNW Living Security.

- **Support of different Operating Systems**
  - Mac OS X
  - Linux
  - UNIX
- **Stealth functionality**
- **Reflect company specific environment & security controls**
  - Typical systems setup
- **Wait for the RE part…**

Analysis – Tools

ERNW Living Security.

- **Registry:** **RegMon (MS Sysinternals)**
- **File System:** **FileMon (MS Sysinternals)**
- **Process:** **ProcMon (MS Sysinternals)**
- **API:** **Autodebug Professional**
- **Network:** **Wireshark (formally known as Ethereal)**

- **Dedicated malware analysis toolkits:**
  - SysAnalyzer
  - Malware Analysis Pack
  - Multipot

ERNW
Living Security.

Reverse Engineering

Reversing – Get prepared

ERNW Living Security.

- **An analysis system (sandboxed) is required**
- **Network traffic must be controlled by a firewall**
- **The system must defeat Anti-RE tricks (remember the introduction)**
- **It must contain the mandatory tools**
- **It must be able to simulate services like SMTP, HTTP, IRC, SMB and DNS, so the malware can fulfill all it's tasks**
- **You must be able to restore the system to it's initial clean state**
- **VMware images can be used**
- **Usage of a dedicated (real) system**

Reversing – Vmware Sandbox

ERNW Living Security.

- **Easy to handle approach because it's running virtual on your system**
- **Snapshots ensure that a clean state can be restored**
- **"Host-only" network settings prevent your network from getting infected**
- **But virtualization can be detected by malware using different techniques**
- **Using a VM requires installation steps to make the detection a little bit harder**

Reversing – Vmware Installation

ERNW Living Security.

- **Install a commonly used operating system (Win XP is fine)**
- **DO NOT INSTALL VMware Tools !!!!!!**
- **DO NOT INSTALL any VMware drivers**
- **Change the MAC address of your NIC to one of the AMD PCNET32 Family**
- **Install your tools**
- **Apply additional settings to your os.vmx file (the VMware configuration file)**

Reversing – Additional VMX Settings

ERNW Living Security.

- isolation.tools.getPtrLocation.disable = "TRUE"
- isolation.tools.setPtrLocation.disable = "TRUE"
- isolation.tools.setVersion.disable = "TRUE"
- isolation.tools.getVersion.disable = "TRUE"
- monitor_control.disable_directexec = "TRUE"
- monitor_control.disable_chksimd = "TRUE"
- monitor_control.disable_ntreloc = "TRUE"
- monitor_control.disable_selfmod = "TRUE"
- monitor_control.disable_reloc = "TRUE"
- monitor_control.disable_btinout = "TRUE"
- monitor_control.disable_btmemspace = "TRUE"
- monitor_control.disable_btpriv = "TRUE"
- monitor_control.disable_btseg = "TRUE"

Reversing – Required Tools

ERNW Living Security.

- **IDA Pro 5.4: Commercial Disassembler available at http://www.hex-rays.com/**
- **Hex-Rays: Commercial Decompiler Plugin for IDA Pro available at http://www.hex-rays.com/**
- **X86emu: x86 Emulator Plugin for IDA Pro available at https://sourceforge.net/projects/ida-x86emu/**
- **Bochs 2.3.7: Virtualizing Software and PC Emulator available at http://bochs.sourceforge.net/**

Reversing – Required Tools 2

ERNW Living Security.

- **OllyDBG: Windows Ring 3 Debugger available at http://www.ollydbg.de/**
- **Ollydump: OllyDBG plugin that dumps a program from memory available at http://www.woodmann.com/collaborative/tools/index.php/OllyDump**
- **Phant0m: OllyDBG plugin for hiding the debugger available at http://www.woodmann.com/collaborative/tools/index.php/PhantOm**
- **xADT: Tool to check if your debugger is detectable, available at http://xchg.info/ARTeam/Tutorials/index.php?dir=ARTeam_Releases/**

Reversing – Required Tools 3

ERNW Living Security.

- **RDG Packer Detector: Program for detecting code obfuscators available at http://rdgsoft.8k.com/IndexIngles.html**
- **ExplorerSuite: Working with EXE Files and detect packers, available at http://www.ntcore.com/exsuite.php**
- **Signsrch: Tool looking for special tricks, available at http://www.aluigi.org/mytoolz.htm**
- **ScoopyNG: Tool to check for virtualization, available at http://www.trapkit.de/research/vmm/scoopyng/index.html**
- **LordPE: Dumping process, available at http://www.woodmann.net/collaborative/tools/index.php/LordPE**

Reversing – Some basics

ERNW Living Security.

- **We are NOT talking about any magic voodoo geek hacker stuff, we're talking about RE in a business context**
- **That means we have a limited amount of time for our analysis**
- **We must follow a structured approach to accomplish the task**
- **If commercial tools are needed, helpful or reducing the analysis time, go and buy them!**
- **Deep knowledge is required for RE, if you don't have it, start learning ☺**

Reversing – Structured Approach

ERNW Living Security.

1. **Get hands on your malware sample**
2. **Prepare your sandbox**
3. **Detect code obfuscation**
4. **Defeat code obfuscation**
5. **Detect anti-reversing tricks**
6. **Defeat anti-reversing tricks**
7. **Analyze what the malware is doing**

Reversing – Get hands on the sample

ERNW Living Security.

- **Of course first you have to get the malware to start analysis**
- **Inform users to send it to you and remind them not to click at the attachment. Give detailed instructions !!!**
- **Acquire the sample from an infected system (some basic forensic knowledge can be helpful)**
- **Get it from a quarantine area, e.g. if catched by your attachment blocker**
- **Store it at a safe place! It shouldn't be accessible by everyone ☺**
- **Rename the extension to something like ".rename2exe"**

Reversing – Prepare Sandbox

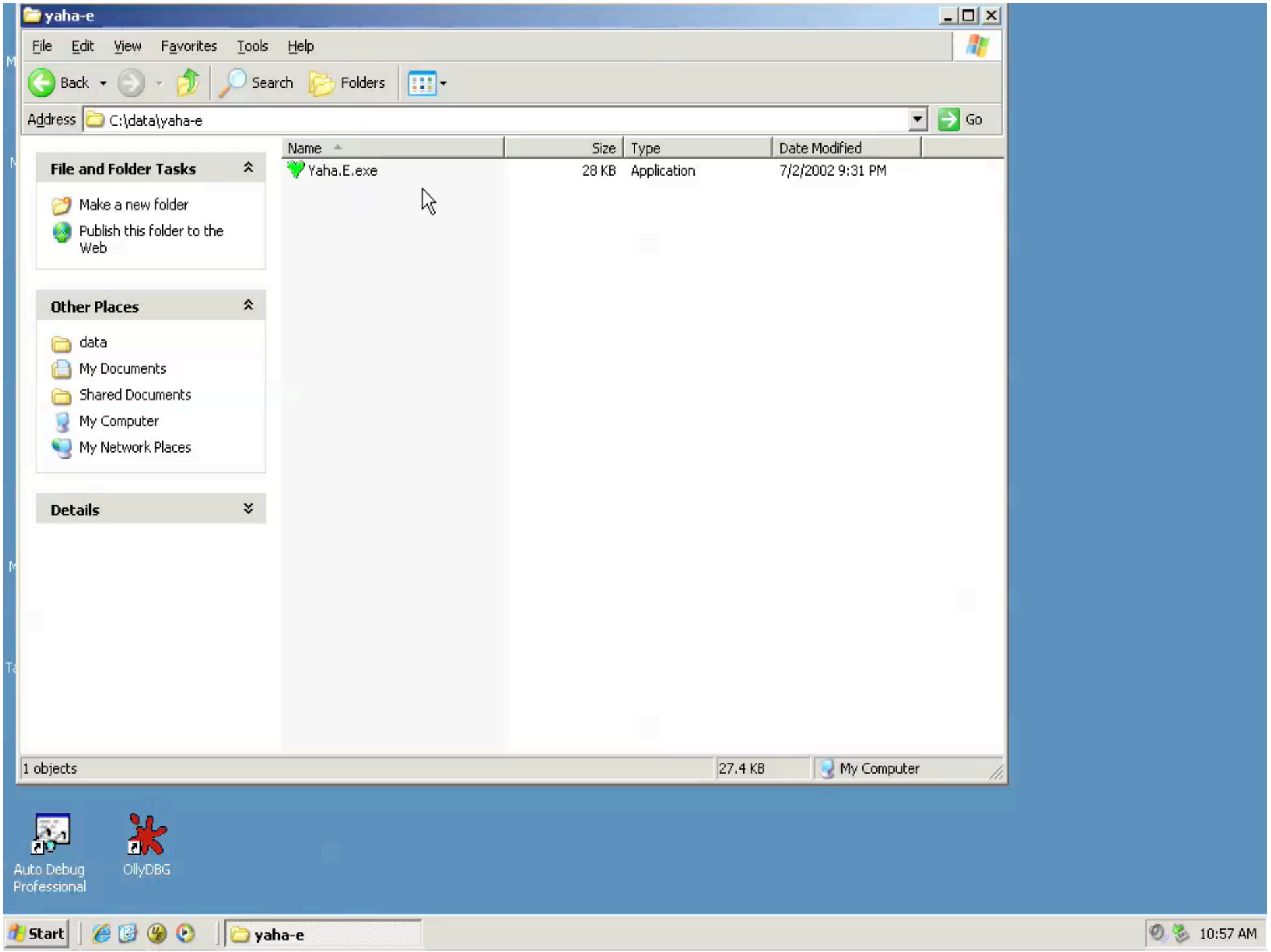
ERNW Living Security.

- **Are the tools updated? E.g. Packer detectors receive signature updates like AV**
- **Check if sandbox is in a clean state, restore it if this is not the case**
- **Ensure that the malware can't escape !!**
- **Enable Firewall**
- **Check network separation**
- **Copy malware to analysis system**

Reversing – Detect Code Obfuscation

ERNW Living Security.

- **There are many approaches to do this and many opinions about it, but we're in a business context, so here's my way**
- **Run at least two packer detectors. I recommend RDG and ExplorerSuite**
- **Use entropy analysis (RDG)**
- **Do the initial disassembly using IDA**
- **Spot typical signs in the disassembly**
- **Make your statement ☺**


yaha-e
File Edit View Favorites Tools Help
Back Search Folders
Address C:\data\yaha-e
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Other Places
data
My Documents
Shared Documents
My Computer
My Network Places
Details
Name Size Type Date Modified
Yaha.E.exe 28 KB Application 7/2/2002 9:31 PM
1 objects
27.4 KB
My Computer
Auto Debug Professional
OllyDBG
Start
yaha-e
10:57 AM

IDA

File Edit Jump Search View Debugger Options Windows Help

Text

Drag a file here to disassemble

```
IDAPython version 1.1.0 final (serial 0)
Copyright (c) 2004-2009 Gergely Erdelyi - http://d-dome.net/idapython/
-------------------------------------------------------------------------
x86emu: No saved x86emu state data was found.
Using FLIRT signature: SEH for vc7/8
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Unloading IDP module C:\Programming\IDAPro\procs\pc.w32...
```

Start | yaha-e | IDA | 1:35 PM

Reversing – Defeat Code Obfuscation

ERNW Living Security.

- **To get a readable disassembly code obfuscation must be defeated**
- **The How To depends on the used packer. Some can be easy to unpack like standard UPX, others are extremely difficult like Armadillo**
- **If a working unpacker is available, USE IT ! (search with Google ☺)**
- **Otherwise you have to run the malware to let it unpack itself. It must unpack to function properly**
- **IDA and it's universal unpacker is one choice**
- **Running the malware and dump the process is another**

IDA - C:\data\yaha-e\Yaha.E.exe - [IDA View-A]

File Edit Jump Search View Debugger Options Windows Help

No debugger

Text

IDA View-A | Hex View-A | Exports | Imports | Names | Functions | Structures | Enums

```
...1:00414B00 ; ---------------------------------------------------------------
...1:00414B00
...1:00414B00                 public start
...1:00414B00 start:
...1:00414B00                 pusha
...1:00414B01                 mov     esi, offset dword_40F000
...1:00414B06                 lea     edi, [esi-0E000h]
...1:00414B0C                 push    edi
...1:00414B0D                 or      ebp, 0FFFFFFFFh
...1:00414B10                 jmp     short loc_414B22
...1:00414B10 ; ---------------------------------------------------------------
...1:00414B12                 align 8
...1:00414B18
...1:00414B18 loc_414B18:                             ; CODE XREF: ...1:loc_414B29↓j
...1:00414B18                 mov     al, [esi]
...1:00414B1A                 inc     esi
...1:00414B1B                 mov     [edi], al
...1:00414B1D                 inc     edi
...1:00414B1E
...1:00414B1E loc_414B1E:                             ; CODE XREF: ...1:00414BB6↓j
...1:00414B1E                                         ; ...1:00414BCD↓j
...1:00414B1E                 add     ebx, ebx
```

00005F00 | 00414B00: ...1:start

```
Copyright (c) 1990-2009 Python Software Foundation - http://www.python.org/

IDAPython version 1.1.0 final (serial 0)
Copyright (c) 2004-2009 Gergely Erdelyi - http://d-dome.net/idapython/
-----------------------------------------------------------------------------
x86emu: No saved x86emu state data was found.
Using FLIRT signature: SEH for vc7/8
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

AU: idle | Down | Disk: 15GB

Start | yaha-e | IDA - C:\data\yaha-e... | TC 9 - Console | 2:30 PM

Reversing – Detect Anti-RE Tricks

ERNW Living Security.

- **Anti-RE tricks are used to prevent a proper analysis of the malware**
- **They try to detect if the malware is analyzed**
- **They make the disassembly less readable and harder to analyze**
- **The malware behaves different if analyzing tools are detected**
- **Tools can be used to detect Anti-RE tricks (signsrch)**
- **Looking at the disassembly can reveal these kind of tricks**

Reversing – Anti-RE Tricks

ERNW Living Security.

- **Detecting Debuggers using the Windows API call IsDebuggerPresent**
- **Detecting Virtualization e.g. looking for specific hardware or registry keys**
- **Detecting Instrumentation e.g. with FindWindow("FilemonClass", NULL)**
- **Dynamically Computed Target Addresses are used to ensure that the execution flow can only be followed at runtime**
- **Targeted Attacks against the Analysis Tools e.g. vulnerabilities in IDA and OllyDBG**

TC 9 - Security Checks

File Edit Tabs View Options Windows Help

Console | User ernw | Security Checks

```
ernw@attack#
```

Start | data | TC 9 - Security Checks | conficker-a | 11:32 PM

Reversing – Debug Check in Conficker

ERNW Living Security.

```
.text:10002F7E ; START OF FUNCTION CHUNK FOR sub_1000296A
.text:10002F7E
.text:10002F7E loc_10002F7E:                          ; CODE XREF: sub_1000296A+24↑j
.text:10002F7E                 call    ds:IsDebuggerPresent
.text:10002F84                 mov     dword_100169F4, eax
.text:10002F89                 mov     edx, [ebp-44h]
.text:10002F8C                 push    edx             ; ucchMax
.text:10002F8D                 push    51E80000h       ; lpsz
.text:10002F92                 call    ds:IsBadStringPtrA
.text:10002F98                 mov     edi, dword_10016588
.text:10002F9E                 mov     [ebp+0Ch], edi
.text:10002FA1                 jmp     loc_100043D0
.text:10002FA1 ; END OF FUNCTION CHUNK FOR sub_1000296A
```

Reversing –Conficker checks for VMware

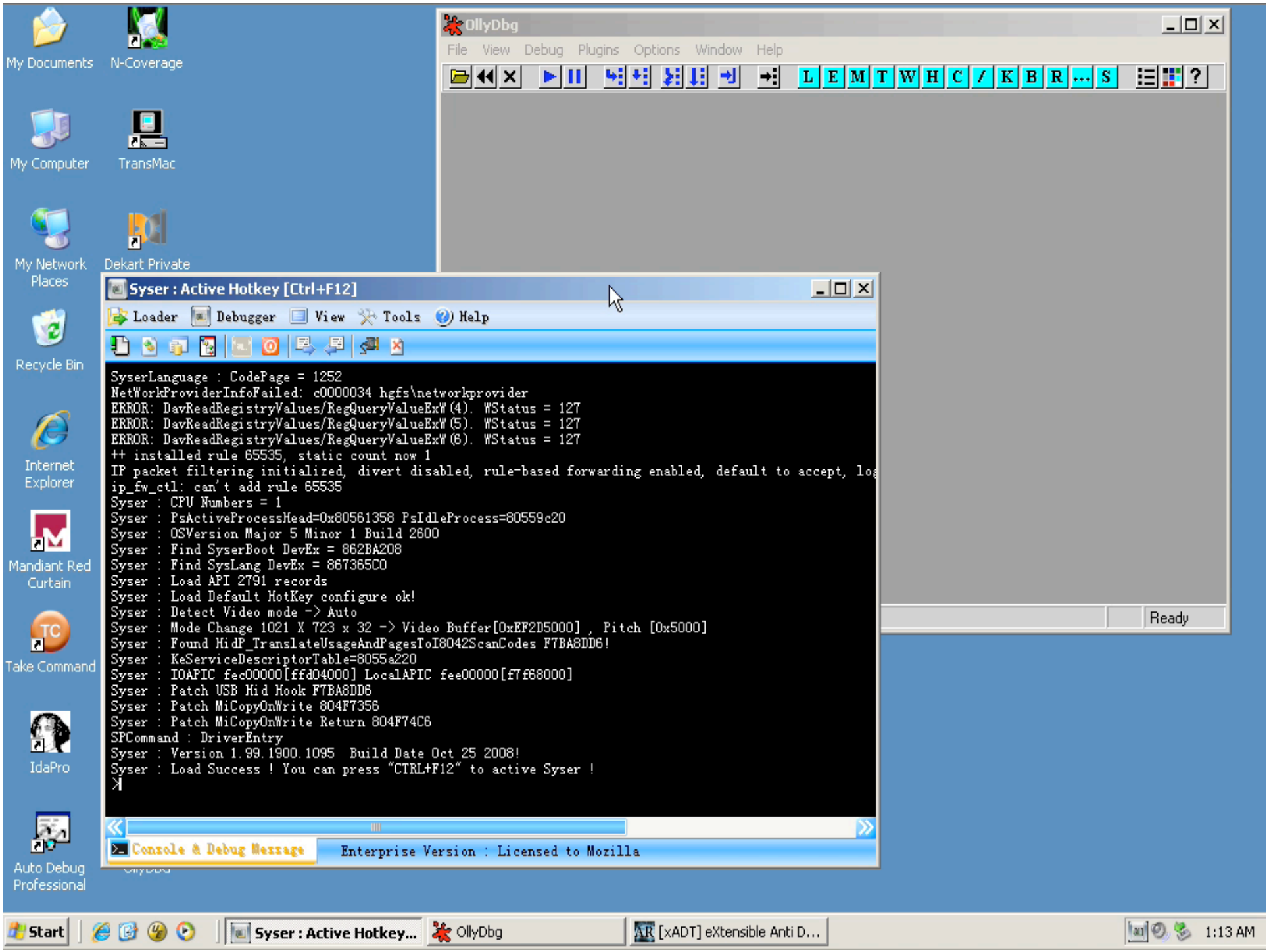
ERNW Living Security.

```
sldt    eax                 ; Store Local Descriptor Table (LDT)
cmp     ax, si              ; Result should be Zero on native os
jz      short loc_3A77C5    ; step over loop if no vm
push    0FFFFFFFFh          ; dwMilliseconds (= -1)
call    ds:Sleep_0          ; sleep forever
```

Reversing – Defeating Anti-RE Tricks

ERNW Living Security.

- **Don’t use your VM or hide it ☺ (remember the installation instructions for the VMware sandbox)**
- **Use Anti-Anti-Debug Tools (e.g. OllyDBG Phant0m Plugin)**
- **Decide which tools are really needed (Syser must be disabled at boot time to prevent it from being detected)**
- **Check your system state with xADT (debugging) und ScoopyNG (virtualization)**


My Documents
N-Coverage
My Computer
TransMac
My Network Places
Dekart Private
Recycle Bin
Internet Explorer
Mandiant Red Curtain
Take Command
IdaPro
Auto Debug Professional
OllyDbg
File View Debug Plugins Options Window Help
Ready
Syser : Active Hotkey [Ctrl+F12]
Loader Debugger View Tools Help
SyserLanguage : CodePage = 1252
NetWorkProviderInfoFailed: c0000034 hgfs\networkprovider
ERROR: DavReadRegistryValues/RegQueryValueExW(4). WStatus = 127
ERROR: DavReadRegistryValues/RegQueryValueExW(5). WStatus = 127
ERROR: DavReadRegistryValues/RegQueryValueExW(6). WStatus = 127
++ installed rule 65535, static count now 1
IP packet filtering initialized, divert disabled, rule-based forwarding enabled, default to accept, log
ip_fw_ctl: can't add rule 65535
Syser : CPU Numbers = 1
Syser : PsActiveProcessHead=0x80561358 PsIdleProcess=80559c20
Syser : OSVersion Major 5 Minor 1 Build 2600
Syser : Find SyserBoot DevEx = 862BA208
Syser : Find SysLang DevEx = 867365C0
Syser : Load API 2791 records
Syser : Load Default HotKey configure ok!
Syser : Detect Video mode -> Auto
Syser : Mode Change 1021 X 723 x 32 -> Video Buffer[0xEF2D5000] , Pitch [0x5000]
Syser : Found HidP_TranslateUsageAndPagesToI8042ScanCodes F7BA8DD6!
Syser : KeServiceDescriptorTable=8055a220
Syser : IOAPIC fec00000[ffd04000] LocalAPIC fee00000[f7f68000]
Syser : Patch USB Hid Hook F7BA8DD6
Syser : Patch MiCopyOnWrite 804F7356
Syser : Patch MiCopyOnWrite Return 804F74C6
SPCommand : DriverEntry
Syser : Version 1.99.1900.1095 Build Date Oct 25 2008!
Syser : Load Success ! You can press "CTRL+F12" to active Syser !
Console & Debug Message
Enterprise Version : Licensed to Mozilla
Start
Syser : Active Hotkey...
OllyDbg
[xADT] eXtensible Anti D...
1:13 AM

[xADT] eXtensible Anti Debug Tester

Start Selected | Clear | About

Enable	TestName	Result	Status	Description of Test	o..
	IsDebuggerPresent()	NaN	NaN	Test using IsDebuggerPresent	Inl
	CheckRemoteDebuggerPresent()	NaN	NaN	Test using CheckRemoteDebuggerPresent	Inl
	PEB.BeingDebugged	NaN	NaN	Controls PEB.BeingDebugged	Inl
	PEB.ProcessHeap	NaN	NaN	Controls PEB.ProcessHeap	Inl
	GetProcessHeap()	NaN	NaN	Controls PEB.ProcessHeap through GetProcessHeap API	Inl
	PEB.NtGlobalFlag	NaN	NaN	Controls PEB.NtGlobalFlag	Inl
	PEB.NtGlobalFlag2	NaN	NaN	Controls PEB.NtGlobalFlag via ZwQueryInformationProcess	Inl
	Debug Registers	NaN	NaN	Test if any of the Debug Registers is not 0	Inl
	Single Step	NaN	NaN	Test if single step bit in EFlags is set	Inl
	CreateFileDrivers()	NaN	NaN	Test some drivers using CreateFile	Inl
	ZwQueryInformationProcess()	NaN	NaN	Test using ZwQueryInformationProcess	Inl
	ZwQueryInformationThread()	NaN	NaN	Test using ZwQueryInformationThread	Inl
	ChupaChu_MY_NtQueryInformationProcess	NaN	NaN	This test uses NtQueryInformationProcess API...	Ex
	ChupaChu_OLLY_PROCESS_HIDEN_Sweep	NaN	NaN	This test will detect active Olly process while hiding its process...	Ex
	ChupaChu_STARTUPINFO(AW)	NaN	NaN	This test will detect Olly (and most ring3 debuggers)..	Ex
	ChupaChu_SYSER_detector	NaN	NaN	This test will detect active syser devices..	Ex
	ChupaChu debugger test v0.3 *final public-plugin edition*	NaN	NaN	This test will run most of "ChupaChu debugger test v0.3" checks to see if debugged.	Ex
	ChupaChu_TICK_TIME_TRICK	NaN	NaN	This test will detect debugger using GetTickTime API...	Ex
	DBG_PRINTEXCEPTION_C	NaN	NaN	Test checking the handling of DBG_PRINTEXCEPTION_C	Ex
	DeleteFiber Test	NaN	NaN	Test using the DeleteFiber API	Ex
	Find Tools Complex	NaN	NaN	Checks blacklisted RCE tools via a lot of different methods	Ex
	FindWindow 4 OllyDbg	NaN	NaN	Check presence of a window with OllDbg caption	Ex
	GetSystemTime and INT3	NaN	NaN	Test using GetSystemTime and INT3	Ex
	RDTSC and INT3	NaN	NaN	Test using RDTSC ASM instruction and INT3	Ex
	int_hooks	NaN	NaN	Tests for hooks in IDT	Ex
	Invalid_Handle Exception Test	NaN	NaN	Test looking if the Invalid_Handle Exception is caught or not	Ex
	Int2ATrick way to KiGetTickCount	NaN	NaN	Test using int 2A to call KiGetTickCount	Ex
	Some Rootkits typical tests	NaN	NaN	Several checks found in some Rootkits and compressors. Several tests, see opened DOS window for details	Ex
	NtQueryInfoProc_hook_detection Test	NaN	NaN	Test creating a debugged child process and querying it using NtQueryInfoProcess	Ex
	NtSystemDebugControl Tests	NaN	NaN	A collection of Tests using the NtSystemDebugControl API	Ex
	NtYieldExecution	NaN	NaN	Test using NtYieldExecution	Ex
	ParentProcess Test	NaN	NaN	Test looking if the ParentProcess is a debugger	Ex
	SICE Presence Tests	NaN	NaN	Uses several ways to detect SICE	Ex
	UnhandledExceptionFilter	NaN	NaN	Test using UnhandledExceptionFilter	Ex
	EnumWindows	NaN	NaN	Detect OllyDBG via EnumWindows	Ex
	GetProcessHeaps	NaN	NaN	Detect ring3 debuggers on NT via GetProcessHeaps	Ex
	PageGuard	NaN	NaN	Detect OllyDBG via PageGuard	Ex

Will display WARNING, POSITIVE Results

Will display UNKNOWN, NEGATIVE results

Start | xADT | [xADT] eXtensible An... | 1:09 AM

Reversing – Analyze the Malware

ERNW Living Security.

- **Finally we have our readable disassembly**
- **To understand what the malware is doing we use IDA and Hex-Rays (commercial tools)**
- **Hex-Rays ensures that we can accomplish our analysis task quite fast (C-Code is easier to read than assembler ☺)**
- **Hex-Rays can be invoked by pressing F5**
- **Start at the new entry point and follow the execution flow**

IDA - C:\Documents and Settings\ernw\My Documents\demo-zerowine\yaha-e\Yaha.E.idb (Yaha.E.exe) - [IDA View-A]

File Edit Jump Search View Debugger Options Windows Help

IDA View-A | Hex View-A | Exports | Imports | Names | Functions | Strings | Structures | Enums | Segmentation

```
.text:00401000 ;
.text:00401000 ; +-------------------------------------------------------------------------+
.text:00401000 ; |   This file has been generated by The Interactive Disassembler (IDA)    |
.text:00401000 ; |        Copyright (c) 2009 by Hex-Rays, <support@hex-rays.com>           |
.text:00401000 ; |                        License info: 48-B27D-7194-CE                    |
.text:00401000 ; |                         Michael Thumann, ERNW GmbH                      |
.text:00401000 ; +-------------------------------------------------------------------------+
.text:00401000 ;
.text:00401000 ; Input MD5   : 0CFC1D3AD017EA6B91DCD8457D81A02A
.text:00401000
.text:00401000 ; File Name   : C:\Documents and Settings\ernw\My Documents\MalwareAnalysis\demo-zerowine\yaha-e\Yah
.text:00401000 ; Format      : Portable executable for 80386 (PE)
.text:00401000 ; Imagebase   : 400000
.text:00401000 ; Section 1. (virtual address 00001000)
.text:00401000 ; Virtual size                  : 0000E000 (  57344.)
.text:00401000 ; Section size in file          : 00000000 (      0.)
.text:00401000 ; Offset to raw data for section: 00000400
.text:00401000 ; Flags E0000080: Bss Executable Readable Writable
.text:00401000 ; Alignment     : default
.text:00401000
.text:00401000                 include uni.inc ; see unicode subdir of ida for info on unicode
.text:00401000
.text:00401000                 .686p
.text:00401000                 .mmx
.text:00401000                 .model flat
.text:00401000
```

UNKNOWN | 00401000: sub_401000

```
--------------------------------------------------------------------------------
Python interpreter version 2.5.2 final (serial 0)
Copyright (c) 1990-2009 Python Software Foundation - http://www.python.org/

IDAPython version 1.1.0 final (serial 0)
Copyright (c) 2004-2009 Gergely Erdelyi - http://d-dome.net/idapython/
--------------------------------------------------------------------------------
x86emu: No saved x86emu state data was found.
```

Python

AU: idle | Down | Disk: 15GB

Start | yaha-e | IDA - C:\Documents a... | 1:01 AM

Recommendations

Recommendations

ERNW Living Security.

- **Malware Analysis in a business context must be accomplished in a reasonable amount of time**
- **Fast analysis procedures ensure minimal damage and impact of a malware outbreak**
- **The presented methods require a different level of the analysts knowledge**
- **Each of them has their individual advantages and disadvantages**
- **Processes and procedures are needed in an enterprise environment**

Recommendations

ERNW Living Security.

- **The different types of malware often require a specific approach**
- **Targeted malware may be analyzed with the RE approach**
- **Worms are analyzed with automatic sandbox systems**
- **Known malware isn't analyzed at all, your AV should do the job**
- **A process should be implemented, that defines which approach has to be applied for the different types of malware**
- **The categories must be defined**

Recommendations - Categories

ERNW Living Security.

1. **Known malware (detected by antivirus solutions), targeting all computer users**
2. **Unknown malware (not yet detected by antivirus solutions), targeting all computer users**
3. **Known (already analyzed) targeted malware, targeting your organization**
4. **Unknown (not yet analyzed) targeted malware, targeting your organization**
5. **Known (already analyzed) targeted malware, targeting VIPs in your organization**
6. **Unknown (not yet analyzed) targeted malware, targeting VIPs in your organization**

Recommendations - Approaches

ERNW Living Security.

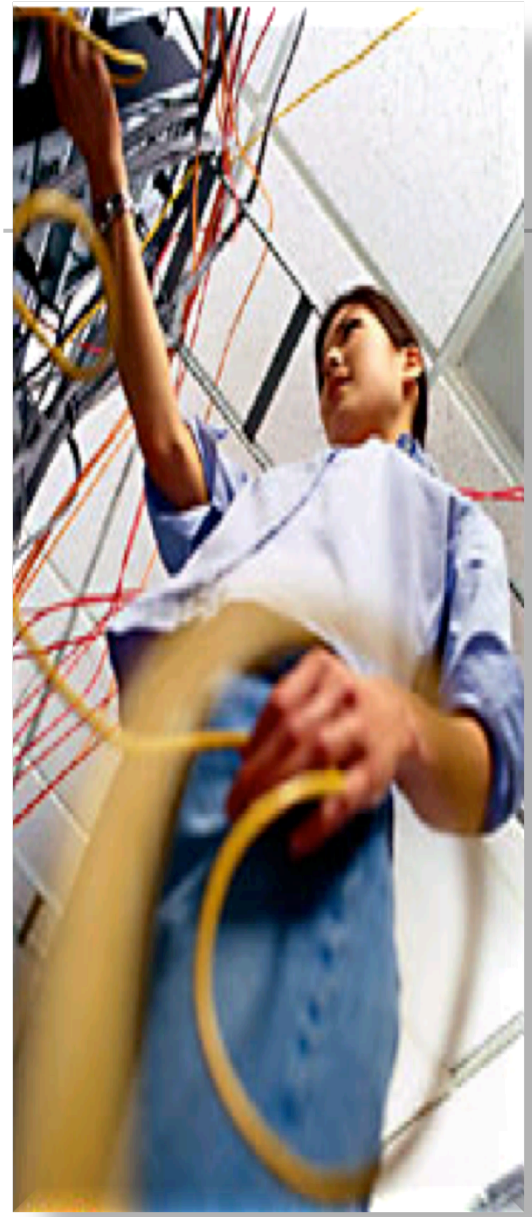
Category	Action	Tool
1.	Nothing, should be detected by AV solution	Antivirus
2.	Acquire sample and analyze	Online sandbox
3.	Inform all users and ensure that AV is up to date	Antivirus
4.	Acquire sample and analyze. Create custom signature for AV and deploy.	Online sandbox (depending on your internal policies) or internal sandbox / Antivirus
5.	Inform targeted users and ensure that AV is up to date	Antivirus
6.	Acquire sample and analyze. Create custom signature for AV and deploy.	Internal sandbox / RE of malware (maybe using partners) / Antivirus

Summary

Final conclusion

ERNW Living Security.

- **Malware analysis isn't just for Antivirus companies anymore**
- **Targeted attacks are forcing enterprises to implement malware analysis procedures**
- **Tools and techniques must be choosen depending on the available knowledge in the organization**
- **Training is needed for all involved personnel**
- **Be prepared and improve your methods and procedures, the blackhats will do**



ERNW
Living Security.

Questions?
And Answers…