# Scripting AntiVirus Signature File Updates and Testing

Randy Abrams
Microsoft Corporation
http://www.microsoft.com

Andreas Marx
AV-Test GmbH
http://www.av-test.org

# Table of Content

- Reasons to script updates
- Types of update mechanisms
- Scripting techniques
- Why test? What to test?
- Sample test scripts
- Conclusion / Q&A section

# Disclaimer

- The examples and code samples in this presentation are provided as is and are only intended for educational purposes and to illustrate concepts
- The code is not suitable for a production environment
- The authors, nor their employers, AV-Test.org, or Microsoft, make any representations or warranties about the suitability of these samples for any other purpose

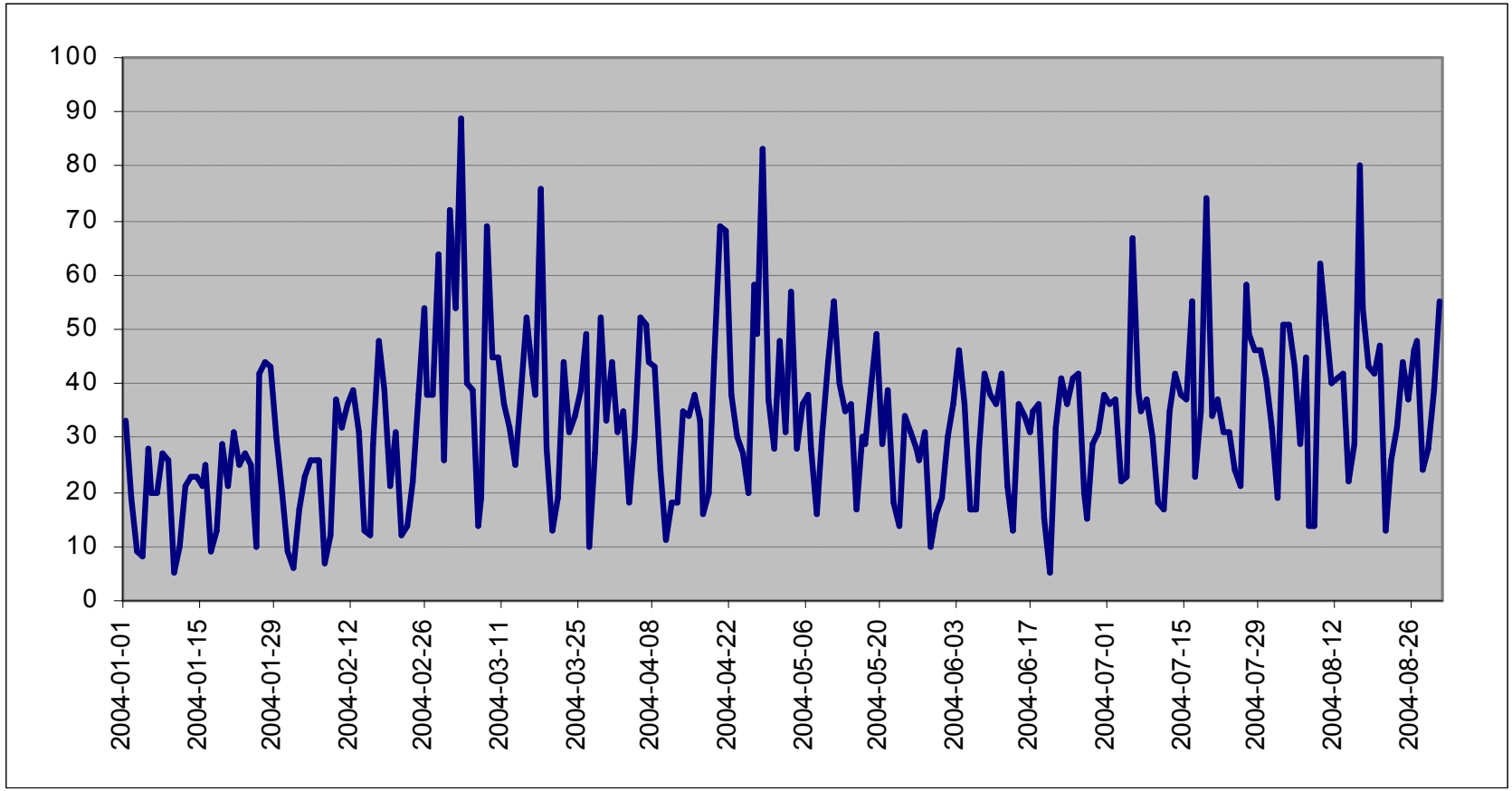
# Reasons to Script Updates (I)

- Testing organizations (e.g. response time tests)
- Antivirus vendor virus name and variant synchronization, cross-reference lists (VGrep)
- PCs not on the internet (e.g. for security reasons)
- Quality Assurance programs & Software Release Services (scan of all incoming and outgoing data)
- Ensure newest signatures

# Reasons to Script Updates (II)

- Multi-scanner systems:
  - Independently developed AV engines have different good and weak points
  - Different response times in case of new virus outbreaks
  - Protection level might still be "too low" for some critical business cases

# Regular Update Releases per Day (I)
# (x = Date, y = Number of Updates)



# Regular Update Releases per Day (II)

- Days with the most update releases:
  - 2004-03-03 (89), 2004-04-28 (83), 2004-08-16 (80), 2004-03-18 (76), 2004-07-19 (74)
- Days with the lowest number of update downloads:
  - 2004-06-20 (5), 2004-01-10 (5), 2004-02-01 (6), 2004-02-07 (7), 2004-01-04 (8)

# Types of Update Mechanisms

- HTTP downloads (public)
- HTTP downloads (username/password)
- FTP downloads (anonymous)
- FTP downloads (username/password)

- Wget.exe handles all

# Scripting Techniques for CMD.EXE

- Filename determination
- Creating unique filenames for archiving
- Determining required files

- Text parsing
- Environment variable parsing
- Subroutines

# Scripting Techniques for CMD.EXE

- Parsing Environment Variables

```
for /f "tokens=1-4 delims=/ " %%i in ("%date%") do (
 set mydate=%%l-%%j-%%l-)
for /f "tokens=1-4 delims=:. " %%i in ("%time%") do (
 set mytime=%%i_%%j_%%k-)
```

%Date%=Thu 11/25/2004  %mydate%=2004-11-25-

%Time%=09:35:37.87  %mytime%=09_35_37

**%mydate%%mytime%= 2004-11-25-09_35_37**

# Scripting Techniques for CMD.EXE

## Finding the file name: McAfee example

***From McAfee's "readme.txt" file: "  - DAT Version:  4297"***

```
wget ftp://ftp.nai.com/pub/datfiles/english/readme.txt
for /f "tokens=4 delims=: " %%i in ('findstr /i /c:"- DAT version:"
    readme.txt') do(
if not exist dat-%%i.zip wget
    ftp://ftp.nai.com//pub/datfiles/english/dat%%i.zip
)
```

***dat-%%i.zip= dat-4297.zip***

# Scripting Techniques for CMD.EXE

```
Set dailycount=0
:daily
wget http://download.nai.com/products/mcafee-
    avert/daily_dats/DAILYDAT.ZIP
if %errorlevel%==0 goto :gethash
set /a dailycount=%dailycount%+1
if %dailycount% GTR 3 del dailydat.z* & exit
goto :daily
```

# Scripting Techniques for CMD.EXE

```
:gethash
for /f %%i in ('md5 dailydat.zip') do set newmcafee=%%i
if exist oldmcafee.bat call oldmcafee.bat
if /i ["%oldmcafee%"]==["%newmcafee%"] del dailydat.zip &
    exit
if /i not ["%oldmcafee%"]==["%newmcafee%"] (
del oldmcafee.bat
echo (set oldmcafee=%newmcafee%) >oldmcafee.bat
ren dailydat.zip %mydate%%mytime%dailydat.zip
echo newfiles>c:\mcafee\newfiles.txt
)
```

# Scripting Techniques for CMD.EXE

Only take what you need: Kaspersky example

- Download file with digital signatures
- Determine which files are required
- Download required files

# Scripting Techniques for CMD.EXE

**Initialize variables**

(set avc_loop=) & (set klb_loop=) & (set newavc=)

(set oldavc=) & (set success=false)

**Using Subroutines (***avp.klb contains digital signatures)*

call :Get_File avp.klb

# Scripting Techniques for CMD.EXE

```
:Get_File
(set avc_loop=0)
(set del_loop=0)
:loop1
(set /a del_loop=%del_loop%+1)
if exist %1 del %1
if not ["%errorlevel%"]==["0"] (
if %del_loop% GTR 2 echo Can't delete %1
    >>c:\kav\download_problem.txt && goto :finalsteps
goto :loop1
)
```

# Scripting Techniques for CMD.EXE

```
(set /a avc_loop=%avc_loop%+1)
wget -aC:\KAV\kavwget.log -t1 -w3 -T360
    ftp://ftp.kaspersky.com/updates_x/%1
if ["%errorlevel%"]==["0"] goto :eof
sleep 30
if %avc_loop% LEQ3 goto :loop1
echo %Date%%time%-Error %1
    download>>c:\kav\update\download_problem.txt
goto :finalsteps
```

# Scripting Techniques for CMD.EXE

findstr /c:".avc" avp.klb >>c:\kav\temp\newklb.txt

C:\UTIL\REP.EXE = , c:\kav\temp\newklb.txt

***The previous line changes entries in the AVP.KLB file that look like this:***

0=kernel.avc,0XLSznpdI71fB300e7Uwj1TdKc+yC8rcaj94jH6EUL9 UuOmxXyKkBjNdD+,8445,09/09/2004

***And writes the line below to a file called "NEWKLB.TXT".***

0,kernel.avc,0XLSznpdI71fB300e7Uwj1TdKc+yC8rcaj94jH6EUL9 UuOmxXyKkBjNdD+,8445,09/09/2004

# Scripting Techniques for CMD.EXE

```
for /f "tokens=1-5 delims=," %%a in (c:\kav\temp\newklb.txt) do (
call :compare1 %%b %%c)

:compare1
set newavc=%2
if exist c:\kav\temp\oldklb.txt (
for /f "tokens=1-3 delims=," %%i in ('findstr /c:"%1" c:\kav\temp\oldklb.txt')
    do (
(set oldavc=%%k)
)
)
if ["%newavc%"]==["%oldavc%"] (set newavc=) & (set oldavc=) & goto
    :eof
echo %1 >>c:\kav\temp\newavcfiles.txt & (set newavc=) & (set oldavc=)
goto :eof
```

# Why Test?

- Update files can be faulty, especially in case of beta definitions:
  - Corrupted virus signatures (e.g. broken archives, interrupted downloads)
  - Download of "old" signatures due to problems with the synchronization
  - Bad or corrupt updates (e.g. the "creation process" at the AV company was erroneous)
- Statistics provided at VB 2004 indicated that of 37,000 updates obtained, approximately 7,000 were non-functioning

# What to Test?

- Different checks are useful to find possible bad updates:
  - Size check: Has the new definition file the "usual" size?
  - Can the ZIP or SFX file be extracted without errors (using digital signature checks or CRC sums)?
  - Does the command-line scanner loads and starts properly?
  - Has the version number increased (this information can be taken from the report file)?
  - Is the scanner able to find Eicar test file under the usual name and does the scanner sets the correct exit code?

# Testing Considerations

- Advanced testing methods with a limited number of "live" viruses (might not be required in every case):
  - Infected files are stored securely on a system with limited access (e.g. limited number of accounts)
  - Infected files should be renamed and/or stored inside of archive files (e.g. ZIP)
  - Deeper test of the internal structures of the AV program
  - Corrupted updates which are not working properly can be found much easier

# Sample Test Script (I)

```
wzunzip -t c:\scanners\mcafee\archives\dailydat.zip
if not "%errorlevel%"=="0" set problem=bad_zip_file & goto :abort
Del /q c:\scanners\mcafee\test\*.*
wzunzip c:\mcafee\archives\dailydat.zip  c:\scanners\mcafee\test
if not "%errorlevel%"=="0" set problem=bad_zip_file & goto :abort
Copy /y c:\scanners\mcafee\test\*.* c:\mcafee
C:\mcafee\scan /sub /secure /unzip /report c:\mcafee\logs\cln.log
    c:\test\cln\*.*
If not %errorlevel%==0 set problem=FP & goto :abort
C:\mcafee\scan /sub /secure /unzip /report c:\mcafee\logs\vir.log
    c:\test\dirty\*.*
```

# Sample Test Script (II)

```
If not %errorlevel%==13 set problem=no_detect goto :abort
C:\mcafee\scan /sub /secure /unzip /report c:\mcafee\logs\vir2.log
    c:\test\dirty c:\test\cln\
If not %errorlevel%==13 set problem=no_detect goto :abort
del c:\scanners\mcafee\update\*.dat
copy c:\scanners\mcafee\test\*.dat c:\scanners\mcafee\update\*.dat
exit
:abort
Echo %date% %time% - %problem%
    >>c:\scanners\logs\Mcafee_Update.log
```

# Conclusion

- Scriptable solutions for single or multi-scanner systems simplify the process of maintaining systems and allow for automated quality testing that can enhance detection in test and production environments
- Virus-infected or suspicious software (regardless, if it's "incoming" or "outgoing") can be stopped early
- More scanners can possibly find more than just one (e.g. due to different response times to new malware and new types of exploits)

# Questions

- Are there any questions?