Workflow

- Execution of Windows-native commands
- ◦ Output is in txt or csv
- Copy all (non-open) event log files
- ◦ Files stored in a directory audit_ with system name
- Also executes some tools from Sysinternals
- ◦ Make sure that they are in the folder **supporttools**
- ◦ ■ autorunsc64.exe
- ◦ ■ csvde.exe
- ◦ ■ psinfo64.exe
- ◦ ■ psloggedon64.exe
- Then ZIPs all files in the directory audit_
- Deletes the audit_ folder

Usage

1. Copy folder security-screening to USB
2. Insert USB in machine to audit
3. Open Explorer, go to USB drive
4. Right click on “auditscript.bat”
5. Choose “Run as administrator”
6. Let the script run until it’s completely finished (the new “bat” window will close). This can take a long (>15 minutes) time
7. Verify that there is a file “audit_.zip”
8. Right click on the USB and eject the USB drive

The zip file “audit_.zip” contains all the evidences.

Script

```
@ECHO OFF

:: Audit script v10
::  v1 : Start
::  v2 : Fixed fetching all users ; include localgroups
::           Removed bugs with jumping to wrong subs from v1
::  v3 : Included scheduled task and startup items
::  v4 : Fix spaces (line wrapping) for systeminfo_inventory.csv
::       Add whoami and gpresult data
::  v5 : Add copy etc/drivers/* files
::  v6 : Add wmic for software list
::  v7 : Add directory listing program file for software list,
list of hotfixes, list of logicaldisks, fw dump
::  v8 : Add wmic for process and service list
::       Add tasklist for loaded modules
::  v9 : Add copy browser info
```

```
::  v10: Add date of exection (to step 2)
::       Handle user and group names with space (step 4)
::       Export Application, Security event log (step 9)
::       Copy all (non-open) logs (step 9)
::       Use msinfo32 to gather systeminfo (IRQs etc) (step 2)
::       Use sysinternal tools to query psinfo, autostart items
and logged-on sessions (step 20)
::       Query active directory for users, groups, orgunits,
sites, domains (step 21)
::       Delete audit directory and create a ZIP archive (step
99)
::
```

Debug

Set this value to **1** if you want to have the progress of the screening displayed in the output screen. This does not negatively impact the performance, it's just additional text output.

```
set debug=1
```

Basic system, user and network information

Gather basic system, user and network information.

```
:: Step 1
:: Get the computer name
:: Needed to create the output directory
if %debug%==1 echo "Fetching system name"
FOR /f "tokens=2,* delims= " %%a in ('IPCONFIG ^/ALL ^| FINDSTR
"Primary Dns"') do set tempsuffix=%%b
FOR /f "tokens=1,2 delims=:" %%a in ('echo %tempsuffix%') do set
dnssuffix=%%b
SET FQDN=%COMPUTERNAME%.%DNSSUFFIX:~1%

ECHO Server FQDN: %FQDN%
set aud_dir=audit_%FQDN%
mkdir %aud_dir%
cd %aud_dir%


:: Step 2
:: Operating system version and system information
date /T > date_of_execution.txt
time /T >> date_of_execution.txt

if %debug%==1 echo "ver"
ver > ver.txt

if %debug%==1 echo "systeminfo"
systeminfo > systeminfo.txt
```

```
systeminfo /fo CSV > systeminfo.csv
msinfo32 /categories +all /report systeminfo_msinfo.xls


:: Step 3
:: Grap info from systeminfo for inventory template
set inventory_hostname=
set inventory_osname=
set inventory_osversion=
set inventory_installdate=
set inventory_boottime=
set inventory_system_manufacturer=
set inventory_timezone=
set inventory_productid=

for /f "usebackq tokens=2 delims=:" %%s in (`type systeminfo.txt
^| findstr /B /C:"Host Name:"`) do (
 set inventory_hostname=%%s
)
for /f "usebackq tokens=2 delims=:" %%s in (`type systeminfo.txt
^| findstr /B /C:"OS Name:"`) do (
 set inventory_osname=%%s
)
for /f "usebackq tokens=2 delims=:" %%s in (`type systeminfo.txt
^| findstr /B /C:"OS Version:"`) do (
 set inventory_osversion=%%s
)
for /f "usebackq tokens=2,3,4 delims=:" %%s in (`type
systeminfo.txt ^| findstr /C:"Original Install Date:"`) do (
 set inventory_installdate=%%s:%%t:%%u
)
for /f "usebackq tokens=2,3,4 delims=:" %%s in (`type
systeminfo.txt ^| findstr /C:"System Boot Time:"`) do (
 set inventory_boottime=%%s:%%t:%%u
)
for /f "usebackq tokens=2 delims=:" %%s in (`type systeminfo.txt
^| findstr /B /C:"System Manufacturer:"`) do (
 set inventory_system_manufacturer=%%s
)
for /f "usebackq tokens=2,* delims=:" %%s in (`type
systeminfo.txt ^| findstr /B /C:"Time Zone:"`) do (
 set inventory_timezone=%%s:%%t
)
for /f "usebackq tokens=2 delims=:" %%s in (`type systeminfo.txt
^| findstr /B /C:"Product ID:"`) do (
 set inventory_productid=%%s
)

for /f "tokens=* delims= " %%G in ("%inventory_hostname%") do
set inventory_hostname=%%G
for /f "tokens=* delims= " %%G in ("%inventory_osname%") do set
inventory_osname=%%G
```

```
for /f "tokens=* delims= " %%G in ("%inventory_osversion%") do
set inventory_osversion=%%G
for /f "tokens=* delims= " %%G in ("%inventory_installdate%") do
set inventory_installdate=%%G
for /f "tokens=* delims= " %%G in ("%inventory_boottime%") do
set inventory_boottime=%%G
for /f "tokens=* delims= " %%G in
("%inventory_system_manufacturer%") do set
inventory_system_manufacturer=%%G
for /f "tokens=* delims= " %%G in ("%inventory_timezone%") do
set inventory_timezone=%%G
for /f "tokens=* delims= " %%G in ("%inventory_productid%") do
set inventory_productid=%%G

echo %inventory_hostname% ; %FQDN% ; %inventory_osname% ;
%inventory_osversion% ; %inventory_installdate% ;
%inventory_boottime% ; %inventory_system_manufacturer% ;
%inventory_timezone% ; %inventory_productid%  >
systeminfo_inventory.csv


:: Step 4
:: User and account information
:: Service information
if %debug%==1 echo "net start"
net start > net_start.txt


if %debug%==1 echo "net user"
net user > net_user.txt
if %debug%==1 echo "net account"
net accounts > net_accounts.txt
if %debug%==1 echo "net use"
net use > net_use.txt
if %debug%==1 echo "net view"
net view > net_view.txt
if %debug%==1 echo "net config server"
net config server >> net_config.txt
if %debug%==1 echo "net config workstation"
net config workstation >> net_config.txt
if %debug%==1 echo "net localgroup"
net localgroup >> net_localgroup.txt
echo > net_localgroup_detail.txt

for /F "tokens=* eol=- skip=2" %%a in (net_localgroup.txt) do
call :processlocalgroup %%a


if %debug%==1 echo "users"
echo > users_detail.txt

for /F "tokens=* delims=  eol=- skip=2" %%a in (net_user.txt) do
```

```
call :processuser %%a



:: Step 5
:: Network information

if %debug%==1 echo "ipconfig dns"
ipconfig /displaydns > ipconfig_dnscache.txt

if %debug%==1 echo "ipconfig"
ipconfig /all > ipconfig_all.txt
if %debug%==1 echo "route"
route print > route_print.txt
if %debug%==1 echo "fw"
netsh firewall show state >> fw_config.txt
netsh firewall show config >> fw_config.txt
netsh advfirewall firewall show rule name=all > fwadv_config.txt
netsh dump > fw_dump.txt

if %debug%==1 echo "rpc"
netsh rpc show >> rpc_config.txt

if %debug%==1 echo "netstat"
netstat -nao > netstat.txt
netstat -naob > netstat_naob.txt

if %debug%==1 echo "netstat stats"
netstat -s > netstat_stats.txt

if %debug%==1 echo "arp"
arp -a > arp.txt
arp -a -v > arp_verbose.txt

if %debug%==1 echo "nbtstat"
nbtstat -n > nbtstat_n.txt
nbtstat -c > nbtstat_c.txt
nbtstat -s > nbtstat_s.txt


:: Step 6
:: Running procecess
if %debug%==1 echo "ps"
tasklist > tasklist.txt
tasklist /v > tasklist_verbose.txt
tasklist /SVC > tasklist_svc.txt
tasklist /v /FO CSV > tasklist.csv
tasklist /SVC /FO CSV > tasklist_svc.csv
tasklist /M /FO CSV > tasklist_loaded_modules.csv

:: Step 7
```

```
:: Installed software
:: Installed services

if %debug%==1 echo "installed"

echo ================ >>software_list.txt
reg export
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
temp1.txt
find "DisplayName" temp1.txt| find /V "ParentDisplayName" >
temp2.txt
for /f "tokens=2,3 delims==" %%a in (temp2.txt) do (echo %%a >>
software_list.txt)
del temp1.txt
del temp2.txt

for /f "usebackq tokens=1,2,3 delims=:" %%i in (`sc query
state^= all`) do (
  rem echo %%i %%j %%k
  if "%%i"=="SERVICE_NAME" call :%%i %%j %%k
)

if %debug%==1 echo "installed - wmic"
:: wmic  /output:software_list_wmic.csv  product get * /
format:"%WINDIR%\System32\wbem\en-US\csv"
wmic  /output:software_list_wmic.csv  product get * /format:csv

dir /a "C:\Program Files" > software_list_programfiles.txt
dir /a "C:\Program Files (x86)" >
software_list_programfiles_x86.txt

wmic /output:software_list_hotfixes.csv qfe list /format:csv
```

Group Policies

Querying the Group Policies can require a lot of time, especially on servers. This function is now disabled by default. Remove the “::” in front of gpresult to enable it again.

```
:: Step 8
:: Policies
::if %debug%==1 echo "policies"
::gpresult /r > gpresult.txt
::gpresult /x gpresult.xml
::gpresult /h gpresult.html
```

Fetch the log configuration

Fetch the log configuration and exports some log files to text. As a final action, it also copies all the evtx files to a seperate directory (log) in the audit folder.

```
:: Step 9
:: Log configuration setup
if %debug%==1 echo "log files"
wevtutil gl Application > log_config_application.txt
wevtutil gli Application >> log_status_application.txt
wevtutil gl Security > log_config_security.txt
wevtutil gli Security >> log_status_security.txt
wevtutil gl Setup > log_config_setup.txt
wevtutil gli Setup >> log_status_setup.txt
wevtutil gl System > log_config_system.txt
wevtutil gli System >> log_status_system.txt


wevtutil qe Application > log_export_application.txt
wevtutil qe Security > log_export_security.txt
wevtutil qe System > log_export_system.txt
wevtutil qe "Windows PowerShell" > log_export_powershell.txt

wevtutil epl Application application.evtx
wevtutil epl System system.evtx
wevtutil epl Security security.evtx
wevtutil epl Microsoft-Windows-RemoteDesktopServices-RdpCoreTS/
Operational rdpcore.evtx
wevtutil epl "Windows PowerShell" powershell.evtx

wevtutil epl "Microsoft-Windows-Windows Firewall With Advanced
Security/ConnectionSecurity" firewall_ConnectionSecurity.evtx
wevtutil epl "Microsoft-Windows-Windows Firewall With Advanced
Security/ConnectionSecurityVerbose"
firewall_ConnectionSecurityVerbose.evtx
wevtutil epl "Microsoft-Windows-Windows Firewall With Advanced
Security/Firewall" firewall_Firewall.evtx
wevtutil epl "Microsoft-Windows-Windows Firewall With Advanced
Security/FirewallVerbose" firewall_FirewallVerbose.evtx

wevtutil epl "Microsoft-Windows-Terminal-Services-
RemoteConnectionManager/Operational"
rdp_RemoteConnectionManager.evtx
wevtutil epl "Microsoft-Windows-TerminalServices-
LocalSessionManager/Operational" rdp_LocalSessionManager.evtx

echo "Attempt to copy all log files"
xcopy /E/H/C/I "%SystemRoot%\System32\Winevt\Logs" logs

:: Step 10
:: USB Information
if %debug%==1 echo "USB"
reg export
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB"
reg_enum_usb.txt
copy %SYSTEMROOT%\inf\setupapi.app.log .
copy %SYSTEMROOT%\inf\setupapi.dev.log .
```

```
:: Step 11
:: Driver Information
if %debug%==1 echo "drivers"
driverquery > driverquery.txt
driverquery /v /FO CSV > driverquery.csv


:: Step 12
:: Get scheduled tasks
if %debug%==1 echo "scheduled tasks"
schtasks /query /FO CSV /V >schtasks.csv


:: Step 13
:: Get startup items
if %debug%==1 echo "startup items"
::wmic /output:wmic_startup.csv startup list full /
format:"%WINDIR%\System32\wbem\en-us\csv"
wmic /output:wmic_startup.csv startup list full /format:csv

:: Step 14
:: Get whoami information
if %debug%==1 echo "whoami"
whoami /user /fo csv > whoami_user.csv
whoami /groups /fo csv > whoami_groups.csv
whoami /priv /fo csv > whoami_priv.csv


:: Step 15
:: Get group policy results
:: Already fetched in a previous setup


:: Step 16
:: Copy files from drivers/drivers_etc_networks
if %debug%==1 echo "network drivers"
copy %windir%\system32\drivers\etc\networks
drivers_etc_networks.txt
copy %windir%\system32\drivers\etc\hosts drivers_etc_hosts.txt


:: Step 17
:: List of logical disks
if %debug%==1 echo "logical disks"
wmic /output:logicaldisk.csv logicaldisk get caption,
description, providername, filesystem,volumeserialnumber /
format:csv


:: Step 18
```

```
:: ProcessList via wmic
if %debug%==1 echo "process list wmic"
wmic /output:process_list_wmic.csv  process get ProcessID,
Caption, ExecutablePath, CreationDate, ParentProcessID,
SessionId,CommandLine /format:csv

:: Service list
if %debug%==1 echo "service list wmic"
wmic /output:service_list_wmic.csv  service get name, pathname,
processid, startmode, state /format:csv

:: Logon list
if %debug%==1 echo "logon list wmic"
wmic /output:logon_wmic.csv logon list full /format:csv
```

Browser data

Get the browser data for Firefox and Chrome.

```
:: Step 19
:: Browser data
if %debug%==1 echo "browser data"
xcopy "C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\" firefox_profiles_user /E /H /C /I
xcopy "C:\Users\%user%
\AppData\Roaming\Mozilla\Firefox\Profiles\" firefox_profiles /E /
H /C /I
xcopy "C:\Users\%user%\AppData\Local\Google\Chrome\User Data\"
chrome_userdata /E /H /C /I
xcopy "C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\"
chrome_userdata_user /E /H /C /I
```

Sysinternals

Execute the tools from sysinternals. This gets additional system information, lists the applications that are automatically started and who recently logged on to the system.

```
:: Step 20
:: Sysinternal tools
if %debug%==1 echo "sysinternals"
"../supporttools/psinfo64.exe" /accepteula -h -d -s -c  >
systeminfo_psinfo.csv
"../supporttools/autorunsc64.exe" /accepteula -c >
sysinternals_autoruns.csv
"../supporttools/PsLoggedon64.exe" /accepteula > psloggedon.txt
```

Active Directory

Execute some AD queryies to get the users, groups, domains and sites.

```
:: Step 21
:: AD-data
mkdir adquery_logs
"../supporttools/csvde.exe" -v -r "(objectClass=user)" -n -j
adquery_logs\ -f adquery_users.csv
"../supporttools/csvde.exe" -v -r "(objectClass=group)" -n -j
adquery_logs\ -f adquery_group.csv
"../supporttools/csvde.exe" -v -r
"(objectClass=organizationalUnit)" -n -j adquery_logs\ -f
adquery_orgunits.csv
"../supporttools/csvde.exe" -v -r "(objectClass=domain)" -n -j
adquery_logs\ -f adquery_domain.csv
"../supporttools/csvde.exe" -v -r "(objectClass=site)" -n -j
adquery_logs\ -f adquery_site.csv
```

ZIP archive

Create a ZIP archive and remove the audit directory.

```
:: Step 99
:: Make a ZIP archive
cd ..
"supporttools/7za.exe" a -bd -tzip %aud_dir%.zip %aud_dir%
rmdir /S /Q %aud_dir%
```

End

```
:: END

exit /b


:: SUBROUTINES

:processlocalgroup
set mygroup=%*
set mygroup2=%mygroup:~1,200%

echo %mygroup2% >> net_localgroup_detail.txt
echo ------------- >>  net_localgroup_detail.txt
net localgroup "%mygroup2%" >> net_localgroup_detail.txt
echo >> net_localgroup_detail.txt
exit /b

:processuser
```

```
if {%1}=={} goto :end_user
net user "%1" >> users_detail.txt

shift
goto :processuser

:end_user


:SERVICE_NAME
::  echo %0 %1 %2
  set a=%1
  set a=%a:(=_%
  set a=%a:)=_%
  if     "%2"==""   call :process-service %a%
  if not "%2"=="" call :process-service %a%$%2


:process-service
::if %debug%==1 echo service "%1"

  set service_display_name=
  set service_name=
  set service_pid=
  set service_properties=
  set service_state=
  set service_type=
:: `sc query` and `sc queryex` will only show DISPLAY_NAME when
no SERVICE_NAME is specified
:: so we have to perform `sc query` for ALL services, then grab
the DISPLAY_NAME for the matching SERVICE_NAME
  for /f "usebackq tokens=1,* delims=:, " %%s in (`sc query
state^= all`) do (
    rem if "%%s"=="STATE" if not !%1!==!! echo %%v state of %1
is %%v
    if "%%s"=="SERVICE_NAME" set service_name=%%t
    if "%%s"=="DISPLAY_NAME" if "!service_name!"=="%1" set
service_display_name=%%t
    set first_char=%%s
    set first_char=!first_char:~0,1!
    if "!first_char!"=="(" if "!service_name!"=="%1" set
service_properties=%%s, %%t
    rem echo "!first_char!", !service_properties!, %%s, %%t
  )
  set service_name=
:: find all services by SERVICE_NAME, then list STATE, TYPE,
DISPLAY_NAME, and "" (this is on the line below STATE)
  for /f "usebackq tokens=1,2,3,4 delims=:, " %%s in (`sc
queryex "%1"`) do (
    rem if "%%s"=="STATE" if not !%1!==!! echo %%v state of %1
is %%v
    if "%%s"=="PID" set service_pid=%%t
```

```
    if "%%s"=="SERVICE_NAME" set service_name=%%t
    if "%%s"=="STATE" set service_state=%%u
    if "%%s"=="STATE" set service_state=%%u
    if "%%s"=="TYPE" set service_type=%%u
    if "%%s"=="STATE" set service_state=%%u
    rem echo "%%s", "%%t", "%%u", "%%v"
  )
  echo %service_pid%, %service_state%, %service_type%,
%service_name%, %service_properties%, %service_display_name% >>
SERVICE_list.txt
```