

Task 1. Log Export

Purpose: understand how is possible to export OS Windows log files.

After the work the student must

- know: what is windows log files, where are they stores, what is log configuration;
- be able to: conduct export of Windows logs.

Tasks:

- get full list of OS Windows logs
- get configuration of Windows logs
- detect non-empty logs
- export all non-empty logs

Material and technical equipping of the workplace

- Suspicious Windows VM
- wevtutil tool (<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil>)

Solution:

Enumerate OS Windows logs.

1. Start CMD console with administrator rights
2. Use “wevtutil el” command to enumerate all logs

Or

1. Start Windows Explorer
2. Open folder “C:\Windows\System32\winevt\Logs\”

Get configuration of Windows logs

1. Start CMD console with administrator rights
2. Use “wevtutil gl LOG_NAME /f:xml” to get configuration of specified log

See several recent events from specified log

1. Start CMD console with administrator rights
2. Use “wevtutil qe LOG_NAME /c:N /rd:true /f:text” to get N recent events from specified log

Export OS Windows logs

1. Start CMD console with administrator rights
2. Use “wevtutil epl LOG_NAME FILENAME.evtx” to

Task 2. Windows Log Analysis

Purpose: understand principle of Windows logs analysis.

After the work the student must

- know: how is possible to analyse OS Windows logs;
- be able to: conduct manual Windows logs analysis, find suspicious actions etc.

Tasks:

- Analyze exported logs
- Find suspicious activities
- Restore timeline of hackers attack

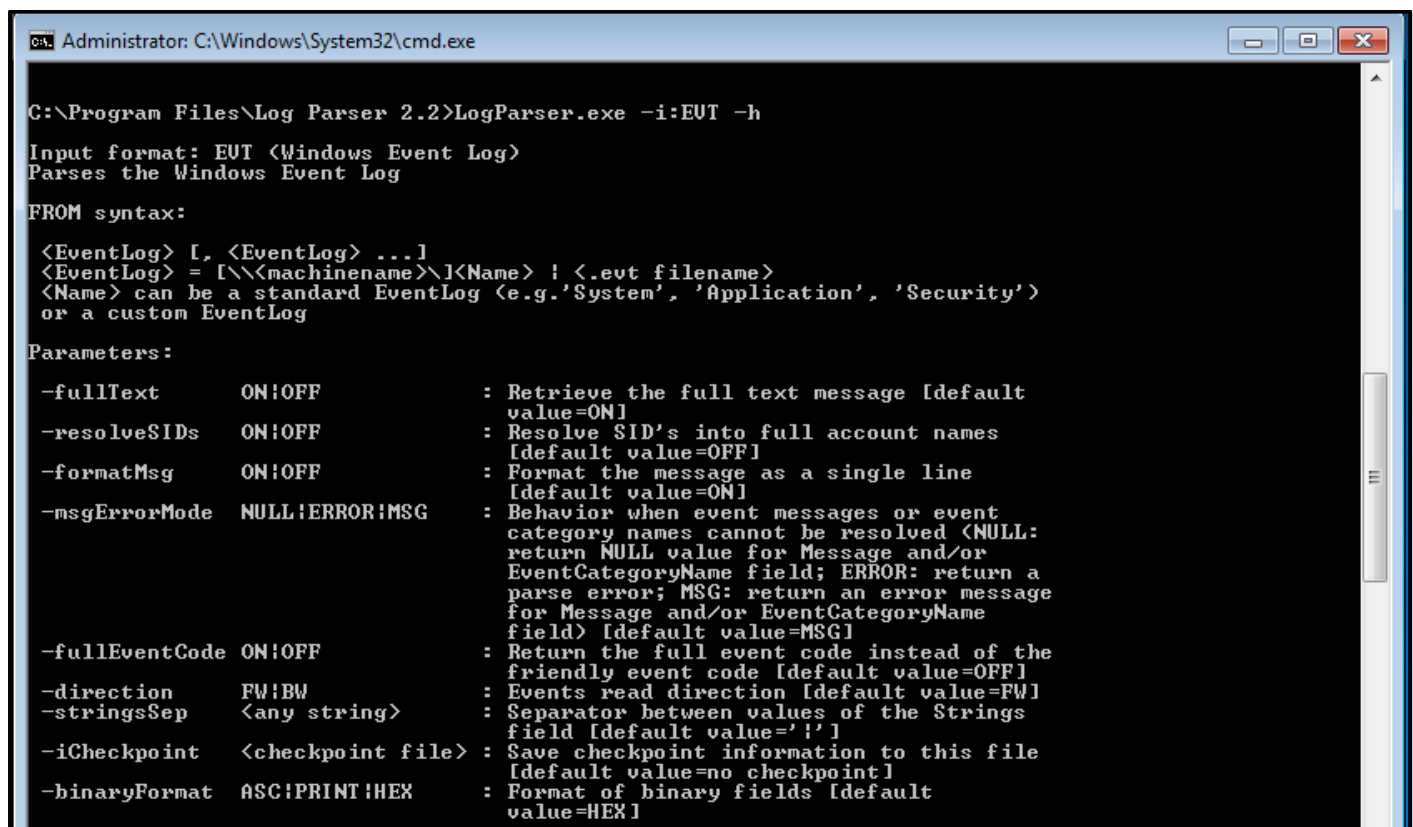
Material and technical equipping of the workplace

- Suspicious Windows VM
- Windows Event Viewer (Control Panel -> System and Security -> Administrative tools -> Event Viewer)
- Microsoft Log Parser (<https://www.microsoft.com/en-us/download/details.aspx?id=24659>)
(already downloaded and installed on suspicious Windows VM into Program Files)

Solution examples:

Enumerate list fields from specified log with Log Parser:

Use command 'LogParser.exe -i:EVT -h'



```
C:\Program Files\Log Parser 2.2>LogParser.exe -i:EVT -h

Input format: EVT <Windows Event Log>
Parses the Windows Event Log

FROM syntax:

<EventLog> [, <EventLog> ...]
<EventLog> = [\\<machinename>\]<Name> | <.evt filename>
<Name> can be a standard EventLog (e.g. 'System', 'Application', 'Security')
or a custom EventLog

Parameters:

-fullText      ON!OFF      : Retrieve the full text message [default
                           value=ON]
-resolveSIDs   ON!OFF      : Resolve SID's into full account names
                           [default value=OFF]
-formatMsg     ON!OFF      : Format the message as a single line
                           [default value=ON]
-msgErrorMode  NULL!ERROR!MSG : Behavior when event messages or event
                           category names cannot be resolved (NULL:
                           return NULL value for Message and/or
                           EventCategoryName field; ERROR: return a
                           parse error; MSG: return an error message
                           for Message and/or EventCategoryName
                           field) [default value=MSG]
-fullEventCode ON!OFF      : Return the full event code instead of the
                           friendly event code [default value=OFF]
-direction    FW!BW       : Events read direction [default value=FW]
-stringsSep    <any string> : Separator between values of the Strings
                           field [default value='!']
-iCheckpoint   <checkpoint file> : Save checkpoint information to this file
                           [default value=no checkpoint]
-binaryFormat  ASC!PRINT!HEX : Format of binary fields [default
                           value=HEX]
```

```
Fields:
EventLog <S>           RecordNumber <I>           TimeGenerated <T>
TimeWritten <T>        EventID <I>           EventType <I>
EventTypeNames <S>     EventCategory <I>    EventCategoryName <S>
SourceName <S>         Strings <S>          ComputerName <S>
SID <S>                Message <S>          Data <S>

Examples:

Create an XML report file containing logon account names and dates from the
Security Event Log messages:

LogParser "SELECT TimeGenerated AS LogonDate, EXTRACT_TOKEN<Strings, 0,
'!'> AS Account INTO Report.xml FROM Security WHERE EventID NOT IN
<541;542;543> AND EventType = 8 AND EventCategory = 2"

Get the distribution of EventID values for each Source:
```

Get all events from exported log with Log Parser:

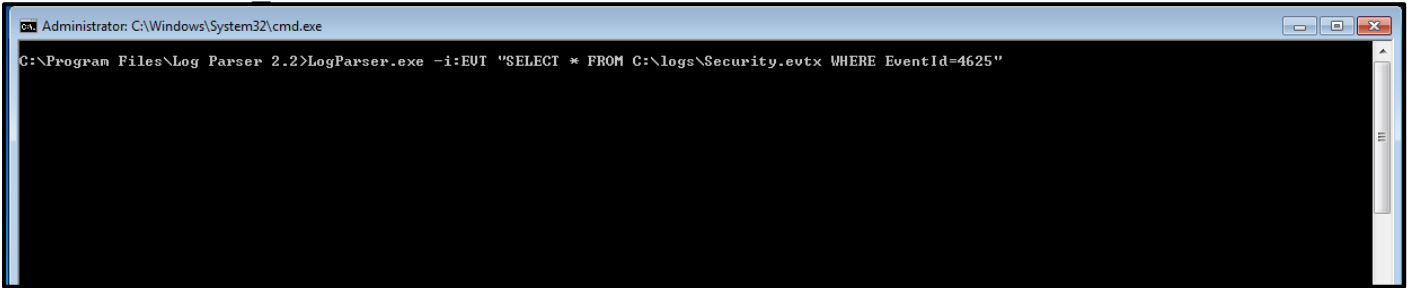
Use command 'LogParser.exe -i:EVT "SELECT * FROM FILE_NAME.evtx"'



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Log Parser 2.2>LogParser.exe -i:EVT "SELECT * FROM C:\logs\Security.evtx" -q_
```

Get specified item from exported log with Log Parser:

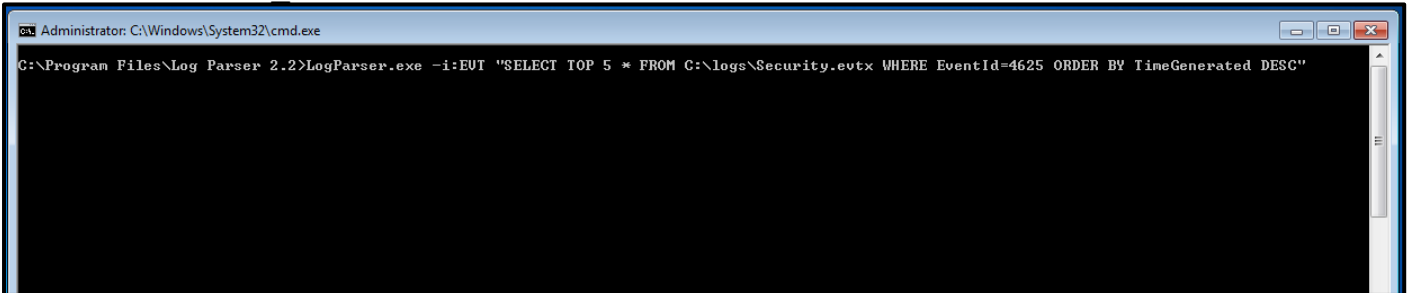
Use command 'LogParser.exe -i:EVT "SELECT * FROM FILE_NAME.evtx WHERE EventId=EVENT ID"'



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Log Parser 2.2>LogParser.exe -i:EVT "SELECT * FROM C:\logs\Security.evtx WHERE EventId=4625"
```

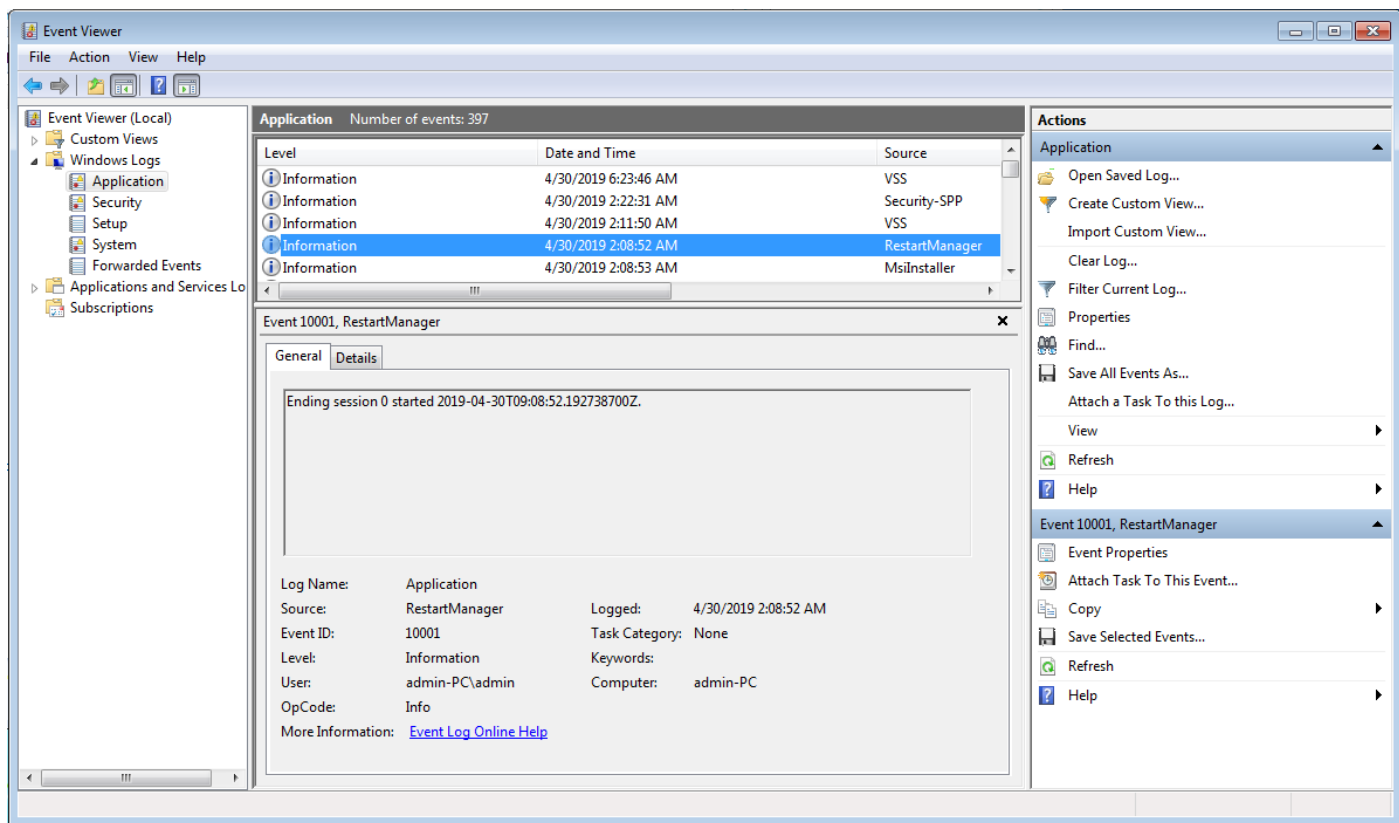
Select limited ordered events with Log Parser

Use command 'LogParser.exe -i:EVT "SELECT TOP N * FROM FILE_NAME.evtx ORDER BY FIELD NAME DESC/ASC"'



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Log Parser 2.2>LogParser.exe -i:EVT "SELECT TOP 5 * FROM C:\logs\Security.evtx WHERE EventId=4625 ORDER BY TimeGenerated DESC"
```

Obtain detailed view of event in Event Viewer:



Get count of events with Log Parser (strongly recommended to start investigation with this approach).

Use command 'LogParser.exe -i:EVT "SELECT *, count(*) as count FROM FILE_NAME.evtx GROUP BY FIELD_NAME"'

