

Audit Viewer

User Guide

Version 1.0.0.0



Audit Viewer

Audit Viewer is an open source tool that allows users to examine the results of Memoryze's (<http://www.mandiant.com/software/Memoryze.htm>) analysis. Audit Viewer allows the incident responder or forensic analyst to quickly view complex XML output in an easily readable format. Using familiar grouping of data and search capabilities, Audit Viewer makes memory analysis quicker and more intuitive.

License

Audit Viewer is released under the BSD license.

```
#Copyright (c) 2008, MANDIANT Corporation
#All rights reserved.
#
#Redistribution and use in source and binary forms, with or without
#modification, are permitted provided that the following conditions are met:
#
# * Redistributions of source code must retain the above copyright notice,
#   this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
# * Neither the name of the MANDIANT Corporation nor the names of its
#   contributors may be used to endorse or promote products derived from
#   this software without specific prior written permission.
#
#THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
#AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
#IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
#ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
#LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
#CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
#SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
#INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
#CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
#POSSIBILITY OF SUCH DAMAGE.
```

Installation Prerequisites

To run Audit Viewer, the user must have Python 2.5 or 2.6 and the wxPython library installed. The user can download these from:

- <http://www.python.org>
- <http://wxpython.org/download.php#binaries>

Feature List

- Process data can be viewed on a per process basis or in its entirety by double clicking the root node, "Processes". For example, when you double click on "Processes" and then click on the Files tab, all the file handles open on the host are displayed from least frequently to most frequently occurring.

- Ability to search Files, Processes, Mutants, Events, Registry Keys, and Strings using plain text or regex.
- Ability to load multiple Memoryze result sets contained in the same directory.
- Handle types are separated out into more abstract types representing the logical type of the handle.
 - Files
 - Directories (part of the Object Manager's namespace)
 - Processes
 - Keys
 - Mutants
 - Events
- Memory sections with names are displayed under the DLLs tab.
- Layered drivers are displayed in a tree view. *This is useful for finding certain types of keyboard sniffers, network sniffers, and file filtering drivers.*
- Integrated with Memoryze to seamlessly acquire drivers and processes from live memory and images.
- Ability to scan all processes for “questionable” executable sections. *These sections have the EXECUTE_READWRITE flag but no name.*

Using Audit Viewer

Setup

Install the prerequisite packages and then double click on Audit Viewer.py from Windows Explorer.

Configuration

First, determine if you are running on live memory or a memory image/snapshot.

If you are running on a memory image, you should check “Running on image” and put the path to the image file in the textbox.

Running on image
Path to image file:

NOTE: “Running on live memory” means that the Audit Viewer is running on the same machine as Memoryze. If you ran Memoryze in a VM and copied the results, selecting “Running on live memory” would not be true, and could lead to acquisition problems.

The next step in the configuration process is to ensure that Audit Viewer has the correct install directory for Memoryze if you want the Audit Viewer to be able to launch acquires interactively. By default, Memoryze is installed to c:\Program Files\Mandiant\Memoryze.

Memoryze Install Directory:

Loading and viewing audits

Once the two previous steps are finished, the user can click the “Open Audit” button in the top right corner of Audit Viewer and open the directory that contains the audit(s) to be examined.

Open Audit

Each top level tab represents a different audit:

ProcessAuditMemory DriverAuditSignature DriverAuditModuleList RootkitAudit

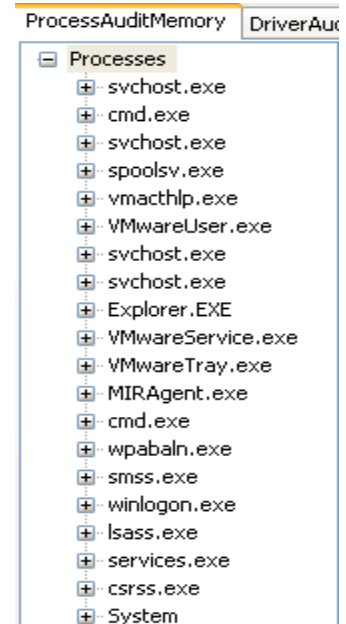
- ProcessAuditMemory – the results of a Process.bat or ProcessAuditMemory.Batch.xml.
- DriverAuditSignature – the results of a DriverSearch.bat or DriverAuditSignature.Batch.xml.
- DriverAuditModuleList – the results of a DriverWalkList.bat or DriverAuditModuleList.Batch.xml.
- RootkitAudit – the results of a HookeDetection.bat or RootkitAudit.Batch.xml.

ProcessAuditModule Tab

Under this tab on the left is a tree with a root node of “Processes”. By expanding this node, the user will see a listing of all the processes that were running when Memoryze ran. Each individual process node is expandable as well and contains process metadata. Process metadata is made up of:

- PID
- Parent PID
- Arguments
- Path
- SecurityID *Not on all images*
- Username *Not on all images*

Double clicking on any process’ name will result in that process’s data filling out the tabs on the right. Clicking on the process metadata will result in no action.



The first six tabs going from left to right represent a handle broken into its different categories. This data will only be present if Process.bat from Memoryze was run with the “–handles true” option.

- Files – will contain all file handles for the selected process.
- Directories – will contain all directory handles from the Object Manager’s namespace for the selected process.
- Processes – will contain all process handles for the selected process.
- Keys – will contain all open Registry Key handles for the selected process.
- Mutants – will contain all open mutex handles for the selected process.
- Events – will contain all open event handles for the selected process.

Files Directories Processes Keys Mutants Events Dlls Strings Memory Sections Ports

The next four tabs represent data acquired by running Process.bat with “-ports true”, “-sections true”, and “-strings true”.

- DLLs – all loaded DLL memory sections. *NOTE: If a process has been injected using certain injection methods, the DLL’s name is not associated with a memory section so the memory section will be listed, but the DLL name will not be shown. Memoryze parameter “-sections true”.*
- Strings –The strings found across the process address space which includes the EXE, DLLs, heap and stack. *Memoryze parameter “-strings true”.*
- Memory Sections – all memory sections used by the process. *Memoryze parameter “-sections true”.*
- Ports – all open/listening ports the process was using. *Memoryze parameter “-ports true”.*

Advanced ProcessAuditModule Usage

- If the user right clicks on a process name, they will be given the option to acquire the process from live memory or from a memory image depending what options have been configured.
- If the user right clicks on an open process in the “Processes” tab on the right, they can acquire processes that the given process has open.
- Double clicking on the root node “Processes” (on the left side) will result in all process information being sorted by least frequent occurrence. This means that all file handles will be shown in the “Files” tab and they will have been sorted. The files that are opened and have the fewest occurrences across all processes will be at the top of the list control. The files that are open within many processes are going to be at the bottom of the list. This is true for almost every tab. This is designed to assist with malware analysis since it weeds out the commonly occurring data and focuses the investigation or analysis on the anomalies in memory.
- Right clicking on the root node “Processes” will give the user three options:
 - Scan Process for executable memory – will scan all processes memory sections looking for memory sections with the permission EXECUTE_READWRITE. It will prompt the user showing them which processes have this memory section attribute. At Hack In The Box Dubai, Jamie Butler talked about how this can be used to find injected DLLs and shellcode.
 - Find – See the section on Searching.
 - Clear Find Results – clear the results from the Find operation.

Searching all processes

When a user right clicks on the process root node and selects “Find”, they will be prompted with a new dialog box. This dialog box is made up of a text box, a bunch of check boxes and a search button.

- Regex Search check box – when this is checked, the search string will be compiled into a regular expression and applied to the specified data set.
- Case Insensitive check box – when this is checked, the search becomes case insensitive regardless of the user regex expression or search text.

Search Criteria is used to determine which data fields to search. **Currently only ONE field will be searched per search request.**

- Search File handles – will search all file names and apply the search text or regex to the file name.
- Search Registry Keys – will search all Registry Key names and apply the search text or regex to the Registry Key.
- Search Mutants – will search all mutex names and apply the search text or regex to the Mutex name.
- Search Events – will search all event names and apply the search text or regex to the Event name.
- Search Strings – will search all strings and apply the search text or regex to the string.
- Search DLLs – will search all DLL names and apply the search text or regex to the DLL name.

When the search is done, the dialog box will disappear. If a match occurred, then the process that contains the hits will have its font color changed to blue in the process tree. Double clicking on the process that contains a hit will load the results. The user will have to navigate to the tab that represents their search criteria. For example, if a search on strings was done, the user would click the “Strings” tab once the data is loaded. The user will notice that within the list there are now fields that are colored blue. These blue rows represent hits and are placed at the top of the list so that the user does not have to look through potentially thousands of rows to find the hits.

Example Search Terms:

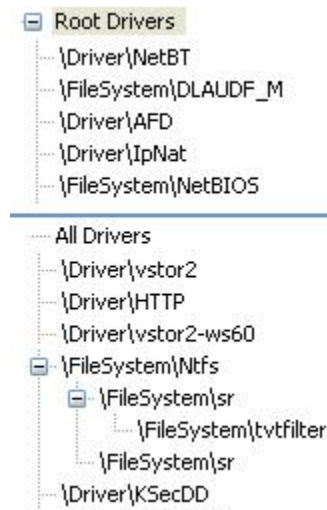
- Check Regex Search and Case Insensitive
 - Check “Search DLLs”
 - Type “.*kernel32.*”
 - Expected result will be all processes excluding system
- Check Regex Search, Case Insensitive, Search File handles
 - Type “.*pagefile.*”
 - Expected result will be in system, “pagefile.sys”
- Check Regex Search, Case Insensitive, Search File handles
 - Type “.**.txt”
 - Expected result will be any process with an open text file

DriverAuditSignature Tab

This tab has two tree views. The top tree view is “Root Drivers”. Root drivers are drivers that have nothing attached to them or are at the highest level of the attachment tree. (Drivers are attached to them, but they are attached to nothing else.) The bottom tree view is all drivers. Some drivers have an expandable feature indicating that something is attached to them. The user can determine what is attached by expanding the tree.

NOTE: DriverName does not necessarily mean the driver’s name on disk.

Double clicking on any driver name will result in the driver’s metadata being loaded into the list control on the right.



By right clicking on any given driver, the user can acquire it in either tree view. By right clicking on the root node in either tree, the user can acquire all drivers loaded on the system.

DriverAuditModuleList Tab

This is a list view of all the drivers and modules in the PsLoadedModuleList linked list- an operating system maintained list. If a user right clicks on any of the items in the list, they can be acquired from memory.

NOTE: ntdll.dll is not included in this list because ntdll.dll is not actually in the PsLoadedModuleList. It is in the MmLoadedUserImagelist.

RootkitAudit Tab

The RootkitAudit tab contains three separate tabs. These three tabs represent the currently detected hooks by Memoryze. *NOTE: Not all hooks are rootkits. Legitimate software often hooks the operating system in order to provide additional security and expanded features.*



- System Service Descriptor Table (SSDT) – this list contains any System Call Table entries that have been hooked.
- Interrupt Descriptor Table (IDT) – report if the page fault handler and the system interrupt have been hooked.
- Interrupt Request Packet (IRP) – displays all loaded driver IRP tables have been hooked. It is common that the operating system will hook its own drivers, which results in a lot of IRP hooks

being reported. However attackers can also hook drivers in order to hide files, to hide sockets, to capture keystrokes, and for other nefarious purposes.

Each tab also contains the name of the Hooked Function if it can be determined, the name of the Hooked Module (or Driver) that was hooked, the name of the Hooking Module (or Driver) that is hooking the legitimate function, and the Hooking Address so that malware analyst can determine what the function hook does.

HookedFunction	HookedModule	HookingModule	HookingAddress
----------------	--------------	---------------	----------------

By right clicking on any item in any of the three lists, a user can acquire either the “Hooking Module” (the attacker’s module/driver) or the “Hooked Module” (usually the system module/driver).

For bugs, questions, comments, and feature requests e-mail peter.silberman@mandiant.com