



Cerbero Journal

"Good decisions come from experience. Experience comes from making bad decisions." – Mark Twain

ISSUE NR. 6

CERBERO LABS

AUGUST 5, 2025

We decided to skip the January issue and are even running late for the summer one. The reason is simple: we were still working on the features we wanted to showcase. To make up for the delay, we've included a summer crossword puzzle for a bit of seasonal fun.

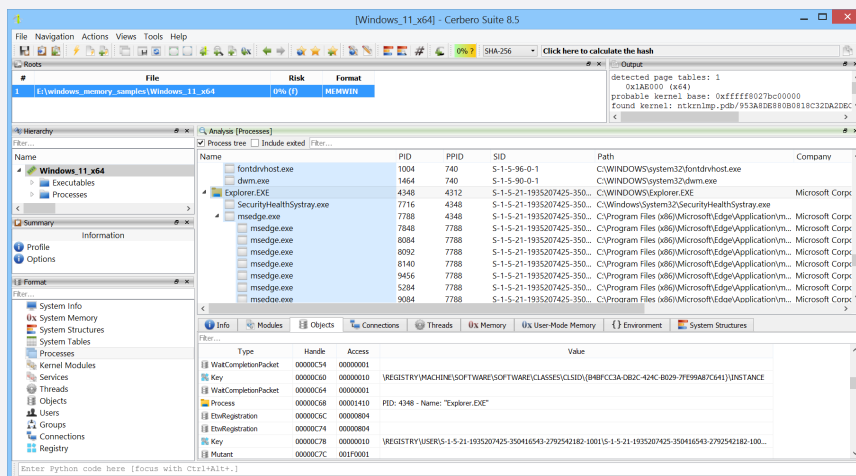
We wish you all a wonderful rest of the summer!

A STROLL DOWN MEMORY LANE

Since the last issue we have released version 8 of Cerbero Suite and, in fact, have already started working on version 9. The main new addition to our product line that we're going to cover in this issue is the **Memory Analysis package** that replaces the now deprecated Windows Memory Analysis package. While this is the main news, there is so much more that has been added.

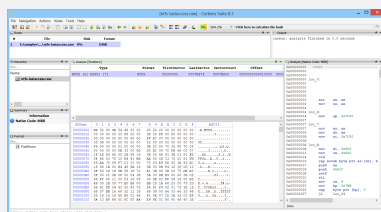
We now support **most file systems**, the **SDK documentation** is now complete, **most tables** can now be sorted and all of

them support filtering, the main window has been improved with the introduction of **customizable panels**, controls allow the **exporting** of their content in various formats and much more.



FILE SYSTEMS GALORE

Need inspect the contents of a file system? We've added support for quite a bit of them. [\[read more\]](#)



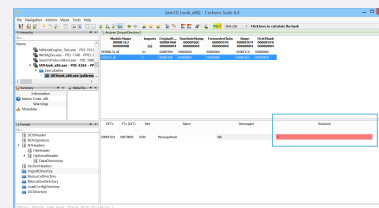
TABLES: SORT & FILTER

This has been a long time coming: most tables in Cerbero Suite can now be sorted and filtered! [\[read more\]](#)

Ordinal (n)	Function RVA	Name Ord	Name RVA	Name
142	003F7300	141	0069AB99	ExAllocateCacheAwareRun...
143	00006A54	142	0069ABBF	ExAllocatePool
144	0022D7E4	143	0069ABCE	ExAllocatePoolWithQuota
145	0005A580	144	0069ABE6	ExAllocatePoolWithQuotaTag
146	0028DBF0	145	0069AC01	ExAllocatePoolWithTag
147	000393E8	146	0069AC17	ExAllocatePoolWithTagPriority
148	00116940	147	0069AC35	ExAllocateTimer
149	000ADABC	148	0069AC45	ExBlockOnAddressPushLock
150	000ADB5C	149	0069AC5E	ExBlockPushLock
151	0022FC74	150	0069AC6E	ExCancelTimer
152	002A4008	151	0069AC7C	ExCompositionObjectType

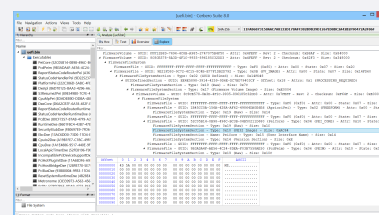
MISSING OFTS & IAT HOOKING

Portable Executables mapped in memory differ from their on-disk versions in several important ways. At the same time, having access to the full process address space provides valuable insight when analyzing a mapped executable. [\[read more\]](#)



UEFI FIRMWARE IMAGES

Support for a variety of UEFI firmware image formats. [\[read more\]](#)



CERBERO STORE

For those not yet familiar with it, Cerbero Store is a simple way to install and update optional packages for Cerbero Suite and Cerbero Engine.

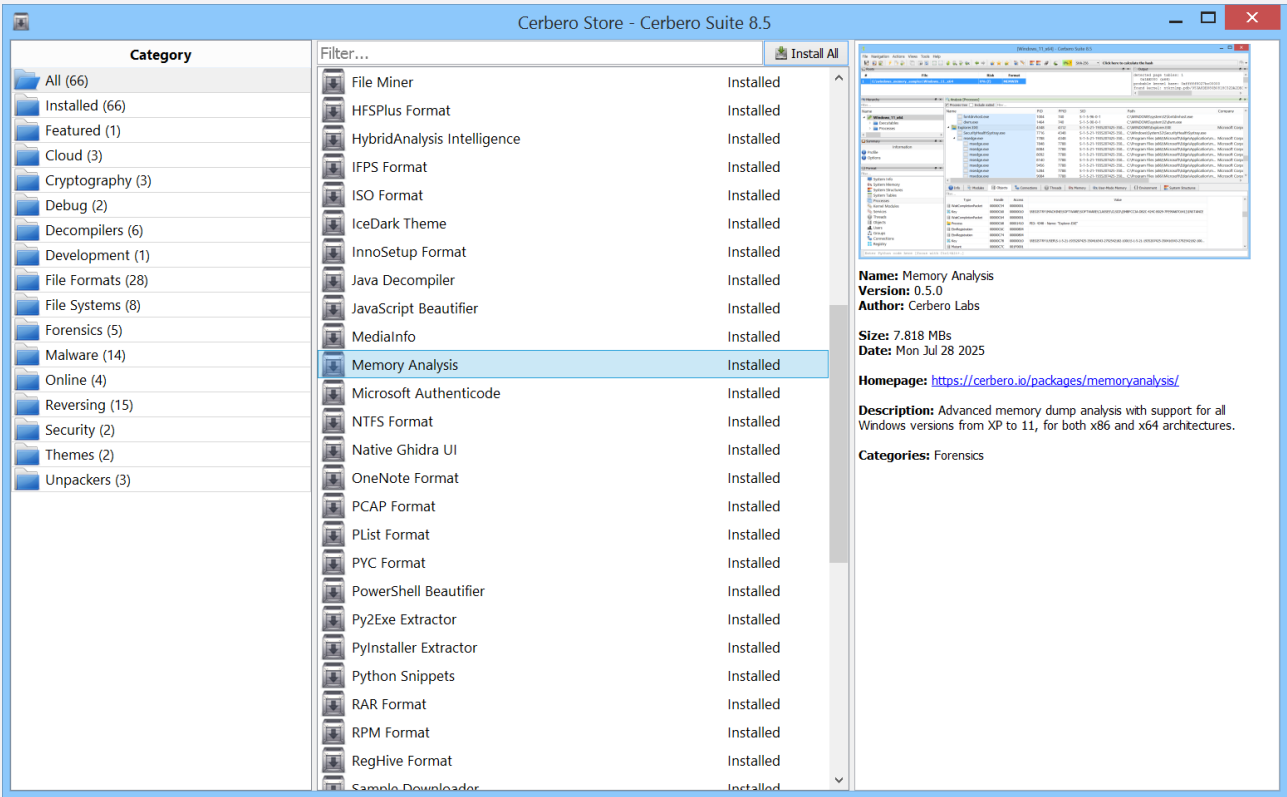
Updating specific parts of an application through Cerbero Store is notably more efficient than updating the entire application. This method also prevents users from having to install functionality they are not interested in.

Furthermore, our software operates across multiple platforms, meaning each update traditionally required the creation of multiple software packages. Cerbero Store addresses this issue effectively, as all platforms use the same package code, simplifying the update process and ensuring consistency across different operating systems.

In fact, our [memory analysis support](#) relies heavily on Cerbero Store to deliver timely updates in response to frequent changes in the operating system. This modular approach ensures that

INDEX	
CERBERO STORE	2
CERBERO SUITE 8	3
TABLES: SORT & FILTER	4
DATA EXPORT	4
MEMORY ANALYSIS	5
PAGING, PROTOTYPES & COMPRESSION	8
CHALLENGE: MEMORY DUMP	8
ENGINE INTERMEZZO	9
MISSING OFTS & IAT HOOKING	10
FILE SYSTEMS GALORE	11
UEFI FIRMWARE IMAGES	12
ONLINE INTELLIGENCE	12
SDK DOCUMENTATION	13
CROSSWORD PUZZLE	14

improvements and fixes can be deployed quickly, without waiting for full product updates.



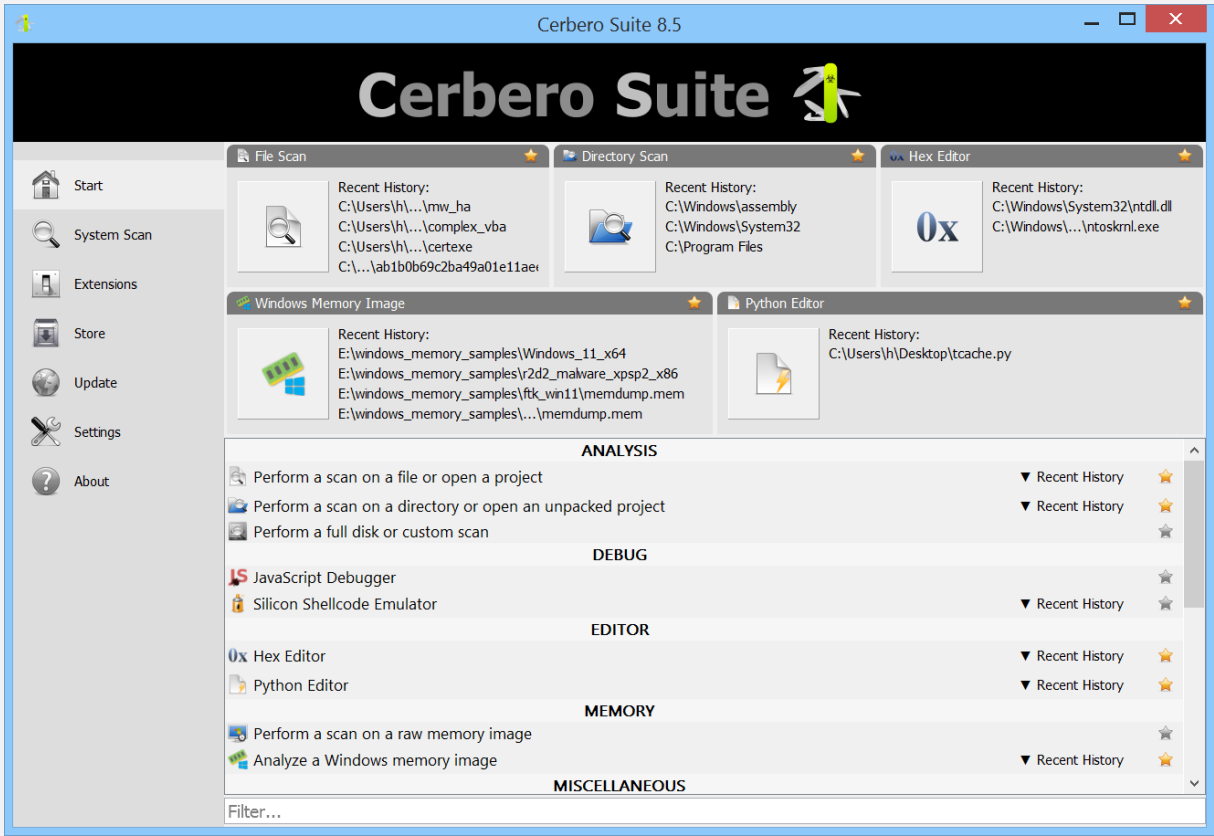
COMMERCIAL LICENSES

Holders of a personal license for Cerbero Suite have access to a wide range of packages on Cerbero Store. However, some packages are exclusively available to commercial license holders, and others may be released to commercial licenses before becoming available to personal ones. We make a deliberate effort to keep the number of commercially restricted packages to a minimum, focusing only on those we consider primarily suited for commercial use.

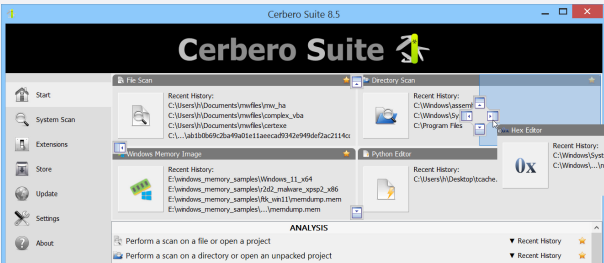
For professionals who rely on Cerbero Suite, commercial licenses offer the best way to unlock the platform’s full potential, including early access to advanced capabilities and exclusive packages.

CERBERO SUITE 8

Back in September 2024, we released Cerbero Suite 8. In this major update, we revamped the start page to enhance accessibility to the various logic providers and introduced customizable panels. Thanks to this change, you can select your preferred tools and position them prominently for quick access upon launching Cerbero Suite.



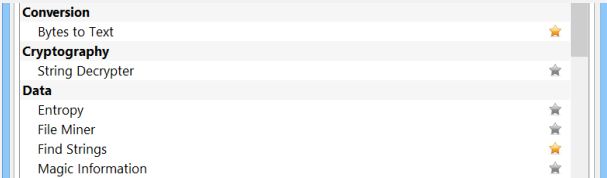
Adding a panel is straightforward: simply click on the star icon next to a logic provider, and a panel for the tool will appear. You can rearrange and resize these panels to tailor the layout to your specific needs.



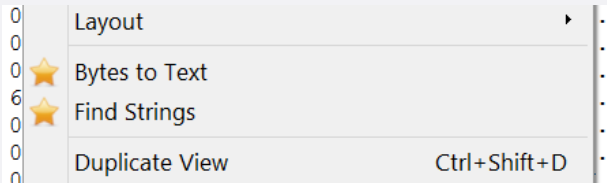
The panels not only provide quick access to tools but also, when available, offer functionalities such as viewing recent history, dragging and dropping files, and accessing tool settings.

With the continuous addition of new tools to Cerbero Suite, the updated start page has become an efficient and customizable launcher.

In alignment with the theme of personalization, you can also set favorite actions.



Designating an action as a favorite places it at the top level of the actions menu and the context menu within the various views.



This enhancement makes the execution of your most frequently used actions quicker and more intuitive.

Cerbero Suite 8 makes it easier to work the way you want. With a start page that lets you organize your tools and favorite actions just how you like them, getting things done feels quicker and more natural. This release and its minor versions are all focused on giving you a more flexible and efficient experience.

TABLES: SORT & FILTER

This has been a long time coming: most tables in Cerbero Suite can now be sorted. Whether you’re browsing through large datasets or looking for specific entries, sorting columns helps you quickly find what you need. If a plugin uses the default table control, sorting works automatically without any extra effort.

Ordinal (n)	Function RVA	Name Ord	Name RVA	Name
142	003F7300	141	0069AB99	ExAllocateCacheAwareRun...
143	000D6A54	142	0069ABBF	ExAllocatePool
144	0022D7E4	143	0069ABCE	ExAllocatePoolWithQuota
145	0005A580	144	0069ABE6	ExAllocatePoolWithQuotaTag
146	0028DBF0	145	0069AC01	ExAllocatePoolWithTag
147	000393E8	146	0069AC17	ExAllocatePoolWithTagPriority
148	00116940	147	0069AC35	ExAllocateTimer
149	000ADABC	148	0069AC45	ExBlockOnAddressPushLock
150	000ADB5C	149	0069AC5E	ExBlockPushLock
151	0022FC74	150	0069AC6E	ExCancelTimer
152	002A4008	151	0069AC7C	ExCompositionObjectType

Filtering works in much the same way. Even if a table doesn’t come with a visible filter, you can now bring one up and apply it instantly. Like sorting, this feature just works out of the box for anything using the default table control.

keb|

Ordinal (n)	Function RVA	Name Ord	Name RVA	Name
	0061D03C 00000004	00621CA8 00000002	0061F674 00000004	00622FF7 0000001F
871	00140490	870	0069F21B	KeBugCheck
872	001404C0	871	0069F226	KeBugCheckEx

Combined with sorting, filters make it far easier to focus on the data that matters most, especially when dealing with long lists or complex results.

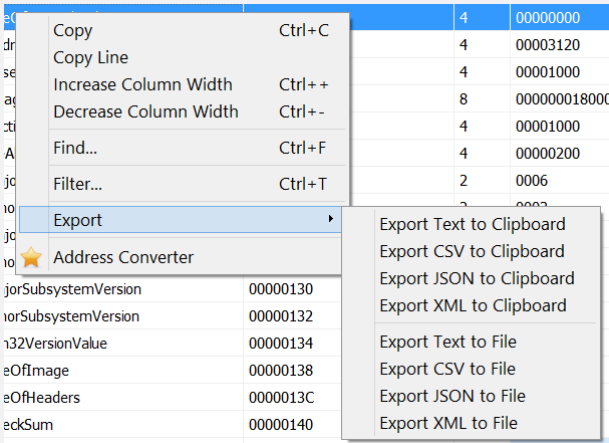
DATA EXPORT

Whether you’re performing an in-depth analysis or just need to extract specific pieces of information, having the ability to export data is a practical necessity. Tables and tree controls often hold valuable details (e.g., lists, search results, parsed structures) and being able to pull that information out in a usable format can save time and simplify your workflow.

This feature comes in handy in many real-world scenarios: maybe you’re preparing a report for a colleague, scripting further analysis in a different tool, or just want to keep a snapshot of your findings. Instead of manually copying data or writing custom scripts, you can now export directly with just a couple of clicks.

Supported export formats include plain text, CSV (for tables), JSON, and XML, covering a variety of use cases from quick notes to structured automation. It’s a small feature with a big

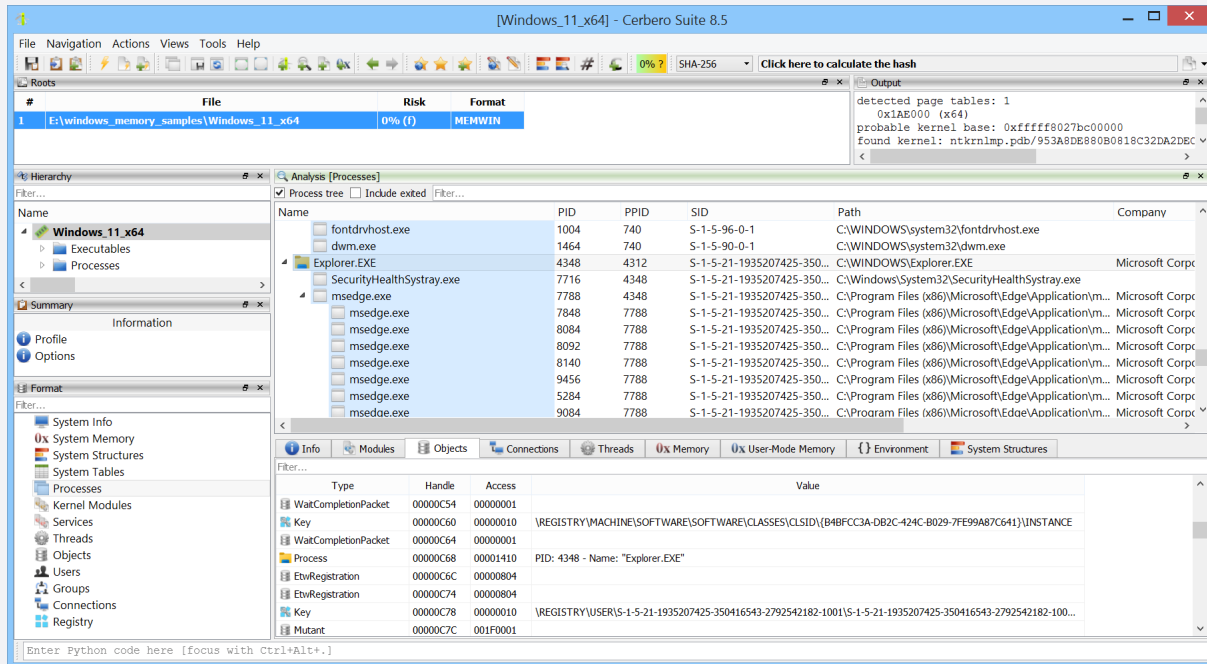
impact. Another long-requested addition that’s finally made its way into Cerbero Suite.



MEMORY ANALYSIS

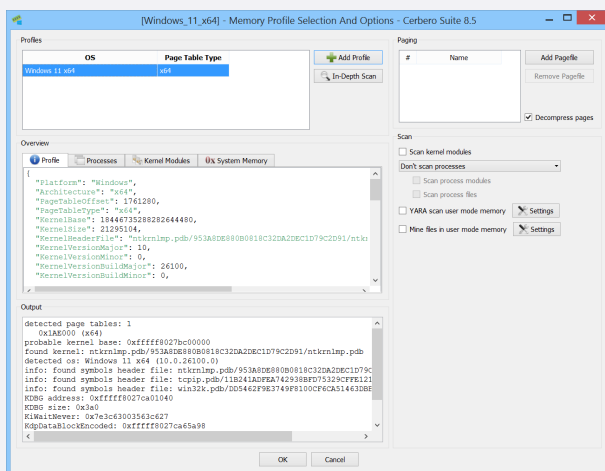
It's time to introduce the biggest feature we've been working on over the past few months. We've deprecated our old Windows Memory Analysis package and completely rewritten it, resulting in the new [Memory Analysis package](#), currently still in beta.

This package is designed to push the limits of memory dump analysis and make the process more accessible. It supports all Windows versions from XP to 11, for both x86 and x64 architectures.

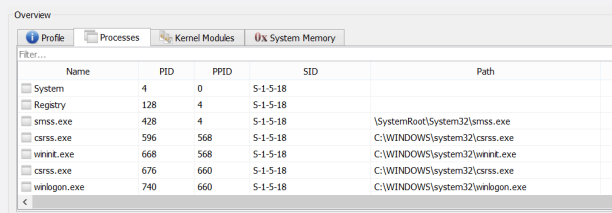


By default, the package supports parsing raw memory dumps, but full kernel crash dumps are also supported by installing the [Windows Crash Dump Format](#) package.

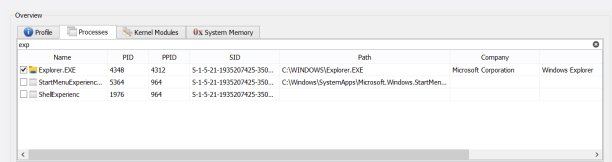
When opening a memory dump, an initialization dialog is displayed, allowing you to select the appropriate profile for the dump, as well as configure paging support and scanning options.



The dialog provides a preview to help confirm the correctness of the selected memory profile.



In addition to skipping the scanning of processes in a memory dump or scanning all of them, you can choose to scan only specific processes of interest, making your analysis faster and more focused.

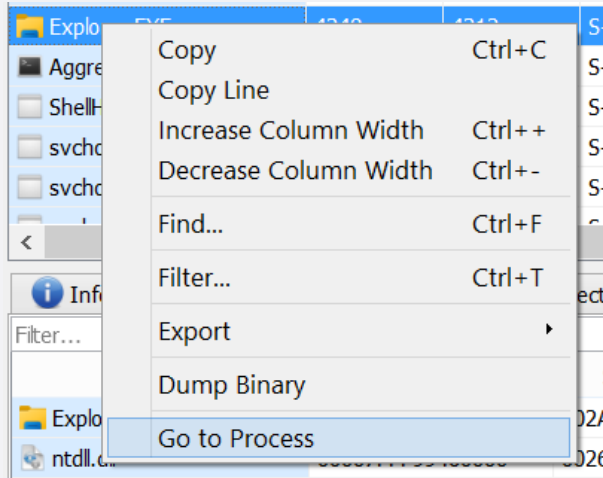


After specifying the profile and options, the memory dump can be inspected in the analysis workspace. Every list view supports filtering for quick access to relevant items.

...continued from page 5.

Name	PID	PPID	SID	Path
wininit.exe	668	568	S-1-5-18	C:\WINDOWS\system32\wininit.exe
services.exe	812	668	S-1-5-18	C:\WINDOWS\system32\services.exe
svchost.exe	964	812	S-1-5-18	C:\WINDOWS\system32\svchost.exe
StartMenuExperienceHost.exe	5364	964	S-1-5-21-1935207425-350...	C:\Windows\SystemApps\Microsoft.Windows.Start...
ShellExperience	1976	964	S-1-5-21-1935207425-350...	
Explorer.exe	4384	4312	S-1-5-21-1935207425-350...	C:\WINDOWS\Explorer.exe

Processes and modules are hyperlinked, allowing you to jump directly to a process or module analysis from any view. When opening a memory dump, you can choose to skip scanning processes and modules for faster inspection, yet still jump directly to a specific module and inspect it.



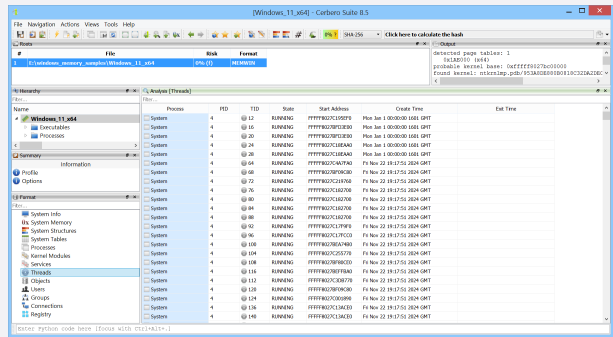
Loaded kernel modules can be examined.

[illegible]

Registered services are enumerated.

[illegible]

Threads from all processes are available.



Referenced objects from all processes can be inspected.

Active network connections can be reviewed.

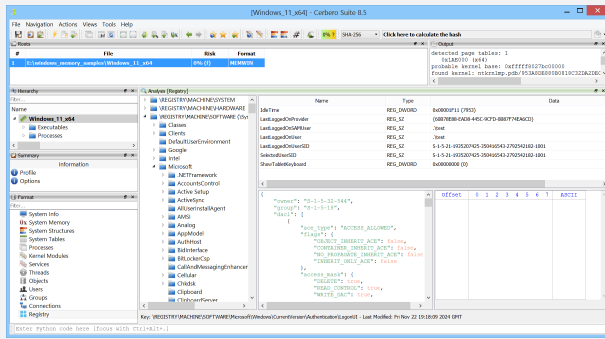
The screenshot displays the Windows Task Manager application. The main window is titled "[Windows 11 x64] - Carbon Sub 8.3". The "Processes" tab is selected, showing a list of running processes. The list includes columns for Name, Process ID, PID, Parent PID, Local Address, Remote Address, Status, and Creation Time. The processes listed include System Idle Process, System, smss.exe, csrss.exe, explorer.exe, and various system services. The "Task Manager" window on the right shows a list of tasks with columns for Name, Status, and Creation Time. The tasks listed include System Idle Process, System, smss.exe, csrss.exe, explorer.exe, and various system services. The "Task Manager" window on the right also shows a "Performance" tab on the left and a "Task Manager" window on the right. The "Task Manager" window on the right shows a list of tasks with columns for Name, Status, and Creation Time.

System users and groups, along with their properties, can be examined.

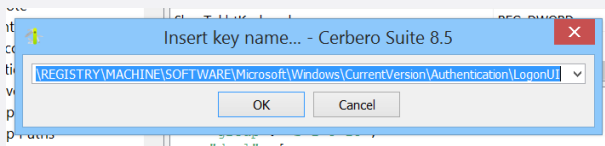
The screenshot shows the Windows Task Manager application. The 'Processes' tab is active, displaying a list of running processes. The 'System Idle Process' is selected. The 'Performance' tab is also visible, showing system metrics like CPU, Memory, Disk, and Network.

Registry hives loaded in memory are displayed in a familiar interface.

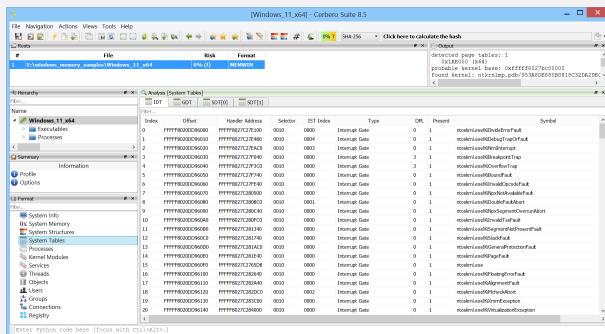
... continued from page 6.



It is also possible to jump directly to specific registry keys.



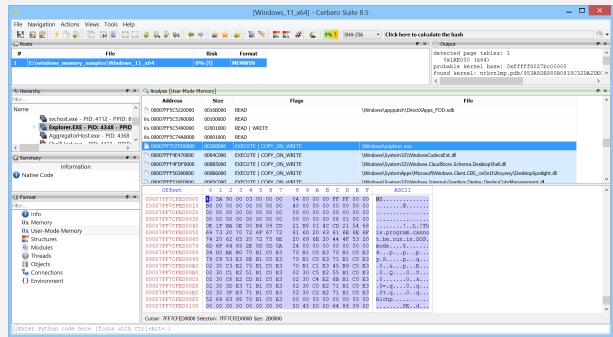
Architecture-specific tables such as the Interrupt Descriptor Table are supported.



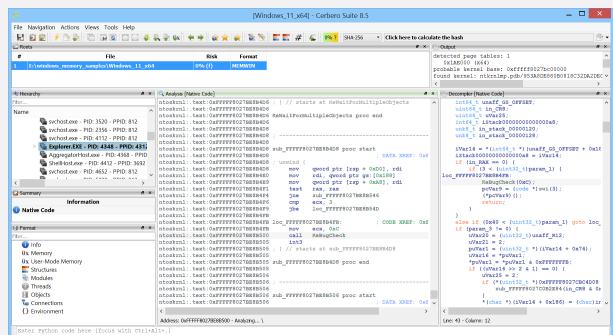
The Windows Service Descriptor Table can also be inspected.

Index	Offset	Handler Address	
0	FFFFF8027BCD4EF0	FFFFF8027BF2EE40	ntoskrnl.exe!NtAccessCheck
1	FFFFF8027BCD4EF4	FFFFF8027C03CA40	ntoskrnl.exe!NtWorkerFactoryWorkerReady
2	FFFFF8027BCD4EF8	FFFFF8027C5C98F0	ntoskrnl.exe!NtAcceptConnectPort
3	FFFFF8027BCD4EFC	FFFFF8027C3D5610	ntoskrnl.exe!NtMapUserPhysicalPagesScatter
4	FFFFF8027BCD4F00	FFFFF8027C45B170	ntoskrnl.exe!NtWaitForSingleObject
5	FFFFF8027BCD4F04	FFFFF8027C273E20	ntoskrnl.exe!NtCallbackReturn
6	FFFFF8027BCD4F08	FFFFF8027C45A790	ntoskrnl.exe!NtReadFile
7	FFFFF8027BCD4F0C	FFFFF8027C488A50	ntoskrnl.exe!NtDeviceIoControlFile
8	FFFFF8027BCD4F10	FFFFF8027C4882C0	ntoskrnl.exe!NtWriteFile
9	FFFFF8027BCD4F14	FFFFF8027C568680	ntoskrnl.exe!NtRemoveIoCompletion
10	FFFFF8027BCD4F18	FFFFF8027C58F990	ntoskrnl.exe!NtReleaseSemaphore
11	FFFFF8027BCD4F1C	FFFFF8027C49C160	ntoskrnl.exe!NtReplyWaitReceivePort
12	FFFFF8027BCD4F20	FFFFF8027C5C2920	ntoskrnl.exe!NtReplyPort
13	FFFFF8027BCD4F24	FFFFF8027C454E40	ntoskrnl.exe!NtSetInformationThread
14	FFFFF8027BCD4F28	FFFFF8027C566D50	ntoskrnl.exe!NtSetEvent
15	FFFFF8027BCD4F2C	FFFFF8027C45AE10	ntoskrnl.exe!NtClose
16	FFFFF8027BCD4F30	FFFFF8027C484C0	ntoskrnl.exe!NtQueryObject
17	FFFFF8027BCD4F34	FFFFF8027C5420A0	ntoskrnl.exe!NtQueryInformationFile
18	FFFFF8027BCD4F38	FFFFF8027C41DC0	ntoskrnl.exe!NtOpenKey
19	FFFFF8027BCD4F3C	FFFFF8027C42CB30	ntoskrnl.exe!NtEnumerateValueKey
20	FFFFF8027BCD4F40	FFFFF8027C4F1710	ntoskrnl.exe!NtFindAtom
21	FFFFF8027BCD4F44	FFFFF8027C4D96B0	ntoskrnl.exe!NtQueryDefaultLocale

Each process can be individually inspected as a child object.



The complete address space of a process can be analyzed using the Carbon disassembler.



Modules and files are scanned using YARA when the **YARA Rules package** is installed. Additionally, the user-mode memory of processes can be scanned using YARA. It can also be mined for files using our advanced **File Miner package**.

Scan

☒ Scan kernel modules

☒ Scan all processes

☒ Scan process modules

☒ Scan process files

☒ YARA scan user mode memory

Settings

☒ Mine files in user mode memory

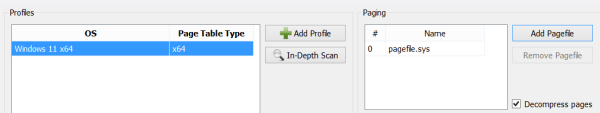
Settings

The new Memory Analysis package introduces a modern and integrated way to investigate memory dumps. From full system overviews to detailed inspection of individual processes, it offers both flexibility and precision. Whether you are performing malware triage, searching for forensic artifacts, or analyzing complex behavior, this package delivers the depth and control needed for serious memory analysis, all within the familiar environment of Cerbero Suite. This is only the beginning, as development is ongoing and new features are already planned.

PAGING, PROTOTYPES & COMPRESSION

To deliver high-quality memory analysis, it is important to resolve memory pages beyond what the standard page table provides.

One such case involves pagefiles, which can be configured through the initialization dialog. Windows theoretically supports up to 16 pagefiles. When adding one, you can assign it an index, with 0 being the default.



This makes it possible to resolve memory that has been paged out to disk.

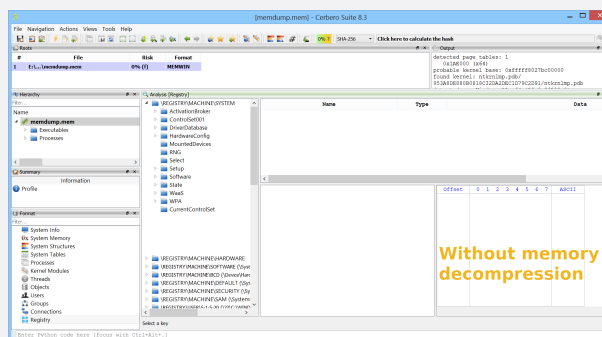
Another Windows-specific artifact related to memory paging is the use of prototype PTEs. These entries are a key component of Windows memory management and represent shared memory pages, often mapped from image files or shared sections. Unlike regular page table entries, prototype PTEs reside in separate memory regions and require dedicated handling to be correctly interpreted during analysis.

Support for prototype PTEs in the Memory Analysis package improves the resolution of virtual memory mappings, particularly in cases involving shared memory or image-backed memory. This significantly enhances the ability to reconstruct executable images, DLLs, and other shared artifacts across processes.

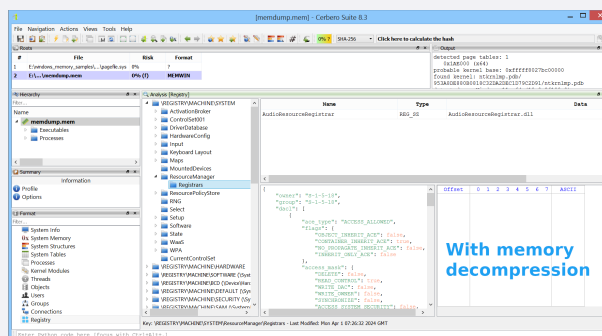
Perhaps the most complex feature to support has been memory decompression. Introduced in Windows 10 version 1507, memory compression allows certain memory pages to be compressed and managed by the 'MemCompression' process. As a result, some pages in memory snapshots may appear unavailable, since they reside in compressed space. Although

memory compression can be disabled if desired, it is enabled by default.

In the example below, registry keys appear to be missing when analyzing a memory snapshot. These keys reside in memory pages that were compressed at the time the snapshot was taken.



After enabling memory decompression, the missing keys are successfully recovered.

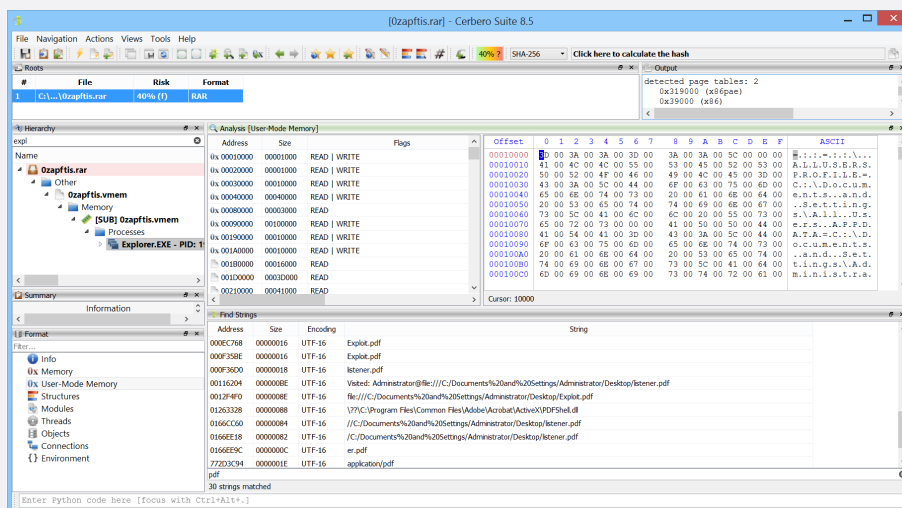


Credit for the [original research](#) into the undocumented Windows 10 memory compression mechanism goes to the team at FireEye, now Mandiant.

CHALLENGE: MEMORY DUMP

This challenge is simple and ideal for beginners. It does not require writing any code and can be completed entirely using the Cerbero Suite user interface. The memory dump was found online and can be downloaded from [this link](#).

The goal is to identify the malware binaries present in memory, along with any other relevant artifacts. Once you have completed the challenge, you can compare your results with this detailed [write-up](#).



ENGINE INTERMEZZO



In case you're not yet familiar with Cerbero Engine, here is a quick introduction. You can read more on our [web page](#).

WHAT IS CERBERO ENGINE?

Cerbero Engine is our solution for enterprise projects such as cloud or in-house services. It offers the same SDK as Cerbero Suite and has already been used to analyze billions of files.

WHAT CAN IT DO?

The SDK is extensive and features support for dozens of file formats, scanning, disassembly, decompiling, emulation, signature matching, file carving, decompression, decryption and much more.

We make sure Cerbero Engine keeps up with the latest threats and challenges presented by file formats which are difficult to analyze. We offer state-of-the-art support for various file types such as Adobe PDF and Microsoft Office.

HOW SECURE IS IT?

Cerbero Engine has been designed taking into account all types of security issues when analyzing malicious files: buffer overflows, integer overflows, infinite loops, infinite recursion, decompression bombs, denial-of-service etc.

WHAT PLATFORMS DOES IT SUPPORT?

Just like Cerbero Suite, Cerbero Engine is cross-platform. Currently we offer it for both Windows (x86, x64) and Linux (x64). It is also compatible with older versions of Windows and Linux.

CAN IT BE EMBEDDED?

Cerbero Engine is deployed as an embeddable module: a Dynamic-Link Library (DLL) on Windows and a Shared Library on Linux. The engine can be loaded from both C/C++ and Python 3.

Loading the engine from Python is extremely simple.

```
from ProEngine import *

# initialize the engine
proEngineInit()

# from here on the SDK can be accessed
from Pro.Core import *
# ...

# finalize the engine before exiting
proEngineFinal()
```

Loading the engine from C/C++ is also very simple: it only requires including the 'ProEngine' header and specifying the location of the engine on disk.

```
#define PRO_ENGINE_INIT
#include "ProEngine.h"

int main()
{
    // initialize the engine
    if (!proEngineInit("/path/to/the/
    ↪ engine", ProEngine_InitPython))
        return -1;

    // from here on the SDK can be
    ↪ accessed

    // finalize the engine before exiting
    proEngineFinal();
    return 0;
}
```

IS IT FAST?

While the SDK is in Python, our engine is written in C++ and is both multi-thread and multi-process. This design decision guarantees maximum speed, while also giving you the capability to write cross-platform code that is compatible across both Cerbero Engine and Cerbero Suite.

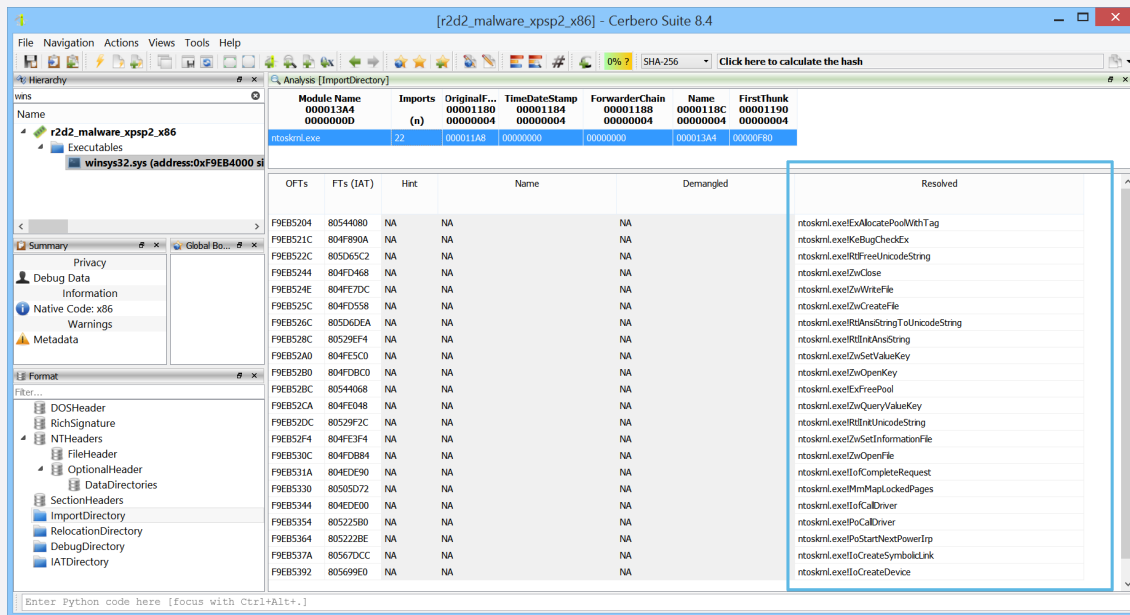
Since the SDK is in Python, you don't need to worry about rebuilding your project when the engine is updated. Moreover, we take great care not to introduce breaking changes to the SDK: we don't want you to worry that an update could cause your code to stop working!

HOW DO YOU LICENSE IT?

We license Cerbero Engine on a per-case basis. Licensing depends on the project's scope. If you are interested in a quotation, please [contact us](#).

Purchasing a license of Cerbero Engine comes with discounted lab licenses for Cerbero Suite. By using Cerbero Suite, your engineers can interactively debug parsing issues, analyze edge cases, use the Python editor for development and create graphical applications that work in conjunction with Cerbero Engine.

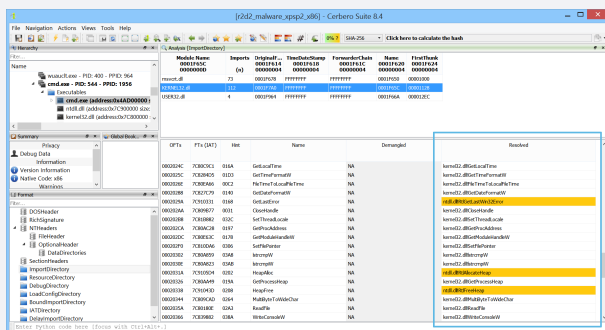
MISSING OFTS & IAT HOOKING



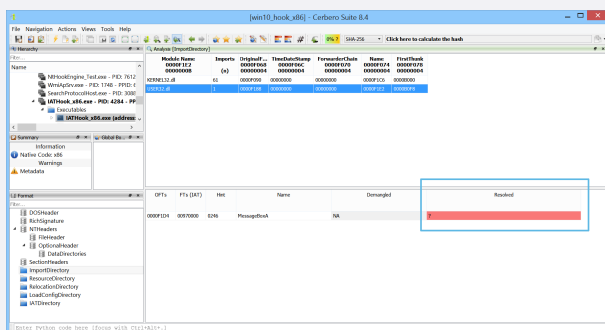
Portable Executables mapped in memory differ from their on-disk versions in several important ways. At the same time, having access to the full process address space provides valuable insight when analyzing a mapped executable.

For example, viewing the import table is helpful, but seeing where the IAT entries actually point to is even more informative, especially when the original import information is missing, as shown in the image above.

Forwarded entries are highlighted for clarity.



Hooked IAT entries are even easier to spot.



When the original and first thunk arrays are identical and import names cannot be recovered, the resulting disassembly can be difficult to interpret.

```

Analysis [Native Code]
winys32: Text: 0xF9EB4A5 int3
winys32: Text: 0xF9EB4A6 sub_F9EB4A6 proc start
winys32: Text: 0xF9EB4A6 push ebp
winys32: Text: 0xF9EB4A7 mov ebp, esp
winys32: Text: 0xF9EB4A8 sub esp, 0x10
winys32: Text: 0xF9EB4A9 push esi
winys32: Text: 0xF9EB4AA mov esi, dword ptr [0xF9EB4F80]
winys32: Text: 0xF9EB4AB push 0xF9EB5044
winys32: Text: 0xF9EB4AC lea eax, [ebp - 8]
winys32: Text: 0xF9EB4AD push eax
winys32: Text: 0xF9EB4AE call esi
winys32: Text: 0xF9EB4AF push 0xF9EB50D4
winys32: Text: 0xF9EB4B0 lea eax, [ebp - 0x10]
winys32: Text: 0xF9EB4B1 push eax
winys32: Text: 0xF9EB4B2 call esi
winys32: Text: 0xF9EB4B3 push 0xF9EB5168
winys32: Text: 0xF9EB4B4 xor esi, esi
winys32: Text: 0xF9EB4B5 push esi
winys32: Text: 0xF9EB4B6 push 0xF9EB50D4
winys32: Text: 0xF9EB4B7 lea eax, [ebp - 8]
winys32: Text: 0xF9EB4B8 push eax
winys32: Text: 0xF9EB4B9 push 0xF9EB50D4
winys32: Text: 0xF9EB4BA call dword ptr [0xF9EB4F80]
winys32: Text: 0xF9EB4BB cmp eax, esi
winys32: Text: 0xF9EB4BC pop esi
winys32: Text: 0xF9EB4BD j1 loc_F9EB4E5F
winys32: Text: 0xF9EB4BE lea eax, [ebp - 8]
winys32: Text: 0xF9EB4BF push eax
winys32: Text: 0xF9EB4C0 push 0xF9EB50D4
winys32: Text: 0xF9EB4C1 lea eax, [ebp - 0x10]
winys32: Text: 0xF9EB4C2 push eax
winys32: Text: 0xF9EB4C3 call dword ptr [0xF9EB4F80]
winys32: Text: 0xF9EB4C4 leave
winys32: Text: 0xF9EB4C5

```

By dynamically resolving IAT entries, the disassembler is able to recover imported function names, making the disassembly much easier to read and understand.

```

Text: 0xF9EB4A5 sub_F9EB4A6 proc start
Text: 0xF9EB4A6 push ebp
Text: 0xF9EB4A7 mov ebp, esp
Text: 0xF9EB4A8 sub esp, 0x10
Text: 0xF9EB4A9 push esi
Text: 0xF9EB4AA mov esi, dword ptr [0xF9EB4F80] -> RtlInitUnicodeString
Text: 0xF9EB4AB push 0xF9EB5044
Text: 0xF9EB4AC lea eax, [ebp - 8]
Text: 0xF9EB4AD push eax
Text: 0xF9EB4AE call esi
Text: 0xF9EB4AF push 0xF9EB50D4
Text: 0xF9EB4B0 lea eax, [ebp - 0x10]
Text: 0xF9EB4B1 push eax
Text: 0xF9EB4B2 call esi
Text: 0xF9EB4B3 push 0xF9EB5168
Text: 0xF9EB4B4 xor esi, esi
Text: 0xF9EB4B5 push esi
Text: 0xF9EB4B6 push 0xF9EB50D4
Text: 0xF9EB4B7 lea eax, [ebp - 8]
Text: 0xF9EB4B8 push eax
Text: 0xF9EB4B9 push 0xF9EB50D4
Text: 0xF9EB4BA call dword ptr [0xF9EB4F80] -> IoCreateDevice
Text: 0xF9EB4BB cmp eax, esi
Text: 0xF9EB4BC pop esi
Text: 0xF9EB4BD j1 loc_F9EB4E5F
Text: 0xF9EB4BE lea eax, [ebp - 8]
Text: 0xF9EB4BF push eax
Text: 0xF9EB4C0 push 0xF9EB50D4
Text: 0xF9EB4C1 lea eax, [ebp - 0x10]
Text: 0xF9EB4C2 push eax
Text: 0xF9EB4C3 call dword ptr [0xF9EB4F80] -> IoCreateSymbolicLink
Text: 0xF9EB4C4 leave
Text: 0xF9EB4C5

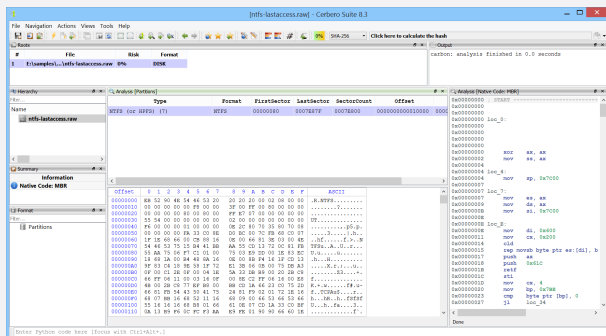
```

FILE SYSTEMS GALORE

Over the years, Cerbero Suite has steadily expanded its support for archive formats. However, support for actual file systems, beyond the initial [ISO Format package](#), was missing.

That changed in recent months with the release of a series of packages introducing comprehensive file system support.

The first step toward handling file systems properly was the introduction of the [Disk Format package](#), which added capabilities for parsing and analyzing disk layouts, including MBR and GPT partition tables.



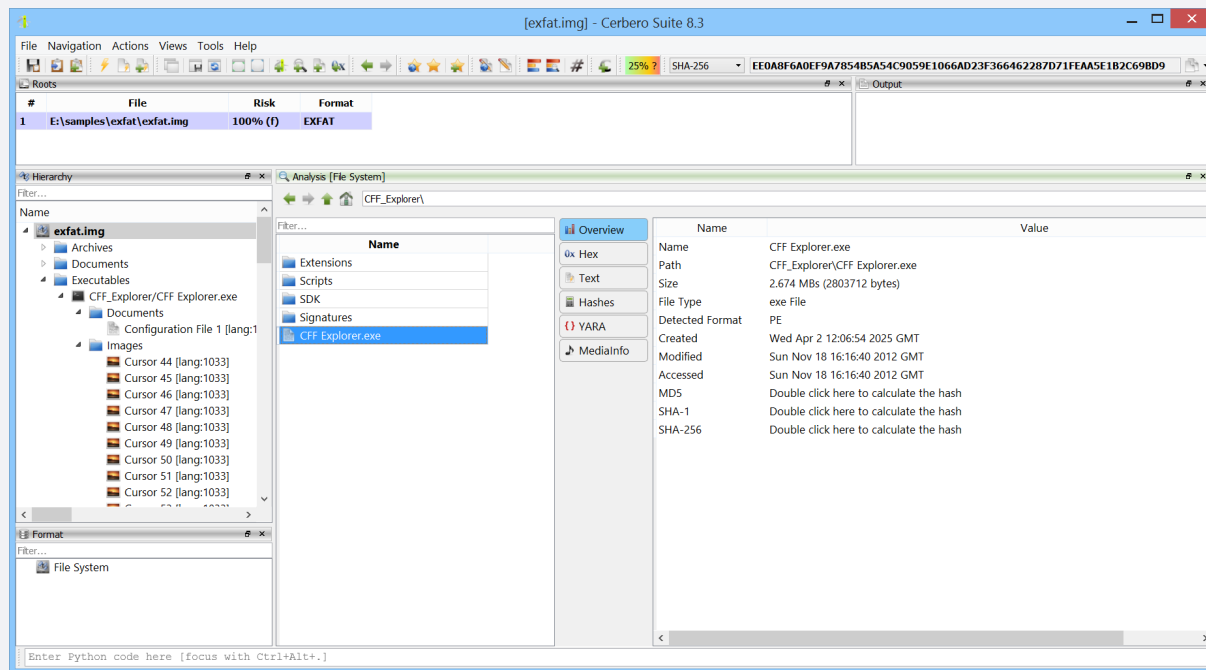
If a partition contains a supported file system, it will

automatically appear as a child object. The package also enables in-depth analysis of MBR boot code using the Carbon disassembler.

Following that, we released support for several major file systems:

- **FAT12/16/32** – Common in disk images, firmware, and USB devices.
- **exFAT** – Often seen in memory cards and modern storage media.
- **NTFS** – Full support for Microsoft's journaling file system.
- **EXT2/3/4** – Widely used in Linux systems and embedded environments.
- **HFS+** – Apple's legacy file system, useful when dealing with older macOS volumes.
- **WIM** – Used in Windows Imaging Format, commonly found in system deployment archives.

All supported file systems are treated like first-class containers in the analysis workspace. You can browse directories, open and analyze files, and view metadata for both files and folders. Everything behaves exactly as you'd expect.



CERBERO LABS

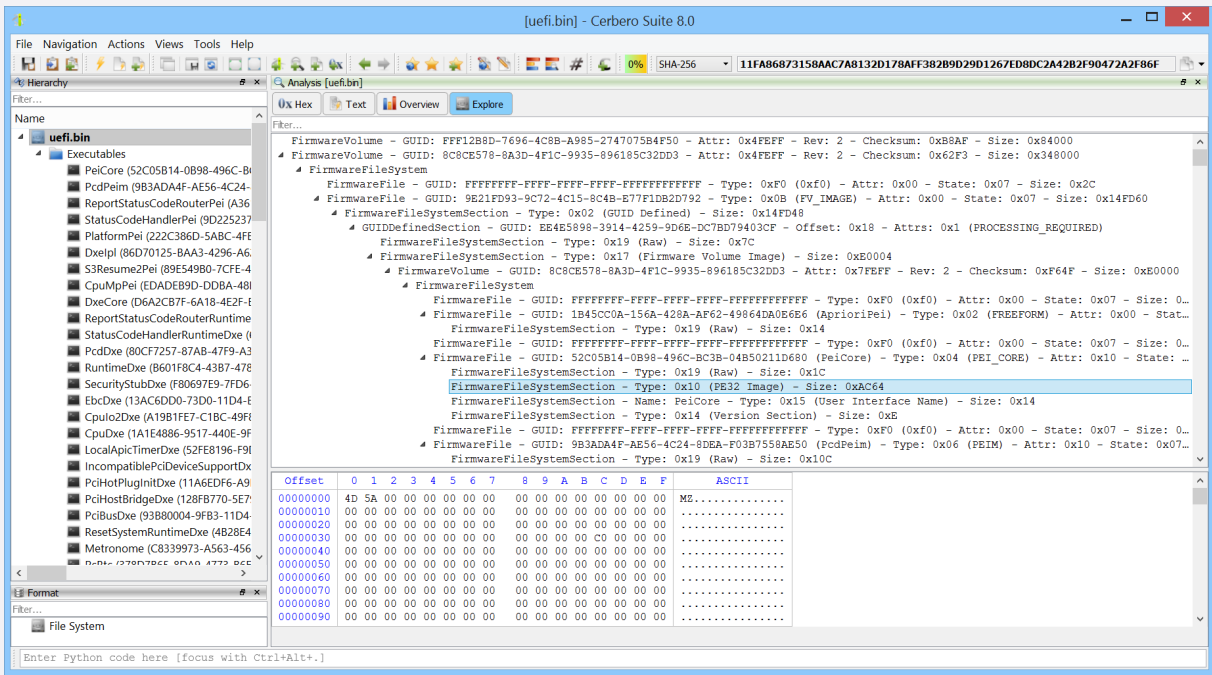


If you have any questions, feel free to contact us at: info@cerbero.io

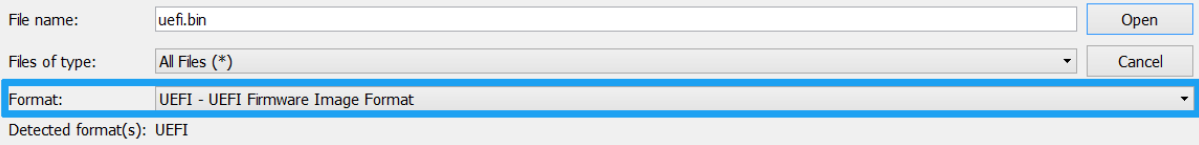
You can follow us on [X](#) and [LinkedIn](#) for the latest updates, or better yet, subscribe to our [newsletter](#) so you don't miss any major news!

UEFI FIRMWARE IMAGES

The [UEFI Firmware Image Format](#) package supports a variety of UEFI firmware image formats and, in addition to allowing you to inspect their structure, it automatically extracts embedded files.



The package usually identifies the UEFI firmware image format automatically. However, if the format is not automatically recognized, you will need to manually specify it when opening the file.

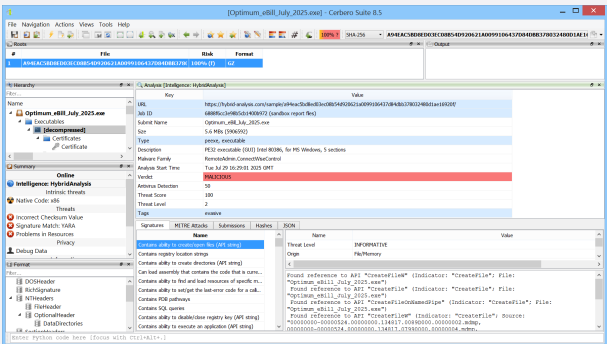


Inspecting firmware images is essential for tasks like vulnerability research, reverse engineering, and digital forensics. UEFI firmware often contains drivers, executables, configuration data, and other critical components that can reveal valuable insights about a system’s behavior, security posture, or manufacturer-specific implementations. With this package, Cerbero Suite makes the analysis of complex firmware images more accessible and efficient.

ONLINE INTELLIGENCE

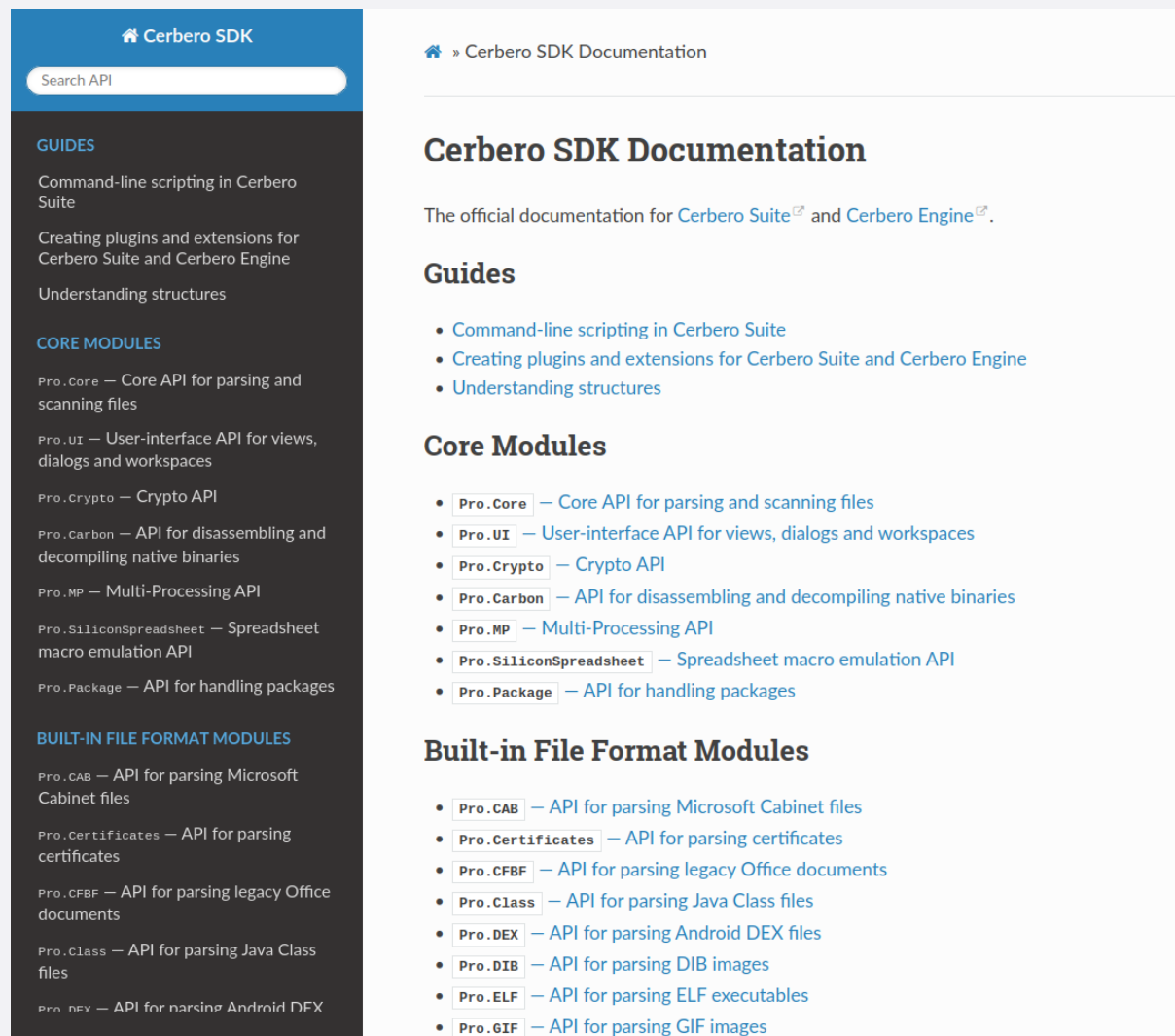
We’ve updated our online intelligence providers to keep up with recent API changes across various services. The [AbuseCH Intelligence](#), [HybridAnalysis Intelligence](#), and [Sample Downloader](#) packages have all been refreshed to ensure continued compatibility and reliability.

These tools are especially useful to researchers looking to gather threat intelligence and access the latest malware samples along with their related artifacts. Whether you’re tracking indicators of compromise or analyzing payloads in context, these integrations help bring external intelligence directly into your workflow.



SDK DOCUMENTATION

We have completed the documentation of the [SDK](#), including built-in file formats, installable packages, and external modules.



Cerbero SDK

Search API

GUIDES

- Command-line scripting in Cerbero Suite
- Creating plugins and extensions for Cerbero Suite and Cerbero Engine
- Understanding structures

CORE MODULES

- Pro.Core** — Core API for parsing and scanning files
- Pro.UI** — User-interface API for views, dialogs and workspaces
- Pro.Crypto** — Crypto API
- Pro.Carbon** — API for disassembling and decompiling native binaries
- Pro.MP** — Multi-Processing API
- Pro.SiliconSpreadsheet** — Spreadsheet macro emulation API
- Pro.Package** — API for handling packages

BUILT-IN FILE FORMAT MODULES

- Pro.CAB** — API for parsing Microsoft Cabinet files
- Pro.Certificates** — API for parsing certificates
- Pro.CFBF** — API for parsing legacy Office documents
- Pro.Class** — API for parsing Java Class files
- Pro.DEX** — API for parsing Android DEX files

Cerbero SDK Documentation

The official documentation for [Cerbero Suite](#) and [Cerbero Engine](#).

Guides

- [Command-line scripting in Cerbero Suite](#)
- [Creating plugins and extensions for Cerbero Suite and Cerbero Engine](#)
- [Understanding structures](#)

Core Modules

- Pro.Core** — Core API for parsing and scanning files
- Pro.UI** — User-interface API for views, dialogs and workspaces
- Pro.Crypto** — Crypto API
- Pro.Carbon** — API for disassembling and decompiling native binaries
- Pro.MP** — Multi-Processing API
- Pro.SiliconSpreadsheet** — Spreadsheet macro emulation API
- Pro.Package** — API for handling packages

Built-in File Format Modules

- Pro.CAB** — API for parsing Microsoft Cabinet files
- Pro.Certificates** — API for parsing certificates
- Pro.CFBF** — API for parsing legacy Office documents
- Pro.Class** — API for parsing Java Class files
- Pro.DEX** — API for parsing Android DEX files
- Pro.DIB** — API for parsing DIB images
- Pro.ELF** — API for parsing ELF executables
- Pro.GIF** — API for parsing GIF images

This means that the entire SDK is now available for auto-completion in the Python editor. Every time you install a package that is exposed to the SDK, auto-completion becomes available for that package as well.

The SDK of Cerbero Suite and Cerbero Engine is unparalleled in scope and depth, offering a vast array of functionalities for developers. With the documentation, you can easily explore all the capabilities the SDK has to offer. Whether you're writing plugins to automate tasks, analyzing complex file formats, or creating new tools, the SDK provides the necessary tools and resources.

The integrated Python editor in Cerbero Suite further enhances your development experience by providing features like syntax highlighting, hints and code completion. To get started, simply navigate to the [SDK documentation website](#), where you'll find extensive guides, API references, and examples.

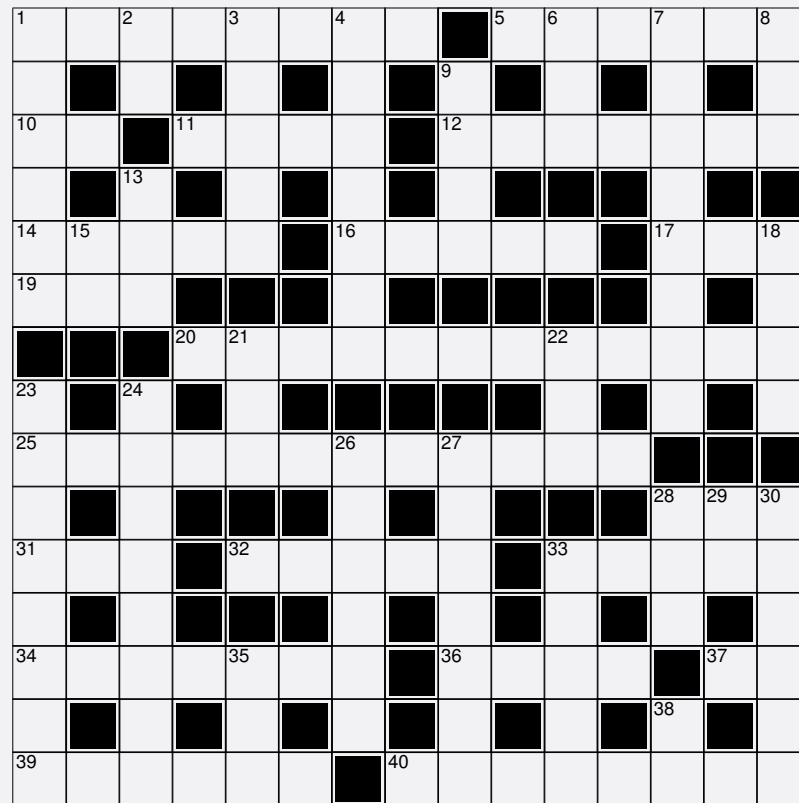
The documentation is organized to help both beginners and advanced users quickly find the information they need.

```

4 def customViewCallback(cv: ProCustomView, ud, code,
5     if code == pvnInit:
6         t = cv.getView(1)
7         t.
8         addAction
9         addDependentView
10        addMenu
11        clear
12        close
13        closeDependentView
14        contextualHeaderDescription
15        vid = view.id()
16        if vid == 1:
17            data.setText(0, str(data.row))
18        elif code == pvnRowSelected:
19            vid = view.id()
20            if vid == 1:
21                e = cv.getView(2)
22                e.setText(str(data.row))
23        return 0
24

```

CROSSWORD PUZZLE



ACROSS: 1. Power state where the PC looks "asleep" but keeps services running (Windows). 5. Box that forwards packets between networks (layer-3). 10. Boolean operator that returns true if either operand is true. 11. Oracle's clustered file system (abbr.). 12. Switch-case branch that runs when no case matches. 14. Entries in a list or menu. 16. Precedes a C++ destructor name. 17. Lightweight C compiler by Fabrice Bellard (abbr.). 19. Windows console command to clear the screen. 20. Science of securing information against adversaries. 25. Firewall that inspects packet headers to decide pass or drop. 28. Automatic Image Annotation (computer algorithm). 31. Slang for fully compromise a target machine. 32. Common string function that divides text by a delimiter. 33. Self-replicating malware. 34. CIDR block; contiguous set of addresses. 36. Google smart-home brand often seen in IoT forensics. 37. What users click and see (abbr.). 39. Variable scope visible across the whole module/namespace. 40. Keeper of servers, services, and uptime.

DOWN: 1. Operation that completes entirely or not at all. 2. Field of machine-driven pattern recognition and generation (abbr.). 3. Apple's desktop operating system. 4. Primary workspace with icons, windows, and taskbar/dock. 6. Open document file standard (abbr.). 7. Common font format for .ttf files. 8. Remote Desktop tool (abbr.). 9. Asymmetric DSL over copper loops (abbr.). 13. Widely used block cipher standard (abbr.). 15. Country-code top-level domain for Timor-Leste (two letters). 18. Print-oriented color model with four inks. 21. Simple lossless compression: counts of repeated runs (abbr.). 22. Worst-case exploit: execute attacker's code on a remote host (abbr.). 23. Impersonating identity or data to deceive systems or users. 24. Structured "what-if" used for testing or threat modeling. 26. Directory in a filesystem. 27. Network delay measured in milliseconds. 28. RISC CPU architecture behind most mobiles. 29. Pair of regex flags: case-insensitive + Unicode mode (two letters). 30. Bind a value to a variable or a task to a user. 33. Windows release between XP and 7. 35. Agreement to keep information confidential (abbr.). 38. Isolated compute environment emulating a machine (abbr.).