

GView

A Smart Generic Visualizer Assistant for Security Analysis

GView

A framework designed to provide **generic visualization** for various file types with **automatic identification** and **artifact extraction** and reanalysis.

GView was designed with **3 major use-cases** in mind:



**Forensics
Investigations**

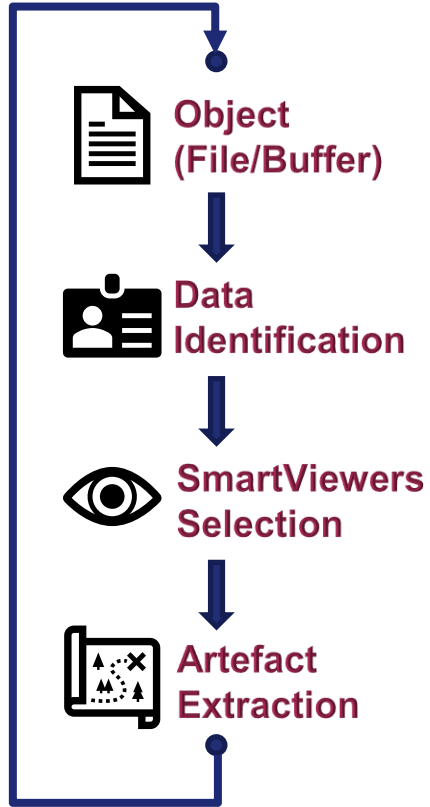


**Security
Operation Centers**

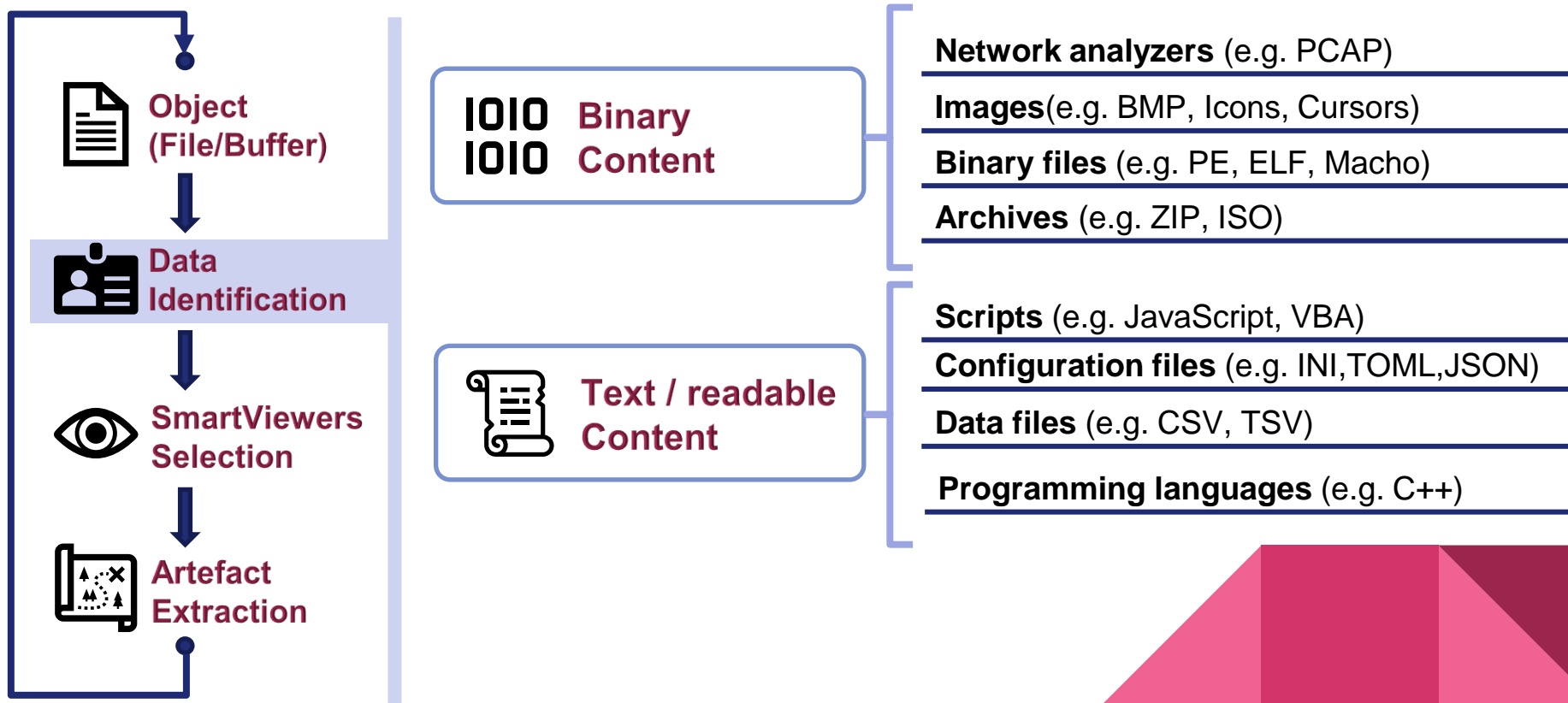


**Security vendors:
In-lab plugins**

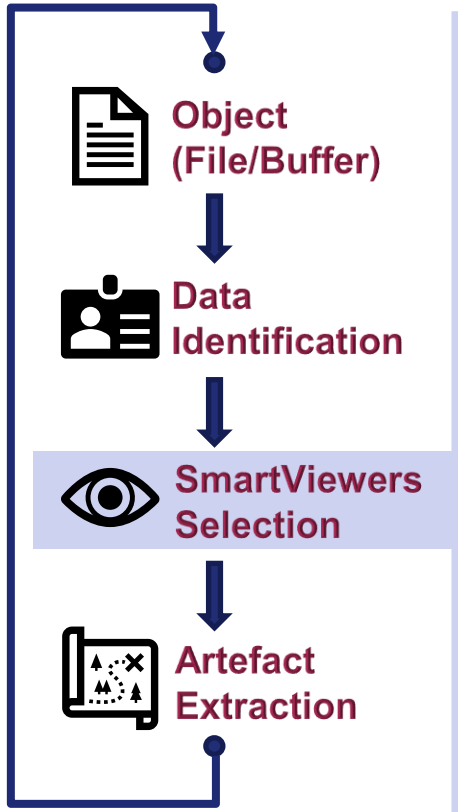
GView architecture overview



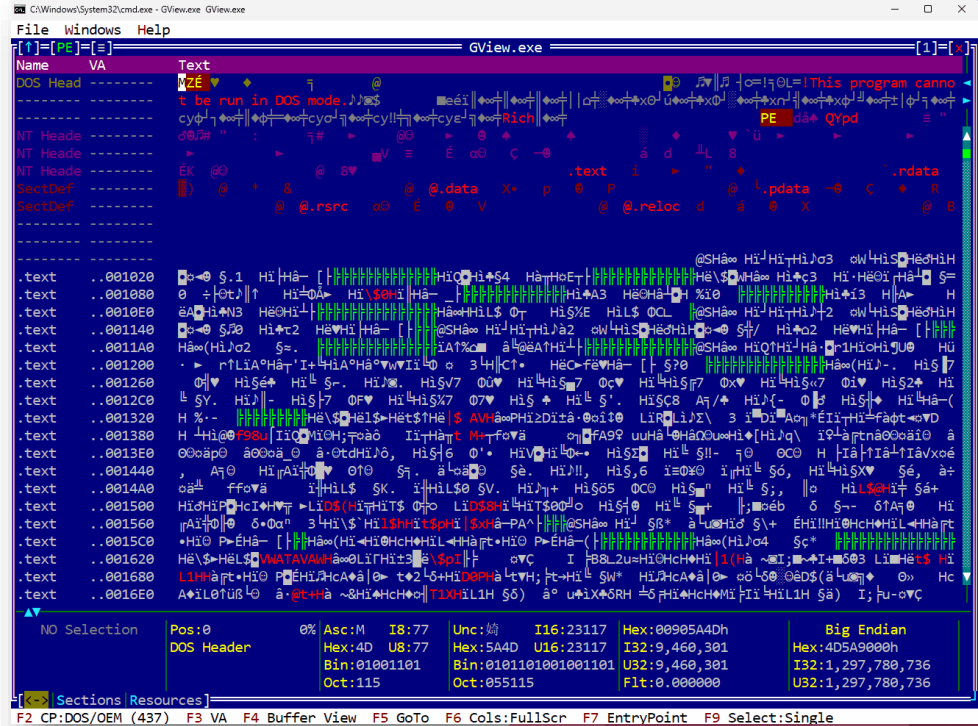
GView architecture overview



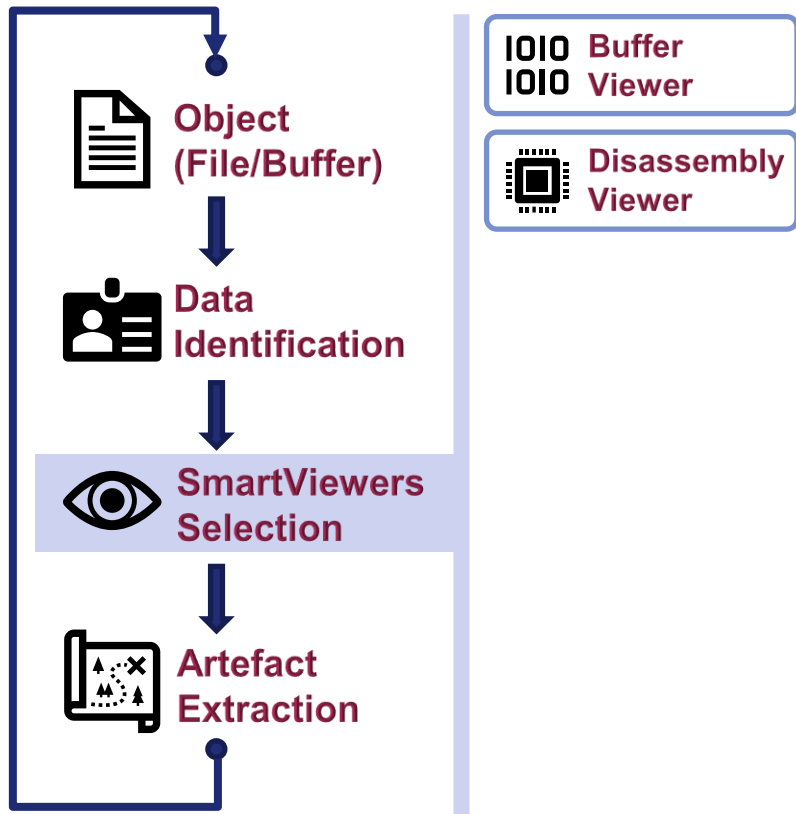
GView architecture overview



IOIO Buffer
IOIO Viewer



GView architecture overview



The screenshot shows the GView application window with the file `poc.bin` open. The **Dissasm zone** is active, displaying assembly code for the `poc.bin` file. The code is as follows:

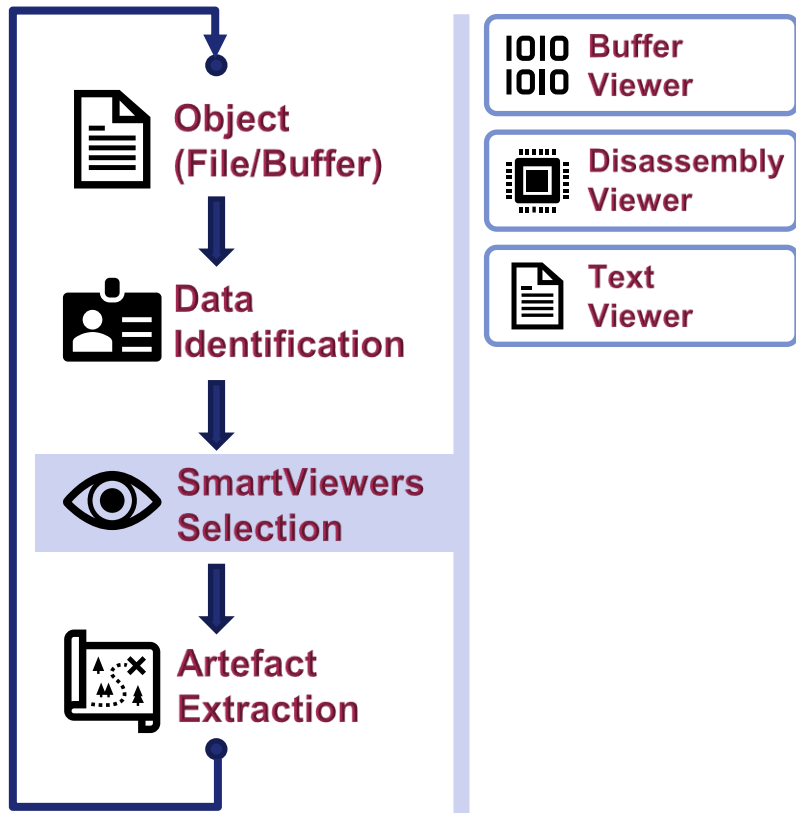
```
0x400: push ebp
0x401: mov ebp, esp
0x403: sub esp, 8
0x406: mov eax, 1
0x40b: test eax, eax
0x40d: je 0x411
0x40f: jmp 0x52
0x411: push 0
0x413: push 0
0x415: push 2
0x417: push 0
0x419: push 2
0x41b: push 0x40000000
0x420: push 0x4022d4
0x425: call CreateFileW
0x42b: mov dword ptr [ebp - 4], eax
0x42e: push 0
0x430: lea ecx, [ebp - 8]
0x433: push ecx
0x434: push 0x1932
0x439: push 0x403000
0x43e: mov edx, dword ptr [ebp - 4]
0x441: push edx
0x442: call WriteFile
0x448: mov eax, dword ptr [ebp - 4]
0x44b: push eax
0x44c: call CloseHandle
0x452: mov esp, ebp
0x454: pop ebp
0x455: ret
0x456: int3
0x457: int3
```

Comments on the right side of the assembly code include:

- `;hTemplateFile`
- `;dwFlagsAndAttributes`
- `;dwCreationDisposition`
- `;lpSecurityAttributes`
- `;dwShareMode`
- `;dwDesiredAccess`
- `;lpFileName "fake_random_note.txt"`
- `;lpOverlapped`
- `;lpNumberOfBytesWritten`
- `;nNumberOfBytesToWrite`
- `;lpBuffer`
- `;hFile`
- `;hObject`

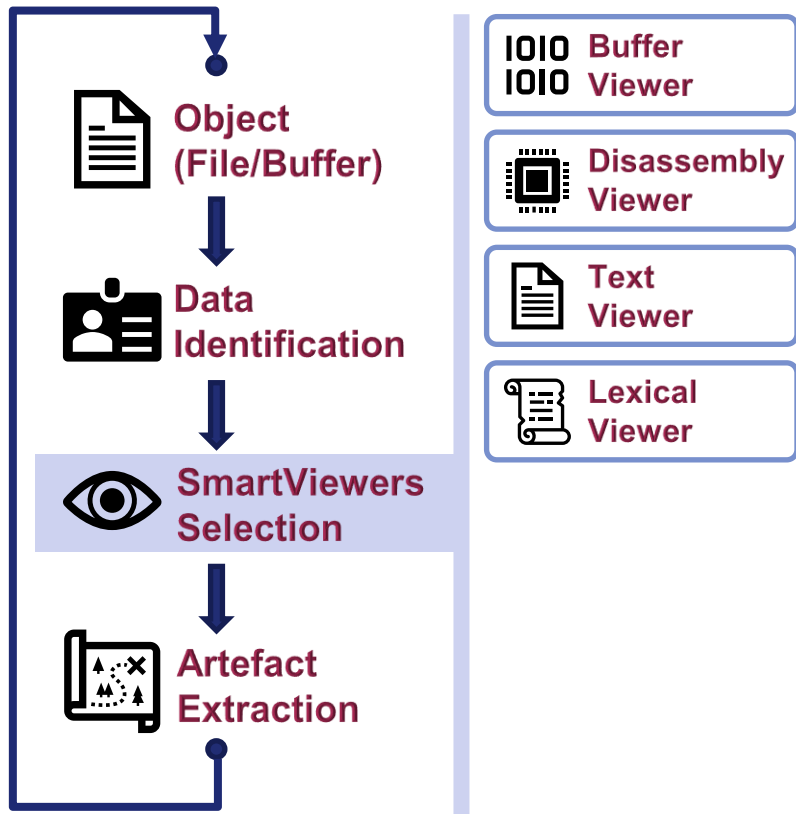
The status bar at the bottom indicates the current position is `Pos:12` at `5%|Line:10`. It also shows the **Sections Resources** tab is selected. The bottom of the window features a menu bar with the following options: **F2 Entry point**, **F4 Dissasm View**, **F5 GoTo**, **F8 Export asm file**, and **F9 ShowFileContent**.

GView architecture overview



A screenshot of the GView application window. The title bar shows the file path: C:\Windows\System32\cmd.exe - GView Z:\Repositories\SampleCollection_Demo\TEXT\Lorem.txt. The menu bar includes File, Windows, and Help. The main text area displays the content of 'lorem.txt', which is a Lorem Ipsum text. The status bar at the bottom shows 'NO Selection', 'Line:1/14', 'Col:1', and 'File ofs: 0'. Below the status bar, there are keyboard shortcuts: F2 Wrap:Bullets, F4 Text View, and F5 GoTo.

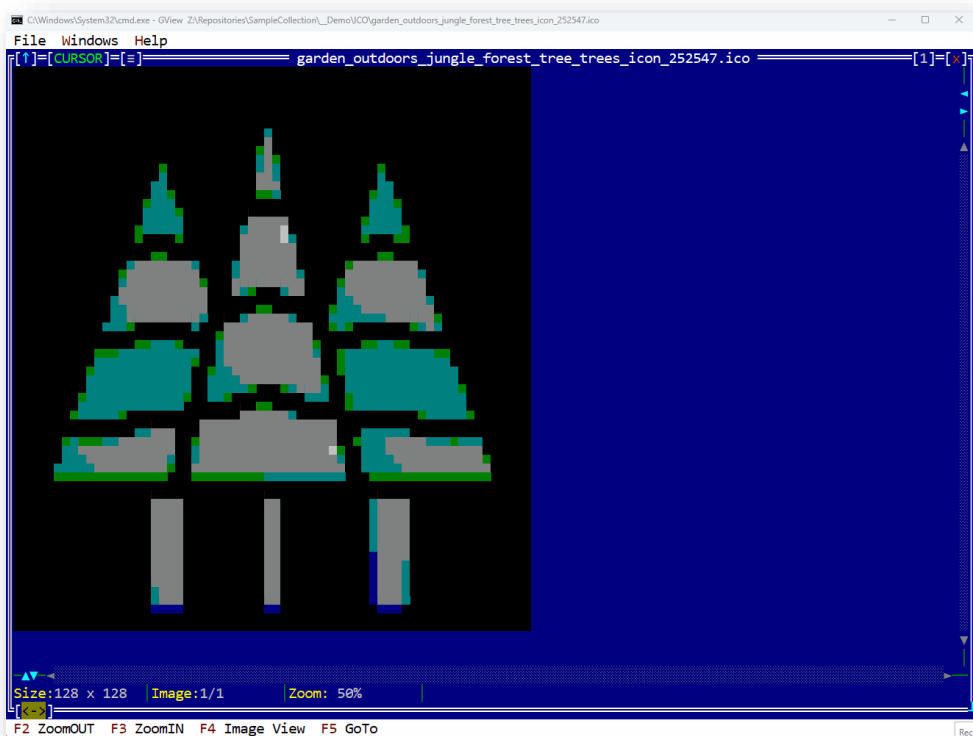
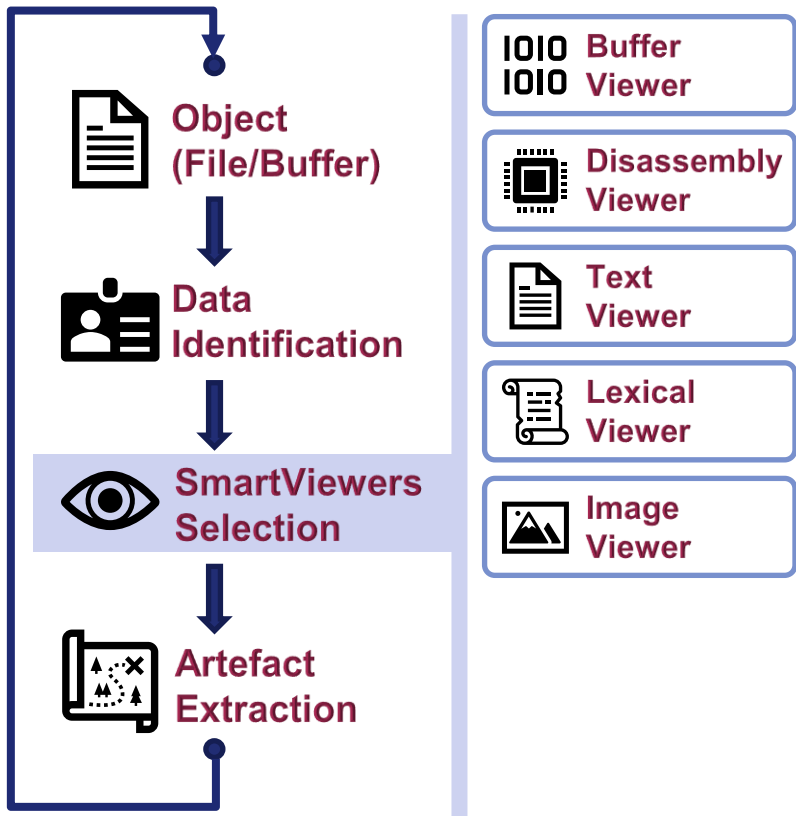
GView architecture overview



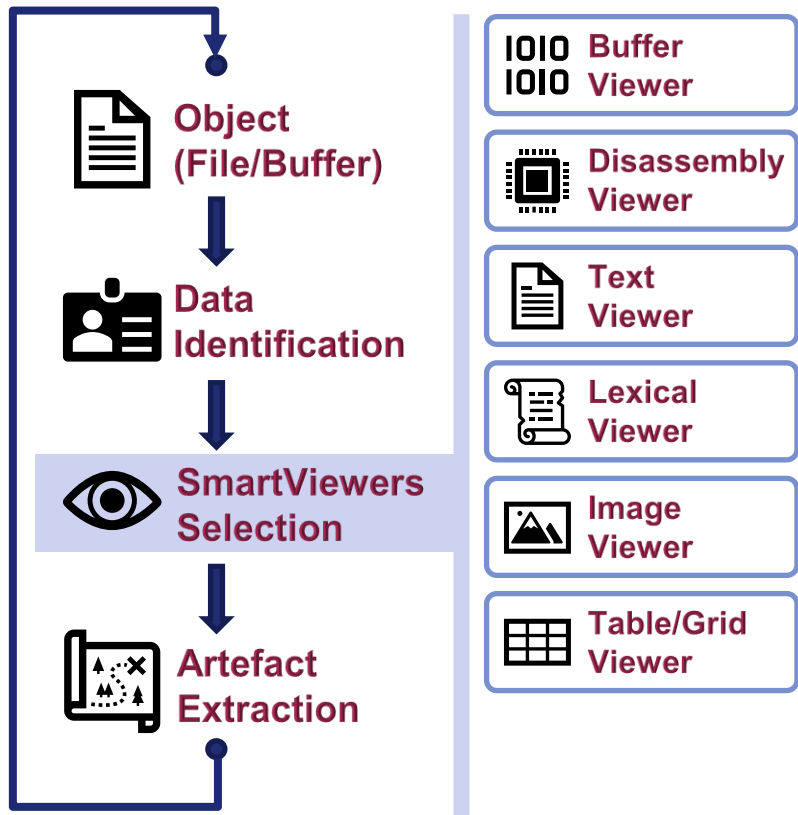
The screenshot shows the GView application window with the file `f1.js` open. The code is displayed in a dark-themed editor with syntax highlighting. The code is a JavaScript function to calculate the factorial of a number. The status bar at the bottom shows the current selection: `Line:1/29, Col:1, Char ofs:0, Token Type: Comment`. The bottom menu bar includes options: `F1 Plugins, F2 Save As, F3 ShowMetaData:ON, F4 Lexical View, F5 GoTo, F8 Fold all, F9 Select:Single, Delete Delete`.

```
1 //Comment-uri: [1]=[x]
2 //program to find the factorial of a number
3 // take input from the user
4 var number = parseInt('12', 10);
5 // checking if number is negative
6 if(number < 0)
7 {
8     //number lower than 0
9     /*(number < 0)*/ console.log('Error! Factorial for negat...');
10 }
11 // if number is 0
12 else
13 if(number === 0)
14 {
15     console.log('The factorial of ${number} is 1.');
```


GView architecture overview



GView architecture overview



CA\Windows\System32\cmd.exe - GView Z:\Repositories\SampleCollection_L_Demo\CSV\test2.csv

File Windows Help

[1]=[-CSV/18]=[-]

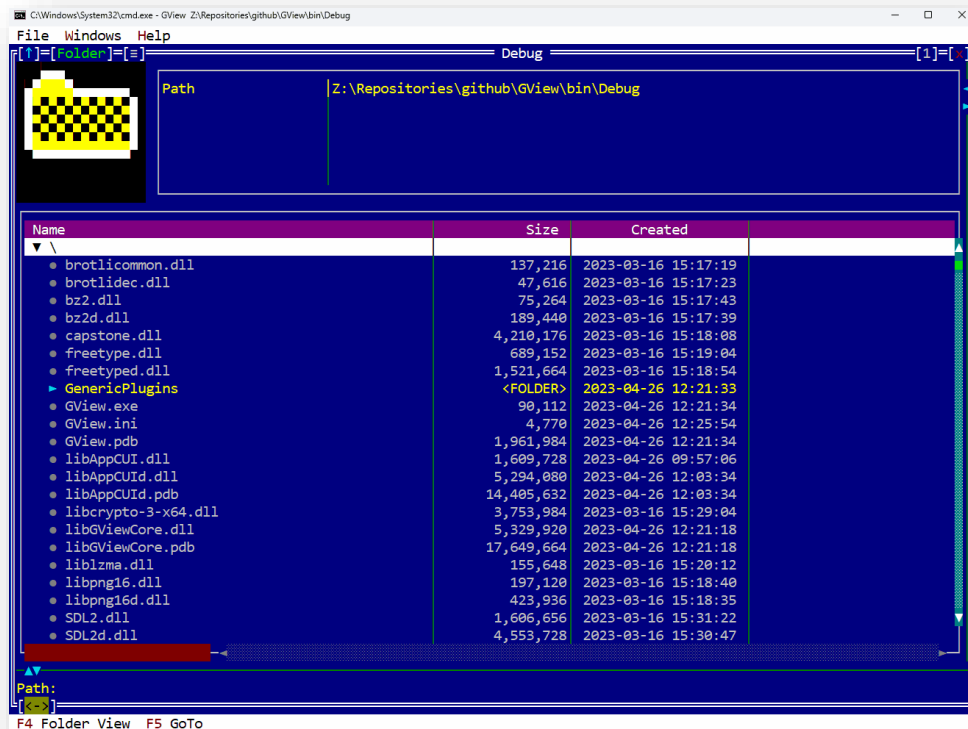
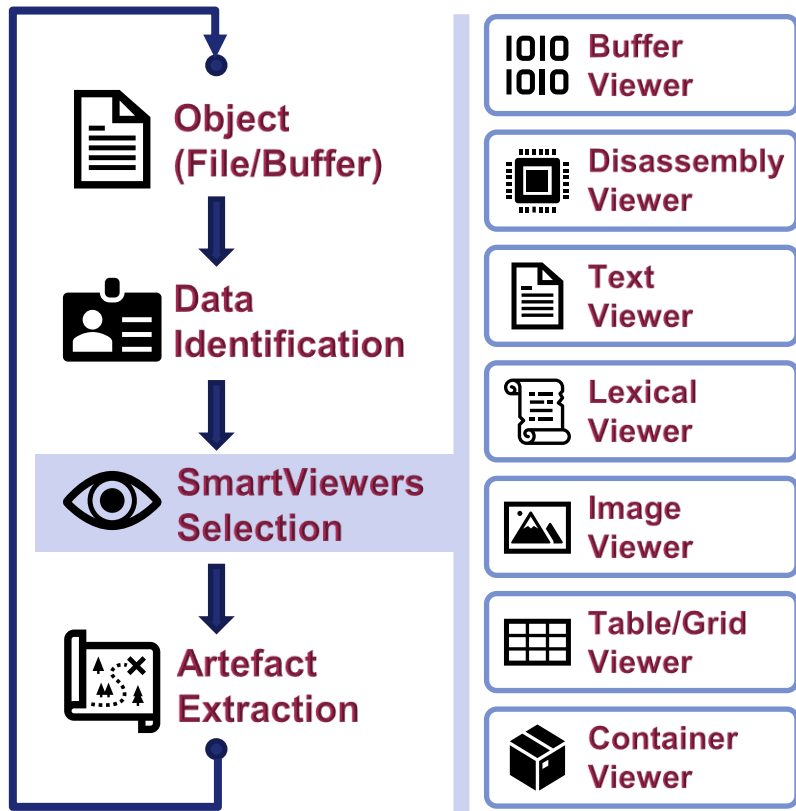
test2.csv [1]=[-]

Column_0	Column_1	Column_2	Column_3	Column_4	Column_5	Column_6	Column_7
0	0	0	0	0	0	0	0
Username	2070	Craig	Booker	Username	2070	Craig	Booker
booker12	2070	Craig	Booker	booker12	2070	Craig	Booker
booker12	2070	Craig	Booker	booker12	2070	Craig	Booker
booker12	2070	Craig	Booker	booker12	2070	Craig	Booker
booker12	2070	Craig	Booker	booker12	2070	Craig	Booker
booker12	4081	First name	Grey	booker12	4081	First name	Grey
grey07	4081	Jamie	Grey	grey07	4081	Jamie	Grey
grey07	4081	Jamie	Grey	grey07	4081	Jamie	Grey
grey07	4081	Jamie	Grey	grey07	4081	Jamie	Grey
grey07	4081	Jamie	Grey	grey07	4081	Jamie	Grey
grey07	5079	Laura	Jenkins	grey07	5079	Laura	Jenkins

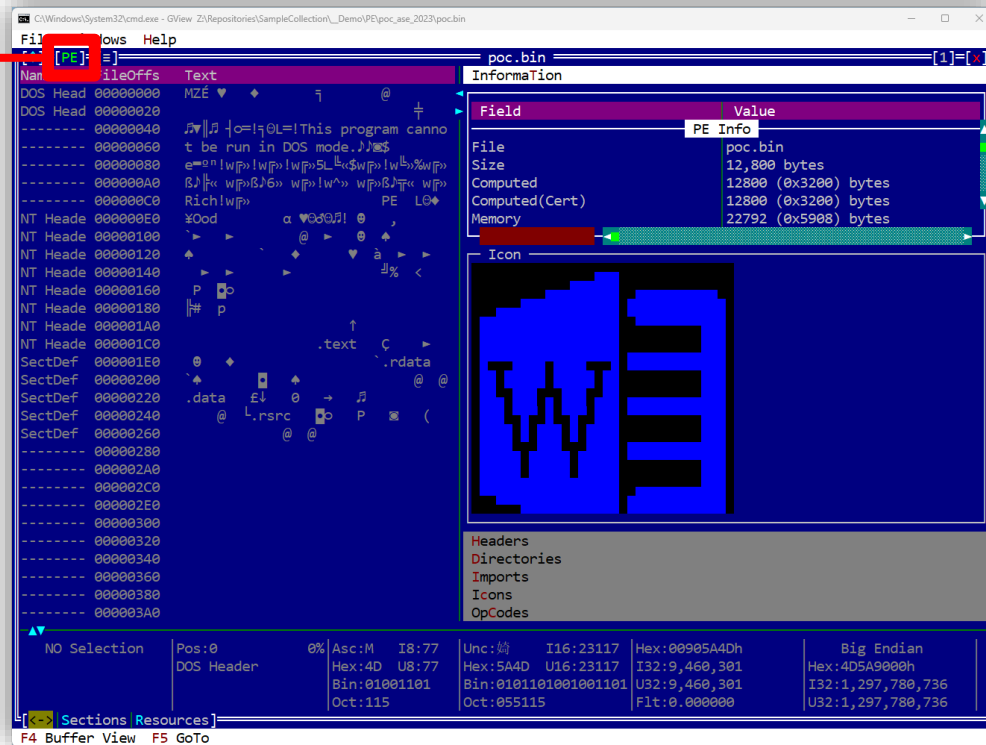
Width:117 Height:34 Cells:248 Hovered:- - Selection:- & - - -

F4 Grid View F5 GoTo Space ReplaceHeader H ToggleHorizontalLines V ToggleVerticalLines

GView architecture overview

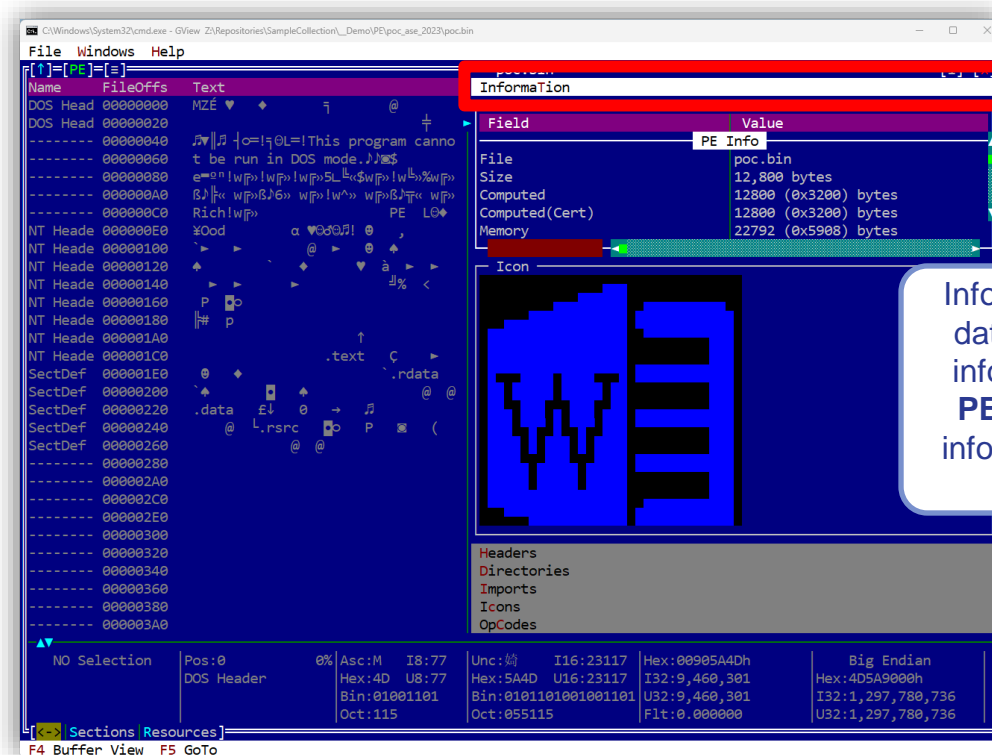


UX Overview



Data type plugin (in this case **PE**)

UX Overview



Information Panel (where each data type plugin adds various information). For example the PE plugin prints here version information, icon, export name, PDB path, ...

UX Overview

This area presents cursor and selection information. For the PE file it also computes different values (such as hex values, bin values, float values, etc)

The screenshot shows a PE file viewer application. The main window displays a hex dump of the file 'poc.bin'. The hex dump is organized into columns: Name, FileOffs, Text, and a hex view. The text column shows the program's header, including 'MZÉ' and 'This program cannot be run in DOS mode.' The hex view shows the raw bytes of the file, with a cursor positioned at the start of the DOS header.

On the right side of the window, there is a 'PE Info' panel. It contains a table with the following data:

Field	Value
File	poc.bin
Size	12,800 bytes
Computed	12800 (0x3200) bytes
Computed(Cert)	12800 (0x3200) bytes
Memory	22792 (0x5908) bytes

Below the PE Info panel is an 'Icon' section, which displays a blue icon with a white 'W' and a black 'E'.

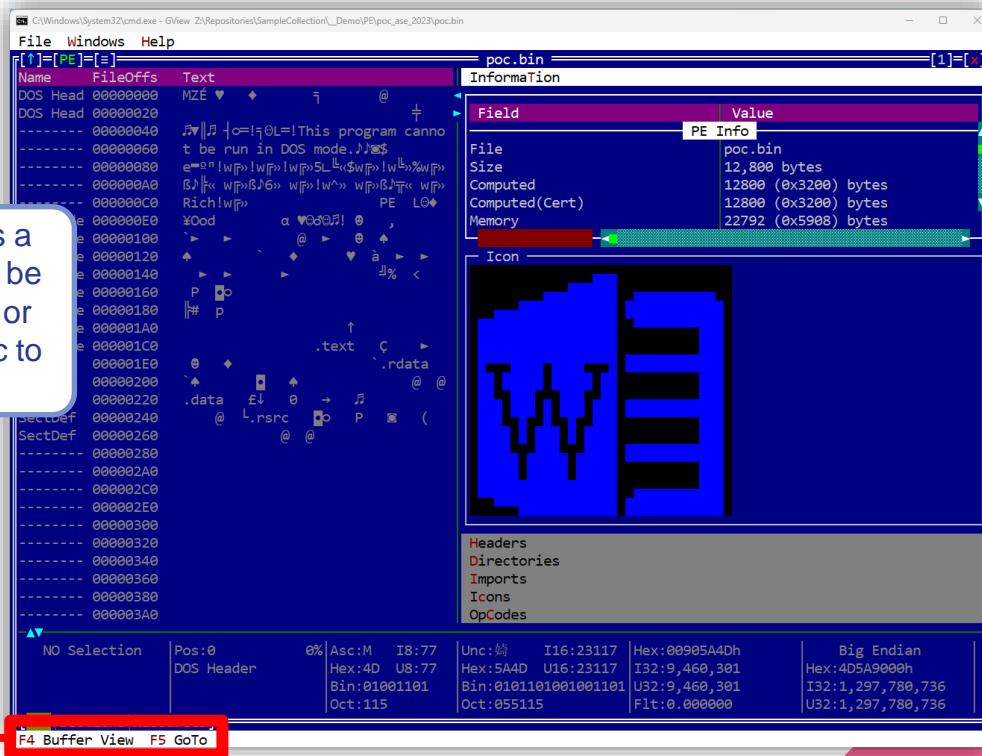
At the bottom of the window, there is a table with the following data:

NO Selection	Pos:0	0%	Asc:M	I8:77	Unc:00	I16:23117	Hex:00905A4Dh	Big Endian
DOS Header			Hex:4D	U8:77	Hex:5A4D	U16:23117	I32:9,460,301	Hex:4D5A9000h
			Bin:01001101		Bin:0101101001001101		U32:9,460,301	I32:1,297,780,736
			Oct:115		Oct:855115		Flt:0.000000	U32:1,297,780,736

A red arrow points from the text box to the table at the bottom of the application window.

UX Overview

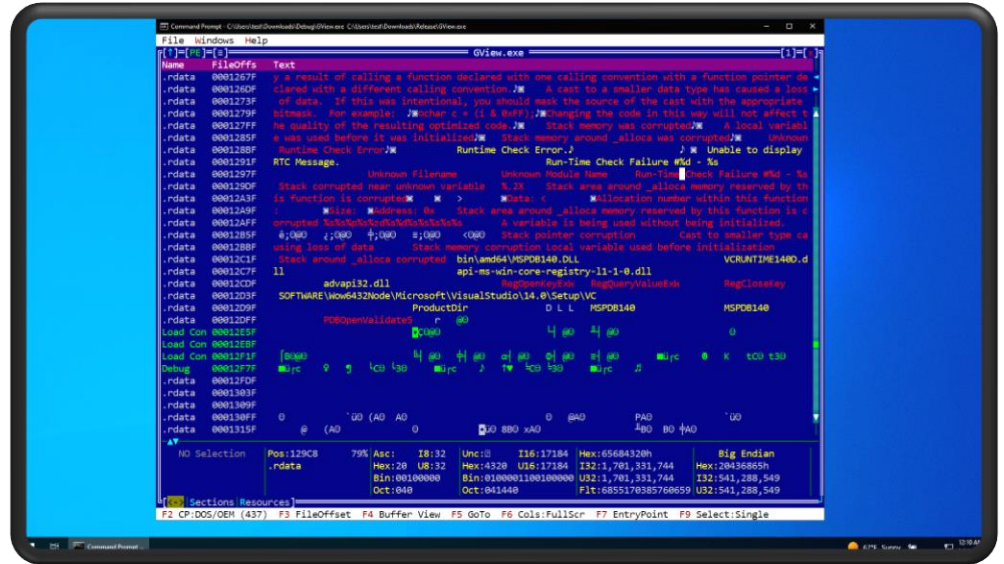
Finally, the bottom bar contains a list of quick commands that can be used to either change the view or to perform various tasks specific to the view or data plugin.



Suported OS architecture



Microsoft Windows



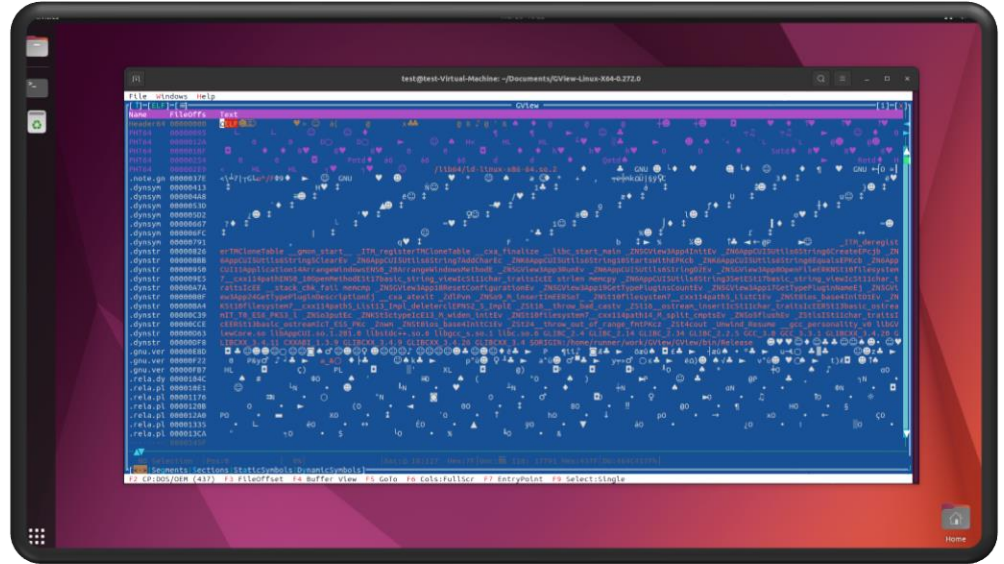
Supported OS architecture



**Microsoft
Windows**



**Linux
(ex. Ubuntu)**



Supported OS architecture



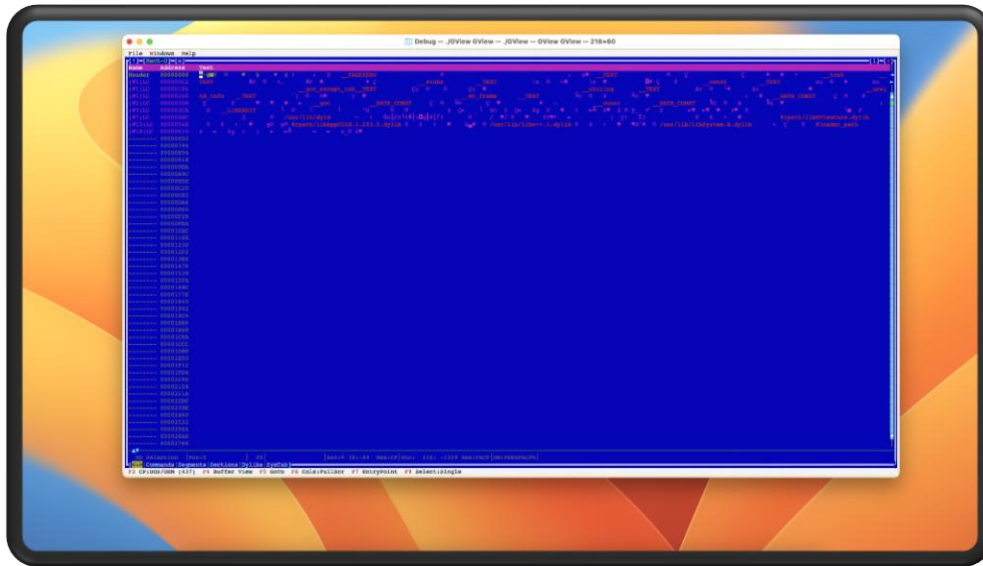
**Microsoft
Windows**



**Linux
(ex. Ubuntu)**



**Apple
MacOS**



Supported OS architecture



**Microsoft
Windows**



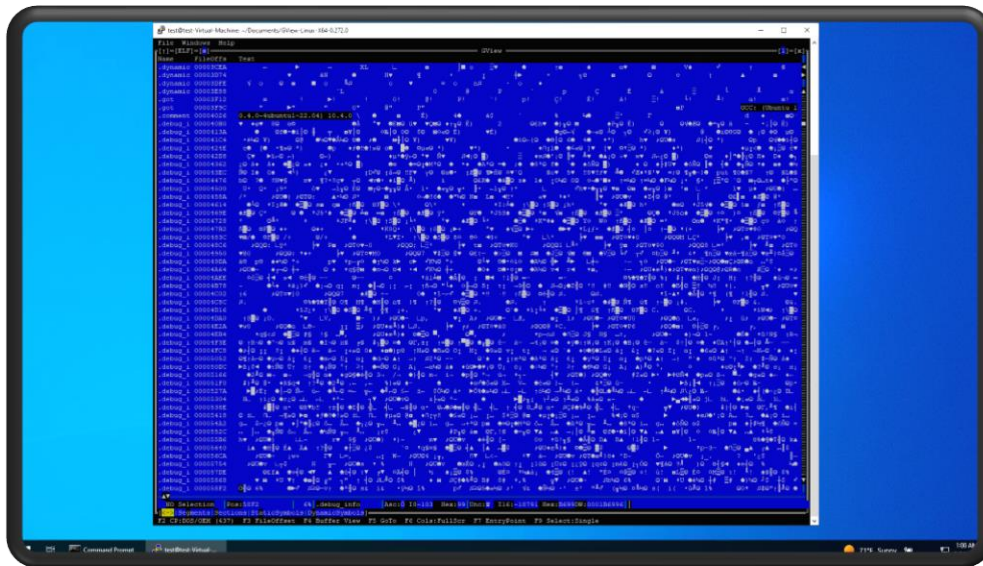
**Linux
(ex. Ubuntu)**



**Apple
MacOS**



**Remote via SSH
connections**



Demo

For the demo purposes we have build a POC traffic capture that resembles a ransomware attack. No actual malware or any piece of malicious code were used in this demo.

Conclusions & Future Work



Conclusions

Security framework for Forensic Engineers and Malware researchers

Cross platform compatibility

Security Operation Center assess solution



Future Development

Email support

Office files (Macro VBA extractor)

PowerShell parser and deobfuscator

Support for in-memory analysis



Thank you !