

Keamanan Jaringan Komputer



Disusun Oleh

Nama : Kusuma Dwi Indriani

NIM : 09011181320017

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA**

2017

EXPLOIT KIT

Exploit kit merupakan kit perangkat lunak yang dirancang untuk berjalan pada server web, dengan tujuan mengidentifikasi kerentanan perangkat lunak di mesin klien yang sedang berkomunikasi. Tidak hanya mengidentifikasi *exploit kit* juga memiliki tujuan menemukan dan mengeksploitasi kerentanan untuk dapat mengupload serta mengeksekusi kode berbahaya pada klien.

Dalam proses Exploit kit hal yang dilakukan diantaranya adalah mengumpulkan informasi tentang mesin korban, menemukan kerentanan dan menentukan sesuai eksploitasi serta memberikan *malware* yang biasanya secara diam-diam pada klient. Diantara berbagai cara *exploit kit* yang ada salah satunya adalah mengintruksi melalui payload.

Hal yang dibutuhkan saat mengintruksi melalui payload adalah file payload yang telah disiapkan sebelumnya. Pada tugas kali ini hal yang dilakukan adalah menganalisa terhadap kedua file payload :

1. payload.exe
2. payload2.exe

Untuk dapat menganalisa kedua file payload tersebut membutuhkan dua tools berupa:

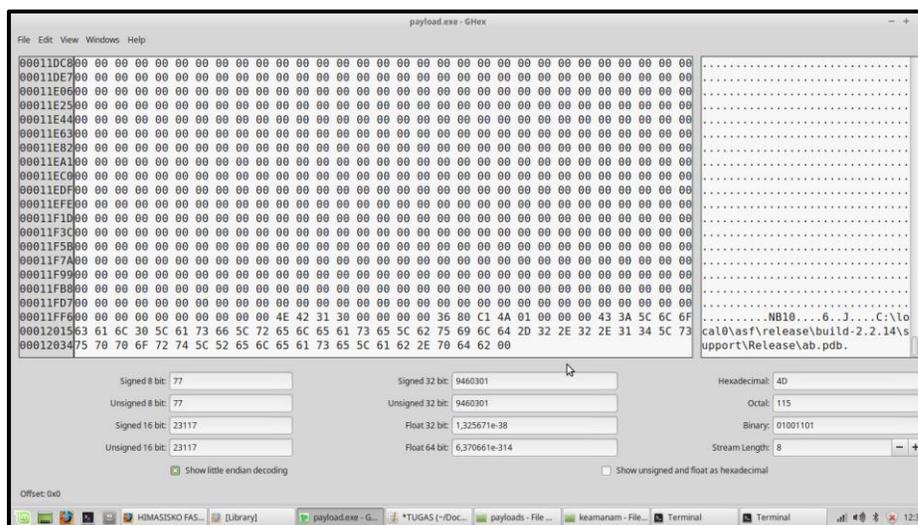
1. ghex, hexdump, strings (linux)
2. ollydbg(win), ida pro(linux,win)

Adapun perintah untuk dapat melihat isi file melalui tool ghex :

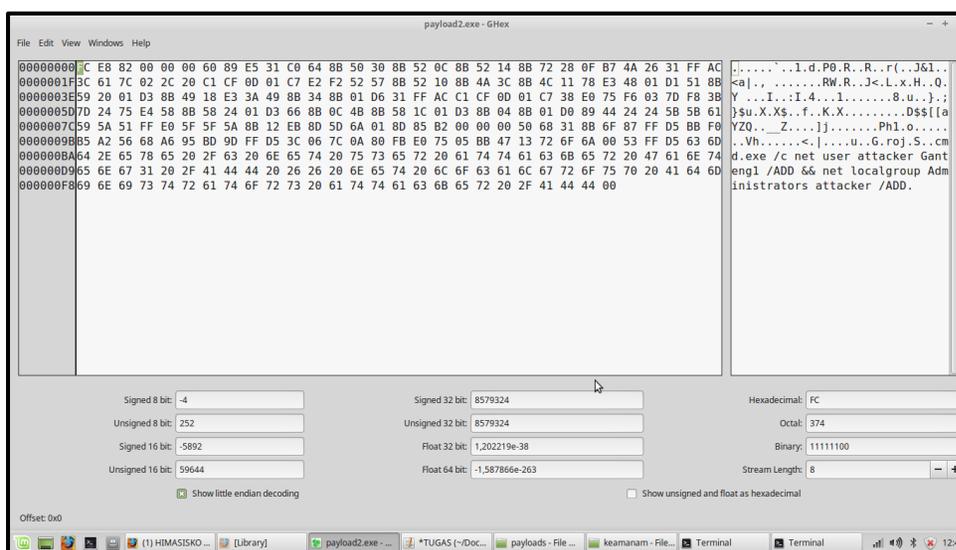
ghex payload.exe

ghex payload2.exe

setelah mengetikkan perintah tersebut maka akan muncul seperti gambar 1 dan gambar 2.

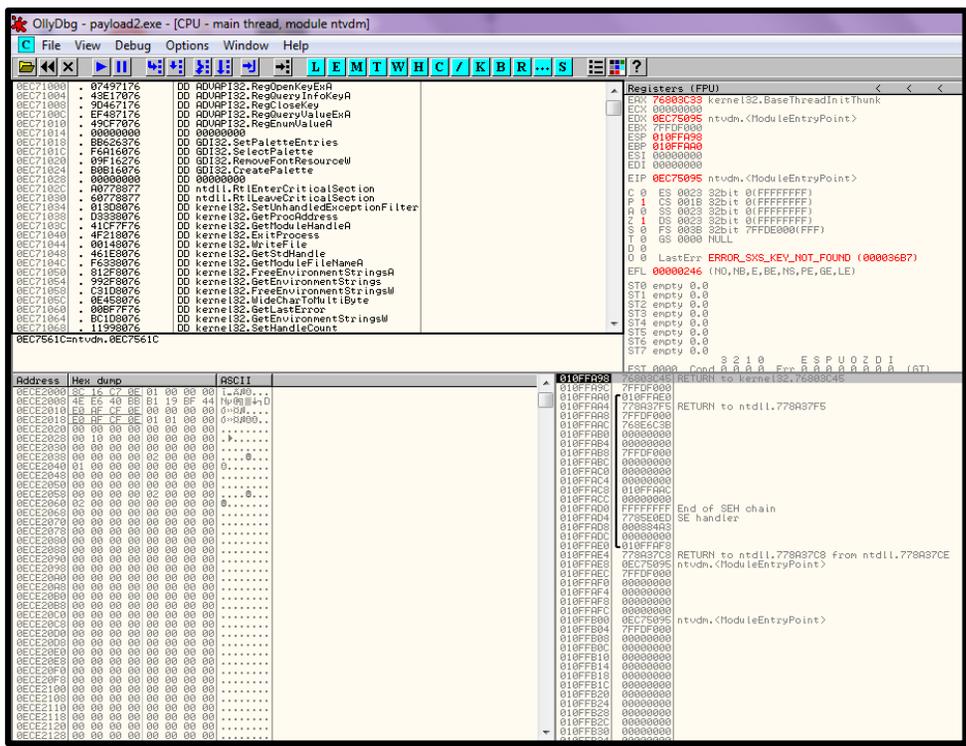


Gambar 1. Hasil file payload.exe menggunakan ghex

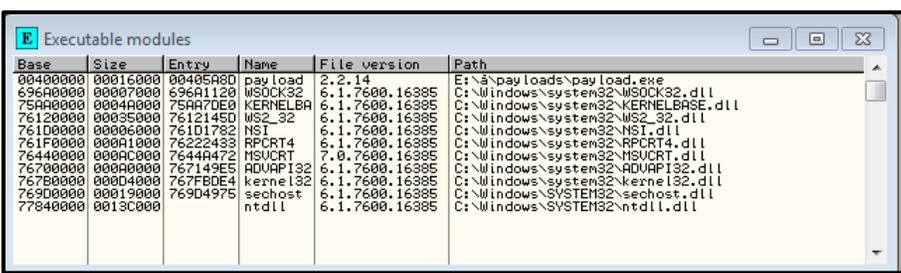


Gambar 2. Hasil file payload2.exe menggunakan ghex

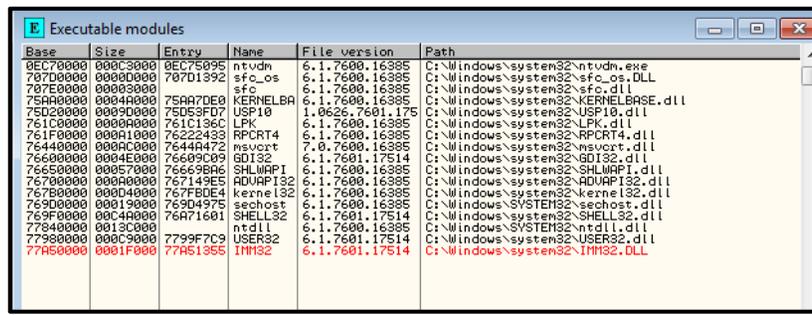
Terlihat dari kedua gambar diatas file ditampilkan dalam bentuk kode-kode pada blok yang menarik. Kode-kode tersebut menandakan bahwa file adalah file ODT(OpenOffice/LibreOffice Document Format). Pada kedua gambar diatas membuktikan bahwa perbedaan kode-kode hexadecimal dalam setiap setiap file. Dari kode-kode hex yang berbeda tersebut kita dapat mengklarifikasi file-file yang dianggap mencurigakan. Multimedia files terkdang memiliki perbedaan dari susunan



Gambar 7. Isi file payload2.exe pada detail CPU



Gambar 8. Isi file payload.exe pada detail executable modules



Gambar 9. Isi file payload2.exe pada detail executable modules

Detail Executable modules berisi bagaimana kedua file payload tersebut dijalankan. Dimana terdapat perbedaan lokasi kedua file tersebut dijalankan kecuali bahwa kedua file sama sama dijalankan pada lokasi windows\system32. Tidak hanya tempat kedua file dijalankan tetapi rincian nama dan ukuran dari file tersebut dijalankan juga tertera.

Log Data pada kedua file tidak berbeda jauh dengan detail dari executable modules dimana berisi proses yang dibuat pada sistem oleh kedua file tersebut. Akan tetapi dikarenakan isi kedua file berbeda maka prose yang terjadi pada file juga berbeda. Telihat pada detail paling bawah kedua gmabar 10 dan gambar 11 dianalisa **8 calls to known** pada payload dan **1440 calls to known** pada payload2

```

Log data
Address  Message
-----  -
OllYdbg v1.10
File 'E:\a\payloads\payload.exe'
New process with ID 00000624 created
Main thread with ID 000007C4 created
00405800 Module E:\a\payloads\payload.exe
63680000 Module C:\Windows\system32\USER32.dll
759A0000 Module C:\Windows\system32\KERNELBASE.dll
76120000 Module C:\Windows\system32\USER32.dll
761D0000 Module C:\Windows\system32\USER32.dll
761F0000 Module C:\Windows\system32\RPCRT4.dll
76440000 Module C:\Windows\system32\USER32.dll
76700000 Module C:\Windows\system32\ADVAPI32.dll
767B0000 Module C:\Windows\system32\kernel32.dll
769D0000 Module C:\Windows\SYSTEM32\sechost.dll
77840000 Module C:\Windows\SYSTEM32\ntdll.dll
00405800 Program entry point
Analysing payload
9 heuristical procedures
8 calls to known, 1 call to guessed functions
1 switches
  
```

Gambar 10. Isi file payload.exe pada detail Log data

```

Log data
Address  Message
-----  -
OllYdbg v1.10
File 'E:\a\payloads\payload2.exe'
New process with ID 00000604 created
Main thread with ID 0000047C created
0EC75095 Module C:\Windows\system32\ntvdn.exe
207D0000 Module C:\Windows\system32\sfco_os.DLL
707E0000 Module C:\Windows\system32\sfco.dll
759A0000 Module C:\Windows\system32\KERNELBASE.dll
75D20000 Module C:\Windows\system32\USER32.dll
761D0000 Module C:\Windows\system32\LPK.dll
761F0000 Module C:\Windows\system32\RPCRT4.dll
76440000 Module C:\Windows\system32\msvart.dll
76600000 Module C:\Windows\system32\GDI32.dll
76650000 Module C:\Windows\system32\SHLWAPI.dll
76700000 Module C:\Windows\system32\ADVAPI32.dll
767B0000 Module C:\Windows\system32\kernel32.dll
769D0000 Module C:\Windows\SYSTEM32\sechost.dll
769F0000 Module C:\Windows\system32\SHELL32.dll
77840000 Module C:\Windows\SYSTEM32\ntdll.dll
77990000 Module C:\Windows\system32\USER32.dll
77A50000 Module C:\Windows\system32\IMM32.DLL
0EC75095 Program entry point
Analysing ntvdn
1385 heuristical procedures
1440 calls to known, 514 calls to guessed functions
1024 loops, 230 switches
  
```

Gambar 11. Isi file payload2.exe pada detail Log data

Threads menunjukkan bagaimana kedua file payload tersebut berisi status aktif dan dapat digunakan dalam mengintruksi melalui payload.

Ident	Entry	Data block	Last error	Status	Priority	User time	System time
000007C4	00405A80	7FFDF000	ERROR_SUCCESS (00000000)	Active	32 + 0	0.0000 s	0.0156 s

Gambar 12. Isi file payload.exe pada detail Threads

Ident	Entry	Data block	Last error	Status	Priority	User time	System time
0000047C	0EC75095	7FFDE000	ERROR_SXS_KEY_NOT_FOUND (80070002)	Active	32 + 0	0.0000 s	0.0156 s

Gambar 13. Isi file payload2.exe pada detail Threads

Detail memori map menjelaskan secara rinci skema dari file payload.exe saat di proses. Begitu rinci mulai dari alamat, ukuran file, bentuk file serta akses yang dapat dilakukan. Akan tetapi untuk file payload2.exe detail memory map tidak bisa ditampilkan.

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00010000	00010000				Map	RW	RM	
00120000	00010000			stack of na	Priv	RW	Gua	RM
00120000	00020000				Priv	RW	Gua	RM
00130000	00040000				Map	R	R	
00140000	00010000				Priv	RW	RM	
00140000	00050000				Priv	RW	RM	
00200000	00050000				Map	R	R	
003C0000	00050000				Priv	RW	RM	
00400000	00010000	payload		PE header	Imag	R	RME	
00401000	00000000			code	Imag	R	RME	
0040C000	00010000	payload	.text	imports	Imag	R	RME	
0040D000	00000000			.idata	Imag	R	RME	
0040D000	00000000	payload	.data	data	Imag	R	RME	
00415000	00010000	payload	.rsrc	resources	Imag	R	RME	
00500000	00010000				Priv	RW	RM	
696A0000	00010000	WSOCK32		PE header	Imag	R	RME	
696A1000	00000000			code, import	Imag	R	RME	
696A4000	00010000	WSOCK32	.data	data	Imag	R	RME	
696A5000	00010000	WSOCK32	.rsrc	resources	Imag	R	RME	
696A6000	00010000	WSOCK32	.reloc	relocations	Imag	R	RME	
75A00000	00010000	KERNELBA		PE header	Imag	R	RME	
75A01000	00040000	KERNELBA	.text	code, import	Imag	R	RME	
75A04000	00020000	KERNELBA	.data	data, import	Imag	R	RME	
75A06000	00010000	KERNELBA	.rsrc	resources	Imag	R	RME	
75A07000	00000000	KERNELBA	.reloc	relocations	Imag	R	RME	
76120000	00010000	WS2_32		PE header	Imag	R	RME	
76121000	00020000	WS2_32	.text	code, import	Imag	R	RME	
76147000	00010000	WS2_32	.data	data	Imag	R	RME	
76148000	00000000	WS2_32	.rsrc	resources	Imag	R	RME	
76153000	00020000	WS2_32	.reloc	relocations	Imag	R	RME	
761D0000	00010000	NSI		PE header	Imag	R	RME	
761D1000	00000000	NSI	.text	code, import	Imag	R	RME	
761D3000	00010000	NSI	.data	data	Imag	R	RME	
761D4000	00010000	NSI	.rsrc	resources	Imag	R	RME	
761D5000	00010000	NSI	.reloc	relocations	Imag	R	RME	
761F0000	00010000	RPCRT4		PE header	Imag	R	RME	
761F1000	00030000	RPCRT4	.text	code, import	Imag	R	RME	
76284000	00000000	RPCRT4	.orpc	code	Imag	R	RME	
76287000	00010000	RPCRT4	.data	data	Imag	R	RME	
76289000	00000000	RPCRT4	.rsrc	resources	Imag	R	RME	
7628C000	00000000	RPCRT4	.reloc	relocations	Imag	R	RME	
76440000	00010000	MSUCRT		PE header	Imag	R	RME	
76441000	00000000	MSUCRT	.text	code, import	Imag	R	RME	
764E0000	00000000	MSUCRT	.data	data	Imag	R	RME	
764E7000	00010000	MSUCRT	.rsrc	resources	Imag	R	RME	
764E8000	00000000	MSUCRT	.reloc	relocations	Imag	R	RME	
76700000	00010000	ADVAPI32		PE header	Imag	R	RME	
76701000	00070000	ADVAPI32	.text	code, import	Imag	R	RME	
76773000	00040000	ADVAPI32	.data	data	Imag	R	RME	
76777000	00020000	ADVAPI32	.rsrc	resources	Imag	R	RME	
76778000	00000000	ADVAPI32	.reloc	relocations	Imag	R	RME	
767B0000	00010000	kernel32		PE header	Imag	R	RME	
767B1000	00000000	kernel32	.text	code, import	Imag	R	RME	
767B6000	00010000	kernel32	.data	data	Imag	R	RME	
76877000	00010000	kernel32	.rsrc	resources	Imag	R	RME	
76878000	00000000	kernel32	.reloc	relocations	Imag	R	RME	
76900000	00010000	sechost		PE header	Imag	R	RME	
76901000	00010000	sechost	.text	code, import	Imag	R	RME	
769E4000	00000000	sechost	.data	data	Imag	R	RME	
769E7000	00010000	sechost	.rsrc	resources	Imag	R	RME	
769E8000	00000000	sechost	.reloc	relocations	Imag	R	RME	
77840000	00010000	ntdll		PE header	Imag	R	RME	
77841000	00000000	ntdll	.text	code, export	Imag	R	RME	
77916000	00010000	ntdll	RT		Imag	R	RME	
77917000	00000000	ntdll	.data	data	Imag	R	RME	
77920000	00050000	ntdll	.rsrc	resources	Imag	R	RME	
77977000	00000000	ntdll	.reloc	relocations	Imag	R	RME	
77980000	00010000				Imag	R	RME	
7FF6F000	00000000				Map	R	R	
7FFB0000	00020000				Map	R	R	
7FFD0000	00010000				Priv	RW	RM	

Gambar 14. Isi file payload.exe pada detail memory map

Penggunaan tool ollydbg terlihat begitu detail dalam mendeskripsikan bagaimana isi dari file payload.exe dan payload2.exe. Detail yang ditampilkan berupa CPU, executable modules, Log data, Threads, dan memory map. Tools ini menampilkan proses dan skema kedua file dengan begitu baik.

sedangkan isi file dengan menggunakan tools ida pro seperti gambar 15 dan seterusnya. Untuk keterangan pada gambar isi hampir sama pada saat menggunakan tools ollydbg.

```

IDA View-A  Hex View-A  Exports  Imports  Names
seg000 ;
seg000 ;+-----+
seg000 ;| This file is generated by The Interactive Disassembler
seg000 ;| Copyright (c) 2010 by Hex-Rays SA, <support@hex-rays.com>
seg000:0100 ;| Licensed to: Freeware version 1.0
seg000 ;+-----+
seg000 ;
seg000 ;Input MD5 : 5148D815F36E
seg000 ;
seg000 ;-----+
seg000 ;File Name : C:\payload
seg000 ;Format : MS-DOS
seg000 ;Base Address: 1000h Range: 10100h
seg000 ;
seg000 ;.68
seg000 ;.m
seg000 ;.model
seg000 ;
seg000 ;-----+
seg000 ;
seg000 ; Segment type:
seg000 seg000 segment byte public
seg000 assume cs:
seg000 org 1
seg000 assume es:nothing, ss:nothing, ds:seg000
seg000
seg000 public s

```

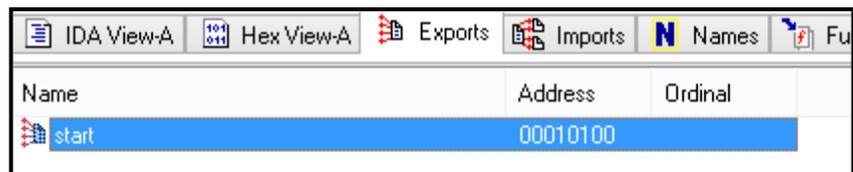
Gambar 15. Isi file payload2.exe menggunakan ida pro

```

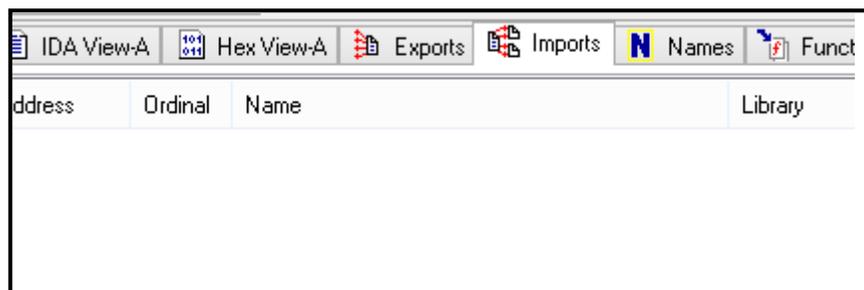
IDA View-A  Hex View-A  Exports
seg000: E800068 E3C68538 |Pr. e01+
seg000 50851872 0B423FA3 |RiirAJ&
seg000 67022CC0 0CEF5585 |ah. -# Å(
seg000 18438417 E40C5852 |i|k|L|*|C
seg000 0C841E34 8380C3FA |W:li4i
seg000 CC00C3E7 F07F3727 |# Å80u
seg000 E58520C6 8048510C |XIX$ E
seg000 8080C842 2556555F |i ÅeD$
seg000 E55581E8 56088B00 |0_zj| à
seg000 0563868F CBFBA56A |.Ph1ioç i+
seg000 9B9FC307 08FE70B4 |òc0 ik-lç'
seg000 176605FC 66626762 |!roj.S icm
seg000 26266727 76726776 |/c net use
seg000 66672466 76663224 |cker Gant
seg000 44222266 72666666 |DD && nel
seg000 76772466 66677767 |roup Admi
seg000 67726776 66672244 |ors attack
seg000 40 C

```

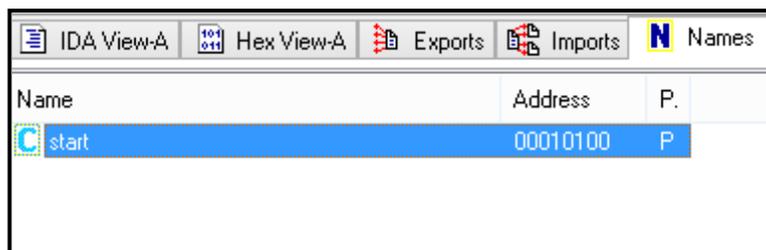
Gambar 16. Isi file payload2.exe menggunakan ida pro



Gambar 17. Isi file payload2.exe menggunakan ida pro



Gambar 18. Isi file payload2.exe menggunakan ida pro



Gambar 19. Isi file payload2.exe menggunakan ida pro



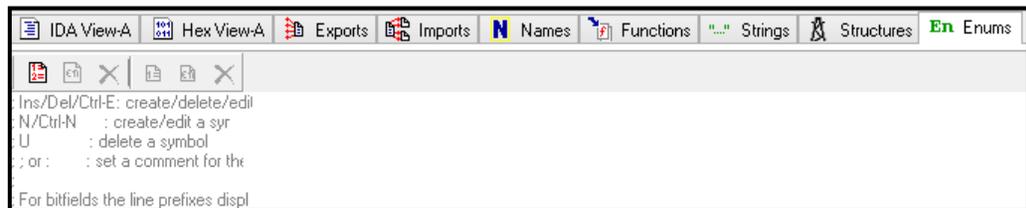
Gambar 20. Isi file payload2.exe menggunakan ida pro



Gambar 21. Isi file payload2.exe menggunakan ida pro



Gambar 22. Isi file payload2.exe menggunakan ida pro



Gambar 23. Isi file payload2.exe menggunakan ida pro