CYDES 2023: Manual Unpacking MPRESS(2.19) - The ESP Trick

Before we get started, I would like to take a moment to thankyou the organizers of the Cyber Warzone CTF challenges. The National Cyber Security Agency Malaysia (NACSA), Velum Labs, etc and the technical team WargamesMY have all worked hard to make this event success.

Warmup



I spend many hours figuring it out, IoI. The challenge can be solve with easy way, i choose the hard road. I love myself.

Challenge 2 Solves		
Nam	е	Date
RootPwner	rs-MY	July 10th, 12:24:44 PM
Team Cir	ncai	July 10th, 12:27:32 PM

Dynamic Analysis



This is a clear indicator that the program is packed.

File name				
C: \Users Des	sktop\CYDES2023\Reverse Engine	ering\Warmup\warmu	up.exe	
File type	Entry point		Base address	
PE32 👻	00406221	> Disasm	00400000	Memory map
PE	Export Import	Resources	.NET TLS	6 Overlay
Sections	TimeDateStamp	SizeOfImage	Resources	;
0003 >	2023-07-08 17:43:24	0008000	Manife	Version
Scan	Endianness	: Mode	Architecture	Туре
Detect It Easy(DiE)	▼ LE	32	I386	Console
packer	М	PRESS(2.19)[-]		S ?
linker	unknown(14	.36)[Console32,co	nsole]	S

By looking at the sample, we know its PEx32 file type and the entry point is 00406221 and the file is packed. The plan here is that, we need to unpacked it first, but how? should we just find any public automated tools? or can we unpacked it manually?

So, the answer is that we will do it manually. Because, why not? :)

Unpacking MPRESS(2.19) - The ESP Trick

In order to successfully reverse engineer packed we need to debug it until we get to the decompressed memory section. Then we can dump that out and analyze that dumped executable.

One trick in doing that is the "ESP trick". So named for the ESP register, we can use this trick to set a hardware breakpoint on the ESP register, and when we get to the breakpoint we should be at the Original Entry Point (OEP) of the program. We can then dump the rest of the executable and we should have our unpacked executable.



Load the sample to any debugger that you can use, make sure it support x32. Run as admin to get full access on it.

	H H → → LEMTWH	C / K B R S ☷ ₩ ?		
CPU - main thread, mo	odule warmup			
	PUBHO PUBHO PUBLOW: 0.00406227 CPDL BAY PUBLOW: 0.00406227 PUBLOW: 0.004 PUBLOW: 0.004 PU		∧ Registers (FPU) End 014FFC0 CFFEI Warmpo.th Warmpo.th To 0140CFEI Warmpo.th CFFEI Warmpo.th Warmpo.th To 0150CFEI Warmpo.th CFFEI Warmpo.th <th></th>	
Hottness Hext during 004467000 00	HSC11 HSC14 00 00 00 00 00 <th>Constraints Constraints Constrain</th> <th>SEGAD SEGAD SECTOR LEVENES, 2004 CPT cellin (TTUNK SETUR) KEPNES, 2004 CPT cellin (TTUNK SETUR) SETUR SETUR) to ntdll.776E7A9E SEGAD</th> <th>Â</th>	Constraints Constrain	SEGAD SEGAD SECTOR LEVENES, 2004 CPT cellin (TTUNK SETUR) KEPNES, 2004 CPT cellin (TTUNK SETUR) SETUR SETUR) to ntdll.776E7A9E SEGAD	Â

Step over to the call

C CPU - main thread module warmun	
00406221 60 004000000 FUSH00 7	Registers: (FU) ()) ())
Horizania Horizania <t< th=""><th>IPFEET Appearance IPFEET userway, CHool in ElectryPoint> IPFEET userway, CHool in ElectryPoint</th></t<>	IPFEET Appearance IPFEET userway, CHool in ElectryPoint> IPFEET userway, CHool in ElectryPoint

Then we select the first four bytes in the dump at the bottom of debugger and set a hardware access breakpoint on the DWord. This will have us break right before we unpack the executable.

CPU - main threa	d, module warmup							
06227 58 06227 58	Backup	>						_
06220 05 9F0200 06220 8B30 0622F 03F0	Сору	>						
06231 28C0 06233 88FE	Binary	>						
06235 66:AD 06237 C1E0 0C	Breakpoint	>	Me	mory, on access				
06230 50 06230 50	Search for	>	Me	mory, on write				
0623E 2BC8 06240 03F1	Follow DWORD in Dum	p 🚺	Ha	dware on access	> Bvt	e		
06242 8BC8 06244 57 06245 51	Go to	>	Ha	dware, on write	> Wo	rd		
06246 49 06247 884439 06	✓ Hex	>	Ha	rdware, on execution	Dw	ord		
0624E ^75 F6 06250 8BD6	Text	>						
06252 8BCF 06254 E8 5C0000	Short	>						
06259 5E 0625A 5A 0625B 2BC0	Long	>						
2625D 890432 36260 B4 10	Float	>						
26262 2800 36264 28C9 36266 38CA	Disassemble							
36227=warmup.004	Special	>						
and they down	Appearance	>					00195554	0040
1 1 1 2 4 0 9 FF5C 80 FF 19 0 9 FF5C 80 FF 19 0 9 FF5C 80 FF 19 0 9 FF6C 21 62 40 0 9 FF6C 21 62 40 0 9 FF6C 21 62 40 0 9 FF7C 10 FA 15 7 9 FF7C 10 FA 15 7 9 FF84 9E 7A 6E 7 9 9FF34 00 00 00 00 9 9FF34 00 00 00 00	21 52 40 00 MDE TDE. 74 FF 19 00 C + t + 21 62 40 00 PX tbe. CC FF 19 00 tbe.F +. 00 50 25 00 1 ≥ 30 + 200 PX. 00 50 25 00 P×30 + 200 PX. 00 50 25 00 P×30 PX. 00 50 25 00 PX. 00 50 PX. 00 50 PX. 00 50 PX. 00 50 PX. 00 50 PX. 00 50 PX. 00 PX.						 0019FF58 0019FF50 0019FF60 0019FF64 0019FF68 0019FF68 0019FF70 0019FF70 0019FF70 0019FF78 0019FF78 0019FF78 0019FF78 0019FF78 0019FF78 	0040 0019 0025 0040 0040 0040 0040 7715 0025 7715 7715

and click run, it will stop at the breakpoint

	C / K B R S 🔚 📰 ?		
CPU - main thread, module warmup			×
COURSE COURSE <thcourse< th=""> <thcourse< th=""> <thcourse< td="" th<=""><td>2 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12</td><td>Persiters (FPU) < <</td><td><</td></thcourse<></thcourse<></thcourse<>	2 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12 CHR 12	Persiters (FPU) < < < < < < < < < < < < < < < < < < <	<
Hodress Her. duro FSC11 0017FFFL 0122 040 052 040 052 050 052 050 052 050 052 050 052 050 052 050 052 050 052 050	A CONSTRUCT OF	ning an Unet to KadriLucz, cristerkov FFCC FFCC AND AND AND AND AND AND AND AND AND AND	Ŷ

next, you can analyst the code

CPU - main thread, module warmup		
COLONIF32 E9 DB E9 00401F33 F4 DB A9 00401F33 F4 DB F4 00401F33 FF DB FF 00401F35 C3 DB FF 00401F35 C3 DB FF 00401F35 C3 DB C3 00401F35 C3 DB C3 00401F35 C3 DB C3 00401F41 C5 DB FF 00401F41 25 DB FF 00401F41 A0 DB A0 00401F41 A0 DB A0 00401F41 A0 DB A0 00401F44 A0 DB B0 00401F45 A	Backup Copy Binary Modify byte Assemble Space Label : Comment ; Breakpoint Hit trace Run trace Go to Thread	 Registers (FP) EFR 0019FF20 EFR 0019FF20 EFR 00255500 EFR 00255500 EFR 00255500 EFR 00255500 EFR 00255500 EFR 00406221 (EFR 00406221 (EFR 00406221 (EFR 00406221 (EFR 00406221 (EFR 00406221 (EFR 00406217) C 1 ES 002B C 1 ES 002B C 0 DS 002
00401F55 00 DB 00 00401F57 00 DB 00	Follow in Dump	> Analyse code Ctrl+A
Address Hex dump ASCII 0019FF54 21.62 40.00 21.62 40.00 10.0 10.0 0019FF50 30.FF 19.00 74.FF 19.00 74.F	Search for Find references to View	Remove analysis from module Scan object files Ctrl+O Remove object scan from module
0019FF6C 21 62 40 00 CC FF 19 00 tb(F+. 0019FF74 29 FA 15 77 00 50 25 00) sw.P%.	Copy to executable	> Remove analysis from selection BkSpc
0019FF7C110 FA 15 77 DC FF 19 00 ▶ Swm ↓. 0019FF84 9E 7A 6E 77 00 50 25 00 Aznw.PX. 0019FF8C B1 B1 B3 32 00 00 00 00 00 %%[]2	Analysis	During next analysis, treat selection as
0819FF94 00 00 00 00 00 00 25 00 ***************	Appearance	 > 001 9F 720 0003 0900 001 9F 950 0003 0000 001 9F FFA 0030 0000 001 9F FFA

It's unconditional JMP, it will jump to address warmup.004013E5

00401537	XE9 09545555	IMP uppmup 004012E5	
00401F3C	C8		^
00401F3D	08	DB 08	
00401F3E	00	DB 00	
00401F3F	. 00	DB 00	
00401F40	\$-FF25 80604000	3 JMP DWORD PTR DS:[<&KERNEL32.GetModule]	KERNEL32.GetModuleHandleH
00401F46	>-FF25 H4604006	JUNE DWORD FIR DS:LK&RERNEL32.GetFrochd	n KERNEL32.GetProchddress
00401F4C	00		
00401F4F	ññ	I DB 00	
00401F4F	йй	DB 00	
00401F50	00	DB 00	
00401F51	00	DB 00	
00401F52	00	DB 00	
00401F53	90	DB 00	
00401F54	99		
00401F56	ÃÃ	DB 00	
00401F57	ด้ดี	DB 00	
00401F58	00	DB 00	
00401F59	00	DB 00	
00401F5A	00	DB 00	
00401F5B	00	DB 00	
00401F5C	99	DB 00	
00401F5D	00	DB 00	
00401F5E	ññ	DB 00	
00401F60	00	DB 00	
00401F61	00	DB 00	
00401F62	00	DB 00	
00401F63	00	DB 00	
00401F64	99	DB 00	· · · · · · · · · · · · · · · · · · ·
00401765	00		
Jump from	-warmup.004013E5 00401F14		
Camp 110P	1 00101211		

Check the address, make sure its on the right place

CPU - main thread, module warmup	
Constraint Constra	Registers (FPU) C <thc< th=""> C <thc< th=""> <</thc<></thc<>
Jump From 00401F37 Inderses Hex dump BSCII 0019FF54 22 42 40 800 21 62 44 000 1888 108.	RETURN to KERKELSZ.7715FR29
001977-02 00197-02 001	D EEPELS2.BaseThreadInitThunk E EETURH to ntdll.776E7A9E

We can use OllyDumpEx to Dump the process

<u>F</u> ile	<u>V</u> iew	<u>D</u> ebug	<u>P</u> lugins	Op <u>t</u> ions	<u>W</u> indow	<u>H</u> elp						
	44 ×	► II	1 B	ookmarks	>	E M	T W H	C / K B R	S	≣∎?		
C	CPU - n	agin three	2 C	ommand li	ine >				_			
00	4013E5	> E8 C503	3 C	llyDumpEx	د >	Du	ump prod	ess				~
00	14013EF 14013EF 14013F0 14013F2 14013F4 14013F4	.~E9 74FE \$ 55 • 8BEC • 6A 00 • FF15 04 • FF75 08	1204000 CA	ISH EBP IV EBP,ESP ISH Ø ILL DWORD P ISH DWORD P	TR DS: [4020	Pli	ugin deb	ug toggle	= NULL	ilter		
	440135FH 1401403 1401408 1401408 1401405 1401415 1401415 1401415 1401417 1401418 1401418 1401422 1401422 1401422 1401422 1401422 1401421	- FF15 20 - FF15 20 - 68 0904 - FF15 00 - 50 - 50 - 55 - 88EC 24 - 81EC 24 - 85C0 - 85C0 - 85C0 - 85C0 - 850 - 64 02 - 59 - 02 29 - 02 1921 - 1921	1204000 CA 100C0 PU 1204000 CA 1204000 CA PU 1204000 CA PU 1030000 SU 1030000 SU 1030000 SU 1030000 SU 10000 SU	IST DWORD P ISH C2020640 P ISH EAX ILL DWORD P ISH EAX ILL DWORD P IP EBP ISH EBP ISH EBP ISH EBP ISH EAX, EAX ISH 17 ISH EBY, SAU ISH 17 ISH CAX ISH 29 ISH 29 IN ECX II DWORD P IN ECX	TR DS: [4020 9 TR DS: [4020 TR DS: [4020 TR DS: [4020 mup.0040143	AL 9081 90001 9101 91	Jour	ExitCode = C0006 GetCurrentFroce hFrocess TerminateProcess KERNEL32. IsProce	onFilte 0409 (-: ess s essorFea	er 1873740791.) aturePresent		
00 00 00 00 00 00 00	401436 140143C 1401442 1401448 1401448 1401448 1401458 1401458 1401458	. 890D 14 . 8915 10 . 891D 00 . 8935 08 . 893D 04 . 66:8C15 . 66:8C0D . 66:8C10	314000 MO 314000 MO 314000 MO 314000 MO 314000 MO 5 303140 MO 5 303140 MO 0 243140 MO 0 203140 MO	U DWORD PT U DWORD PT U DWORD PT U DWORD PT U DWORD PT U WORD PTR U WORD PTR U WORD PTR U WORD PTR	R DS: [4031] R DS: [4031] R DS: [40310 R DS: [40310 DS: [403130 DS: [403130 DS: [403100 DS: [403100	41,ECX (01,EDX (01,EDX (01,EBX (01,EBX (01,EBX (01,EBX (01,EBX (01,EBX (01,EBX (01,EBX (01,EDX (01,EDX (01,ECX (01,ECX (01,ECX (01,ECX (01,ECX (01,ECX (01,EDX) (01,EDX (01,EDX) (01,EX)						<

Copy the Entry point before and put it somewhere note, next click to get the current EIP and dump it

Save the file(WarmUp_Dump.exe).

List Section: © Base Only © All Memory © Address Range 00400000 - 03	03400000 File Home Share View
Dump Mode: • Rebuild O Binary (Raw) O Binary (Virtual) Header Source: • Memory • File C Dummy PE32 EXE • Res	ReScan Image 🛛 🗧 🔶 👻 🕇 🧧 > CYDES2023 > Reverse Engineering > Warm-Up-Writeup
Search	
Search Mode: OUMP OK	
Search Result:	
Image Base: 004(ES2023\Reverse Engineering\Warm-Up-Writeup\WarmUp_Du	
Image Size: 0000 Saved Size: 32768 Requested Size: 32768	• warmup.exe warmup.zip warmup_bump .exe
Entry Point: 0000 Header	
Section Finish Back to Menu	
Address Size Owner Section Type Access VirtualOffset V	VitualSize Char

Get the current EIP and click IAT Auto Search to automatically find the Import Address Table of the executable. After that click "Get Imports" to get a list of the imports that the

executable has. Choose the dump file and fix it.

		4	Import REConstructor v1.7e FINAL (C) 2001-2010 MackT/uCF -	
CPU - main thread, module warmup			Attach to an Active Process	
004013E5 > E8 C5030000 CALL warmup.004017AF 004013EH > E9 74FEFFFF JMP warmup.00401263			c:\users desktop\cydes2023\reverse engineering\warm-up-writeup\warmup.exe	Pick DLL
004013EF \$ 55 PUSH EBP 004013F2 . 8BEC MOV EBP,ESP 004013F2 . 6R 00 PUSH 0		Б	Imported Functions Found	
004013F4 . FF15 04204000 CALL DWORD PTR DS: [402004] 004013FA . FF75 08 PUSH DWORD PTR SS: [EBP+8] 004013FD . FF15 20204000 CALL DWORD PTR DS: [402020]				Show Invalid
00401403 . 68 090400C8 PUSH C0000409 00401408 . FF15 08204000 CALL DWORD PTR DS:[402008]				Show Suspect
0040140F . FF15 0C204000 CALL DWORD PTR DS: [40200C] 00401415 . 5D POP EBP				
00401415 - C3 REIN 00401417 > 55 PUSH EBR 00401418 - 88EC MOV EBP, 55P				Auto Trace
0040141A . 81EC 24030000 SUB ESP,324 00401420 . 6A 17 PUSH 17 00401422 . FF15 10204000 CALL DWORD PTR DS:[402010]	Ц			
00401428 . 85C0 TEST EAX,EAX 0040142A74 05 JE SHORT warmup.00401431 0440142C 64 02 PUBL 2				Clear Imports
0040142E · 59 POP ECX 0040142F · CD 29 INT 29			Log	
00401431 > A3 18314000 MOV DWORD PTR DS:[403118],EAX 00401436 . 890D 14314000 MOV DWORD PTR DS:[403114],ECX			Module loaded: c:\windows\system32\kernelbase.dll	
0040143C . 8915 10314000 MOV DWORD PTR DS:[403110],EDX 00401442 . 891D 0C314000 MOV DWORD PTR DS:[40310C],EBX			Module loaded: c:\windows\system32\apphelp.dll	ClearLog
00401448 . 8935 08314000 MOV DWORD PTR DS:[403108],ESI 0040144E . 893D 04314000 MOV DWORD PTR DS:[403104],EDI	4		Module loaded: c: \windows\system32\ucrtase.ali Module loaded: c:\windows\system32\ucrtase.ali	
00401454 . 66:8C15 30314(MOV WORD PTR DS:[403130],SS 00401458 . 66:8C0D 24314(MOV WORD PTR DS:[403124].CS	\searrow		Getting associated modules done.	
00401462 . 66:8C1D 00314(MOV WORD PTR DS:[403100],DS		\mathbf{i}	Image Base:00400000 Size:00008000	
Jump from 00401F37				Options
		1	IAT Infos needed New Import Infos (IID+ASLII+LUADER)	
Address Hex dump ASCII 0019FF54 21 62 40 00 21 62 40 00 *b@,*b@,	F	-	OEP 000013E5 [IAT AutoSearch] 2/VA 00000000 Size 00000000	
0019FF5C 80 FF 19 00 74 FF 19 00 C +.t +. 0019FF64 00 E0 3A 00 21 62 40 00 .~:.!b@.			RVA 00000000 Size 00001000	About
0019FF6C 21 62 40 00 CC FF 19 00 *b@.⊫ ↓. 0019FF74 29 FA 15 77 00 E0 3A 00) Sw.«:.			→ → → → → → → → → → → → → → → → → → →	Exit
0019FF7C110 FA 15 77 DC FF 19 00 ▶ Swm ↓. 0019FF84 9E 7A 6E 77 00 E0 3A 00 №znw.«:. 0019FF8C 88 E0 3D 81 00 00 00 00 & «=:			Load Tree Get Imports 3	

This is the unpacked version, look at the size. Rename it to make things easier.



Now everything looks clear on the strings, we succesfully unpacked the sample.



Next, we just need to crack the software and get the flag.

Run and go to breakpoint software



Go to the false instruction address

🔟 CPU 📝 Log 📋 Notes 🔹 Breakpoints 📟 Memory Map 📋 Call Stack 🗠 SEH 😥 Script 🔮 Symbols 🛇 Source	re 🔎 References 🖙 Threads 💼 Handles 🦸 Trace
X All Modules (Strings) 🛛	
Address Disassembly	String
004010A6 push unpacked.402108	"Hi, here for some warmup?\n\n"
00401080 push unpacked.402124	"Please enter the flag\n>"
004010CB movups xmm0,xmmword ptr ds:[402140]	"d~cbt 31?eab5ec7c4>f>17e7afae4c6b4?>bbz"
004010E2 movups xmm0, xmmword ptr ds:[402150]	"C4>T>1/e/aTae4C6D4?>DDZ"
00401060 move adv uppacked 400160	UNICA WARMUNI Lat's dat startad!"
00401160 mov eax.unpacked.402188	"Uh-oh, bad warmup can cause injury."
004011D5 push unpacked.401A67	"5\%\\&@"
004011FE push unpacked.4017FA	"3AA3A@A, "
0040165C cmp word ptr ds:[400000],ax	"MZ@"
0040166A cmp dword ptr ds:[e8x+400000],4550	"MZ®"
7495157 mov ecx, unpacked, 40000	MAC WAR AND A CONTRACT OF A CO
749F4011 cm dword ptr ds:[<& DestructExceptionObject5].0	"Daytasyt * myt "
749F401A push <vcruntime140. destructexceptionobject=""></vcruntime140.>	"OaŸt@%Ÿt *mŶt"
749F402D mov esi,dword ptr ds:[<&DestructExceptionObject>]	"OaŸt@%Ÿt'mŸt"
749F4180 cmp dword ptr ds:[<&DestructExceptionObject>],0	"Oaÿt@%ÿt°mÿt"
749F4189 push <vcruntime140.&destructexceptionobject></vcruntime140.&destructexceptionobject>	"Oayt@%yt°myt"
749F419A moves, aword ptr ds: [<a_bestructexceptonobject>]</a_bestructexceptonobject>	U attest int
749F453D rea cax, dword ptr ds. [ccx*8+7400196]	 "DG&"
749F6349 push vcruntime140.74A01034	"a#Yt"
749F67AA push vcruntime140.74A01050	"ā#Ÿt"
749FGD1E mov dword ptr ds:[esi],vcruntime140.749F1268	"°mŸt"
749F6D35 mov dword ptr ds:[ecx+4],vcruntime140.749F1270	"bad exception"
749F603C mov dword ptr ds: [ecc], vcruntime140, 749F1268	"TMYL" # a mode #
749E6083 mov dword ptr dst cert vcruntime140 749E1248	" • mV+ "
749F60B9 mov dword ptr ds:[esi].vcruntime140.749F1248	"'mŶt"
749F7207 mov eax,vcruntime140.749F1250	"Unknown exception"
	10.000

The comparison will be happen on the top

E CPU	🗋 Log 📋	Notes	Breakpoints	Memory Map	🛛 Call Stack 🛛 🗠 📆	SEH 🗾 Script	🐏 Symbols	<> Source	e 🖉 References	🛸 Threads	晶 Handles	🐔 Tra
	1 •	0040114D	3A51 01	cmp	dl, byte ptr ds	[ecx+1]						~
		00401150	✓ 75 OE	Jne	unpacked.40116	o						
		00401152	83C0 02	add	eax,2			"bit woon"				
		00401155	8402	auu	ecx,2		ecx:	EA \XUS				
		00401158	0 7E E 4	ina	uppacked 40114	0						
		00401150	3309	xor	ecx.ecx	<u>v</u>	ecx:	"èå\x03"				
		0040115E	FB_05	imp	unpacked, 40116	5	- cent	0.1(100				
		00401160	1BC 9	sbb	ecx.ecx	-	ecx:	"èÂ\x03"				
		00401162	83C9 01	or e	ecx,i		ecx:	"èÂ\x03"				
	└→ ₀	00401165	85C9	test	ecx,ecx		ecx:	"eA\x03"				
	•	00401167	BA 682140	000 mov	edx, unpacked. 4	02168	edx:	"èÅ\x03",	402168:"Nice war	mup! Let's g	et started!"	
	•	0040116C	B8 882140	000 mov	eax, unpacked. 4	02188	4021	.88:"Uh-oh,	bad warmup can	cause injury	- " · · · · · · · · · · · · · · · · · ·	
	•	00401171	0F44C2	cmov	/e eax,edx		edx:	"eA\x03"				
	•	00401174	50	push	eax							
	•	00401175	E8 AGFEFF	FF Call	unpacked. 4010	20						
		004011/A	884D FC	mov	ecx, aword ptr	ss:[epp-4]						
		00401170	3304 04	adu	esp,4							
		00401180	2200	XOI	eax eax							
		00401182	E8 070000		unpacked 4011	90						
		00401104	200700000		b-	<u></u>						*
	-	<										>
ecx=0040:	13E5 "èÅ\x03"											

.MPRESS1:00401165 unpacked.exe:\$1165 #1165

Do analyze the instruction carefully

🕷 Unpacked.exe - PID: 3468 - Modu	ile: unpacked.exe - Thread: M	ain Thread 2764 - x32dbg [Elevated]					
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)							
😑 😏 🔳 🔿 🖩 😤 🎭 🎍 💲 🕰 🔳 🥜 😓 🕢 🎵 🌮 🕼 📕 🖉 🖉 🖉 🖉							
🖾 CPU 🚺 Log 🖺 Notes 🔮	Breakpoints 🛛 🛲 Memory Mag	💿 🗐 Call Stack 🛛 🛒 SEH 🛛 🖸 Script 🔗	Symbols 🗢 Source	🖉 References 🛛 🛸 Thread	s 🔒 Handles	🐔 Tra	
	68 08214000 E8 70FFFF 68 24214000 E8 60FFFFF 80 85 70FFFFF 50 68 3C214000 E8 85FFFF 0F1005 40214000 83C4 10 33C0 0F2800 80214000 0F145 C0 0F2800 80214000 0F1145 C0 0F1005 50214000 0F1145 C0 0F10445 C0 66:0FD645 E0 66:0FD645 C0 66:0FD645 C0 66:0FEFC1	push unpacked.402108 call unpacked.402124 call unpacked.402124 call unpacked.402124 call unpacked.402132 call unpacked.402132 call unpacked.402132 call unpacked.402132 call unpacked.402132 call unpacked.402130 movups xmmo, xmmword ptr ds:[402140] add esp.10 xor eax, eax movups xmmuord ptr ds:[402180] movups xmmuord ptr ds:[402180] movups xmmuord ptr ds:[402180] movq xmm0, xmmword ptr ds:[402180] movq xmd, qword ptr ss:[ebp-30], xmm0 movq xmd, qword ptr ds:[402180] movq xmm0, xmmword ptr ds:[402180] movq xmd, qword ptr ds:[402180] movq xmd, trd ss:[ebp-30], xmm0 nop word ptr ds:[eax+eax], ax movups xmm0, xmmvord ptr ss:[ebp+eax-40 pxor xmm0, xmm1	402108:"H1, here 402124:"Please er 00402140:"d~cbt : 00402150:"c4>f>17 00402160:"b47>bbz	for some warmup?\n\n" iter the flag\n>" 81?eab5ec7c4>f>17e7afae4 ?e7afae4c6b4?>bbz" "	IC6b4?>bbz"	~	
• 00401109	0114405 98	movups xmmword ptr ss:[ebp+eax-68],xmm	0			> [×]	

The false will break at this address, keep step over.

Elle <u>Vi</u> ew <u>D</u> ebug Tracing Plugins Favourites Options <u>H</u> elp Apr 17 2021 (TitanEngine)	
😑 😇 🔳 🜩 🖩 🍷 🚓 🛬 🎍 ‡ 🕰 🚺 🖉 😓 🕢 🌮 fx # Az 🖺 🗮 💇	
🔟 CPU 📝 Log 📋 Notes 🔹 Breakpoints 💻 Memory Map 🧊 Call Stack 🧠 SEH 🐻 Script 🍳 Symbols ᄾ Source 🖉 References 🐄 Threads 📥 Handl	s 👔 Trac
12 00401007 00401007 00401007 00401007 00401007 00401007 00401002 832(4 10 332(0) 0071145 C0 0040102 0040010 000 000 000 000 000 000 000 000	d.exe

You will find instructor that compare between ECX and EAX. This is where comparison is made.

W Unpacked.exe - PID: 5884 - Module: unpacked.exe - Thread: N	lain Thread 1908 - x32dbg [Elevated]		- 0 ×
	🥠 fx # A1 🆺 🗐 👮		
💹 CPU 📝 Log 📋 Notes 🔹 Breakpoints 📟 Memory Ma	p 🧊 Call Stack 🧠 SEH 🗵 Script	length Symbols ◇ Source 🖉 References 🎐 Threads 🐇 Handles 🦿	Trace
EIP 00401140 8A10 00401142 3A11	mov dl,byte ptr ds:[eax]	eax: "cookies" ecx: "cydes(468bfe2bd0d39a960b0fafb3d1e389ee}"	A Hide FPU
00401146 47 25 1A 00401146 442 00401146 74 12 00401140 74 12 00401150 851 01 00401150 850 02 00401152 850 02 00401152 850 02 00401155 850 02 0040155 850 02 0040155 850 02 0040155 850 02 0040155 850 02 0040155 850 02 0040155 850 02	<pre>ime unpacked.40150 IM unpacked.40156 Um unpacked.40156 Um unpacked.40156 Um unpacked.40156 Um unpacked.40156 add eax.2 add d1.2 mm unpacked.401160 Som exx.2 mm unpacked.401165 Som exx.2 Som e</pre>	<pre>exx:1:"okites" exx:1:"ydes(468bfe2bdod39a960b0fafb3d1e389ee)" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee)" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960b0fafb3d1e389ee]" exx:"cydes(468bfe2bd0d39a960bfe4bd3d3a960bfe4bfe3d1e389ee]" exx:"cydes(468bfe2bd0d39a960bfe4bfe3d1e389ee]" exx:"cydes(468bfe2bd0d39a960bfe4bfe3d1e389ee]" exx:"cydes(468bfe2bd0d39a960bfe4bfe4bd4d39a960bfe4bfe4bfe4bfe4bfe4bfe4bfe4bfe4bfe4bfe4</pre>	FAX 0019FE90 "cookies" ∧ EBX 0019FE88 "cydes [468bfe2b] EDX 0000001 EBF 0019FE80 "cookies" ESP 0019FE30 "cookies" EC ESP 0019FE30 "cookies" ESF ESF <t< td=""></t<>
• 00401174 50 • <	push eax	eax:"cookies"	✓ Default (stdcall) ▼ 5 ↓ Unlocked 1: [esp+4] 00736569 2: [esp+8] 27530EF3 ucrtbase.77530E^
d]=1 byte ptr ds:[eax]=[0019FE90 "cookies"]=63 'c' .MPRESS1:00401140 unpacked.exe:\$1140 #1140			3: [esp+C] 0000000 4: [esp+10] 006AA72F "«««««««««" 5: [esp+14] 0000000

That's the Flag.



Ok challenge. but I'm surprised it only got 2 solves! I think rev/pwn tend to scare people away sadly.

```
cydes{468bfe2bd0d39a960b0fafb3d1e389ee}
```

Power of Rewind

This one got the code, i just brute force(AI) the code until i get the flag, Because i have no time to waste.

```
$base64String =
"FgJqAMKJ5ePgsWMLneXHLrXKhmjNwCYUDCpD3u8sbiT8sEJ9M1GmdzrYkXP64PYv"
$encryptedBytes = [System.Convert]::FromBase64String($base64String)
$key = 145,96,34,150,165,222,211,99,165,119,17,98,225,14,249,255
$iv = 251,122,202,111,165,48,247,134,32,88,101,199,33,154,190,56
$aes = New-Object System.Security.Cryptography.RijndaelManaged
$aes.Mode = [System.Security.Cryptography.RijndaelManaged
$aes.Mode = [System.Security.Cryptography.PaddingMode]::PKCS7
$aes.Fadding = [System.Security.Cryptography.PaddingMode]::PKCS7
$aes.Key = $key
$aes.IV = $iv
$decryptor = $aes.CreateDecryptor()
$decryptedBytes = $decryptor.TransformFinalBlock($encryptedBytes, 0,
$encryptedBytes.Length)
$decryptedString = [System.Text.Encoding]::UTF8.GetString($decryptedBytes)
$decryptedString
```

cydes{ce65c25c5bd0fa669bd3bdef7aa9bdac}