

Cátedra de Ciberseguridad CiberUGR, INCIBE-UGR UGR CTF 2024 by jtsec



### CÁTEDRA DE CIBERSEGURIDAD CIBERUGR, INCIBE-UGR

Nombre	Protected File
Categoría	MISC (CRIPTO + FORENSE)
Dificultad	MEDIA
Puntos	300

#### **DESCRIPCIÓN DEL RETO**

Un cliente nos ha enviado un archivo protegido con contraseña, pero no nos han proporcionado la clave para ver el contenido de este. ¿Serás capaz de recuperar el contenido que se encuentra dentro del archivo?

#### WRITEUP

- 1. Inicialmente, al leer la descripción del reto se nos proporciona un fichero adjunto. Este fichero se trata de un zip con contraseña.
- 2. Para ello, vamos a intentar conseguir la contraseña a través de ataques de fuerza bruta, usando "zip2john" para sacar el hash del zip y "john" para crackearlo.
- 3. Sacamos el hash del zip usando zip2john.

*zip2john Protected File.zip > hash.txt* 

s zip2john Protected\_File.zip > hash.txt ver 2.0 efh 5455 efh 7875 Protected\_File.zip/file\_transfer\_protoco l.pcapng PKZIP Encr: TS\_chk, cmplen=43574, decmplen=115656, crc=17 25AD31 ts=5A2B cs=5a2b type=8

4. Usamos la herramienta "John the Ripper" y el diccionario "rockyou.txt" para crackear la contraseña de este zip.

john hash.txt -wordlist=/usr/share/wordlists/rockyou.txt



















UGR CTF 2024 by jtsec

-\$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt Using default input encoding: UTF-8 Loaded 1 password hash (PKZIP [32/64]) Will run 8 OpenMP threads Press 'q' or Ctrl-C to abort, almost any other key for status (Protected\_File.zip/file\_transfer\_protocol.pcapng !!!secret!!! 1g 0:00:00:00 DONE (2024-04-24 17:35) 1.923g/s 27583Kp/s 27583Kc/s 27583KC/s "2parrow" .. \*7; Vamos! Use the "--show" option to display all of the cracked passwords re liably Session completed.

5. Ahora que tenemos la contraseña "!!!secret!!!" vamos a descomprimir el archivo. NOTA: Es necesario escapar los caracteres especiales como en este caso son todas las exclamaciones.

unzip -P \!\!\!secret\!\!\! Protected File.zip

-\$ unzip -P \!\!\!secret\!\!\! Protected\_File.zip Archive: Protected\_File.zip inflating: file\_transfer\_protocol.pcapng

- 6. Una vez hemos descomprimido el zip, vamos a ver qué es lo que contiene dentro.
- 7. En este caso vemos un archivo pcap, el cual se trata de una captura de tráfico.
- 8. Abrimos el pcap usando "Wireshark".

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> aptu	re <u>A</u> nalyze <u>S</u> tatistics Telepho	ny <u>W</u> ireless <u>T</u> ools <u>H</u> elp	
	🖹 🙆 Q K → A K	» 🌉 🔳 🛛 🖬 🖬	
Apply a display filter <ctr< td=""><td>rl-/&gt;</td><td></td><td></td></ctr<>	rl-/>		
No. Time	Source Des	tination Protocol	Length Info
284 94.829691052	::1 ::1	FTP	108 Request: EPRT [2]::1[50269]
285 94.829808751	::1 ::1	FTP	139 Response: 200 EPRT command successful. Consider using EPSV.
286 94.829875502	::1 ::1	FTP	94 Request: NLST
287 94.830085179	::1 ::1	TCP	96 20 - 50269 [SYN] Seq=0 Win=65476 Len=0 MSS=65476 SACK_PERM TSval=310154
288 94.830119519	::1 ::1	TCP	96 50269 - 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65476 SACK_PERM 1
289 94.830129568	::1 ::1	TCP	88 20 → 50269 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3101549314 TSecr=310
290 94.830174257	::1 ::1	FTP	127 Response: 150 Here comes the directory listing.
291 94.830195104	::1 ::1	FTP-DA	100 FTP Data: 12 bytes (EPRT) (NLST)
292 94.830197131	::1 ::1	TCP	88 50269 - 20 [ACK] Seq=1 Ack=13 Win=65536 Len=0 TSval=3101549314 TSecr=31
293 94.830205405	::1 ::1	TCP	88 20 - 50269 [FIN, ACK] Seq=13 Ack=1 Win=65536 Len=0 TSval=3101549314 TSe
294 94.830344034	::1 ::1	TCP	88 50269 - 20 [FIN, ACK] Seq=1 Ack=14 Win=65536 Len=0 TSval=3101549314 TSe
295 94.830357788	::1 ::1	TCP	88 20 - 50269 [ACK] Seq=14 Ack=2 Win=65536 Len=0 TSval=3101549314 TSecr=31
296 94.830388753	::1 ::1	FTP	112 Response: 226 Directory send OK.
297 94.830400456	::1 ::1	TCP	88 54590 → 21 [ACK] Seq=304 Ack=1200 Win=65536 Len=0 TSval=3101549314 TSec
298 95.437275080	::1 ::1	FTP	96 Request: TYPE I
299 95.437348765	::1 ::1	FTP	119 Response: 200 Switching to Binary mode.
300 95.437422794		FIP	105 Request: SIZE secret.png
301 95.437447420	::1 ::1	FTP	99 Response: 213 442/6
1 302 95.437572989	::1 ::1	FTP	108 Request: EPRI [2]::1[52489]
303 95.437642964	.:1 .:1	FIP	139 Response: 200 EPRI command successful. Consider using EPSV.
304 95.437736274		FIP	105 Request: REIR secret.png
305 95.437987335		TOP	96 20 - 52409 [STN] Seq=0 WIN=554/6 Len=0 MS5=654/6 SACK_PERM ISVAL=310154
306 95.437995774	111	TOP	96 52469 - 26 [STN, ACK] SEG=0 ACK=1 WIN=65535 LEN=0 MS5=65476 SACK_PERM 1
307 95.438003493		TCP	00 20 → 52409 [ACK] Sed=1 ACK=1 WIN=05330 Len=0 15Val=3101549922 [SecI=310
300 95.430061298		ETP-DA	159 Response: 150 Opening BINARY mode data connection for secret.png (442/0
<ul> <li>Frame 309: 32855 bytt</li> <li>Linux cooked capture</li> <li>Internet Protocol Vei</li> <li>Transmission Control</li> <li>FTP Data (32767 byte)</li> <li>[Setup frame: 3027</li> <li>[Setup method: EPRT]</li> <li>[Command frame: 304</li> <li>[Current working direction]</li> </ul>	es on wire (262840 bits), v1 Protocol, Src: ::1, Dst: : Protocol, Src Port: 20, s data) t.png] ectory: ]	32855 bytes captured (20 :1 Dst Port: 52489, Seq: 1,	2840 bits) on interface any, id 0       0000       00 00



GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA













- UGR CTF 2024 by jtsec
- 9. Al abrir la captura de tráfico, se ve bastante tráfico FTP, por lo tanto, es necesario hacerle un análisis para identificar que es lo que está pasando en esta comunicación. NOTA: Para analizarlo, podemos seguir el flujo TCP.

										file_transfer_prot	ocol.pcapng
Fil	e <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> ar	pture <u>A</u> nalyze <u>S</u> t	atistics Telephony <u>W</u> ireless <u>T</u> oo	ıls <u>H</u> elp							
	[□⊿⊚ ▫ ▮	🗎 🔀 🙆 🔍	e > n .e > 📑 📕								
	Apply a display filter <	Ctrl-/>									
No.	. Time	Source	Destination	Protocol	Length Info						
Г	1 0.000000000	::1	::1	TCP	96 40004 → 21 [SYN] Seq=0 Win=6	5535 Len=0 MSS=65476 SAC	CK_PERM 1	TSval=3	101454484 TSecr=	0 WS=4	and the second second
	2 0.00008051	::1	::1	TCP	96 21 → 40004 [SYN, ACK] Seq=0	Ack=1 Win=65464 Len=0 MS	SS=65476	SACK_P	ERM TSval=310145	4484 TSecr=3101	454484 WS=128
	3 0.000014383	::1	::1	TCP	88 40004 → 21 [ACK] Seq=1 Ack=1	. Win=65536 Len=0 TSval=3	310145448	84 TSec	r=3101454484		
	4 0.001288002	111	::1	FTP	108 Response: 220 (VSFIPO 3.0.3)	H Win-REE26 Lon-0 TOwnla	-2101454	405 TC-	0 F = 2101 4E 440E		
	5 0.001297402	::1	::1	ETP	98 Request: USER off	1 WIN=05530 Len-0 TSVat-	31014344	485 136	CF=3101454465		
	7 0.001414915	111	::1	TCP	88 21 → 40004 [ACK] Seg=21 Ack	Mark/Unmark Packet	Ctrl+M	1	ecr=3101454485		
	8 0.001493646	::1	::1	FTP	122 Response: 331 Please specif	Ignore/Unignore Packet	Ctrl+D				
	9 0.041999057	::1	::1	TCP	88 40004 - 21 [ACK] Seq=11 Ack	Cot lines Time Deferment	CHUT		ecr=3101454485		
	10 4.300352919	::1	::1	FTP	113 Request: PASS Badpasswordje	Set/Unset Time Reference	Cutter				
	11 4.345959544	::1	::1	TCP	88 21 → 40004 [ACK] Seq=55 Ack	Time Shift	Ctrl+Si	hift+T	ecr=3101458784		
1	12 4.649366982	10.0.35.104	239.255.255.250	SSDP	219 M-SEARCH * HTTP/1.1	Packet Comments					
1	13 5.001212039	10.0.35.104	239.255.255.250	SSUP	219 M-SEARCH * HTTP/1.1	C dia Descrito de Norma					
	14 6.000240411	10.0.35.104	239.200.200.200	SSUP	110 Response: 520 Login incorre	Edit Resolved Name					
	16 7.130453453			TCP	88 48994 - 21 [ACK] Seg=36 ACK	Apply as Filter			ecr=3101461614		
	17 7.662242376	10.0.35.104	239.255.255.250	SSDP	219 M-SEARCH * HTTP/1.1	Propage as Filter					
1	18 21.497738807	::1	::1	FTP	94 Request: QUIT	Prepare as Filter					
	19 21.497752253	1 ::1	::1	TCP	88 21 → 40004 [ACK] Seq=77 Ack	Conversation Filter			ecr=3101475981		
	20 21.497819663	. ::1	::1	FTP	102 Response: 221 Goodbye.	Colorize Conversation					
	21 21.497827578	:::1	::1	TCP	88 40004 - 21 [ACK] Seq=42 Ack	5070			ecr=3101475981		
	22 21.497834601	::1	::1	TCP	88 21 - 40004 [FIN, ACK] Seq=	SLIP		_	121 TCoor-2101475	0.9.1	
	23 21.497858777	111	::1	TCP	88 40004 - 21 [FIN, ACK] Seq=	Follow		18	TCP Stream	Ctrl+Alt+Shift+T	
	24 21.497050100	::1		TCP	88 21 → 40004 [ACK] Seq=52 ACT	Copy		,	UDP Stream	Ctrl+Alt+Shift+U	
	26 23 107532511		111	TCP	96 21 - 41696 [SYN ACK] Seg=6	cop)					77591 WS=128
	27 23.107538994	4 111	::1	TCP	88 41606 → 21 [ACK] Seg=1 Ack=	Protocol Preferences					11001 110 100
	28 23.108636553	1 ::1	::1	FTP	108 Response: 220 (vsFTPd 3.0.3	Decode As					
	Frame 6: 98 bytes c	on wire (784 bi	ts), 98 bytes captured (784	bits) on	interface any, id 0	Show Packet in New Window					30 00 86 dd
	Linux cooked captur	e v1				Silow Packet in their damage		00 01			30 00 00 00
1.3	Internet Protocol V	Version 6, Src:	::1, Dst: ::1					00 00			30 00 00 00
	Transmission Contro	l Protocol, Sr	c Port: 40004, Dst Port: 21	., Seq: 1,	Ack: 21, Len: 10			00 00			j2 8a ce a0
•	File Transfer Proto	col (FTP)						4C 9,	SIP Call		31 01 08 0a L
	[Current working di	rectory: ]						Od Ga			20 03 74 00
								00 00			

10. En el primer paquete, se ve como el usuario "ctf" se ha intentado autenticar con una contraseña invalida.



11. Para seguir viendo las comunicaciones podemos ir aumentando el valor de "stream".



















UGR CTF 2024 by jtsec

<u> </u>	ireshark · Follow TCP Stream (t	cp.stream eq 1	) • file_transfer_pro	otocol.pcapng		0 0 8
220 (vsFTPd 3.0 USER ctf 331 Please spec PASS esta mal 530 Login inco QUIT 221 Goodbye.	0.3) cify the password. la contrasea? En rrect.	serio?				
3 <mark>client</mark> pkts, 4 <mark>server</mark> pkt Entire conversation Find:	s, 6 turns. (146 bytes) * Sho	ow data as	ASCII		• St	ream 1 🛟 Find <u>N</u> ext
	Filter Out This Stream	Print	Save as	Back	Close	Help

12. En el flujo 16 podemos observar como el usuario ha listado un directorio en el que se encuentra un PNG con el nombre de secret.png.

<b>4</b> W	Vireshark · Follow TCP	Stream (tcp.stream e	16) - file_transfer_protocol	pcapng	• • 8
-rw-rr	1 1001 10	001 44	276 Mar 22 10:08	secret.png	
1 <mark>client</mark> pkt, 0 <mark>server</mark> pkt	ts, 0 turns.				
Entire conversation	n (68 bytes) 👻	Show data as	ASCII	- Stream	n 16 🗘
Find:				Fir	nd <u>N</u> ext
	Filter Out This	Stream Print	Save as Ba	ck Close	Help

















UGR CTF 2024 by jtsec

13. Si seguimos aumentando el "stream" hasta el número 18, se puede observar la estructura de un PNG, el cual tiene todo el sentido que sea el PNG que se ha listado anteriormente, secret.png.

🚄 Wi	reshark · Follow TCP !	Stream (tcp.stream	eq 18) · file_transfer_	_protocol.pcapng	008
. PNG					
IHDR5		ATx^	% /	!K"KR.i.	.}0\${e-
	@@(	\$!9	буб		
.DU.PsT8					.j@@
.@@@					
	···e····e····e		U.B.Q.g		
	@.	.k@	000.		.000.
@^@.9	L@@		.@@@	9	@5S
			f@		@@
000	.@	.q@@.		.000	
.@5					.QU~6
				)	
3 client pkts, 0 server pkt	s, 0 turns.				
Entire conversation	(44 kB) 👻	Show data as	ASCII		Stream 18 🗘
Find:					Find Next
Find:					Find Next
	Filter Out This !	Stream Prir	nt Save as	Back	Close Help

14. Pasamos el contenido de este paquete a raw, para ver si los magic bytes coinciden con los de un PNG.

















15. Comparamos los primeros bytes de la imagen con los magic bytes de un PNG, en este caso coinciden, por lo que ya podemos asumir que esto es un PNG.

Addition	Additional scanning signature analysis								aly	/sis	5						
Possible: [	ΡN	G f	ile,	sig	g: 8	895	504	E4	701	D0/	۹1 <i>/</i>	40 <i>A</i>	]	Cou	nt:	1	
File hex	ch	ara	cte	ers													
[0000000]	89	50	٨F	47	ØD	94	1	94	99	99	99	ØD	19	18	11	52	
[000000016]	00	00	4L 00	20	00	00	00	20	01	00	00	00	49 00	40 5B	01	47	
[00000032]	59	00	00	00	04	67	41	4D	41	00	01	86	A0	31	E8	96	YgAMA1
[00000048]	5F	00	00	00	5B	49	44	41	54	78	9C	2D	сс	B1	09	03	[IDAT×
[00000064]	30	0C	05	D1	EB	D2	04	B2	4A	20	ØB	7A	34	6F	90	15	0Jz4o
[00000080]	ЗC	82	C1	8D	ØA	61	45	07	51	F1	EØ	8A	2F	AA	EA	D2	<ae.q <="" td=""></ae.q>
[00000096]	Α4	84	6C	CE	A9	25	53	06	E7	53	34	57	12	E2	11	B2	1%SS4W
[00000112]	21	BF	4B	26	3D	1B	42	73	25	25	5E	8B	DA	B2	9E	6F	!.K&=.Bs%%^o
[00000128]	6A	CA	30	69	2E	9D	29	61	6E	E9	6F	30	65	FØ	BF	1F	j.0i)an.o0e
[00000144]	10	87	49	2F	DØ	2F	14	C9	00	00	00	00	49	45	4E	44	I/./IEND
[00000160]	AE	42	60	82													
Sector 1	1	- As	sur	ninį	g 51	12	Byte	25 -									

16. Por último, podemos ver la imagen usando CyberChef (https://gchq.github.io/CyberChef/), aplicando el "from Hex" y como sabemos que es una imagen, añadimos el "Render Image", para finalmente poder visualizar la imagen que contiene la *flaq*.













UGR CTF 2024 by jtsec

	Last build: 10 days ago - Version 10 is	s here! Read about the new features here Options 🤹 Abou						
Recipe	~ B 🖿 🖠	Input	+ 🗅 🖯 🕯 🚍					
From Hex Delimiter None	^ ⊗ II	ff43ad4ebd1bbd9093597bf7cde3c272c3fb8dc2c8aefa1f99c5031674547c49 d611d0c6803f9f1889f4798bd61372d37d6593dc2a8ces5df5337078188 ad37bc5bf7a834d6d7cc3bF08c3dbcceffc573df83c9be80229db8 ce6e3dbbba7l0cfe1570e39a292dc28c64b9443cde1dc33fdf737b1bff66 8be4933db84e1bacf755cf01eb3455rf1f843379f81ce5e3c2b063f2d280c7 cd7baf93ff800bac5bbae7b75379c7993f1cf76s2c32bc6db2e7eef1	144645e171706554faea6ac04e5bf7d2dd65c8da212650ac35ba7 3a3e2b652ad3759a659bf5df4ac19d082af25ad1cea9101d3d 7c8a371e9532e74f4faeccf9452be6b806b1b1016047f1aaf ee9ce330849abd67f50e3039f7b851e3356d7d6f7367d2fd4d91 f1e34ab9f3cb3b2139c2dfde58f680bcbcf485f323be6443e9 hear538038fe8ddsf78e63dfdfff7d91c28823a5ec443e9					
Render Image Inout format Raw	^ ⊘ Ⅱ	<pre>f73747b49acdasd3970140104000104108080001410808000a1c6e0bb3555 f366471562268712bd4fcd6f2b7e453525f3abf372857855bb5f3045712 59156f228573570d641770695711547462370deeba156710611565571041 592bf5as24566da08b75215476045310412bd208b5541046058571041 592bf5as24568246da08b7231f75f0454514da5331012bd208b55410f64054 a8b5d17ad5967af2349fasf7d63abf3ad53ac57a0d1cfaf3892cdab7084 a8f515b0372b412b5567t5b604a75c5c2a872aebd3f7ab73a13ec86777be3 a6f515b0372b412b5567t5b604a375ec2a872aebd3f7ab73a13ec86777be3 a6f31490dac266f74c15343724da12c86f122c8fc78ba55880128572c4 8ab88ab2bf74s63476bd24476c433c543e340972345bc155736f3a752 7ab894c48c5e6f34c565478c448e7ab6f2070f4412a6f6d7d287b88 3ae8f729808h436adccac56973dc17ff157b57a1f4fc21a815beffd72bf85 m 65316 ₹ 2 Detext <b>x</b></pre>	410408901041840890104124C28113C20F4C7eC088788607999 202F68977978236C47d3806908063C37d51C8722837C7998955 39059504598277d3284F1101C398077429348C285785476F1518480 9905950420479782592C0332255540056622850b3c04599561 3060ceef779479479451802804064317521377767588980 30264757142942915028254096046317521377767588980 30264757142942915028254096046317521377767588980 3026475142942915028254996046317521377767588980 3026475142942915028254996046317521377767588980 302647514924915028254996046317521377767588980 30264751492491928497598062932267627647047438598758157 3056171045202925348073506042392267627647047438598758157 5056173454259234807350642494838746140583576C68384758 5056738455583402304448387461436835761281468435826847680 3026170452029214972840428545444688376126424444848 3026174522049754444483874614368357612844483874614385847680 3026174522049754444838746143683576128444838746143858476444848 50567384555834023044483874614385847548464838546768474844848 3026174952204975444484837464484834764484834764648444848 5056738455583402304448387461438484748464834564648 302617445220497544448488476448484854764848484748484848474848484848484848484					
		UGR_ETSIIT_CTF24{P4ss	w0rd_Cr4ck1ng_4nd_F					
STEP	BAKE!		-					



>incibe\_











