

# Ethical Hacking and Vulnerability Assessment

Roll No.: 18BCE152

Date: 10/11/2021

## Practical 9

### OBJECTIVE

- Attack graph implementation and generating attack paths

### INTRODUCTION

An attack graph is a succinct representation of all paths through a system that end in a state where an intruder has successfully achieved his goal. An attack path is the identification of one or more vulnerabilities that can be exploited by attackers to gain access to specific assets and move between them in a network, thus, forming an exploitable path between the assets.

Here I have implemented attack graph in c++ language.

#### CODE:

```
#include<bits/stdc++.h>
using namespace std;

int n,e,st,en;
map<int,string> mp;
vector<int> attack_graph[100];
vector<vector<int>> paths;
bool vis[100];
void find_path(int v,vector<int> &p){
    if(v==en){
        p.push_back(v);
        paths.push_back(p);
        p.pop_back();
        return;
    }
    p.push_back(v);
    vis[v] = true;
    for(auto &x:attack_graph[v]){
        if(!vis[x]){
            find_path(x,p);
        }
    }
    vis[v] = false;
    p.pop_back();
}
```

# Ethical Hacking and Vulnerability Assessment

```
int main(){  
  
    printf("Enter number of nodes : ");scanf("%d",&n);  
    for(int i=0;i<n;i++){  
        string s;  
        printf("Node %d : ",i);  
        cin>>s;  
        mp[i] = s;  
    }  
    printf("\nEnter number of edges : ");scanf("%d",&e);  
    printf("Enter %d connctions : \n",e);  
    for(int i=0;i<e;i++){  
        int s,d;  
        printf("Edge %d : ",i+1);  
        scanf("%d%d",&s,&d);  
        attack_graph[s].push_back(d);  
    }  
  
    printf("\n ++ For Attack Path ++\n");  
    printf("Enter source and destination node : ");  
    scanf("%d%d",&st,&en);  
    printf("\n");  
    vector<int> p;  
    memset(vis,false,sizeof(vis));  
    find_path(st,p);  
    if(paths.size()==0) printf("NO path found!!\n");  
    else{  
        for(auto &x:paths){  
            for(auto &y:x){  
                cout<<mp[y]<<" ";  
            }  
            printf("\n");  
        }  
  
        printf(" Total %d paths found",paths.size());  
    }  
    return 0;  
}
```

# Ethical Hacking and Vulnerability Assessment

```
File Edit Selection View Go Run Terminal Help • 18BCE152_practical_9.cpp - Visual Studio Code

C: > cp > sem 7 > Ethical hacking and vulnerability assesment > lab > Practical_9 > 18BCE152_practical_9.cpp > find_path(int, vector<int>>&)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

cd "c:\cp\sem 7\Ethical hacking and vulnerability assesment\lab\Practical_9\" ; if ($?) { g++ 18BCE152_practical_9.cpp -o 18BCE152_practical_9 } ; if ($?) { .\18BCE152_practical_9 }

Enter number of nodes : 15
Node 0 : (ftp,1,0)
Node 1 : (user,0)
Node 2 : (ftp,2,0)
Node 3 : (ftp_rhosts,0,1)
Node 4 : (ftp_rhosts,0,2)
Node 5 : (rsh,0,1)
Node 6 : (rsh,0,2)
Node 7 : (ftp_rhosts,2,1)
Node 8 : (rsh,2,1)
Node 9 : (sshd_bof,0,1)
Node 10 : (sshd_bof,2,1)
Node 11 : (ftp_rhosts,1,2)
Node 12 : (local_bof,1,1)
Node 13 : (rsh,1,2)
Node 14 : (local_bof,2,2)

Enter number of edges : 22
Enter 22 connctions :
Edge 1 : 1 3
Edge 2 : 1 4
Edge 3 : 1 5
Edge 4 : 1 6
Edge 5 : 1 9
Edge 6 : 3 5
Edge 7 : 4 6
Edge 8 : 5 11
Edge 9 : 5 12
Edge 10 : 6 7
Edge 11 : 6 10
Edge 12 : 7 8
Edge 13 : 8 11
```

```
File Edit Selection View Go Run Terminal Help • 18BCE152_practical_9.cpp - Visual Studio Code

C: > cp > sem 7 > Ethical hacking and vulnerability assesment > lab > Practical_9 > 18BCE152_practical_9.cpp > find_path(int, vector<int>>&)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Edge 14 : 8 12
Edge 15 : 9 11
Edge 16 : 9 12
Edge 17 : 9 13
Edge 18 : 10 11
Edge 19 : 10 12
Edge 20 : 10 13
Edge 21 : 11 13
Edge 22 : 13 14

++ For Attack Path ++
Enter source and destination node : 1 14

(user,0) (ftp_rhosts,0,1) (rsh,0,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (ftp_rhosts,0,2) (rsh,0,2) (ftp_rhosts,2,1) (rsh,2,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (ftp_rhosts,0,2) (rsh,0,2) (sshd_bof,2,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (ftp_rhosts,0,2) (rsh,0,2) (sshd_bof,2,1) (rsh,1,2) (local_bof,2,2)
(user,0) (rsh,0,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (rsh,0,2) (ftp_rhosts,2,1) (rsh,2,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (rsh,0,2) (sshd_bof,2,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (rsh,0,2) (sshd_bof,2,1) (rsh,1,2) (local_bof,2,2)
(user,0) (sshd_bof,0,1) (ftp_rhosts,1,2) (rsh,1,2) (local_bof,2,2)
(user,0) (sshd_bof,0,1) (rsh,1,2) (local_bof,2,2)
Total 10 paths foundPS C:\cp\sem 7\Ethical hacking and vulnerability assesment\lab\Practical_9>
```

# Ethical Hacking and Vulnerability Assessment

## **CONCLUSION**

In this practical we gain knowledge about attack graph and attack paths. Also learn implementation of attack graph.