IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

Prevention against Hacking using Trusted Graphs

Yash Sanzgiri[1], Kevin Garda[2], Arush Vichare[3]

Student, Dept of Information Technology, Fr Conceicao Rodrigues College of Engineering, Mumbai, India[1, 2, 3]

Abstract: Nowadays, accessing information and exchanging of data in business industry is increasing but it also increases the risk of security. The state of the security on internet is bad and becomes worse. The explosive growth of internet has brought many good things, but there is also a dark side: Criminal hacker. The initial design for common communication protocols indicates that the technology was proposed to meet main requirements such as speed, performance, efficiency and reliability but security was not a concern at that stage. Hacking is the practice of modifying the features of system, in order to accomplish a goal outside of the creator's original purpose. Number of solutions is provided against hacking but they are unable to address those issues. This project provides security for entire infrastructure to protect against hacking. The proposed infrastructure avoids the three pre-hacking steps. It generates the trusted graph and creates the confusion in front of hacker. Hacker cannot understand the current communication infrastructure and it is difficult for him to break the system easily.

Keywords: Hacking, Trusted Graph, honey pot, Dijkstra's algorithm.

I. INTRODUCTION

The world of internet is growing at an enormous pace and so is the concern of the security of the data over the internet. [2]Since there isn't any restriction on the users who can access the internet, the vulnerability of the data over the internet is high. People can access someone else's data and manipulate it for their own good. Hacking is descriptive term used to describe the attitude and behaviour of group of people who are greatly involved in technical activity which results in gaining unauthorized access.Hacking is a technique of maliciously attacking someone else's computer/network with an intention to steal or manipulate the data.

This proposed security approach is designed to eliminate the possibility of hacking by using trusted Graphs. These graphs are generated dynamically and would determine the amount of trust factor associated with each node. This paper aim to design a dynamic security approach that is mainly directed to defend hacking.

II. LITERATURE SURVEY

A. Hacking

Hacking is a descriptive term used to describe the attitude and behaviour of group of people who are greatly involved in gaining unauthorized access. [2][7]Hacking on computer systems might lead to loss of money, leak of sensitive information and loss of reputation.In computer networking; hacking is any technical effort to manipulate the normal behaviour network connections and connected systems. A hacker is any person engaged in hacking. The term "hacking" historically referred to constructive, clever technical work that was not necessarily related to computer systems. Today, however, hacking and hackers are most commonly associated with malicious programming attacks on the Internet and other networks.

Malicious attacks on computer networks are officially known as cracking, while hacking truly applies only to activities having good intentions. Most non-technical people fail to make this distinction, however.

Outside of academia, it's extremely common to see the term "hack" misused and be applied to cracks as well. A few highly skilled hackers work for commercial firms with the job to protect that company's software and data from outside hacking. Cracking techniques on networks include creating worms, initiating denial of service (DoS) attacks, or in establishing unauthorized remote access connections to a device.

III.RELATED PREVENTION TECHNIQUES

A. Honeypots

In computer terminology, a honeypot is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Honeypots can be classified based on their deployment (use/action) and based on their level of involvement. Based on deployment, honeypots may be classified as: [3]

Production honeypots and Research honeypots

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Research honeypots are run to gather information about the motives and tactics of the Black hat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats that organizations face and to learn how to better protect against those threats. Based on design criteria, honeypots can be classified as:[3]

1. pure honey pots
2. high-interaction honeypots
3. low-interaction honeypots

Pure honeypots are full-fledged production systems. The activities of the attacker are monitored by using a casual tap that has been installed on the honeypot's link to the network. No other software needs to be installed.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

High-interaction honeypots imitate the activities of the production systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste his time. By employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high-interaction honeypots provide more security by being difficult to detect, but they are expensive to maintain. If virtual machines are not available, one physical computer must be maintained for each honeypot, which can be exorbitantly expensive. Example: Honeynet.

Low-interaction honeypots simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the virtual system's security. Example: Honeyd.

B. Firewalls

A firewall is a system designed to prevent unauthorized access to from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. The evolution of the firewalls is given below [6]

- Access Control Lists (ACLs) were early firewalls implemented, typically on routers. They are useful for scalability and performance, but can't read more than packet headers, which provide only rudimentary information about the traffic.[5]
- Proxy firewalls, the computer's response is sent to the proxy, which passes the data with the origin address of the proxy server.[5]
- Statefull packet filter firewalls were the next major evolutionary step. They classify and track the state of traffic by monitoring all connection interactions until a connection is closed.[5]
- Unified Threat Management (UTM) solutions consolidate statefull inspection firewalls, antivirus, and IPS to a single appliance. They are also generally understood to include many other network security capabilities.
- Next-generation firewalls (NGFWs) were created to respond to increasing capabilities of malware and applications. They bring together the key network security functions, including advanced firewall, IPS/IDS, URL filtering and threat protection.

C. Intrusion Detection System

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. Intrusion detection systems are of two main types, network based (NIDS) and host based (HIDS) intrusion detection systems.

Network Intrusion Detection Systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet

Host Intrusion Detection Systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected.

All Intrusion Detection Systems use one of two detection techniques:

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network- what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other- and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline.

A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware

IV. PROPOSED SYSTEM

The Project would prevent hacking with the help of trusted graphs. We are going to prevent hacking using trusted graph. Trusted graph is used represent the map of the network. The trusted graph helps to find which path from the source to the destination is the trusted path. The node with the highest priority is selected as the trusted node to reach the destination. If two or more nodes have the same priority we will use Dijkstra's algorithm. History of all attacks which have occurred on the network is maintained. Priority to each node is assigned depending on how much secure the node is. The Security of the node would be determined based on the attacks that have occurred on the node. If no attack has occurred on the node it is assigned the highest priority. As and when attack occurs from a particular node the history is updated and priority is changes. If an attack has not occurred through a particular node it is assigned priority 1(Highest priority) .The priority for each node can be assigned by the organization based on the intensity of the attack. For example Dos attack could be assumed as a greater threat than Spoofing. However the assumption of which attack should be considered a higher threat will differ from organisation yo organisation. Similarly a database would be maintained to keep a record of the attacks occurred on the node helping in generating suitable statistics.

The structure of the network will be changed frequently to prevent attacker from attacking the network. Thus, the approach contains a dynamic feature that updates the trusted graph architecture in a period of time which is not enough for hackers to understand the current architecture of the trusted graph. This creates a honeypot which traps attacker to attack thus preventing any attack in future.

A. Dijkstra's Algorithm

Single-Source Shortest Path Problem is the problem of finding shortest paths from a source vertex v to all other vertices in the graph. Dijkstra's algorithm is a solution to the single-source shortest path problem in graph theory. It works on both directed and undirected graphs. However,

IJARCCE

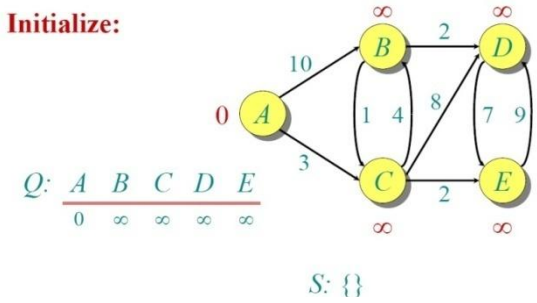
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

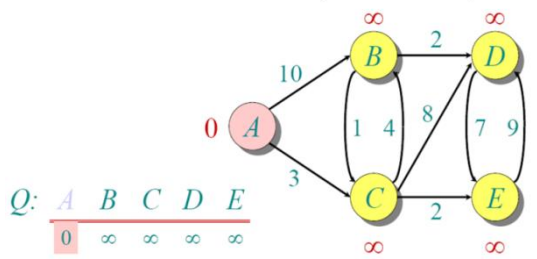
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

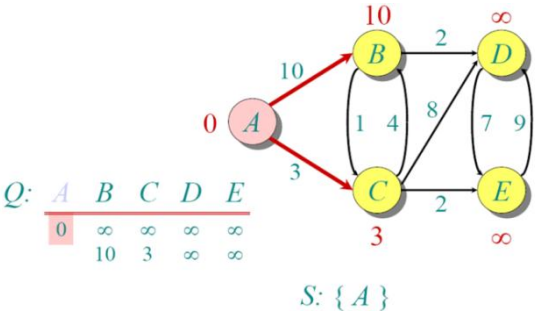
All edges must have nonnegative weights. The algorithm uses a greedy approach.
Input: Weighted graph G= {E, V} and source vertex v∈V, such that all edge weights are nonnegative
Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertexv∈V to all other vertices. Dijkstra's algorithm psuedocode [4]

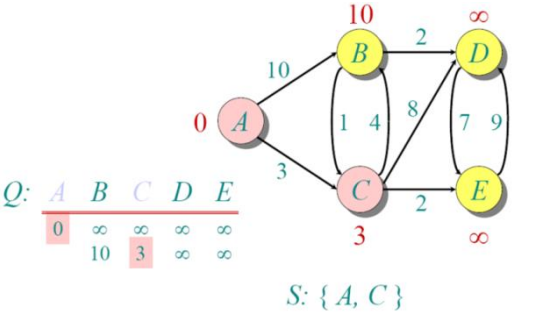
```
dist[s] ←0                              (distance to source vertex is zero)
for all v ∈ V-{s}
    do dist[v] ←∞                       (set all other distances to infinity)
S←∅                                     (S, the set of visited vertices is initially empty)
Q←V                                     (Q, the queue initially contains all vertices)
while Q ≠∅                              (while the queue is not empty)
do  u ← mindistance(Q,dist)             (select the element of Q with the min. distance)
    S←SU{u}                             (add u to list of visited vertices)
    for all v ∈ neighbors[u]
        do if  dist[v] > dist[u] + w(u, v)          (if new shortest path found)
           then  d[v] ←d[u] + w(u, v)               (set new value of shortest path)
                 (if desired, add traceback code)
return dist
```

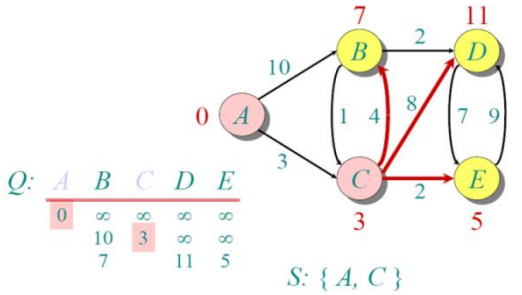
The working of the Dijkstra's algorithm is explained below [4]

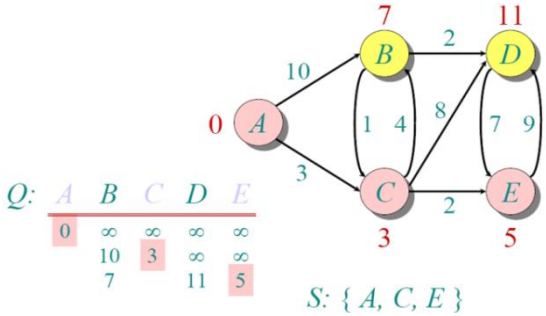


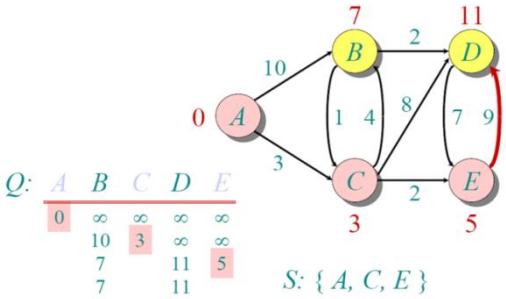


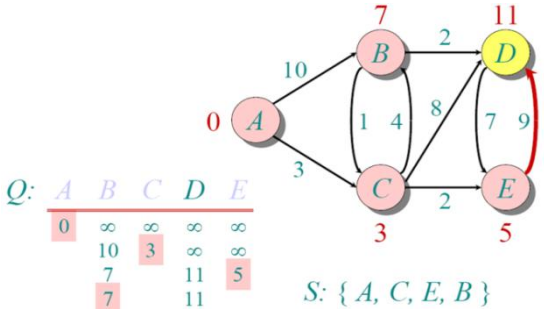


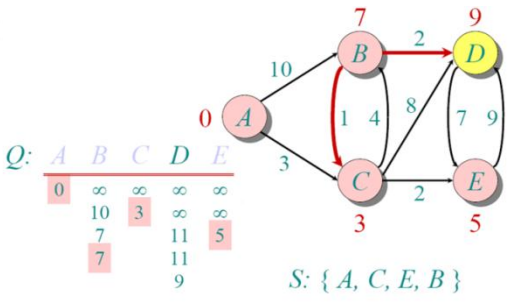


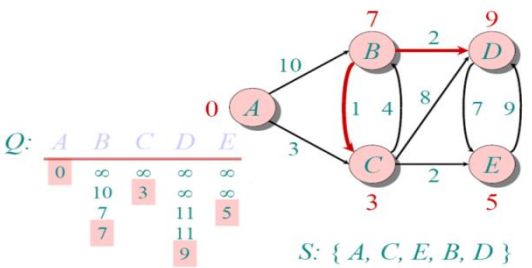












DOI 10.17148/IJARCCE.2015.41140

B. Trusted Graphs

It is a definition of the logic communication sequences within all endpoints inside the infrastructure. It is behaviour of a trusted user. The trusted graph forces all endpoints within the infrastructure to follow a sequence of communication which facilitates obfuscation and meaninglessness of the communication. The illusion of randomization concept is applied in the communication sequence between all nodes inside the infrastructure. That concept creates confusion since all nodes communicate with each other in a formal connection definition and continually changes. For generating the trusted graph, it uses any randomized algorithm. Trust network: A trust network can be formed based on transitive trust, with each link representing the trust relationships between two participants. Trusted graph: A trusted graph is a sub-network of a trust network and connected by a set of trusted paths.

C. Dynamically changing structure of the network

The basic use of a dynamically changing structure over its static counterpart is better security. Dynamic networks mainly consists of four major aspects, which are Network dynamics, Input Dynamics, Duration and control.[8]

- Network Dynamics: Being a dynamic structure, the network topology keeps on changing overtime. Parallel to this change, the nodes and edges may or may not still be a part of the network. It also checks the reliability of a specific set of network.[8]
- Input Dynamics: The load over the network keeps on changing. In a truly balanced network it is highly required that the load gets distributed equally between all active nodes. Thus, a fast approximate balancing is done by preprocessing step in parallel computation. The packets to be routed enter and exit a node, so it has to be taken care that no such packets are lost in the hop from one node to other. Objects in the application itself are dynamically changed by addition and deletion.
- Duration: This can further be categorized into two parts namely 'Transient' and 'Continuous'. In a transient change, the dynamic changes occur for a specific amount of time after which the network remains static for a stipulated period of time. On the other hand, a continuous change keeps on changing the network. So it doesn't maintain static network for some period of time.
- Control: This is basically the entity which keeps a check on the whole network. Depending on the requirement, one can make use of three different control structures such as Adversarial, Stochastic and Game-Theoretic.
- Adversarial: Here, the dynamics of the network are changed by an adversary. She/he decides when to and where the change in the network should be made. Edge crashes, recoveries, node arrivals and departures, packet arrival rates at source and destination. For a meaningful analysis, the adversary needs to be constrained. Some level of connectivity has to be maintained and the packet rate has to be kept below a certain specified rate.
- Stochastic: Here, the dynamics of the network are worked by a probabilistic function. The neighbors of a new node are generated randomly. The packet arrivals itself are drawn from a previously decided probability distribution. Some of the parameter of processes needs to be constrained like, the distribution moment of service time and the mean arrival rate of packets. Just like adversarial approach, even this method needs to have maintained some level of connectivity in network.
- Game-Theoretic: An implicit assumption of the previous two models is made. One administration takes control of the whole network. The dynamics however are changed by the external affairs. Here each node is in standalone mode. They are independent agents which behave rationally according to the situation and respond actions of other nodes in network. Thus, the dynamic changes are done via the interaction within nodes.

V. CONCLUSION

A thorough study of related papers has been done and the report is presented. The aim of our project is to develop a system that prevents hacking with the use of dynamic trusted graphs. Each node in the graph would be assigned a priority based on which the node is selected. Thus the hackers would find it difficult to get the structure of the network since it would change continuously, thereby preventing hacking.

ACKNOWLEDGMENT

We have great pleasure in presenting the report on "Prevention against Hacking using Trusted Graphs". We take this opportunity to express our sincere thanks towards the staff of Fr C.R.C.E, Bandra (W), Mumbai. We would specially like to thank **Prof Prajakta Bhangale** Assistant Professor, Fr CRCE, Bandra West for providing the technical guidelines, and the suggestions regarding the line of this work. We would also like to thank our friends and family members for their constant encouragement and support.

REFERENCES

[1] "A defencesecurity approach for infrastructures against hacking"Saad Alsunbul, Phu Le, Jefferson Tan 2013 IEEE DOI 10.1109/TrustCom.2013.197, pp.1600-1606.
[2] Ethical hacking: the security justification redux,in Technology and Society"B. Smith, Yurcik, W., Doss, D.2002. (ISTAS'02). 2002 International Symposium on, 2002, pp. 374-379
[3] [Online] https://en.wikipedia.org/wiki/Honeypot_(computing)
[4] [Online]http://www.cs.utexas.edu/~tandy/barrera.ppt
[5] [Online]https://www.paloaltonetworks.com/network-infrastructure/types-of-firewall-in-network-security.html
[6] "Firewall Policy Query"Alex X. Liu, Member, IEEE, and Mohamed G. Gouda, Member, IEEE IEEE transactions on Parallel and Distributed Systems Vol 20, No 6 June 2009
[7] "Ethical Hacking: The Security Justification Redux " Bryan Smith,William Yurcik, David Doss, Illiosis State university
[8] [Online]"Introduction to dynamic networks models,algorithms and analysis" Rajmohan Rajaram , Northeastern University http://www.ccs.neu.edu/home/rraj/Talks/DynamicNetworks/DYNAMO