Ant Colony Optimization to Detect Network Risks

Amanjot Kaur

ABSTRACT

Security of the information in the computer networks has been one of the most important Research Area. Network Security is becoming an important issue for all the organizations, and with the increase in knowledge of hackers and intruders they have made many successful attempts to bring down high-profile company networks and web services. Ant Colony Optimization algorithm is an evolutionary learning algorithm which could be applied to solve the complex problems. ACO algorithm fundamental idea has been inspired by the behavior of the real ants. Ants deposit pheromone as a trace to direct the other ones in finding foods. They choose their path according to the congestion of the pheromone. One of the most surprising behavioral patterns exhibited by ants is the ability of certain ant species to find what computer scientists call shortest paths. Biologists have shown experimentally that this is possible by exploiting communication based only on pheromones. ACO algorithms are the most successful and widely recognized algorithms techniques based on ant behaviors.

KEYWORDS

Network Security, Network Vulnerability, Vulnerability Attributes, Ant Colony Optimization, Operating System Fingerprinting

1. NETWORK SECURITY

Network Security is the process of preventing and detecting unauthorized access to your network. Prevention measures help you to stop unauthorized users (also known as “intruders”) from accessing any part of your Network. Detection helps you to determine whether or not someone attempted to break into your system, if they successful, and what they may have done. The closest we can get to an absolutely secure machine is one unplugged from the network, power supply, locked in a safe, and thrown at the bottom of the ocean. Network Security refers to all hardware and software functions, characteristics, features, operational procedures, accountability measures, access controls, and administrative and management policy required to provide an acceptable level of protection for hardware, software and information in a network.

2. NEED OF NETWORK

There are great numbers of threat to a network’ security; there are fortunately many preventative techniques to properly secure a network against those threats. There is some of the fact about the network security.

- Evolution of technology focused on ease of use.
- Increasing complexity of computer infrastructure administration and management.
- Decreasing Skill level needed for exploits.
- Direct impact of security breach on corporate asset base and goodwill.
- Increased networks environment and network based applications.

A major security objective is measuring the costs and benefits of security. If the cost is to be measure for securing an entity, whether it is data on networks, data on computers, or other assets of an organization, something has to be known about risk assessment.

3. VARIOUS THREATS TO NETWORK SECURITY

Logic attacks: are famed for taking advantage of already extant vulnerabilities and bugs in programs with the stated intention of causing a system to crash. There are cyber criminals who exploit this attack with the intention of willfully gaining illegal access to the system, or alternatively of downgrading the performance of a given network.

Resource Attacks: The second classification of network security threats are ***resource attacks***. Such assaults are primarily meant to overwhelm important system resources, like RAM and CPU resources. This is principally accomplished via dispatching numerous forged requests or IP packets to the network in question.

Trojan Horse: It proves to be malware which is not self replicating. Typically, such viruses are terribly cunning, in that they seem like they are performing a desirable task for the user. In reality though, they are making possible illegal access on to the user in question’s computer system. The term itself comes from the Trojan Horse story in Homer’s Illiad from Greek mythology.

Computer worms: They are computer program malware which are self-replicating. They utilize a computer network in order to dispatch copies of themselves to other computers using the network. They are different from computer viruses in that they are not required to be attached to any existing programs.

Eavesdropping: In general, the majority of network communications occur in an unsecured or "clear text" format, which allows an attacker who has gained access to data paths in your network to "listen in" or interpret (read) the traffic. When an attacker is eavesdropping on your communications, it is referred to as sniffing or snooping. The ability of an eavesdropper to monitor the network is generally the biggest security problem that administrators face in an enterprise. Without strong encryption services that are based on cryptography, your data can be read by others as it traverses the network.

Identity Spoofing (IP Address Spoofing): Most networks and operating systems use the IP address of a computer to identify a valid entity. In certain cases, it is possible for an IP address to be falsely assumed— identity spoofing. An attacker might also use special programs to construct IP packets that appear to originate from valid addresses inside the corporate intranet.

Denial-of-Service Attack: Unlike a password-based attack, the denial-of-service attack prevents normal use of your computer or network by valid users. After gaining access to your network, the attacker can do any of the following:

- Randomize the attention of your internal Information Systems staff so that they do not see the intrusion immediately, which allows the attacker to make more attacks during the diversion.
- Send invalid data to applications or network services, which causes abnormal termination or behavior of the applications or services.
- Flood a computer or the entire network with traffic until a shutdown occurs because of the overload.
- Block traffic, which results in a loss of access to network resources by authorized users.

Man-in-the-Middle Attack: As the name indicates, a man-in-the-middle attack occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data.

Sniffer Attack: It is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet. Even encapsulated (tunneled) packets can be broken open and read unless they are encrypted and the attacker does not have access to the key.

4. NETWORK VULNERABITITY

Network vulnerabilities are present in every system. Network technology advances so rapidly that it can be very difficult to eradicate vulnerabilities altogether; the best one can hope for, in many cases, is simply to minimize them. Networks are vulnerable to slowdowns due to both internal and external factors. Internally, networks can be affected by overextension and bottlenecks, external threats, DoS/DDoS attacks, and network data interception. The execution of arbitrary commands can lead to system malfunction, slowed performance, and even failure. Indeed, total system failure is the largest threat caused by compromised system-understanding possible vulnerabilities is critical for administrators.

5. VULNERABILITY ATTRIBUTES

Vulnerability is a complex and by definition it encompasses many attributes or multiple stresses (social, economic, environmental) which change at different speeds (slow and rapid change) - therefore, it is dynamic. If this is the case, methodologically, we cannot assume to be able to capture a vulnerability state per se, using inappropriate methods such as static indicators as it cannot be bounded, even if we attempt to incorporate many differing viewpoints of vulnerability using participatory processes. The system changes faster than it can be assessed (or perceived in many cases) and indicators do not capture the functional processes of the system or the interrelationships between these processes as they are often poorly understood.

One point of departure in attempting to assess dynamic vulnerability are the six attributes discussed in Downing et al. (2006). The existence or lack thereof of the following attributes contribute to the degree of 'lock-in' to a particular development pathway or to the adaptation of responsive coping cycles which are more likely to lead to sustainability and resilience.

1. Vulnerability is the differential exposure to stresses experienced or anticipated by different exposure units.

2. Vulnerability is not static - it is constantly changing on a variety of inter-linked time scales.

3. Social vulnerability is rooted in the actions and multiple attributes of human actors.

4. Social networks drive and bound vulnerability in the social, economic, political and environmental interactions.

5. Vulnerability is constructed simultaneously on more than one scale (e.g. economic impacts at the national or international scale can have cascading and unpredictable impacts at the local, micro-economic scale).

6. Multiple stresses are inherent in integrating vulnerability of peoples, places and systems.

These six attributes of vulnerability mentioned apply a holistic perspective to try and address the complexity and uncertainty inherent in such systems and the potential pathways of transitions to resilience and sustainability or to decline and degradation.

6. ANT COLONY OPTIMIZATION

The Ant Colony Systems or the basic idea of a real ant system is illustrated in Figure 1. In the picture, the ants move in a straight line to the food. The middle picture illustrates the situation soon after an obstacle is inserted between the nest and the food. To avoid the obstacle, initially each ant chooses to turn left or right at random. Let us assume that ants move at the same speed depositing pheromone in the trail uniformly. However, the ants that, by chance, choose to turn left will reach the food sooner, whereas the ants that go around the obstacle turning right will follow a longer path, and so will take longer time to circumvent the obstacle. As a result, pheromone accumulates faster in the shorter path around the obstacle. Since ants prefer to follow trails with larger amounts of pheromone, eventually all the ants converge to the shorter path around the obstacle, as shown in Figure 2.1

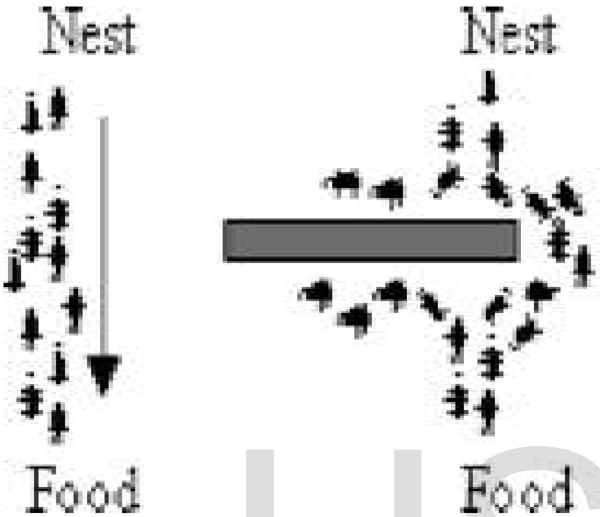

Figure 1 Illustrating the behavior of real ant movements.

An artificial Ant Colony System (ACS) is an agent-based system, which simulates the natural behavior of ants and develops mechanisms of cooperation and learning. ACS was proposed by Dorigo *et al.* (Dorigo and Gambardella, 1997) as a new heuristic to solve combinatorial optimization problems. This new heuristic, called Ant Colony Optimization (ACO) has been found to be both robust and versatile in handling a wide range of combinatorial optimization problems. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Artificial ants as if walk on this graph, looking for cheaper paths. Each ant has a rather simple behavior capable of finding relatively costlier paths. Cheaper paths are found as the emergent result of the global cooperation among ants in the colony. The behavior of artificial ants is inspired from real ants: they lay pheromone trails (obviously in a mathematical form) on the graph edges and choose their path with respect to probabilities that depend on pheromone trails. These pheromone trails progressively decrease by evaporation. In addition, artificial ants have some extra features not seen in their counterpart in real ants. In particular, they live in a discrete world (a graph) and their moves consist of transitions from nodes to nodes. The ACO differs from the classical ant system in the sense that here the pheromone trails are updated in two ways. Firstly, when ants construct a tour they locally change the amount of pheromone on the visited edges by a local updating role. Secondly, after all the ants have built their individual tours, a global updating rule is applied to modify the pheromone level on the edges that belong to the best ant tour found so far.

In computer science and operations research, the **Ant Colony Optimization** algorithm **(ACO)** is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

This algorithm is a member of the **Ant Colony Algorithms** family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.

7. IMPLEMENTATION AND RESULTS

7.1 Operating System Fingerprinting

OS fingerprinting describes the method of utilizing gathered information of a target host to find out what operating system the machine is running on. TCP/IP stack fingerprinting (or OS fingerprinting) is the process in computing of determining the identity of a remote host's operating system by analyzing packets from that host.

When doing penetration testing today the tester starts to gather as much information of the target machine as possible. One major key information is the operating system the target is running on. As long as this information is not revealed, the attacker is limited in the variety of attacks, probes and exploits. Therefore the focus on initial information gathering is put on finding out the operating system. There are several approaches to finding out the running operating system of an unknown host without having an account or any other way of logging in directly on this machine. Their range is from simple banner grabbing to highly sophisticated TCP- and/or ICMP-header analyses. This exposition will give a rough overview on some of them with a little in-depth dissection and description.

7.2 OS Fingerprinting through NMAP

Nmap OS fingerprinting works by sending up to 16 TCP, UDP, and ICMP probes to known open and closed ports of the target machine. These probes are specially designed to exploit various ambiguities in the standard protocol RFCs. Then Nmap listens for responses. Dozens of attributes in those responses are analyzed and combined to generate a fingerprint. Every probe packet is tracked and resent at least once if there is no response. All of the packets are IPv4 with a random IP ID value. Probes to an open TCP port are skipped if no such port has been found. For closed TCP or UDP ports, Nmap will first check if such a port has been found. If not, Nmap will just pick a port at random and hope for the best.

One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting. Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its nmap-os-db database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match. Each fingerprint includes a freeform textual description of the OS, and a classification which provides the vendor name (e.g. Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (general purpose, router, switch, game console, etc). Most fingerprints also have a Common Platform Enumeration (CPE) representation, like cpe:/o:linux:kernel:2.6. If Nmap is unable to guess the OS of a machine, and conditions are good (e.g. at least one open port and one closed port were found), Nmap will provide a URL you can use to submit the fingerprint if you know (for sure) the OS running on the machine. By doing this you contribute to the pool of operating systems known to Nmap and thus it will be more accurate for everyone.

8. CONCLUSION AND FUTURE SCOPE

Network vulnerabilities are the most critical for any network. This dissertation work investigated Network Vulnerability detection process and its current status. Use of Banner Grabbing, OS fingerprinting and Vulnerability detection has been successfully demonstrated. Nessus_Java and ACO integration has been successfully implemented and demonstrated, on isolated university Network. The results obtain show Network Vulnerability detection using. **Future Work** Ant Colony Optimization is a vast area of research. In future, work can be extended to include real network, augmentation based on Intrusion detection can also be applied.

9. REFERENCES

[1] A Brief History of Security and the Need for Adherence to the software Process Model by Paul Innella, www.tdisecurity.com/resources/assets/NetSec.pdf
[2] Network Security Fundamentals, by Gert De Laet, Gert Schauwers, Cisco press
[3] Efficient countermeasures for software vulnerabilities due to memory management errors, Prof. Dr.ir.W.JOOSEN, Prof.Dr.ir.F.PIESSENS.
[4] Computer Vulnerabilities, Written by Eric Knight, C.I.S.S.P.Original Publication: March 6, 2000. www.ussrback.com/docs/papers/general/compvuln.pdf
[5]http://en.wikipedia.org/wiki/Ant_colony_ optimization
[6] Using Ant Colony Optimization to Modellling the Network Vulnerability Detection and Restoration system, Xie Hui 1,Wu Min 2, Zhang Zhi-ming
[7] M.Dorigo, G.D.Caro, and L.M. Gambardella," ant algorithms for discrete optimization," Artificial life, Vol.5, No.3,pp.137-172,1999
[8] An Ant Colony Optimization Algorithm for Network Vulnerability Analysis, M.Abadi and S.Jalili, Iranian Journal of Electrical Electronic Engineering, Vol.2,nos 3 4, July 2006
[9] Ant Colony Optimization, Ahmad Elshamli, Daniel Asmar, Fadi Elmasri www.scribd.com/doc/22599034/Ant-Colony

Amanjot Kaur
Assistant Professor
S.D.S.P.M College For Women, Rayya(Asr)
amanjotkbhullar@yahoo.co.in