A Framework for Network Vulnerability Analysis

Guangzhi Qu, JayaPrakash, Ramkishore, Salim Hariri
ITL Lab, The University of Arizona
{qug, prakash, kishore, hariri}@ece.arizona.edu

C.S Raghavendra
University of Southern California
raghu@halcyon.usc.edu

Abstract

With increasing faults and attacks on the Internet infrastructure, there is an urgent need to develop techniques to analyze network and service vulnerability under organized fault attacks. Network vulnerability refers to the impact of attacks and faults on network and system behaviors. An accurate vulnerability analysis requires a deep understanding of failure modes and effects on each of the network components and the knowledge of how these components are inter-related at each point in time to various applications in a networked system. In this paper we present an agent based network vulnerability analysis framework and show how our framework can be used to analyze and quantify the system vulnerability under a Distributed Denial of Service (DDOS) attack scenario. Our approach can be described in terms of three steps: 1) Vulnerability Metrics – In this step we identify the metrics to be used to analyze the network vulnerability; 2) System State Characterization – In this step we define the thresholds to be used to characterize the node/system state to be in one of three states: Normal State, Uncertain State, and Vulnerable State and 3) Vulnerability Index Evaluation – In this step we evaluate the vulnerability of the network or application with respect to the vulnerability metrics defined in the first step. The vulnerability index can also be used as an indicator to trigger proactive and survivable methodologies to aid fast recovery at the earliest possible stages.

Keywords: Network Vulnerability, Vulnerability Index, Vulnerability Metrics.

1. Introduction

The Internet has grown exponentially over the last few years with the introduction of the World Wide Web and its applications that have expanded commensurately in both scope and variety. However the Internet's reliance on a few key nodes makes it especially vulnerable to organized attacks by hackers and terrorists, according to a new study on the structure of the worldwide network [1]. It has been shwon that, the average performance of the Internet would be reduced by a factor of two if only 1 percent of the most connected nodes are disabled. If 4 percent were shut down, the network would become fragmented and unusable [1].

Furthermore, the Internet will be increasingly heterogeneous in terms of speed, delay, capabilities, security, and mobility. Also the provision of QoS in the Internet (i.e., the Diffserv and Intserv architectures) assumes a reliable underlying network, which needs a high survivability to failures and attacks [2]. With such dependence on network infrastructure and services by military and commercial activities, it is critical that these services operate with high reliability and availability. The commission on critical infrastructure reports that the potential for disaster in the U.S. as a result of network attacks is catastrophic [3].

In this paper we present a framework to network vulnerability analysis, that can be used to discover attack points (or vulnerabilities) and to characterize the behavior of critical infrastructure networks under attacks and faults. We also discuss an approach to quantify their impact on network performance and services by an underlying agent based monitoring. Our approach can be described in terms of three steps: 1) Vulnerability Metrics – In this step we identify the metrics to be used to analyze the network vulnerability; 2) System State Characterization – In this step we define the thresholds to be used to characterize the node/system state to be in one of three states: Normal State, Uncertain State, and Vulnerable State); and 3) Vulnerability Index Evaluation – In this step we evaluate the vulnerability of the network or application with respect to the vulnerability metrics defined in the first step. Such a vulnerability analysis framework is an impending necessity for proactive and survivable architectures that provide fast recovery and convergence stability for networks susceptible to programmed attacks and faults.

This paper is organized as follows: Section 2 is related work, where we describe various methodologies developed for network security analysis. Section 3 gives the overview of the vulnerability analysis framework. Section 4 presents our agent-based network vulnerability analysis architecture. Section 5 explains our vulnerability analysis methodology at nodes, connections and system level. Section 6 provides a prototype DDOS

attack scenario analysis as a proof of concept of our network vulnerability analysis architecture.

2. Related Work

Several tools based on behavior modeling, fault trees, test & failure models provide host based analysis by checking logs, versions of system software and by monitoring performance metrics [4]. Vulnerability analysis in the networked environment and Internet infrastructure analysis is still in its infancy. Several Methods have been proposed but each has its own strengths and weaknesses. In this section we will examine some of the popular methods.

1) **Survivability Analysis of Network Specifications** [5] presents a system architecture, which injects fault and intrusion events into a given specification of a network and then visualizes the effects of the injected events in the form of scenario graphs. Using model checking, Bayesian analysis and probabilistic analysis, they provide a multi-faceted view of a network with respect to a desired service.

2) **Attack Trees** [6] determine which attacks are most feasible and therefore most likely in the environment. Vulnerability is quantified by mapping known attacks scenarios into trees. Attack trees assume that all vulnerability paths are known and can be defined as possible or impossible. This can change as new attacks are discovered, thereby rendering a previously impossible node suddenly possible.

3) **Passive monitoring of Internet links** [7] uses publicly available monitoring tools (CoralReef and NetraMet), to provide valuable data on a wide variety of network performance measures. These measurements are used to analyze Internet connection at a large university.

One of the main drawbacks of the above methods is that they predict occurrence of an event alone but doesn't quantify vulnerability with some certainty. Forecasting and predicting the occurrence of an attack or fault scenario with quantifiable amount of certainty provides results used for predicting attacks and to provide proactive security mechanisms.

3. Motivation and Overview

The overall goal of our approach is to formulate a theoretical basis for the construction of global metrics that can be used to analyze, manage and control complex network systems. The model concept of our approach is illustrated in Figure 3.1.

Any network or system's behavior can be characterized by using a set of focused analysis

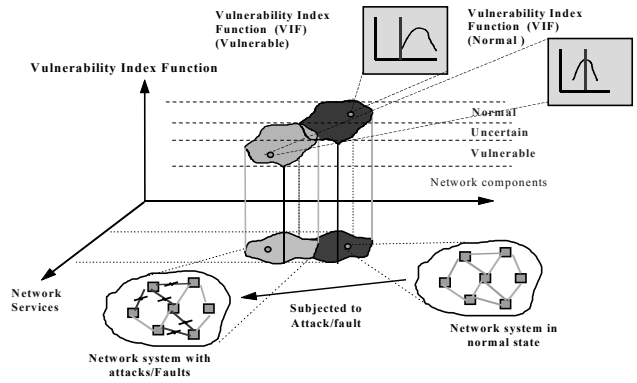

Figure 3.1 vulnerability index function model

functions such as vulnerability *Index function* (VI*F*), dependency *analysis function* (D*AF)*. These functions model the behavior of the individual network components and network services. Using the appropriate function (e.g. VIF), we can define regions of operations (normal, uncertain, and vulnerable) that can accurately characterize and quantify the operations of network services under various attacks and faults (see Figure 3.1). In fact, the concept of using index functions to quantify and characterize the behavior of networked systems and network services is analogous to the biological concept that uses metrics such as temperature or blood pressure to figure out the activity of an organism [8]. We conclude that under average conditions, the VIF will be within the normal region whereas when faults or attacks occur the VIF values will shift beyond normal region into uncertain region and vulnerable region. This provides the mathematical basis to proactively respond to failures at much higher speed.

4. Vulnerability Analysis Framework

The dynamics of network services and complexity of network connectivity can be represented with the traditional state machine approach. In fact all attacks or faults in a system causes the network to evolve along sequence of network states before they finally bring the network system to its knees. Such a representation of network states needs a constant monitoring of the network with respect to metrics that measure the utilization of network resources, traffic flow behaviors for all the running network services. Consequently, the vulnerability analysis architecture requires identification of a set of global metrics that could be associated with various faults

and attacks in a network and then compute the vulnerability index of the network system using these metrics.

The main modules of the vulnerability analysis are shown in Figure 4.1. The Agent module consists of agents that monitor various metrics at each entity in a networked environment (node, link) and generate events that are sent to the Event Correlation module.

The metrics that are monitored by an agent depend on the fault and/or attack scenario the network is subjected to. A single network attack and/or fault will affect the performance of the whole network and could result in multiple events captured by agents at various levels.

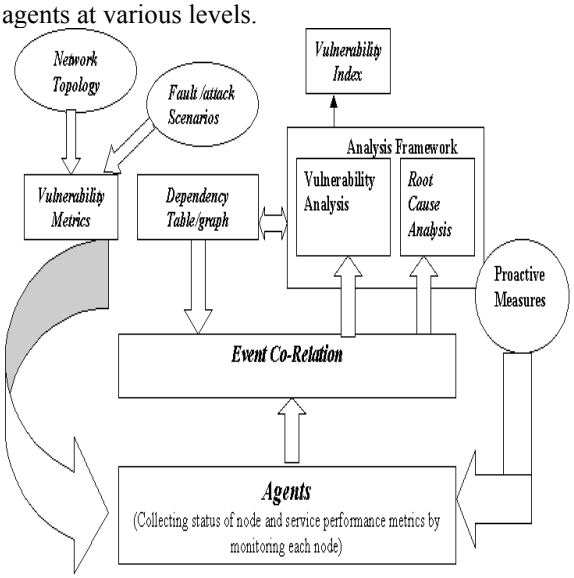

Figure 4.1 Agent based Vulnerability Analysis Architecture

The *Event Correlation module* gathers the various events from *Agents* and correlates the events with the information from the dependency module and creates unique fault or attack events

The *Dependency Module* builds an information model representing various network paths, resources and services at each node.

The Vulnerability Analysis module uses these correlated events and computes the vulnerability index at the system and connection and node level.

Identifying root cause problems in complex networked systems is a difficult, error-prone and time-consuming task. Each managed element can fail in many different ways, and each failure can propagate from the faulty element to other related network, system or application elements, making them appear faulty as well [9]. *Root Cause Analysis Module* then uses these correlated events for further analysis and identification the main bottleneck or root cause of failures and attacks. This analysis is used to trigger the agents to execute the corresponding proactive and recovery routines.

5. Vulnerability Analysis Methodology

Vulnerability metrics are characteristic parameters that represent attacks or faults. We use these vulnerability metrics in our analysis approach to quantify the impact of the attacks and/or faults on a network or a service. The attack scenarios that will be analyzed using our approach are the following:

- **TCP Sync attack**: The attack is characterized as congestion based attack that starves the resources of the victim host by consuming the service request queue and buffers. A vulnerability metric can be quantified based on the number of service requests in a pending/established state, the queue length that indicates congestion levels, packet drop rate at victim node interface that can be used as an indicator of such an attack.

- **IP fragmentation attack**: This uses the fragmentation technique in variable MTU network paths due to the heterogeneous bandwidth requirements. An attacker can spoof false fragmentation packets through the core network leading to wastage of resources such as CPU time, buffers at core routers and leading to denial of service to legitimate packets introducing delays and performance degradation. The CPU time and buffer outage and large stream of fragment packets are metrics of such an attack scenario.

- **Web server attack**: A web server that processes a CGI request will spawn a new process for every request. A malicious stream of request can spawn a lot of processes leading to a heavy usage of OS resources and CPU time of the system.

- **Ping of death and Smurf attack**: Both these attacks are due to the high possibility of spoofing. The vulnerability metrics for such attacks could be the suspicious foreign traffic rate through known interfaces, high protocol specific traffic rate like ICMP and queue length.

In this paper we apply our vulnerability analysis framework to monitor, detect and analyze the DDOS of TCP Sync attacks as a proof of concept.

The networked system behavior can be characterized by three operational states – normal state (NS), uncertain state (US), and vulnerable state (VS). At any moment, a given network or a connection or a node must be in one of these three states. Thresholds are used to differentiate these three states. To quantify network vulnerability, we define a new measure, the vulnerability index (VI), which models the vulnerability level of the components in a given network. Vulnerability metrics at each node can be classified into two types: 1) Computational metrics such as CPU time, number of existing services, buffer usage, file system size, etc; and 2) Communication metrics such as bandwidth, interface speed, number of connections, protocol – *IP/ICMP/TCP/UDP* traffic flow rate, packets drop rate, connection utilization, service queue length, etc.

The vulnerability analysis can be applied to any level of abstraction within the network. We can determine the vulnerability of a node (router, gateway, computer) under a given fault attack, or we can determine the vulnerability for a set of nodes that we refer to as a connection, or for all the components in a networked system environment. Hence, our approach is hierarchical and can be applied to any of three levels of abstractions (nodes, connection, system).

The vulnerability index for a node, connection or a system can be evaluated using the following three mapping functions:

$\overrightarrow{VI}_{node}(t)$ vulnerability index vector of a node in the system.

$\overrightarrow{VI}_{connection}(t)$ vulnerability index vector of a connection.

$VI_{system}(t)$ vulnerability index of a given network system.

The vulnerability index for each level is obtained by applying the appropriate vector computing functions: max-plus computing, minus-plus computing, division-plus computing, little plus computing, and big-plus computing that are briefly explained below.

$\oplus(\vec{A})$ **Max-plus** computing on vector A to get one numerical value between 0 and 1 from different entities. For example, suppose vector $\vec{A}$ is {0.6, 0.3, 0.6, 0.8, 0.1} we can apply the max-plus computing to get the value 0.8, i.e., $\oplus(\vec{A}) = 0.8$

$\Theta(\vec{A},\vec{B})$ **Minus-plus** computing between vector A and vector B. Suppose we have two vectors as $\vec{A}$ ={0.4, 0.5, 0.3, 0.8, 0.9} and $\vec{B}$ ={0.2, 0.2, 0.2, 0.6, 0.4}, by using the Minus-plus computing we can get $\Theta(\vec{A},\vec{B})$ ={0.2, 0.3, 0.1, 0.2, 0.5}

$\varnothing(\vec{A},\vec{B})$ **Division-Plus** computing between vectors A and B.

Our vulnerability analysis approach can be outlined in the following steps:

Step1: Vulnerability Metrics*:* In this step, We define ϵ as the set of all computational metrics and Φ is the set of all communication metrics that identify an attack and/or fault to be monitored at each node by agents.

ϵ computational metric vector $\{\epsilon_{n1}, \epsilon_{n2}, \epsilon_{n3}\ldots\}$.
Φ communication metric vector $\{\zeta_{c1}, \zeta_{c2}, \zeta_{c3}\ldots\}$.

Step2: System State Characterization*:* We define the thresholds for each node that can characterize the state of a node to be one of three states: normal, *uncertain*, and *vulnerable*.

$\overrightarrow{NU}_{metric}$ Threshold between normal state and uncertain state.

$\overrightarrow{UV}_{metric}$ Threshold between uncertain state and vulnerable state.

Step3: Vulnerability Index Evaluation: We provide mapping functions that quantify the impact of faults/attacks on component behavior (e.g. node, connections or system). We define vulnerability index associated with each metric for each component by mapping the value at any given point of time in a network system into three ranges set by vector NU_{metric} and vector UV_{metric}. In mathematical form, it can be represented as follows,

$$\overrightarrow{VI}_{node}(t) = \oplus\left\{\varnothing\left[\vec{c}, \Theta(\overrightarrow{NU}_{metric}, \overrightarrow{UV}_{metric})\right]\right\}$$

Where $\vec{c}$ is the vector of specific metrics values measured or monitored by a software entity (e.g. agent).

By using the max-plus computing we can calculate the vulnerability index at the connection level as follows:

$$\overrightarrow{VI}_{connection}(t) = \oplus\left(\overrightarrow{VI}_{node}(t)\right)$$

Finally, we can get the system's vulnerability index by the following equation:

$$VI_{system} = \oplus\left(\overrightarrow{VI}_{connection}\right)$$

To illustrate how to apply our vulnerability analysis approach, we apply it to analyze the vulnerability of the network under a DDOS scenario.

Step1 – Vulnerability Metrics: We could measure the vulnerability of a server under an organized fault /attack such as DDOS TCP Sync attack using the number of service connections, packet drop rate, and queue length.

Step2 – System State Characterization: The thresholds of the selected metric (number of connection requests) at the node level is chosen to be 50 or less for normal state operation, and 85 or more for being in vulnerable state. That means, the threshold values are:

$$\overrightarrow{NU}_{metric}\{\varepsilon_{connections}, \varepsilon_{packetdroprate}, \xi_{queuelength}\} = \{50, 30, 60\}$$

$$\overrightarrow{UV}_{metric}\{\varepsilon_{connections}, \varepsilon_{packetdroprate}, \xi_{queuelength}\} = \{85, 60, 95\}$$

Step3 – Vulnerability Index Evaluation: The VI with respect to the number of service connections is defined as the ratio of the number of connections to the maximum no of connections available. The metrics increase as the attack progresses by moving the node state from normal state (number of connection =< 50) to uncertain (between 50 and 85 connections) and vulnerable state (number of connections => 85). That is to say when the number of connections is below 50, we can say the node is in normal state and VI =0. When the number of connections is between 50-75, the node is in uncertain state and its VI =c/100 where c is present the number of connections and above 85, we conclude the node is in vulnerable state and VI=1.

Other vulnerability metrics can be used in a similar way to analyze their impact on the behavior of node, connection and/or the overall system under different faults and attacks scenarios.

6. Simulation Results

We simulated a DDOS attack scenario using SSFnet network simulator (www.ssfnet.org) in which we designed a network topology of three sub networks connected through a core router backbone as shown in Figure 6.1. Each subnet is composed of edge routers and several hosts with multiple interfaces configured as either an http server or an http client. The target and source configuration of the traffic generated during the simulation is assigned by specifying the valid source and destination ***ip*** address for a given http server and http clients, respectively. At any given point of time the server networks are subjected to DDOS client and regular http client traffic so that we can study the impact of attacks on the regular traffic along with the resource starvation at the victim server in the form of service queue outage.

The Network vulnerability analysis framework is embedded in the simulator so that agents collect various metrics required for the vulnerability analysis. One particular node on the network is subjected to DDOS attack and all packets are routed to that particular node during the attack. From our system's architecture, we know every node in a given network system runs an agent, so any node subjected to a DDOS attack will give a similar response.

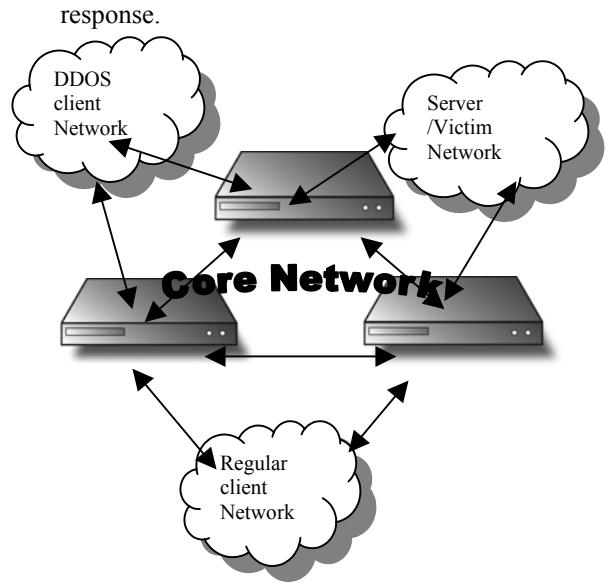

Figure 6.1 Vulnerability Analysis testbed

The figure 6.2 shows the service Connections outage of victim server in a Network under DDOS attack. The X-axis indicates the time for which the network is observed and Y-axis indicates the Vulnerability index of the system. During the simulation time, the server network is subjected to packets from regular http client's and also DDOS attack configured clients. At the beginning of the simulation run the victim server allows new connections from both the networks. As the attack period begins we see a sudden increase in the

connection rate due to the TCP SYN connection requests from the attack network the connections from regular http client network are dropped as their packets are rejected due to buffer overflow at the interface level and lead to time out of established connections. Subsequent legitimate new connection requests from the regular http client's network get dropped as the number of service connections reaches the vulnerable limit. After the Attack has stopped the metrics do not drop to Normal limit but drops exponentially to operational limits s depending on the time out period for service connections and processing time of queues that are consumed by the attack packets.

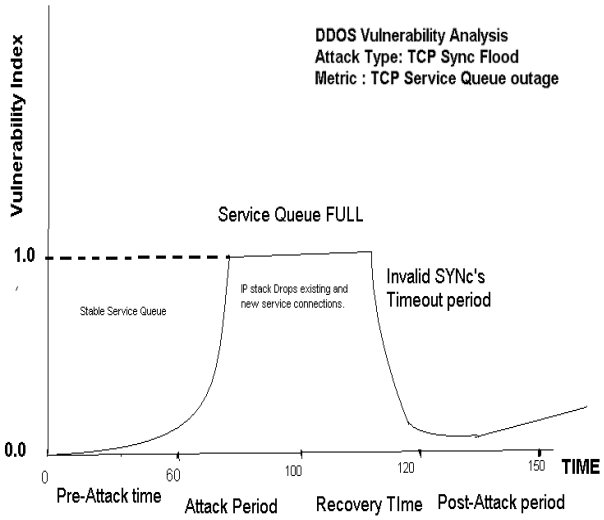

Figure 6.2 simulation of DDOS vulnerability Analysis

After the recovery time we could again observe normal http traffic between the client network and the victim server that was otherwise rejected due to resource consumption of DDOS attack. This simulation proves our idea to capture vulnerability at node level, connection level using the metrics to formulate the VI of the whole system subjected to DDOS attack. Considering Dependencies among other metrics and various nodes we are confident that our approach can accurately capture other vulnerabilities and attack scenarios.

7. Conclusion and Future work

In this paper we presented a framework to analyze the vulnerability of various attacks and/or faults on a network system. We also presented a methodology to quantify the network vulnerability at different levels of abstractions that include node, connection and system levels. Our future work will model all other well-known attacks and faults and then integrated with an agent-based control and management system to achieve proactive and reactive response to these attacks and/or faults.

References:

[1] The Internet's Achilles' Heel: Error and attack tolerance of complex networks, Albert, Hawoong Jeong, Albert-László Barabási, 2000

[2] Self-configuring Survivable Techniques for Quality of Service Enabled Internet, Yu Liu, David Tipper, Deep Medhi, Anotai Srikitja, 2000

[3] Report to the President's Commission on Critical Infrastructure Protection James Ellis, David Fisher, Thomas Longstaff, Linda Pesante, Richard Pethia, January 1997

[4] Quantifying Vulnerabilities In The Networked Environment: Methods and Uses Char Sample, Symantec and Ian Poynter , Jerboa Inc., 2000

[5] S. Jha, R. Linger, T. Longstaff, and J. Wing, Survivability Analysis of Network Specifications,Workshop on Depenedability Despite Malicious Faults, June 2000

[6] "Attack Trees", Bruce Schneier, Doctor Dobbs Journal, December 1999

[7] Brownlee, N., k claffy, M. Murray and E. Nemeth, "Methodology for Passive Analysis of a University Internet Link", 2001

[8] Manish Parashar. Interpretive Performance Prediction for High Performance Computing. PhD thesis, Department of Computer Engineering, Syracuse University, June 1994

[9] White paper Cisco Root Cause Analysis and Its Role in Event Management, 2000