Abnormality Metrics to Detect and Protect against Network Attacks

Guangzhi Qu, Salim Hariri, Santosh Jangiti, Suhail Hussain, Seungchan Oh , Samer Fayssal
ITL Lab, the University of Arizona
{qug, hariri} @ece.arizona.edu

Mazin Yousif
Intel Corporation, USA
mazin.s.yousif@intel.com

Abstract

Internet has been growing at an amazing rate and it becomes pervasive in all aspects of our life. On the other hand, the ubiquity of networked computers and their services has significantly increased their vulnerability to virus and worm attacks. To make pervasive systems and their services reliable and secure it becomes highly essential to develop on-line monitoring, analysis, and quantification of the operational state of such systems and services under a wide range of normal and abnormal workload scenarios. In this paper, we prevent several abnormality metrics that can be used to detect abnormal behaviors and also can be used to quantify the impact of attacks on pervasive system services. Our online monitoring approach is based on deploying software agents on selected routers, clients and servers to continuously monitor the measurement attributes and compute the abnormality metrics. Further, we use this metrics to quantify the impact of attacks on the individual components and on the system as a whole. This analysis leads to identify the most critical components in the system. We have built a test bed to experiment and evaluate the effectiveness of these metrics to detect several well-known network attacks such as MS SQL slammer worm attack, Denial of Service attack, and email worm spam.

1. Introduction

The Internet is experiencing a dramatic growth in connectivity, use, and in the services being offered over the past several years. As this growth continues and the Internet has become the most important and cost effective method of moving/sharing data across a wide range of geographically dispersed heterogeneous information systems. At the same time the Internet has become a virtual breeding ground for attackers who exploit the trust placed by the users in the network. This has increased the vulnerabilities of networked systems and their applications. Hence, it's crucial to make the network system operate in the normal state and guarantee the quality of service. Most of the current research focuses on developing intrusion detection techniques to well-known attacks (signatures) that can be used to protect against such attacks. However, it is very difficult to predict future attacks and their behaviors. Different techniques are required to handle the increase in complexity and intensity of future attacks on networks and their services (e.g., pervasive systems and their services).

In this paper we present a mathematical methodology and a framework to analyze network system operational state in real time so it can be used to accurately discover attack points. Furthermore, our analysis framework identifies the critical components that can severely impacts the operations of networked systems and their services.

The paper is organized as follows. In Section 2, we discuss briefly related works and network attack classification. Sections 3 and 4 describe our abnormality metrics and describe how these metrics can be used to quantify the impact of attacks on networks and services. Section 5 presents the experiment results of using the proposed metrics to characterize and quantify the operational states of systems and their services when subjected to attacks. Section 6 discusses the conclusion and future work.

2. Network Attack Types

Network attacks exploit the vulnerabilities in software and network protocols. They either consume node resource or network resource to severely degrade network performance or shutdown the entire network. Network attacks can be classified into three main categories based on

0-7803-8577-2/04/$20.00 ©2004 IEEE

its behaviors: 1) Denial of Service attacks, 2) Virus/Worm Attacks, and 3) Application Level Attacks.

***DoS* Attacks**: In *DoS* attacks, the attackers focus on one or a few network components and consume all the available resources in order for the denial of service to occur to legitimate services. There are numerous *DoS* attack methods [6] aimed at servers such as *TCP SYN* attack [4], *Smurf IP* attack [5], *Ping of Death* attacks [7]. In some of these attacks the attacker makes overwhelming connection requests to a victim server with spoofed source IP addresses. Due to the vulnerabilities in TCP/IP protocol stacks, the victim server can't complete the connection requests and wastes all of its system resources. As a result, the victim server will not be able to service legitimate traffic. The Denial of Service (*DoS*) attacks are increasingly focusing on routers and switches. Similarly to the server attacks, the attackers aim at consuming all router resources in order to force it dropping all incoming packets. In this way the network performance will be severely impacted due to abnormal behaviors of the victim router and other affected routers.

Virus/Worm Attacks: In virus/worm attacks, the propagation of attacks is random and can reach any network resource that has the vulnerability exploited by the virus/worm. If the number of resources that have this vulnerability is large, then within 10 minutes more than 90 percent of the vulnerable server all over the world got infected [1], most of these resources will be infected and severely degrade the overall network performance as experienced in the CodeRed, Nimda, SQL Slammer Worm attack, MSBlaster, SoBig and other typical worm/virus attacks [3].

Application Level Attacks: Application-level attacks target application servers by deliberately causing a fault in a server's operating system or applications. This results in the attacker gaining the ability to bypass normal access controls. Some programs and network services were not originally designed with strong security in mind and are inherently vulnerable to attacks. The BSD remote services (rlogin, rexec, etc.) and Internet Anywhere Mail Server are some examples. The attacker takes advantage of this situation by gaining control of the application, system, or network, and can do any combination of the following: Abnormally terminate applications; Consume operating system resources; and Disable other security controls to enable future attacks.

3. Abnormality Monitoring and Analysis

The main goal of the online monitoring and analysis is to compute abnormality metrics that characterize and quantify the operational state of actual network systems at runtime.

A Measurement Attribute (*MA*) denotes the value of some attributes that can be measured online during the observation period. For example, *MA*s can be the $TCP_{syn,\ out}$ (the rate of outgoing *TCP SYN* packets) of a network node, UDP_{out} (the total number of outgoing *UDP* packets) for the network system, and CPU_{util} (the *CPU* utilization).

We have identified different *MA*s for all network and system components that can be used to characterize the operational states of applications (FTP, Telnet, Web surfing, emails, etc.) down to physical device level (*CPU*, Memory) as shown in Table 1.

Table 1: Measurement Attributes

Protocols/Layers	*MA*s
App layer	*NIP/NOP*: number of incoming/outgoing PDUs. *IF*: Invocation Frequency
HTTP, DNS, SMTP, POP	
Transport layer	*NIP/NOP*
TCP, UDP	
Network layer	*NIP/NOP* *AR*: ARP request rate
IP, ICMP, ARP	
resource	CPU_{util}: CPU utilization M_{use}: Memory Usage
CPU, Memory	

In our approach, any application and resource (network or pervasive system) is assumed to be in one of three states; 1) Normal Sate, 2) uncertain State, or 3) Abnormal Sate (see Figure 1).

Normal State – when an application, computer system or network node operates normally. We use *the appropriate measurement attributes* at each level to quantify whether or not the component operates normally; For example, if the number of unsuccessful TCP sessions in the network system is 0, we assume the network system is operating normally.

Uncertain State – In a similar way, we use the *MA*s to describe the behavior of a component and/or a resource of being in uncertain state when the measurement attributes associated with it are between the normal and abnormal thresholds.

Abnormal State – A component or a resource is considered to be operating in an abnormal state when its measurement attributes are significantly higher than the levels observed for normal operational states; when the number of incoming

packets per second is order of magnitude larger than the normal observed rate.

We introduce a new metric, Abnormality Distance Ratio (*ADR*) to quantify the component/resource operational states (e.g. *normal, uncertain, and abnormal*) with respect to one measurement attribute (ADR_{MA}). The ADR_{MA} can be calculated as the ratio of the current distance from normal level with respect to one measurement attribute (*MA*) divided by the normal value for the *MA* as shown in Equation 1.

$$ADR(MA,t) = \begin{cases} 1 & if \quad AD_{MA}(t) \geq \Delta_{MA} \\ \dfrac{AD_{MA}(t)}{\Delta_{MA}} & otherwise \end{cases} \quad - (1)$$

where, $AD_{MA}(t)$ is the online calculated operational index specified to *MA* (see Table 1). Δ_{MA} is the threshold that denotes the minimal distance from a normal operational state to the abnormal operational state quantified by the operational measurement attributes *MA*. Figure 1 shows that when the network operates in normal state, the value of *ADR* is around 0.1. When the node endures an attack, the *ADR* value (1.0) shows it operates in abnormal state.

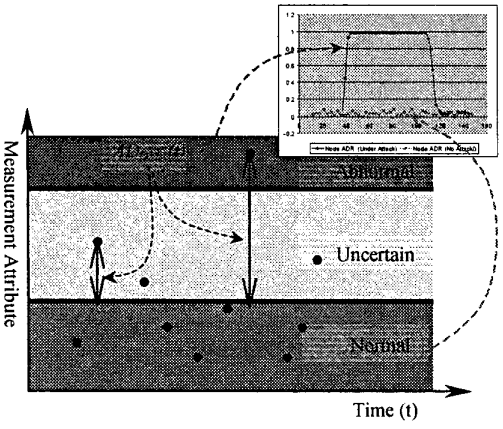

Figure 1: The operational states of a component/resource with respect to MAs

Based on this methodology we have developed an agent based operational analysis framework shown in Figure 2.

The online monitoring engine is capable of monitoring the multiple measurement attributes (*MAs*) for a single node or the whole pervasive system. The data mining statistic engine is designed to automatically extract effective measurement attributes in data collection for profile modeling of operational states and attacks. The adaptive analysis engine calculates the abnormality distance $AD_{MA}(t)$ in real time based on the MA_{normal} which denotes the normal value for *MA* when the application or a network node operates normally. When $AD_{MA}(t)$ deviates from the threshold that determines the operational state of the network system and network node, events will be generated to inform the self protection module. The self healing engine will take actions such as shutting down node interface or itself to prevent the propagation of the attack impact, save the environment information of the breaking point and resume the tasks unfinished through other nodes in the system.

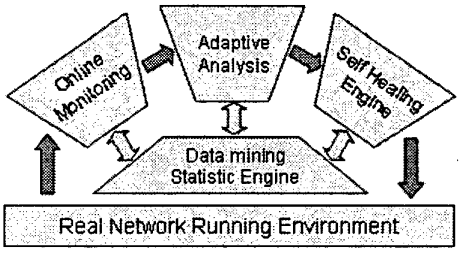

Figure 2: Online Monitoring and Analysis Framework

4. Operational Analysis Application – Impact Analysis

When we do online monitoring of the multiple measurement attributes and analysis of the operational states of the network system and network node, we can quantitatively identify critical resource points in the network system that their failures will severely impact the overall behavior of the system. The results of the impact analysis can be used to design network system and proactively protect the network system under attacks/faults.

The *ADR* characterizes and quantifies the impact of the network attack on a single network node such as client, server, or router. The *ADR* is defined as shown in Equation (1). For example, we assume that in the normal operation of the network system, a network client's data transfer rate is 100Kbps and threshold abnormality distance specified to operational measurement attribute data transfer rate Δ_{TR} is set to 95Kbps. If the client's data transfer rate decreases beyond 5Kbps due to the network attack, the client operational state is then considered to be

abnormal. If the client's data transfer rate is 60Kbps at moment t, then $ADR(TR, t) = 42.1\%$ can be used to quantify the impact of the network attack on this node. Similarly, we can compute the *ADR* of a router or a server in network attack scenarios from the Equation (1) on measurement attributes, such as buffer utilization the number of flows open or being processed, total number of flows, request processing rates, etc. Furthermore, the threshold abnormality distance used to define the boundary between normal and abnormal behavior can be dynamically adjusted to accurate characterize the operational state of any network component.

4.1 System Impact Factor (SIF)

The *SIF* identifies the impact of the network attack on a whole network system. The *SIF* is obtained by evaluating the weighted abnormality distance ratio metric of all the individual network components. That means, *SIF* can be evaluated by determining the percentage of the components that are operating in unacceptable states (i.e., the $ADR_j(MA, t)$ of node j is greater than some threshold value d_j) to the total number of components in the network system. In Equation (2), we can compute the overall impact of a given fault/attack on the network system.

$$SIF(t) = \frac{\sum_{\forall j. ADR_j(MA,t) > d_j} COS_j}{numberOfNodes} \quad (2)$$

Where, d_j is the operational threshold when the $ADR(MA, t)$ for network node j beyond d_j then the network node is regarded operating in an unacceptable state. COS_j is a binary variable denoting the operation state of network node j. The COS_j is equal to 1 when the network node j operates in an unacceptable state (i.e. $ADR_j(MA, t) > d_j$ from Equation 2.) and equal to 0 when it operates in an acceptable state (i.e. $ADR_j(MA, t) < d_j$).

5. Validation and Experimental Results

We have set up an instrumented test bed environment using the resources in the Internet Technology Laboratory at The University of Arizona to validate and demonstrate our approach in achieving efficient network attack detection and quantifying the operational state of the network element as shown in Figure 3. Cisco routers are used as the core network's backbone routers. In addition, several Linux routers are used as the access routers and are programmed using Autonomia online monitoring and analysis engines. For further information about Autonomia, please refer to [2]. The test bed consists of 4 Cisco 7500 series routers and 5 10/100M switches and 40 PCs. All these computers are configured into 5 sub networks. Network services and applications such as web browsing, email service are running on the test bed. Attack library is used to inject viruses, worms, and different kinds of attacks within the test bed. The network services are monitored and analyzed using Autonomia agents. When we inject the viruses, worms or attacks into the test bed, the Autonomia agents installed on each node continuously collect the appropriate *MA*s on every second and compute the $ADR(MA, t)$ for each network node. The operational index $ADR(MA, t)$ will show the operation state of the network node. At the same time, the system impact factor - $SIF(t)$ are continuously computed according to Equation (2) through the exchanging information among the Autonomia agents.

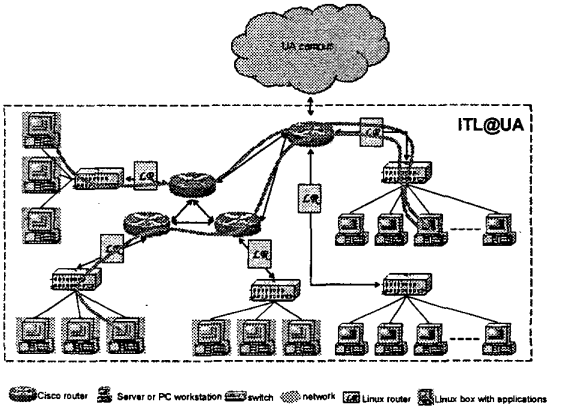

Figure 3: Test bed at the University of Arizona

In what follows, we demonstrate our methodology by quantifying the impact of network worm attack, denial of service, email spam on the network node.

5.1 MS SQL Slammer worm

In our network attack library, we have built the vulnerable *SQL* server program. The worm is a piece of code which will exploit the vulnerability within the vulnerable *SQL* server, the vulnerable *SQL* server will send 376 bytes packets via *UDP* port 1434 to the original attack launcher and other random destinations to propagate the worms.

By monitoring the network activities of the network nodes in the test bed, we use operational measurement attributes outgoing *UDP* packet rate and outgoing *ARP* packet rate to demonstrate our

approach. During the attack, the *SQL* slammer worm generates a lot of worm packets that are sent towards random destinations which will then cause *ARP* protocol to be extremely active shown in Figure 4a. The Autonomia agents continuously compute $ADR(UDP_{out}, t)$ (the abnormality distance ratio for the node specified to the measurement attribute - *UDP* packet outgoing rate), $ADR(ARP_{out}, t)$ (the abnormality distance ratio for the node specified to the operational measurement attribute - the outgoing rate of *ARP* packets), and $SIF_j(t)$. Figure 4b shows the abnormality distance ratio specified to measurement attribute ARP_{out}. When the SQL slammer worm launched on the network node, the $ADR(ARP_{out}, t)$ shows the network node enters abnormal state at around 260 seconds.

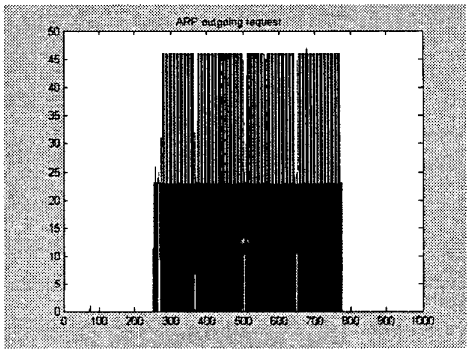

Figure 4a: ARP outgoing request during attack

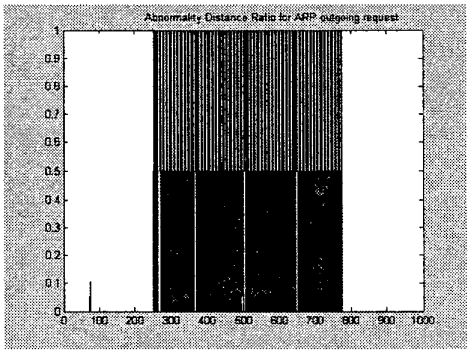

Figure 4b: $ADR(ARP_{out}, t)$ during the attack

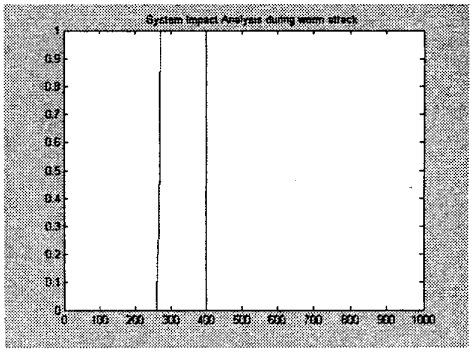

Figure 4c: System impact factor during MS Slammer SQL worm attack

During the experiment, we have 10 vulnerable servers among the 50 computers on the test bed. Figure 4c shows the system wide analysis on the impact of the worm attack, within around 6 seconds the 10 servers got infected and generate huge amount of worm packets (*UDP*) to 'random' *IP* address destination. These packets will consume the bandwidth of the network system and cause the servers themselves to deny legitimate service requests. Within 10 seconds, the system impact factor reaches 1.0 that means the whole network system is crashed down by the worm attack.

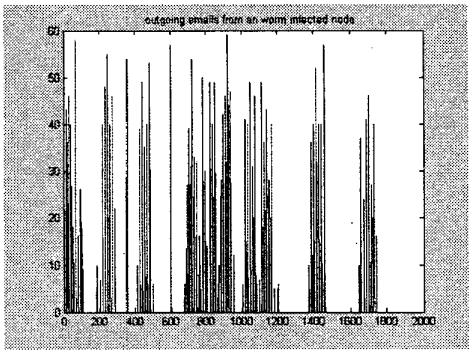

Figure 5a: Abnormal Email behavior

5.2 Email Worm Spam

Email worm is another disruptive network attack. When a node got infected by the email worm, the worm will collect all email addresses from client email program (e.g. MS Outlook) or from the files on the local disk. The worm program then sends hundreds of emails to all of the recipients in the email address list, with virus program as attachment, flooding the network with

traffic and also propagating the virus to other computers on the Internet.

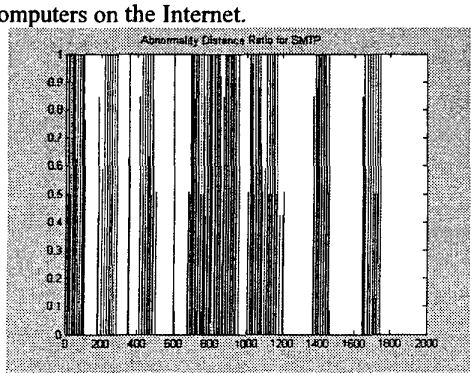

Figure 5b: Abnormality Distance Ratio

We monitored client computer email behaviors on our test bed. The data collected by Autonomia agents are shown in Figure 5a. We characterize the behavior of the Email service by using the abnormality distance ratio with respect to the measurement attribute – email invocation frequency shown in Figure 5b. Figure 5a shows the clients email program usage over a 30 minute period under attack scenario. The analysis engine sets the operational threshold for email service to be 10 emails per second, and computes the abnormality distance ratio. Figure 5b reflects clearly if the number of emails sending out greater than 10 emails per second, the node operates in abnormal state.

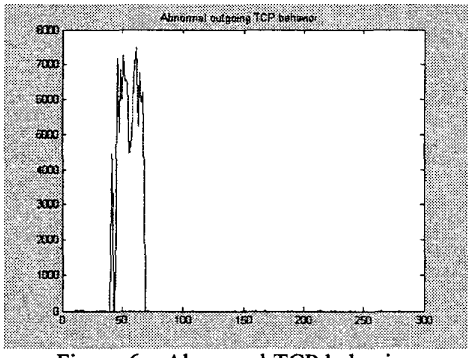

Figure 6a: Abnormal TCP behavior

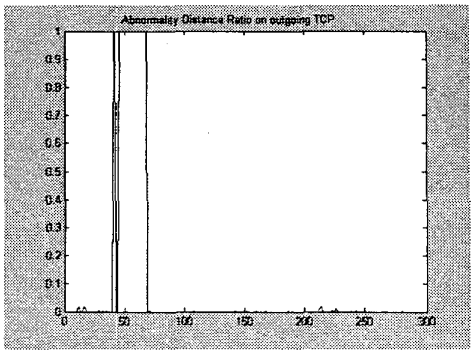

Figure 6b: Abnormality Distance Ratio for TCP outgoing packet rate during *DoS* attack

5.3 Denial of Service Attack

In this scenario, we emulate one of the very popular *DoS* attack methods – *TCP SYN* attack. We launch the *DoS* attack with spoofing IP address. In our test bed, we set up a Web Server with Apache 2.0.47. During the attack, one of the network nodes will start the *TCP SYN* attack towards the *HTTP* Server. The *SYN* packets rate can be adjusted as constant or random. The monitors installed on the server computer will sense the sheer deviation of the incoming *TCP SYN* packets rate. As shown in Figure 6a, the web server is most active from 40 seconds to 70 seconds. We can see the incoming *TCP* traffic increases dramatically around 40 seconds, and this phenomenon will continue quite a while, this is different from the normal *TCP* burst pike. Around 70 seconds, when the attack is stopped, we can see the *TCP SYN* request rate drop back to the normal level. Figure 6b shows the operational states change reflected by the abnormality distance ratio on the outgoing *TCP SYN* packet rate.

6. Conclusion & Future Work

In this paper, we presented a mathematical methodology in operational analysis of the network system. We define the operational mode for single network node, victim network and attacker/victim mixed network system. A framework is proposed for on-line monitoring and analysis of the operational state of the network systems. We have developed an operational index – abnormality distance ratio to quantify the operational state of the network system and its component. Also, the impact of network attacks on network system is quantitatively analyzed. We have validated our methodology on the test bed through emulation the *SQL* worm attack, email

worm spam, and *TCP SYN* Dos attack. We are currently studying new techniques to proactively mitigate the impact of attacks/faults when we analyze the network system operational state has turned into unacceptable or abnormal state.

References:

[1]. Q1Lab3 (2003) "The SQL Slammer Worm Incident".http://www.q1labs.com/resources/documents/q1_slammer_whitepaper.pdf

[2] S. Hariri, L. Xue, H. Chen, M. Zhang, S. Pavuluri, S. Rao (2003) "AUTONOMIA: An Autonomic Computing Environment". Submitted to *International Performance Computing and Communications Conference*.

[3] S. Gaudin (2003). "2003 Worst Year Ever for Viruses,Worms".http://www.internetnews.com/infra/article.php/3292461

[4] CERT® Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks http://www.cert.org/advisories/CA-1996-21.html

[5] CERT® Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks http://www.cert.org/advisories/CA-1998-01.html

[6] Kevin J. Houle, et. al. (2001) "Trends in Denial of Service Attack Technology" http://www.cert.org/archive/pdf/DoS_trends.pdf

[7] CERT® Advisory CA-1996-26 Denial-of-Service Attack via ping http://www.cert.org/advisories/CA-1996-26.html