ISSN: 2278-4632
Vol-12 Issue-12 No.02, December 2022

NICE: NETWORK INTRUSION DETECTION AND COUNTERMEASURE SELECTION INVIRTUAL NETWORK SYSTEMS

Ms. T.Pushpa Latha, Assistant Professor, Department of MCA, St.Ann's College for Women, Mehdipatnam, Hyderabad, Telangana
Dr. G. Anitha Mary, Dean of Informatics, Dept of MSc Data Science, Loyola Academy, Old Alwal,Secendrabad-10, Telangana
Ms.P.Madhuri Paul, Assistant Professor, Department of MCA, St.Ann's College for Women, Telangana

Abstract

Basically every industry and even a few sections of the general population part are tackling distributed computing today, either as a supplier or as a buyer. Regardless of being young it's not been unbroken untouched by programmers, hackers and other "criminals" to break into the web servers. Once debilitated these internet servers will function a place to begin for leading any assaults against purchasers within the cloud. One such assault is the Denial of Service (DoS) [1] or its form Distributed Denial of Service (DDoS) attack. Particularly, aggressors can investigate vulnerabilities of a cloud framework and trade off virtual machines to facilitate substantial scale Distributed Denial-of- Service (DDoS). DDoS assaults more often than not include early stage activities, for example, multi- step misuse, low recurrence vulnerability examining, and compromising recognized defenseless virtual machines as zombies, lastly DDoS assaults through the compromised zombies. Inside of the cloud framework, particularly the Infrastructure-as-Service (IaaS) mists, the discovery of zombie investigation assaults is to a great degree troublesome. To keep vulnerable virtual machines from being traded off in the cloud, a multi-stage disseminated vulnerability identification, estimation, and countermeasure determination system called NICE has been proposed, which is based on assault diagram based logical models and reconfigurable virtual system based countermeasures is proposed. The framework and security assessments show the proficiency and adequacy of the proposed arrangement. In our, we will attempt and execute NICE.

KEYWORDS: Security in virtual systems, Cloud Computing, Intrusion Detection and mitigation.

I. Introduction:

Nowadays usage of cloud has become extremely common amongst technological users as well as corporate users. This extensive usage of cloud has led to an increasing concern in security. Improper use of cloud and cloud resources to access private data and also to deploy attacks on systems is the top trending security issue. Hackers and illegal traders employ vulnerable applications and the reservoirs in cloud to create loopholes in the systems and to the data stored in the cloud. This makes the detection of the hackers difficult in cloud and on the other hand an easy passage for them to escape successfully and unnoticed. In the conventional data centre, where data is stored and monitored in a centralized fashion by a system administer, the security holes arising due to illegal measures can be patched. However, patching of these holes in cloud, where the users have personalized control over their virtual machines, can lead to the violation of Service Level Agreements (SLAs). Again, the cloud users may themselves install and use vulnerable applications in their data centre which can lead to nefarious use of cloud.

NICE (Network Intrusion detection and Countermeasure selection in virtual network systems)[2] has been proposed to set up a protection inside and out interruption identification structure. For better assault recognition, NICE consolidates obstructing of the specific system address into the interruption identification forms. We must note that the outline of NICE does not plan to enhance any of the current interruption discovery calculations; for sure, NICE utilizes a reconfigurable virtual systems administration way to deal with distinguish and counter the endeavors to trade off VMs, consequently anticipating zombie VMs. In view of the aggregate conduct of VMs, NICE can choose suitable activities. Utilizing this methodology, NICE does not have to piece activity streams of a suspicious VM in its initial assault stage. The commitments of NICE are exhibited as takes after:

- A new multi-stage circulated system interruption identification and anticipation structure in a virtual systems administration environment that catches and reviews suspicious cloud movement without intruding on clients' applications and cloud administrations.
- NICE consolidates a product changing answer for isolate and review suspicious VMs for further examination and insurance. Through programmable system approaches, NICE can enhance the assault discovery likelihood and enhance the strength to VM misuse assault without interfering with existing ordinary cloud administrations.
- NICE enhances the usage on cloud servers to minimize asset utilization. Our study demonstrates that NICE devours less computational overhead contrasted with intermediary based system interruption recognition arrange.

Additionally, NICE occasionally checks the virtual framework vulnerabilities inside of a cloud server, and after that in light of the seriousness of distinguished vulnerability towards the shared assault objectives, NICE will choose whether or not to put a zombie in system assessment state. Once a VM enters assessment state, Deep Packet Investigation (DPI) is connected, and/or virtual system reconfigurations can be conveyed to the assessing VM to make the potential assault practices unmistakable. NICE is supposed to identify and alleviate collective assaults in the cloud virtual systems administration environment. The arrangement researches how to utilize the programmability of programming changes based answers for enhance the discovery exactness and annihilation casualty abuse periods of synergistic assaults. The framework execution assessment exhibits the attainability of NICE and demonstrates that the proposed arrangement can altogether lessen the danger of the cloud framework from being abused and mishandled by inside and outer assailants. NICE just examines the system IDS way to deal with counter zombie explorative assaults. So as to enhance the discovery precision, host-based IDS arrangements are expected to be consolidated and to cover the entire range of IDS in the cloud framework. We will explore the versatility of the NICE arrangement by examining the decentralized system control and assault investigation model.

We develop a virtualized environment and install Xen server in it to create a cloud environment. Then we deploy two or three virtual machines in them. Each virtual can either run an application or not. These virtual machines are Xen clients. They access the operating system and all the software of the Xen server by sharing their ip address with the server. The basic objective is that at a particular time we will trigger all the virtual machines to launch an attack on a target whose positionis irrespective. The target can be in a different server.

The main goal is to synchronize the virtual machines and to make them act as DDOS agents so that they can launch an attack together on the target. The attack needs to be triggered so that the virtual machines need be told when to start sending DDOS attacks. The increased count of attacks coming from the IP addresses within a stipulated time period will lead to the automatic blocking of that IP address. However, if that time span is over, that particular IP address will be able to send requests. Similarly if the load of requests increases, it will be blocked and denied access. At the same time when one or two IP addresses are blocked due to their large number of requests or attacks, other IP addresses can freely access the target.There is also a monitor which will run in the background keeping a record and also displaying the requests come from which IP addresses along with the timeof the current and last incoming request.

Distributed Denial Of Service (DDoS)

A DDoS assault is a dangerous attempt to make a server or a system asset distracted to clients, for the most part by briefly intruding on or suspending the administrations of a host joined with the Internet. Distributed Denial of Service (DDoS) attack, in which one PC and one web association is utilized to surge focused on asset with bundles, a DDoS assault utilizes numerous PCs and numerous Internet associations, regularly disseminated internationally in what is alluded to as a botnet.

DDoS attacks are broadly speaking divided into 3 types:

- Volume Based Attacks: As cited by Tao Peng, Christopher Leckie, and Kotagiri Rama mohana rao Department of Computer Science and Software Engineering, The University of Melbourne, Australia, volume based attacks Incorporates UDP surges, ICMP surges, and other spoofed packet surges. The assault's objective is to immerse the data transfer capacity of the assaulted site.
- Convention Attacks: As referred by Jelena Mirkovic, Computer and Information Sciences Department University of Delaware Newark, Peter Reiher Computer Science Department UCLA Los Angeles, CA, convention attacks incorporates SYN surges, divides bundle assaults, Ping of Death, Smurf DDoS. This kind of assault devours genuine server assets, or those of moderate correspondence hardware, for example, firewalls and burden balancers, and is measured in Packets every second.
- Application Layer Attacks : As referred by Yonghua You, Queen's University, Kingston and Zulkernine M, application layer attacks incorporates Slowloris, Zero-day DDoS assaults, DDoS assaults that objective Apache, Windows or OpenBSD vulnerabilities. Contained apparently true blue and honest solicitations, the objective of these assaults is to crash the web server, and the size is measured in Requests every second.

Specific DDoS Attacks Types

Some specific and notably in style and dangerous styles of DDoS attacks include:

UDP Flood

This DDoS assault influences the User Datagram Protocol (UDP), a session less systems administration convention. This kind of assault surges arbitrary ports on a remote host with various UDP parcels, creating the host to over and over check for the application listening at that port, and (when no application is discovered) answer with an ICMP Destination Unreachable parcel. These procedures saps host assets, and can eventually prompt distance.

ICMP (Ping) Flood

Comparable on a fundamental level to the UDP surge assault, an ICMP surge overpowers the objective asset with ICMP Echo Request (ping) bundles, for the most part sending pings or requests as quick as could be expected. This sort of assault can expend both active and approaching transmission capacity, since the casualty's servers will frequently endeavor to react with ICMP Echo Reply bundles, coming about a noteworthy general framework stoppage.

SYN Flood

A SYN surge DDoS assault misuses a known shortcoming in the TCP association arrangement (the "three-way hand shake"), whereby a SYN solicitation to start out a protocol association with a bunch should be replied by a SYN-ACK reaction from that hosts, and afterward affirmed by an ACK reaction from the requester. In a SYN surge situation, the requester sends different SYN asks for, yet either does not react to the host's SYN-ACK reaction, or sends the SYN asks for from a satirize IP address. In any case, the host framework keeps on sitting tight for affirmation for each of the solicitations, tying assets until no new associations can be made, and at last bringing about dissent of administration.

Ping of Death

A ping of death ("POD") assault includes the assailant sending numerous distorted or vindictive pings to a PC. The greatest bundle length of an IP parcel (header) is 65,535 bytes. Nonetheless, the Data Link Layer typically stances points of confinement to the greatest casing size - for instance 1500 bytes more than an Ethernet system. For this situation, a vast IP bundle is part over numerous IP bundles (known as parts), and the beneficiary host reassembles the IP pieces into the complete parcel. In a Ping of Death situation, taking after vindictive control of section substance, the beneficiary winds up with an IP parcel which is bigger than 65,535 bytes when reassembled. This can flood memory cushions designated for the bundle, bringing on disavowal of administration for genuine parcels.

Slowloris

Slowloris is a profoundly focused on assault, empowering one web server to bring down another server, without influencing different administrations or ports on the objective system. Slowloris does this by holding whatever number associations with the objective web server open for whatever length of time that could reasonably be expected. It finishes this by making associations with the objective server, yet sending just a fractional solicitation. Slowloris always sends more HTTP headers, yet never finishes a solicitation. The focused on server keeps each of these false associations open. This in the end floods the greatest simultaneous association pool, and prompts disavowal of extra associations from honest to goodness customers.

NTP Amplification

In NTP Amplification assaults the culprit misuses publically-available Network Time Protocol (NTP) servers to overpower the focused on server with User Datagram Protocol (UDP) activity. In a NTP enhancement assault, the question to-reaction proportion is anyplace somewhere around 1:20 and 1:200 or more. This implies that any aggressor that acquires a rundown of open NTP servers (e.g., by utilizing apparatus like Metasploit or information from the Open NTP Project) can without much of a stretch create an overwhelming high-transfer speed, high-volume DDoS assault.

HTTP Flood

In HTTP surge DDoS assaults the aggressor abuses apparently genuine HTTP GET or POST solicitations to assault a web server or application. HTTP surges don't utilize twisted parcels, ridiculing or reflection strategies, and require less data transfer capacity than different assaults to cut down the focused on location or server. The assault is best when it constrains the server or applicationto designate the greatest assets conceivable because of every single solicitation.

Zero-day DDoS Attacks

"Zero-day" is simply obscure or new assaults, abusing vulnerabilities that no patch has however been discharged. The term is surely understood amongst the individuals from the programmer group, where the act of exchanging Zero-day vulnerabilities has turn into a prevalent action.

II. Literature Survey:

This segment manages a percentage of the current systems, a review of the work did via analysts in the space of security difficulties and unwanted distributed denial of service attacks crosswise over decentralized systems. The overview of papers is done to know the current methods being utilized for security difficulties like mitigation against such attacks and strategies to prevent them in future. This extensive usage of cloud has led to an increasing concern in security. Improper use of cloud and cloud resources to access private data and also to deploy attacks on systems is the top trending security issue. Hackers and illegal traders employ vulnerable applications and the reservoirs in cloud to create loopholes in the systems and to the data stored in the cloud. This makes the detection of the hackers difficult in cloud and on the other hand an easy passage for them to escape successfully and unnoticed. In the conventional data centre, where data is stored and monitored in a centralized fashion by a system administer, the security holes arising due to illegal measures can be patched. However, patching of these holes in cloud, where the users have personalized control over their virtual machines, can lead to the violation of Service Level Agreements (SLAs). Again, the cloud users may themselves install and use vulnerable applications in their data centre which can lead to nefarious use of cloud.

As proposed by Duan et al[3] of Florida State University in OpenFlow-Based Intrusion Prevention System in Cloud Environment methods Volume 9 Issue 2, March 2012, ISSN: 1545-5971, techniques were proposed to identify the compromised VM or VMs or the spam zombies. Care was taken to identify the IP addresses and put them in blocked lists. After that, no requests were tended to from those blocked IP addresses. At the same time when requests came from other sources they were allowed easy access. This blocking of IP address temporarily ensured that the target site does not crash and also that the resources are not consumed unnecessarily by attackers and hackers. The strategies are so devised that the server is not throttled by unending requests, this helps prevent distributed denial of service [4]attacks and helps the server to tend to actual authentic requests.

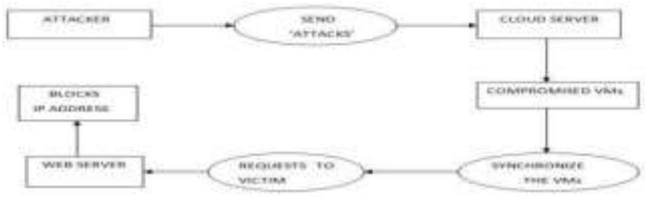


Figure1: Architectural Diagram of theProposed System

In this system, a stream of requests or attacks is sent one after the other to the target site. The target site is residing on a different server; rather the location of the target site is not important. The main attacker uses compromised virtual machines to coordinate, synchronize and launch an attack on the victim site. After the attack command is issued, the compromised VMs launch an attack. Once the target site notices the innumerable attacks using up all the resources, it blocks those particular IP addresses for a dedicated time span. There are many other techniques to prevent DDoS attacks. After that time span those IP addresses are free to send requests (attacks) again. However, if the same continues, they will be blocked again. During that time when a particular IP address is blocked, other IP addresses are free to send requests. All these steps are taken to ensure that a particular server or site is not blocked by innumerable requests or attacks coming from compromised VMs or zombie machines.

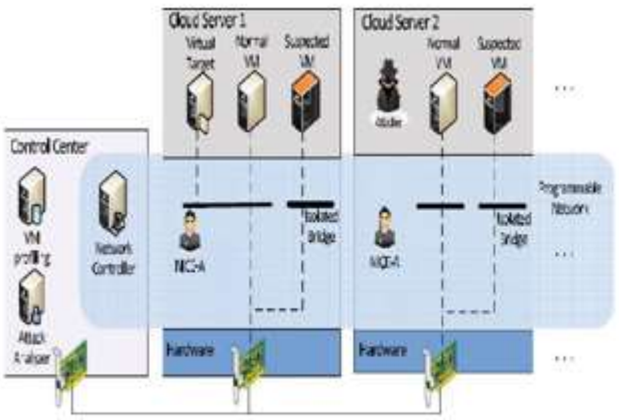
NICE aims to implement a thorough in built defense framework which aims to mitigate DDoS attacks and also take measures to prevent them in the future. It consolidates a product changing answer for isolate and review suspicious VMs for further examination and insurance. Through programmable system approaches, NICE can enhance the assault discovery likelihood and enhance the strength to VM misuse assault without interfering with existing ordinary cloud administrations. It enhances the usage on cloud servers to minimize asset utilization. Our study demonstrates that NICE devours less computational overhead contrasted with intermediary based system interruption recognition arrangement.

III. System Design:

Architectural Design

The architectural design is concerned with establishing a basic framework of a system. It involves identifying the major modules or sections of the system and communications between these components. In the following sub-sections we delve into the design aspects and the sub systems involved in this architecture.

Block Diagram



The block diagram consists of three modules and their functions are described below:

Attacker module

The most significant job of the attacker is to synchronize the compromised VMs to launch an attack on the target. Italso sets the time duration for the attack. Also, it receives notifications whether the attack was successful or not.

Compromised VM module

The compromised VMs or the zombie machines wait for the attack command from the attacker. The command also contains other parameters, namely, which site to attack and the duration of the attack. These VMs wait for response from the target site implying whether the attack was successful or not. This response is then forwarded to the main attacker.

Web server or target site module

The target site after receiving the request from any sender checks the IP address of the sender. Target server checks the IP address is present in the database or not. If the IP is already present, it checks the last visit time and increment the visit count by one for the IP. If the IP is not present, it stores the IP details and its current time as the last visit time. The target server finds the number of visit count from the IP within five minutes, and if the count is more or equal to the pre-defined threshold value, the target server puts the IP address in the block list and suspends the IP for 10 minutes. For the next ten minutes any request from the compromised VMs IP will be blocked. This disables redundant loss of information and resources since the IP address from where the pings were coming are now blocked [5].

IV. Implementation:

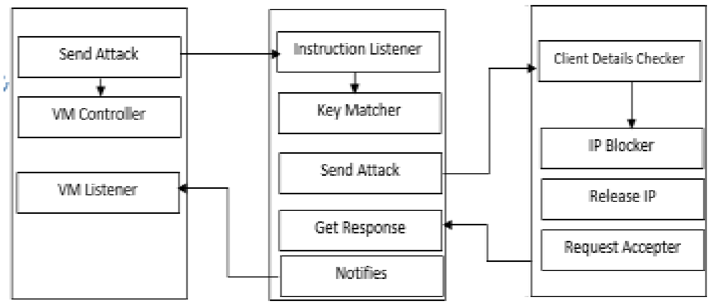


Figure2: Implementation of the ProposedSystem

Algorithms between the Attacker, Compromised VMs and the Target

Algorithm to authenticate the Attacker

- **Step1:** The Attacker launches an attack command to the compromised VMs.
- **Step2:** The compromised VMs or the zombie machines listens to the main Attacker and receivesthe URL to be attacked and the time duration and the key.
- **Step3:** The compromised VMs verify the key with the Key Server.
- **Step4:** If the key is verified then the authentication is confirmed and the zombies start sendingthe request to the URL Step5: The VMs waits for the notification or response from the target server and updates to the attacker.
  - **Step5:** If the key is not matched the compromised VMs does not attack the target server.

Algorithm for communication between attacker, compromised VMs and Victim

- **Step1:** Attacker begins the procedure in the traded off VMs and the procedure begins listeningat port number 5001, 5002 and 5003 separately.
- **Step2:** Attacker sets the objective URL, time span of assault and sends the direction to the traded off VMs utilizing TCP/IP convention.
- **Step3:** The traded off VMs gets the direction through the port number.
- **Step4:** The traded off VMs subsequent to accepting the objective URL and term of assault begins sending solicitation to the Victim server utilizing HTTP convention.
- **Step5:** Victim begins getting the solicitations and sends reaction to the traded off VMs utilizing HTTP convention.
- **Step6:** If the quantity of solicitations sent by the zombies surpasses a specific edge, the casualty puts the arrangement of those IP addresses in the blocked rundown for a specific time of time.

- **Step7:** After that time farthest point is over, the traded off VMs or zombies are allowed to send asks for once more. However in the event that they again begin throttling the objective and superfluously devour discriminating assets, they are blocked once more.

  Algorithm for setting the time, duration and URL

- **Step1:** Attacker sets the time of attack as t sec.
- **Step2:** Attacker sets the time duration of attack as d sec.
- **Step3:** Attacker sets the target server URL as url as http request. Step4: Attacker sets the port number 2000 that it is listening to.
- **Step4:** If the time of the compromised VMs are different from the attacker then they set thetime difference accordingly so that all of them have the same time

Attacker waits for compromised VMs response and starts listening at port number 2000. Compromised VMs sends response to the attacker through the port number 2000 using TCP/IP protocol.

The above mentioned are the algorithms which have been used to develop the proposed system. The first algorithm was for authentication purpose. We use a key to determine whether the attack command has actually come from the main attacker or not. The compromised VMs check for the key on receiving the attack command, if the key matches, the attack is synchronized and launched,however if the match is unsuccessful, the VMs sit tight.

The second algorithm deals with the communication between the main attacker, the zombie machines and the victim or the target server. We have used socket programming, java and TCP/IP connection to establish a communication between the above mentioned counterparts. The compromised VMs are at port numbers 5001, 5002 and5003 respectively. They all have different IP addresses too. These attackers keep listening through their port numbers for the attack command. The main attacker sends the target URL or site, the duration of attack to the zombie machines with the help of TCP/IP convention. Upon receiving the attack command, the VMs synchronize and launch an attack on the target URL with the help of HTTP request/response service. That the target site is successfully hit by innumerable pings, the main attacker gets notifications or responses that the attack was successful, this communication is done via HTTP. The target site continuously monitors all the requests it receives. If it discovers that there are innumerable requests coming from one or more than a single set of IP addresses, it then puts those IP addresses in the blocked list for a particular time period. After that time span expires, those IP addresses are free to send requests again. However if they again start throttling the server, and redundantly using all the critical resources so that the target
cannot tend to authentic requests, those IP addresses are again blocked for a designated time span. While those IP addresses are blocked, other users can easily access and avail the resources of the target site without any hindrance.

The main attacker which monitors over all the compromised VMs, synchronizes them, makes sure that they are all up and running, listens to port number 2000 for the notifications received if the attack was successful or not.

V. Future Work:

In future we will try to provide better authentication schemes in the target server. This authentication scheme is for the zombie machines to know that the attack command is actually coming from the authentic source. One better scheme will be if we try to implement a priority based customer ID based blocked scheme. The priority based scheme can be achieved based on the premium server. We can spot a premium server over time and by its behavior. This will enable scalability and efficient working since all the data and software will be stored in the premium server.

VI. Conclusion:

NICE (Network Intrusion detection and Countermeasure selection in virtual network systems) has been proposed to set up a protection inside and out interruption identification structure.

For better assault recognition, NICE consolidates obstructing of the specific system address into the interruption identification forms. NICE utilizes a reconfigurable virtual systems administration way to deal with distinguish and counter the endeavors to trade off VMs, consequently anticipating zombie VMs. Additionally, NICE occasionally checks the virtual framework vulnerabilities inside of a cloud server, and after that in light of the seriousness of distinguished vulnerability towards the shared assault objectives, it is supposed to identify and alleviate collective assaults in the cloud virtual systems administration environment. Our study demonstrates that NICE devours less computational overhead contrasted with intermediary based system interruption recognition arrange

NICE just examines the system IDS way to deal with counter zombie explorative assaults. So as to enhance the discovery precision, host-based IDS arrangements are expected to be consolidated and to cover the entire range of IDS in the cloud framework. We will explore the versatility of the NICE arrangement by examining the decentralized system control and assault investigation model.

VII. REFERENCES:

1. Detecting Ddos Attacks In Cloud Computing Environment By A.M. Lonea, D.E. Popescu, H. Tianfield. INT J COMPUT COMMUN, ISSN1841-9836 8(1):70-78, February, 2013.
2. Comparison Of Network Intrusion Detection Systems In Cloud Computing Environment By Vanathi, R. Dept. Of Computer. Science, Coimbatore Inst. Of Eng. & Technol., Coimbatore,India And Gunasekaran, S. Computer Communication And Informatics (Iccci), 2012 International Conference. Published By Ieee.
3. Network Intrusion Detection And Countermeasure Selection By Chun-Jen Chung Dept. Of Computer Science, Arizona State Univ, TianyiXing, Jeongkeun Lee, Dijiang Huang, 2013. Ieee Publisher.
4. B. Joshi, A. Vijayan And B. Joshi ,Securing Cloud Computing Environment Against Ddos Attacks, Proc. IEEE International ConferenceComputer Communication And Informatics (ICCCI ',12), Jan. 2012.
5. Ritika Saroha M.Tech (Network Security) Assistant Lecturer In CSE/IT Dept. BPSMV Khanpur Kalan And Sonipat BPSMV Khanpur Kalan, Sonipat , Intrusion Detection In Virtual Systems International Journal Of Computer Science Engineering And Technology IJCSET ,May 2014,Volume 4, Issue 5,158-160