International Journal of Advanced Computer Technology (IJACT)
ISSN:2319-7900

SOME ABSTRACTIVE CONCEPTUAL DEDUCTION OF MOBILITY BASED NETWORK INTRUSION DETECTION SYSTEM FOR LIVE MITIGATION OF VIRTUAL MCHINES

[1]**Waziri Onomza Victor, PhD**, [2]Dogonyaro Noel Moses, [3] Matthew O. Adigun, PROF
[1, 2] Departement of Cyber Security, Federal University of Technology Minna, Nigeria,.
[3]University of Zululand, Centre for Excellence, Computer Science Department, KwaDklangezwa Campus, South Africa,

Abstrct: *Cyber crimes activities have become a worst incriminating part of everyday life of both corporate world and the general public. The digital crime phenomenon has achieved what one may call the overwhelming factor. A few years ago, incidents of this kind were few and almost entirely the works of computer and telecommunications experts that individually, or as members of groups, came to define what is now identified as the underground hacker culture due to the inherent vulnerabilities in the early computers designs and software development. Such acts were carried out, as is often claimed, to prove and not to harm the computing devices; but today, it is worrisome to observe that the criminals of the digital age are driven by rather sinister motives and the numbers of incidents have increased with the multiplicity and to dangerous intentions to match in the most cases in financial and political institutions both at local and international levels. In this paper, we propose more efficient solution to perform safe screening of the target systems and take only the relevant data and systems to the lab. Such screening can be performed by using the local computational and communication resources of the targets to execute spam filtering. uctn abstractive ded reviews various digital forensics investigations and uses open source tools that give justification for their applications. We present the Bluepipe concept for an on-the-sop forensic investigation in a non-invasive way that is sensitive to privacy instead of removing suspect machines from site for a Laboratory based investigation.*

Keywords: Network Intrusion Detection Systems (NIDS), Virtual Machines, Mobility-based NIDS evasion, IP fragmentation

1 Introduction

A virtual Machine is software implemented on abstractive framework of the underlying hardware which is presented to the application layer of the system. Live migration of these VMs involves moving an entire hardware machine state (this include migrating the CPU state, memory content, storage contents) of a running VM from one physical machine on which it is currently executing (called the source machine) to another physical machine (called the target or destination machine) with significant minimal service disruption assail.

Migration of a VM in a Wide Area Network (WAN) requires that the migration is done outside the local subnet. In this type of migration, the operating system will have to obtain a new IP address which is within the destination subnet. In migrating network and storage connections, TCP connections survive VM migration and the applications encapsulated with the VM is not interrupted as regards network connections as far as the source and destination machines are on the same subnet. However, TCP connections disconnect break when VM migrations occurs across subnets. Sumit K.B et al (2011) Modern technologies (Service providers) in the 21st century try to ensure that live migration of VMs are done with minimal downtime on their services. However, these models require that the VM retains its network address in order to keep the migration transparent from network perspective and maintain uninterrupted network connectivity. Internet Protocol assumes that a host IP address uniquely identifies the host's point of attachment to the internet. Thus if migration is not restricted to movement within an IP subnet, stringent measures must be put in place to allow a VM to seamlessly use its pre-migration IP on a different subnet rather than the previous one.

In this paper we would describe three (3) evasion tactics based on node mobility in a WAN infrastructure, and would show practical applicability of the proposed evasion strategy through a proof of various attacks that may occur during a live migration of VM.

The Paper is structured as follows: Section 2 reproduces some literature reviews as related to the topic, Section 3 describes forms of NIDS evasion attacks with various

ISSN:2319-7900

subsections that involves the ICP Tramspot layer etc. In section 4, gives solutions to. We point out how an attacker can take advantage of these lapses to attack his victim. Section 6, discusses virtual machine migration process. This process is illustrated in figure 1. The figure illustrates the various phases of memory page and CPU transfer operation within a specific time frame. Prototype design architecture of live virtual machine migration is discussed in section 7. The system is composed of three main entities; source subnet agent, target subnet and migration manager. Each of these entities has its main function in the design. Section 8 describes mobility-based NIDS evasion (attacker model). IP fragmentation attacks based on reassembly timeout model was demonstrated in section 9 with figures to illustrate how an attacker can exploit the different in time of fragmentation reassemble of an NIDS to lunch its attack by sending false packets on the network. Possible solutions to all forms of attack scenario were presented in section 10, while concluding remark is in section 11.

2. Related Work

As virtualization continues to be gaining popularity in enterprise and organizational networks, most operators are switching to live migration of virtual machines for the simple facts that it brings about load balancing and management of network resources. Jon O. et al (2004) demonstrate how the most popular VMM, Xen and VMware are vulnerable to practical attacks targeting their live migration functionality.

Live migration of virtual machines as discussed by Christopher C. et al (2005) shows that by integrating live operating system into the Xen virtual machine monitor, there is bound to be an improvement in work load within clusters and data centers with a huge among of dedicated bandwidth, there is less downtime and service failure.

Michael C. et al (2011) in their paper present a novel NIDS evasion strategy that allows attackers to exploit network mobility-based evasion by comparing traditional evasion techniques and node mobility. A new attack strategy called mobility-based NIDS evasion was discussed in which an attacker can exploit that only a modern NIDS can detect.

In as much as user transparent live migration is one of the most interesting features of virtual machines environment, modern migration techniques requires that the VM retains its IP network address; and typically movement is restricted within the same IP subnet. Ezra S. et al (2009) introduced a new frame work that will efficiently support live migration of virtual machines across IP subnets. They also study numerous approaches for IP mobility.

Boris D. et al (2011) discussed on the integration of trusted computing technologies into virtualization. Their discussion was focused on the problems of enabling secure migration of Virtual Trusted Platform Module (V-TPM-based) virtual machines in private clouds. Requirements for the VM-vTPM migration in internal virtualized environment were analyzed.

3 Forms of Attacks

There are many forms of attacks on a network infrastructure these days. In this paper will would briefly discuss there (3) major attacks. All attacks entails that an attacker specifically manipulate his network usage to create an abnormal or pathological streams of traffic.

a) Insertion

An insertion involves an attacker stuffing the system with some fake or invalid packets on its victim. An IDS can accept a packet that an end-system rejects. An IDS can do this costly mistake because it might assume that the end-system has accepted and processed the packet when actually it has not. The attacker can exploit this condition by sending packets to an end-system that it will reject, but that the IDS will think are valid. By doing this, the attacker is "Inserting" data into the IDS. An experienced attacker can apply the insertion technique to defeat signature analysis of an IDS

b) Evasion:

Evasion is a term used to describe techniques of by-passing an information security device in a network in order to deliver an exploit, attack or other malware to a target network or system without detection. Evasion attacks foil pattern matching in a manner quite similar to insertion attackers. In this type of attacks, the attacker causes the IDS to see a different stream of data than the end-system can see. At this time, the end-system sees more than the IDS, and the information that the IDS misses is critical to the detection of the attack.

c) IP Fragmentation:

IP fragmentation signifies the process of breaking down of IP into smaller packets and reassembled at the destination by a decoder. This process allows the same information to travel over different types of network media (which may have different packet size limits) without limiting the entire protocol to an arbitrary small maximum packets size. IDS that do not correctly reassemble fragments can be attack simply by ensuring that all data is exchange between machines using artificially fragmented packets. Streams of IP fragments usually arrive in order. However, a destination system must be able to reassemble a datagram from fragments that arrive out of order. Since fragments usually arrive in order, it's easy to make the mistake of assuming that they are correct. An IDS that does not properly handle fragments that arrive out-of-order is prone to attacks; an attacker can intentionally scramble his fragment streams to elude the IDS. It is therefore advisable for an IDS not to reconstruct packets until all fragments have fully arrive and seen. Another strategy that an IDS can adopt is to store received fragments until the stream of fragments can be reassembled into an entire IP

ISSN:2319-7900

datagram. An IDS can be attacked by flooding the network with partial, fragmented datagram which will never be completed. A naïve IDS may run out of memory as it attempts to cache each fragment, since the fragmented packets are never completed. An IDS that is found of dropping old, incomplete fragment stream so as to reduce overload, must do so in a logical and consistent way because, it might be vulnerable to insertion or evasion attacks. Thomas H.P et al (2005)

d) TCP Transport Layer Problem

A lot of attacks detected by most IDS occur over TCP connections. This voice down to say that an IDS must be able to reconstruct the flow of data passing through a stream of TCP packets. If an IDS is lacking in doing this consistently, it will expose the target system it is monitoring, it will expose the system or network to sever attacks.
TCP implements a 'reliable' sequenced stream protocol. By 'reliable' we mean each end of a connection can determine whether data that was sent was successfully received or not. TCP is 'sequenced' because it employs the use of sequence numbers to know the location of any piece of data within a stream. In order for an IDS to reconstruct an information flowing through a TCP connection, it must figure out what sequence numbers are being used. This process is called "synchronization". A scenario in which an ID becomes confused about the current sequence numbers is called "de-synchronization". When an IDS is de-synchronized from a connection, it cannot accurately reconstruct the data being passed through the connection. In this way, we can say that the IDS system is blind. Thus a major goal of an attacker is to see that this condition occurs. Apart from sequence numbers, TCP also keep track of other pieces of information about a connection. TCP defines a flow-control mechanism that prevents one side of the connection from sending too much data for the other side to process ortherwise, this would result data linkage; this is tracked through each side 'window'. TCP also allows out-of-band data to be sent in a stream, by using the 'urgent pointer'.

e) Node Mobility in Live Migration of Virtual Machines

In recent times, live migration of Virtual Machines (VMs) technologies has a better advantage in the sense that migration of VM is done with significantly reduction of downtime on the part of their service providers. However, these technologies require that the VM keep to its network address for transparency reasons and to maintain un-interrupted network connectivity. Internet Protocol required that a host's IP address uniquely identifies the host's point of attachment to the internet. Thus, if migration is not to be restricted to movement within an IP subnet, additional measures must be taken to allow a VM to use its pre-migration IP on a different subnet.

f) Virtual Machine Migration Process

A virtual machine migration consists of several phases. Each phase is unique and has particular characteristics in terms of VM and service availability, duration etc.
Figure 1 below illustrates the main process of migration events. Most often the migration starts with a memory transfer phase in which memory pages of the VM are moved from the source machine (Hypervisor) to the target machine. During this phase, the VM is fully operational at the original location and its services are available to users. Note that from time T_1 to T_4 there is bound to be service outage after this stage, the VM is suspended at the source location after which all the remaining memory pages and CPU state are transferred to the target hypervisor. Finally, when all the necessary information is on the target machine (hypervisor) the VM is brought up on the target machine while the VM at the source machine is suspended.

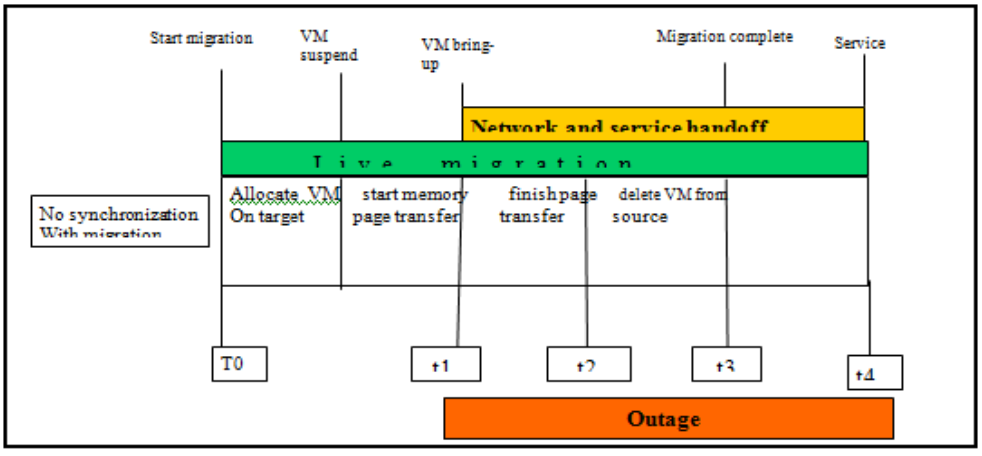

Figure 1: A typical subnet event

g) Design Architecture of Live Virtual Machine Migration

Figure 2 below represents the design architecture of a live VM migration. The system is composed of three (3) main entities: source subnet agent; target subnet agent and a migration manager. The migration manager major function is to perform all necessary network configuration settings and changes that is made during the migration process. In addition, it also monitors all relevant migration events and it is also responsible for the coordination of all operations needed for the network configuration changes during these events. The manager interacts with both source and target tunnel agents and with the source hypervisor (to initiate migration command) using Standard Secure Shell (SSH) sessions. The main role of the subnet agents are to manage the IP tunnel (i:e create/destroy) and perform necessary forwarding and routing to send relevant packets to their destination.

In an intra-subnet case as proved by Ezra S. et al (2005), the VM maintains its MAC address during migration and therefore only the switch needs to be updated with the new physical location of the VM. In order to reduce time and fast transition of this phase, a VM initiates an unsolicited Address Resolution Protocol (ARP) message announcing its new location. Like in mobile IP, all the machines which reside on the VM' original subnet need to update the Hypervisor Workstation (HW) address associated with the MAC address of the source agent.

The second 'update' issue occurs on the cross-subnet setup after the VM is up in the new subnet, it must send all its traffic through the target agent, in order to communicate with other machines. In a bit to achieve that, the target agent acts as an ARP proxy for all machines residing in the VM's original subnet (including the gateway). This type of issue is not common in the intra-subnet case, because the VM stays in the same subnet and communicates with all machines directly; therefore it does not need to update its ARP table with the new MAC addresses. The migration manager is to initiate such ARP messages as soon as the VM resumed on the target side.

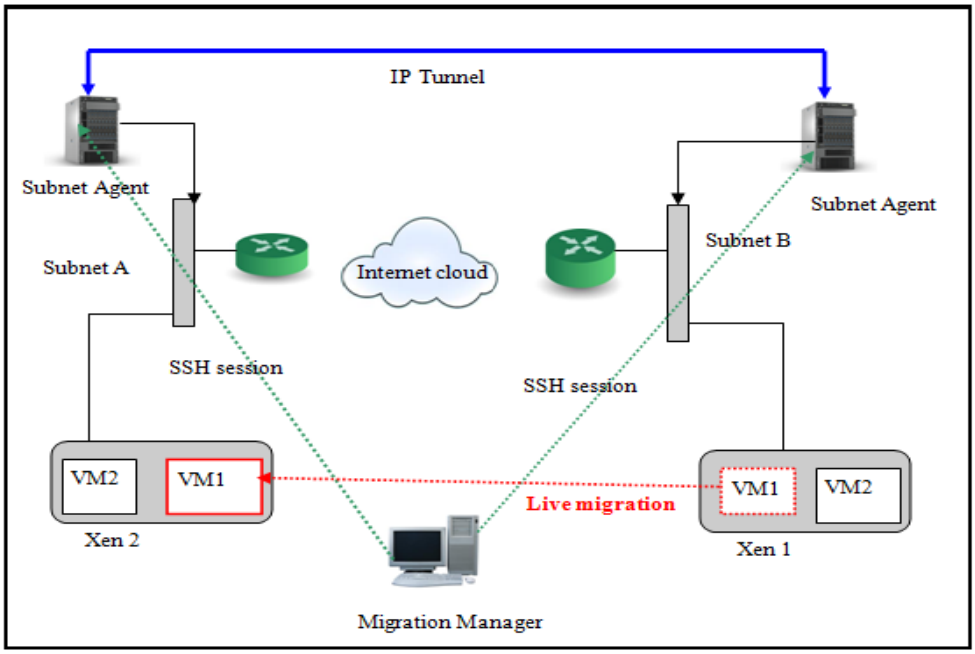

Figure 2: Prototyping Design Architecture

h) Mobility–Based NIDS Evasion- (Attacker Model)

In this section we are to describe a modern evasion technique, known as mobility-based NIDS evasion that an attacker can exploit and attack his victim during live migration of virtual machine across two subnets as it can be shown in figure 3 above. The main idea of mobility-based evasion deployed in this paper is for the attacker to coordinate and monitor node mobility during migration and the traditional NIDS evasion techniques based on fragmentation of the attack payload during migration of huge number of pages on a Virtual Machine.

Let us consider our protype network illustrated in figure 2 above in which two nodes are communicating through the internet (Xen 1 and Xen 2). We assume that Xen 1 and Xen 2 are equipped with state- of-the art NIDS that monitors all the traffic involved during VM migration and that but

ISSN:2319-7900

Xen1 and state implementing the same IP versions (ie IPV6 or IPV4). We assume the presence of an attacker 'A' that can eaves drop, modify, insert or delete configuration settings in the network via the subnets agents at the source and destination end. Note that we had said earlier that the main function of the subnet agents are to manage IP tunnel (i:e, create /destroy), and also the subnet agents is responsible for the forwarding, routing and push relevant packets to their desire destination.

We assume that the attacker 'A' is interested in abusing the migration protocol to increase her bandwidth in the network (that is starting his own virtual machine, acquiring information about the transferred VM etc).

In figure 3 below, the attacker aim at exploiting remote vulnerability of both the source and target VM by sending packets containing malicious payload. The attack is in two portions, Network 1 (source) and Network2 (destination. Since fragmentation of IP packets is discouraged in IPV6 protocol, (Michael C. et al 2011) and can easily detected by modern NIDS as an abnormal network activity, the attacker sends the two attack portions inside two non-fragmented TCP packets having the same sequence numbers via the subnet agents. The sequence of activities performed by the attacker can be shown as follows:

We will describe two different attack scenarios. Note that for a transparent VM migration, a VM keeps its original IP address while moving to a different subnet and this is done to maintain network connectivity and to reduce downtime throughout the migration process.

a) The mobile node (attacker) sends the first attack portion;
b) The mobile node (attacker) roams to the destination network (Network 2);
c) The mobile node (attacker) sends a second (last) attack portion.

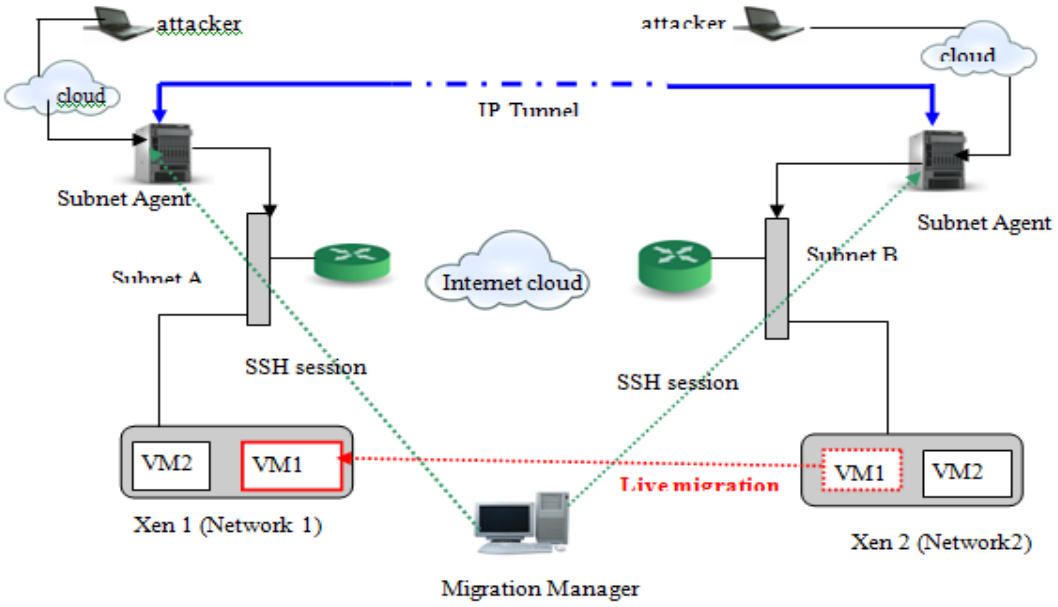

Figure 3: Mobility-based NIDS Evasion Attacker Model

In a typical network environment that does not involve live migration of VM, the attacker sends the first portion of his attack and this will be intercepted and analyzed by an NIDS. The NIDS in coping with this attack updates its state information. Depending on how node mobility is implemented, only a state-full NIDS installed in the network can be able to detect an intrusion attempt.

i) IP Fragmentation Attacks Base On Reassembly Time-Out Model

The second attack model that an attacker can exploit is IP fragmentation reassembly time-out that exists between an NIDS and that of its victim. IP fragmentation reassembly time-out refers to the maximum amount of time that a fragment will be held (unassembled) before its expiration and then been flushed out. These time lapse differ from one operating system to another. Most NIDS that does TCP reassembly will also has an IP fragment reassembly timer installed on them.

A typical attack scenario that an attacker can exploit is when the NIDS fragmentation reassembly time-out is less than fragmentation reassembly time-out of the victim. Let assume that the NIDS fragmentation reassembly time-out is 15 seconds and the NIDS is

ISSN:2319-7900

monitoring a Virtual Machine Monitor (VMM) with all the subnet agents(refer to figure 3) which has a default fragmentation reassembly time-out of 30 seconds. As we illustrate in figure 4 above, after sending the first fragment the attacker can send the second fragment with a delay of 15 seconds but still within 30 seconds.

Now, the victim reassembles the fragments whereas at the NIDS the fragmentation reassembly time-out parameter starts in and there is a time-out that will occur. We also not that the second fragment received by the NIDS will automatically drop because the NIDS has lost the first fragment already due to the time-out. This will make the victim receive the attack while the NIDS will not.

A second scenario is when the NIDS fragmentation reassembly time-out is more than the fragmentation reassembly timeout of the operating system running of the VMM. If we choose to use 'Snort' as our NIDS, by default Snort has a fragmentation reassembly timeout of 60 seconds.

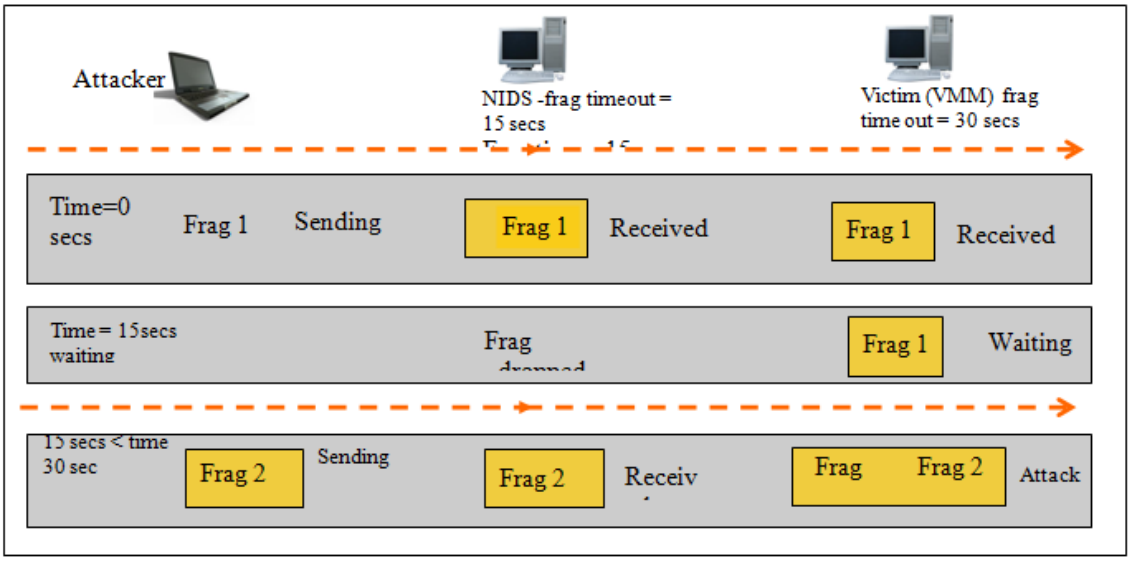

Figure 4: Attack where the NIDS fragmentation re-assembly timeout is less than the Victim's fragmentation reassembly timeout

Also, we can compare this with Linux operating system that has fragmentation reassembly timeout of 30 seconds. An attacker can exploit this as well. As illustrated in figure 5 above, the attacker has fragmented his attack packets into four segments i:e 1,2,3,4.

The attacker sends frag-2 and frag-4 with a false payload (refer to as 2^*,4^*) which are received by both the NIDS and the victim respectively. The attacker will patiently wait until the fragments reassembly timeout occurs at the victim's end and then he will drop the initial fragments (this case in 30 seconds time). Note that the victim has not received frag1, so it will quietly drop the fragments and there will be no any error message at the victim's machine. After these attempts, the attacker would send packets (1, 3) with a legitimate payload.

At this point, the victim has only fragments (1, 3) where as the NIDS has fragments (1, 2^*, 3, 4^*) respectively. Recall that the 2, 4 fragments sent by the attacker has a false payload. Since the NIDS has all the four (4) fragments, it will do a TCP reassembly. Also, since fragments 2 and 4 have false payloads, the net checksum will be invalid. This will cause the NIDS to drop the packet. However, the victim now has only fragment 1, 3. Para venture the attacker now sends fragment 2, 4 again with a valid payload, the NIDS will have only these two fragments (2, 4) with a valid payload as the previous fragments has been reassembled and dropped. Meanwhile the victim will have all the four (1, 2, 3, 4) fragments with a valid payload, and after reassembly, it will read the packet as an attack on the virtual machine.

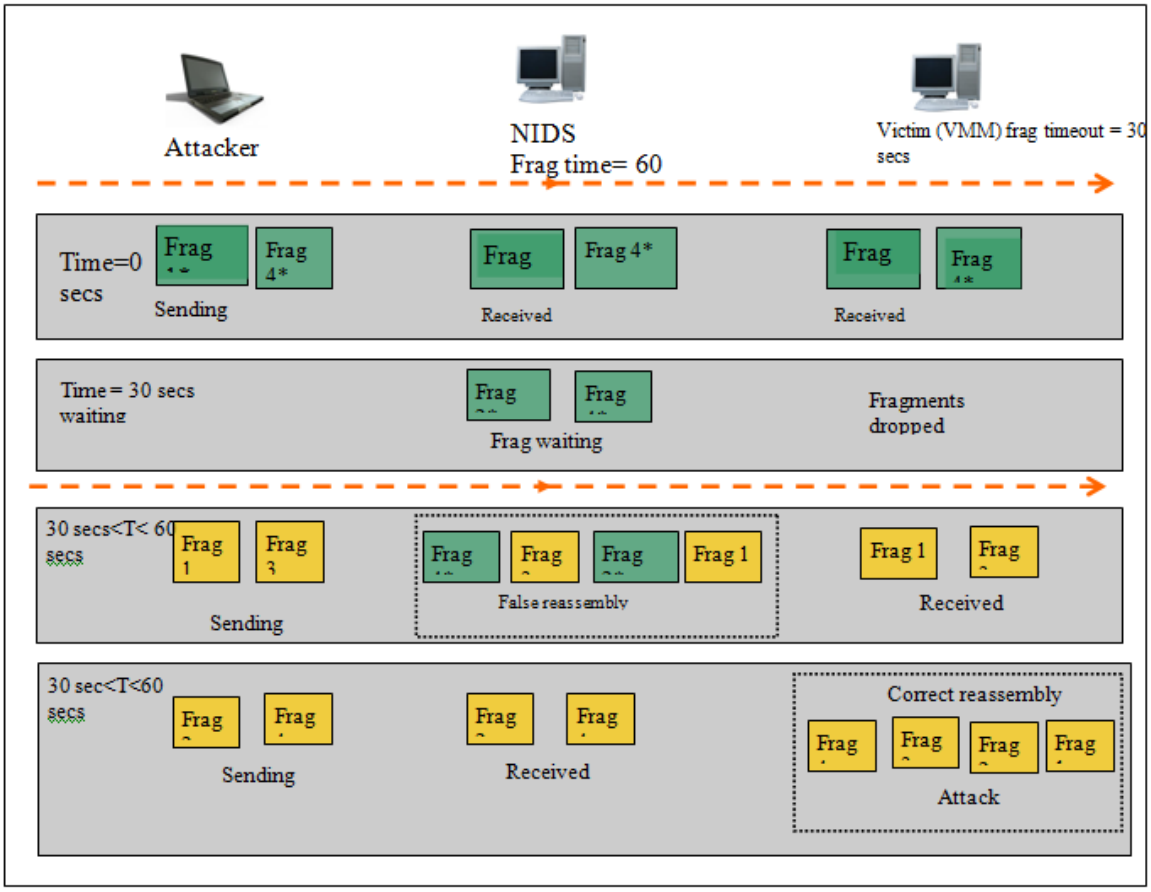

Figure 5: NIDS fragmentation re-assembly timeout is greater than the victim's fragmentation reassembly timeout.

The second 'update' issue occurs on the cross-subnet setup after the VM is up in the new subnet, it must send all its traffic through the target agent, in order to communicate with other machines. In a bit to achieve that, the target agent acts as an ARP proxy for all machines residing in the VM's original subnet (including the gateway).

This type of issue is not common in the intra-subnet case, because the VM stays in the same subnet and communicates with all machines directly; therefore it does not need to update its ARP table with the new MAC addresses. The migration manager is to initiate such ARP messages as soon as the VM resumed on the target side.

4. Solution to the Problem

A common challenge associated with mobility-based NIDS evasion is caused by fragmentation of relevant state information among geographically distributed NIDS deployed in different networks. This fragmentation pose a problem to modern NIDS from building complete state information thereby exposing them to attacks. During live migration of VMs across two subnets, this paper resolved these issues by adopting an implementing the following measures:

a) Fragmentation and reassembly timeouts of both the source and destination VMM (Xen 1, Xen 2) and that of the operating systems running on them must be the same as that of the NIDS

b) There should be enough memory space (buffer) between the source and destination VMM. The reason is that if the machine run out of memory during the migration, the VM will drop or discard incoming packets. The NIDS will find it difficult to comprehend the actual machine that the packets are been dropped (either the VMM it is monitoring or an attack).

c) There should be absolute exchange of state information among various NIDS installed on the different subnets. If this is implemented

the attacker would find it difficult to send his attacks.

d) Establishing trust relationships among cooperative NIDS as well as designing mechanisms to provide confidentiality, authentication and non repudiation of exchange of state management operations need to be compatible with live analysis of network traffic and state migration process has to be robust.

5 CONCLUDING REMARKS

Live migration of virtual machines is very key to every organization. It is of great important because it helps organization in load balancing of their network infrastructures, reduced cost of acquiring large number of infrastructures etc. However, today live migration suffers setbacks due to security challenges and service down time might arise during the process if stringent measures are not put in place for a successful migration.

In this paper we have looked at a few NIDS evasion attacks and different methodologies involved with such attacks. We also illustrate how different IDS and operating systems perform fragmentation reassembly processes. We also discussed attacks based on fragmentation reassembly by NIDS and the operating systems running on the virtual machine. An attacker can exploit these lapses to perform his attack on his victim. A prototype design architecture that would reduce all these forms of attack was illustrated in figure 2.

REFERENCES

[1]. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A ., Neugebauer, R., Pratt, I., and Warfield, A. (2003): "Xen and the art of virtualization". *Proceedings of the nineteenth ACM symposium on Operating systems principles, pages 164–177, 2003.*

[2]. Boris D, Ramya J.M, Ghassan O.K, Sidjan C. (2011): "Enabling secure VM-v TPM migration in private clouds". *Proceeding of the 27th annual computer security applications conference.*

[3]. Bose S.K and S. Sundarrajan, S. (2006). "Optimizing Migration of Virtual Machines across Data-Centers". *In Proceedings of the IEEE International Conference on Parallel Processing (ICPP) Workshops, Vienna, Austria.*

[4]. Christopher C, Keir F, Steven H, Jacob G.H, Eric J, Christian L, Ian P, and Andrew W. (2005): "Live migration of virtual machines". *Proceeding of the 2nd conference on symposium on network systems design and implementation-vol.2 (2005).*

[5]. Craig, I. (2006): "Virtual machines"

[6]. Ezra S, Gilad S, Dean L, Inbar S. (ACM 2009): "IP mobility to support live migration of virtual machines across subnets". *Proceeding of SYSTOR 2009: The Israeli experimental systems conference.*

[7]. Jacob G.H and Eric J.(2004): "Self migration of operating systems". *In proceedings of the 11th ACM SIGOPS European workshop (EW2004).*

[8]. Jon O, Evan C, Farman J. (2004): "Empirical exploitation of live virtual machine migration"

[9]. Melvin V. (2011): "Dynamic load balancing based on live migration of virtual machines: security threats and effects"

[10]. Michael C, Luca D.Z, Mirco M, Michele M. (2011): "The problem of NIDS evasion in mobile networks". *Proceeding of the 2011 IEEE conference.*

[11]. Nelson M, Lim B.H and Hutchins G. (2005): "Fast transparent migration for virtual machines". *Proceeding of the USENIX 2005 annual technical conference.*

[12]. Perkins, C. (2002). " IP Mobility Support for IPv4". *RFC 3344, IETF Network Working Group, August 2002.*

[13]. Smith J, Neir, R. (2006): "An overview of virtual machine architecture"

[14]. Sumit K.B, Scott B, Ronald S, Shrisha R. (2011): "Cloud spider combining replication with scheduling for optimizing live migration of virtual machines across Wide Area Networks". *Proceeding of the 11th IEEE/ACM international symposium on cluster, cloud and grid computing.*

[15]. Thomas H.P, Timothy N.N. (2005): "Insertion, Evasion and Denial of Service: eluding Network Intrusion Detection"

[16]. Travostino,. F, Daspit., P, Gommans,L., Jog, C., de Laat, C., Mambretti, J., Monga,I., van OB., Raghunath,S. and. Yonghui W.P (2006): " Seamless live migration of virtual machines over the MAN/WAN". *Future Generation Computer Systems, 22(8):901–907, 2006.*

[17]. Umesh D, Xiaoshuany W, and Kartic G.(ACM 2011): "Live Gang migration of virtual machines".

a. Proceedings of the 20th international symposium on high performance distributed computing.

[18]. Wood, T., Shenoy, P., Venkataramani, A. and Yousif, M. (2007): "Black-box and Gray-box Strategies for Virtual Machine Migration" *Proceedings of the 4th USENIX Symposium on Networked Systems, Design and Implementation, 2007.*

[19]. Zhang X., Huo J., and Meng D. (2010): "Exploiting data duplication to accelerate live virtual machine migration". *In proceeding of International Conference on Cluster Computing (ICCC, 2010)*

Bibliography of Authors

	Dr. Victor O. Waziri is an Associate Professor of Cyber Security Science in the Department of Cyber Security Science, Federal Federal University of Technology, Minna-Nigeria. His Computational Research is based on Computational Intelligence with Applications on Cyber Security related problems. In most cases, Matlab, Maple and Mathematica are the basis for his accessory in modeling and Simulations in Modern Cryptographic analyses. His researches area also extends into Computational Optimization and in Zero-day Malware Detections. He has published many papers in reputable Journals at both International and Local Levels. He Lectures various courses in the Department that include Cryptography, Network Security, Clouds Security, Data Mining, Computational Theory, Automata and Programming
	Moses Dogonyaro Noel obtained his B.Tech. (Physics/Computer Science) from Federal University of Technology Minna (1998), PGD in Project Management (IT projects) from International Academy of Management, Kaduna centre (2010), Advance Certificate in Base Station Sub-systems (2004) at ZTE University of Telecommunications, Shenzhen, China. He has these professional certifications to his credit: CCNA & CCNP (2004,2005) at Digital Bridge Institute, Abuja; ITIL (2007) respectively. Currently, he is a Masters Student (M-Tech.) at the Department Cyber Security Science and works as Systems Analyst in the Department, Federal University of Technology, Minna. His research works include: Network security, Cyber security, Malware detection and Mobile cloud computing security.
	Prof. Matthew O. Adigun is an outstanding and ccomplished Professor of Computer Software Engineering. He obtained his BSC, MSC and PhD at the renounced University of Ile IFE; now Obafemi Owolowo ILE IFE, Nigeria He the current Head of Computer Science at the University of Zululand, Republic of Soth Africa. Has published over one hundred academic papers at both local and at International Scene. He has attended many International Conferences and has made tremendous researches the Republic of South Africa.