Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 6, Issue 8, August (2016)

Analysis of VMware Hypervisor Security

Peter Sungu Nyakomitta
School of informatics and innovative systems, Jaramogi Oginga Odinga University of Science & Technology

Abstract – Most organizations worldwide are moving to the cloud, where their computing resources are secured and provided by a third party. The cloud vendors make intensive usage of virtualization so as to cater for the growing need for clients to share the same physical computing resources. The security of the virtual environment in which the various clients operate then becomes of great significant. In this paper, the researchers sought to analyze the security vulnerabilities of the VMware hypervisor. The objectives were to practically establish the various weaknesses of the VMware hypervisor and therefore investigate if this virtual machine monitor actually offers any protection to the guest operating system running in it. An experimental research design was used to achieve these objectives. The approach was to install a Windows 2007 host operating system, VMware hypervisor and Windows server 2003 inside this hypervisor. The Metaspoilt software was then used to test the vulnerability of the hypervisor. The results indicated that the hypervisor offered little protection to the guest operating system as indicated by the windows server 2003 fingerprinting. This study is significant in the sense that it exposes the hypervisor weaknesses which its developers and the research community can try to fix so as to protect the guest operating systems running inside the hypervisors. This will eventually ensure the security of the cloud clients whose computing services and resources run within the hypervisors.

Index Terms – Hypervisor, virtualization, guest, host, cloud, server, vulnerabilities.

1. INTRODUCTION

The hypervisor, also called the virtual machine monitor, runs on the host Operating System and allocates emulated resources to each guest operating system. According to Chandramouli, (2014), Hypervisor is a software which provides abstraction of virtually all physical computing resources. These computing resources can be the central processing unit, memory, network or storage. In so doing, it enables numerous computing stacks consisting of operating systems, middleware and application programs, which are collectively referred to as called virtual machines to be executed on a single physical host.

Moreover, hypervisors can be utilized to define a network within the single physical host, commonly referred to as virtual network. This network can then be employed to enable communication among the virtual machines that reside on that host as well as with physical and virtual machines exterior to the host Foley (2014). Under this architecture, the hypervisor functions to mediate access to physical resources, offer run time isolation among the virtual machines and facilitate a virtual network that gives security-preserving communication flow among the virtual machines and between the virtual machines and the external network.

In his study, Randell (2015) noted that majority of the hypervisor security issues come up not from the virtualization infrastructure itself but from operational issues such as the adaptation of the current security processes and solutions to work in the virtualized environment, major security solutions do not take into consideration the fact that machines can be physical or virtual, the idea that the hypervisors make the datacenter and its traffic became a much more dynamic and flexible place, and the very risk of mis-configuration which calls for the usage of best practices specific to virtualization domain.

Kovacs (2014) explains that to advance the security of VMware products, the manufacturers of this hypervisor make use of a number of techniques during its software development cycle. These typical techniques utilize both internal and external security expertise and include threat modeling, static code analysis, incident response planning, and penetration testing. As Cleary (2015) found out, the manufacturers have also established a software security engineering group that incorporate these techniques into the software development cycle and provides security expertise, guidance on the latest security threats and defensive techniques. This group also offers training within the development organization.

In Section 2 we will present the VMware vulnerabilities, which are abusing a lack of access control in a VMware 3D graphics driver, directory traversal vulnerability in VMware tools, dangling pointers due to a bug in the hardware emulation layer that can be attributed to vulnerabilities such as bug in the backdoor application programming interface (for communication between VMware tools and host) channel between VM and host, and just to mention few. Taxonomy for experimental set of virtual environment is presented in Section 3 and results finding of the study in Section 4 are discussed. Finally, the paper is concluded in Section 5.

2. VMWARE VULNERABILITIES

The VMware hypervisor has a number of weaknesses. As Matthias (2013) illustrated, numerous flows exist in VMware. These include abusing a lack of access control in a VMware 3D graphics driver, directory traversal vulnerability in VMware tools, dangling pointers due to a bug in the hardware emulation layer that can be attributed to vulnerabilities such as bug in the backdoor application programming interface (for communication between VMware tools and host) channel between VM and host, bug in the SCSI device registration, potentially due to a bug in hardware emulation layer, and buffer overflow in floppy driver, potentially because of a bug in hardware emulation layer. Bart (2015) noted that design flaw in the VMware ESXi hard disk handling can also be a major security hole.

Moreover, Kovacs (2014) further notes that hypervisors form an important part of enterprise environments and while they should normally reduce the attack vectors, they are actually beleaguered by security vulnerabilities that could be exploited by malicious actors.

In his study, Steven (2014) pointed out that any code that processes attacker-controlled input makes VMware potentially vulnerable. The central parts of the hypervisor, device model, additional privileged hypervisor-related services are all attack points. Aneesh (2016) argues that the compromise of the hypervisor core instantly gives an attacker the full control over the system. The exploitation of weaknesses in other VMware components could also be considered critical.

In situations where the hypervisor is employed to isolate un-trusted code executing in a virtual machine from the rest of the system, successful exploitation of hypervisor susceptibility shatters this isolation. In so doing, the attacker gains access to all the resources available to the hypervisor Karpouzas (2013). Ultimately, this provides the attacker complete control over the targeted machine.

3. PROCEDURE

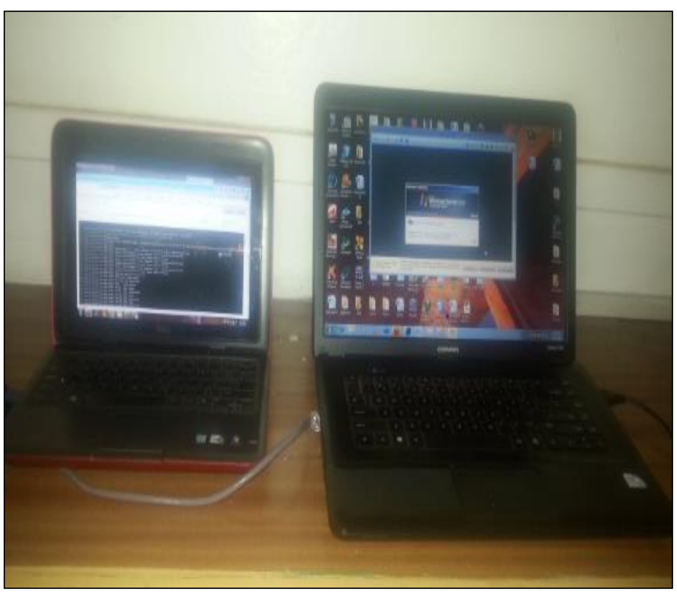


Figure1: Experimental Setup

In this paper, the researchers used Windows 2007 as a host operating system, VMware as a hypervisor, Windows Server 2003 as a guest network operating system. The intention was to investigate whether VMware hypervisor can protect the guest operating system from intruder activities such as port scanning, fingerprinting, and service identification. Figure 1 shows the experimental set up that was utilized.

As this figure shows, the set up consisted of two laptop computers. The Metaspoilt software was installed in one machine directly connected to the other laptop containing virtualized Windows Server 2003 network operating system. In this study, the *attacker* was the laptop in which the Metaspoilt software was installed, while the *target* was the laptop in which virtualization was done. In this perspective, class C network was purposively chosen to assign internet protocol (IP) addresses. Table 1 shows how these addresses were assigned.

Table 1: IP Address Assignments

Attacker	**Host Operating System**	**Guest Operating System**
192.168.1.10	192.168.1.30	192.168.1.20

The Metaspilt Web User Interface was then launched and activated through online License Key. After this, two accounts, with user names and corresponding passwords were created and were used to gain as shown in Figure 2. These were the details that were used to authenticate the researchers to the Metaspoilt functionalities.

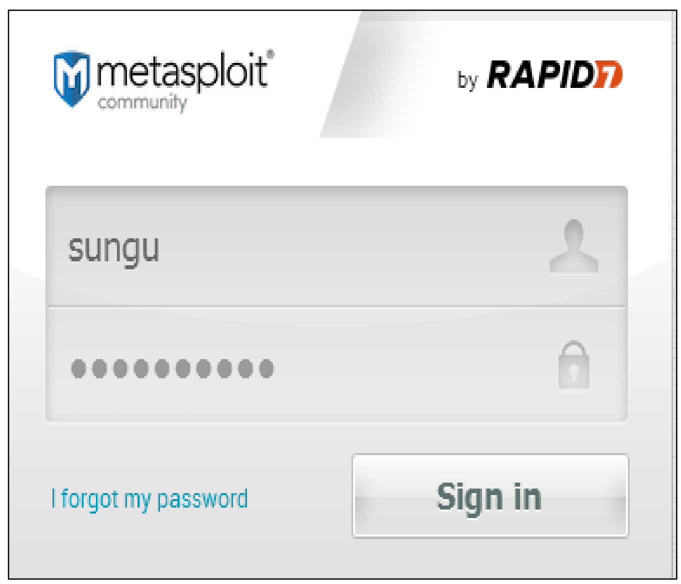


Figure 2: Authentication Interface

The *Ping* commands were carried out among the three entities and the connections were fund to be good and therefore all the three entities could exchange information. After successful login, the information in Figure 3 was displayed.

Figure 3: Project Menu Interface

This interface contained various options to choose from. For this study, the reserachers were inerested in *Hosts* and *Services* menus. To begin with, the Hosts option was selected and as aresult, the information in Figure 4 was shown. This figure shows that two hosts were discovered and a total of 14 services were deteted to be running in these two hosts.

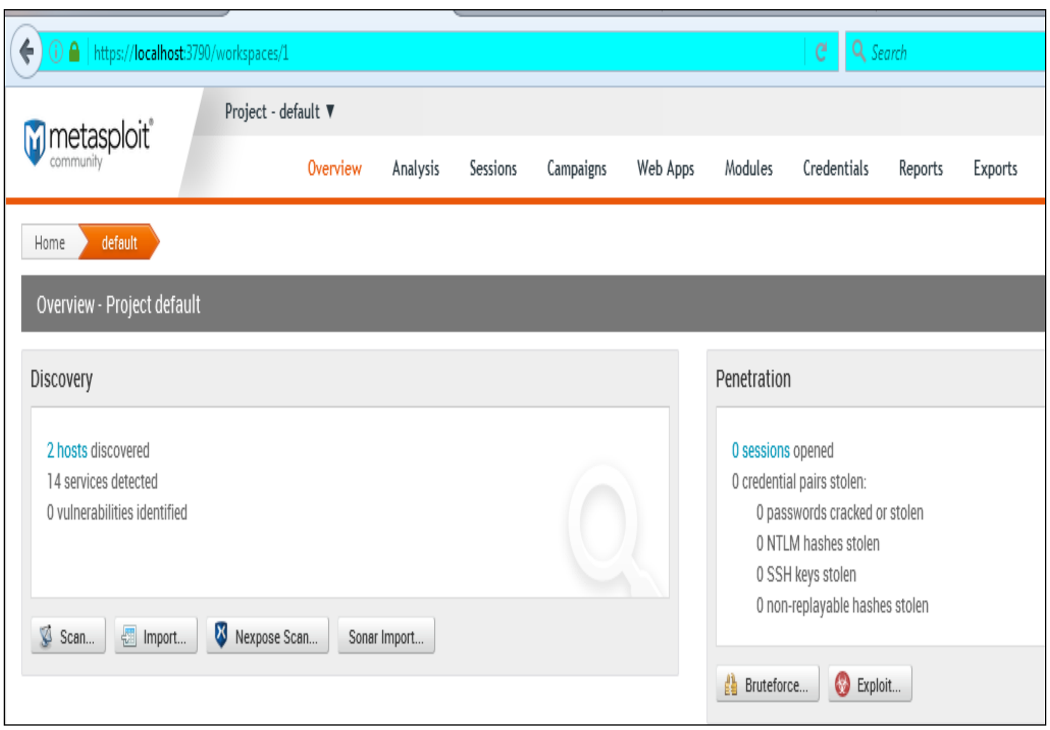


Figure 4: Host and Services Identification

The first step in our investigation was to carry out a network scan on the target guest operating system, whose IP address was 192.168.1.20. To accomplish this, the *scan* option in the above figure was selected.

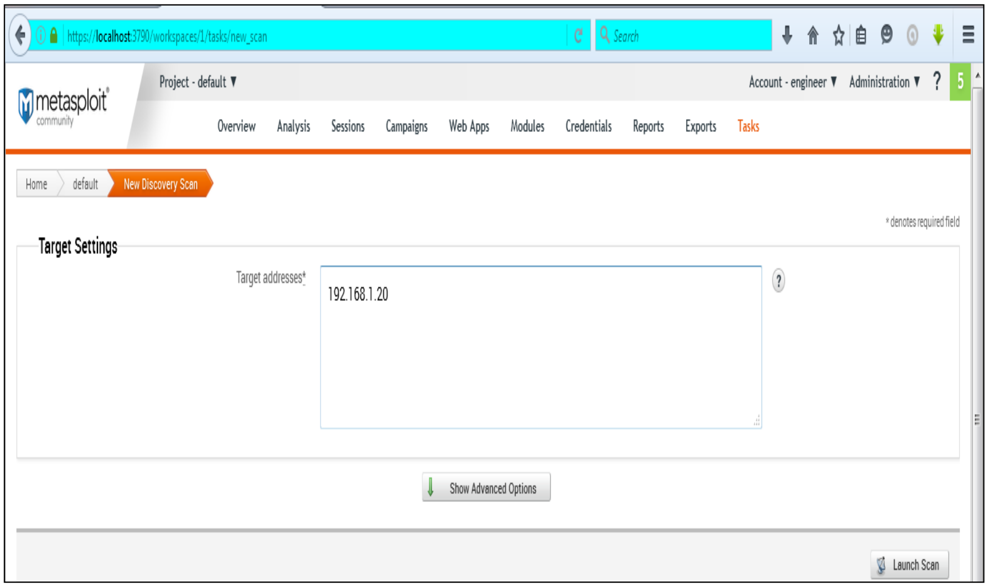


Figure 5: Target Scanning

This address was entered in the text box shown above and the *'launch scan'* command button was clicked. The Metaspoilt software then called an inbuilt version of Nmap which began scanning the virtualized guest operating system as shown in Figure 6 that follows.

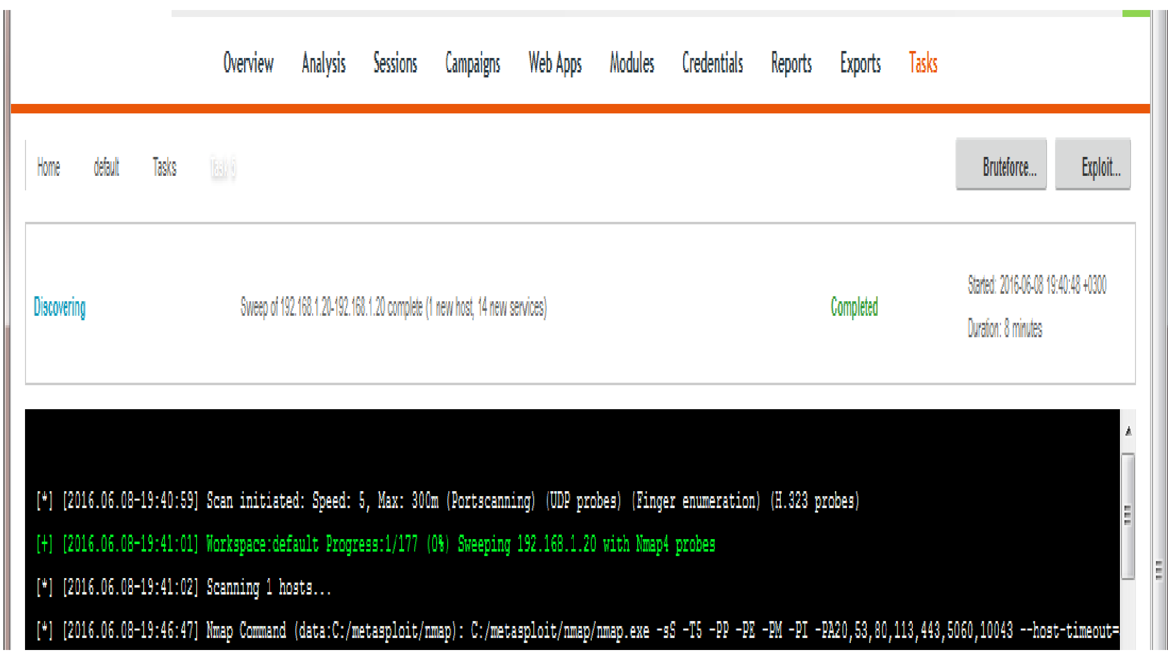


Figure 6: Target Scanning

This diagram shows that Metaspoilt has launched port scanning, the time the scan was started and the IP address of the target being scanned. To investigate the services running on the target, the *services* option was selected from the *Analysis* menu. Figure 7 shows the interface displayed after this selection.

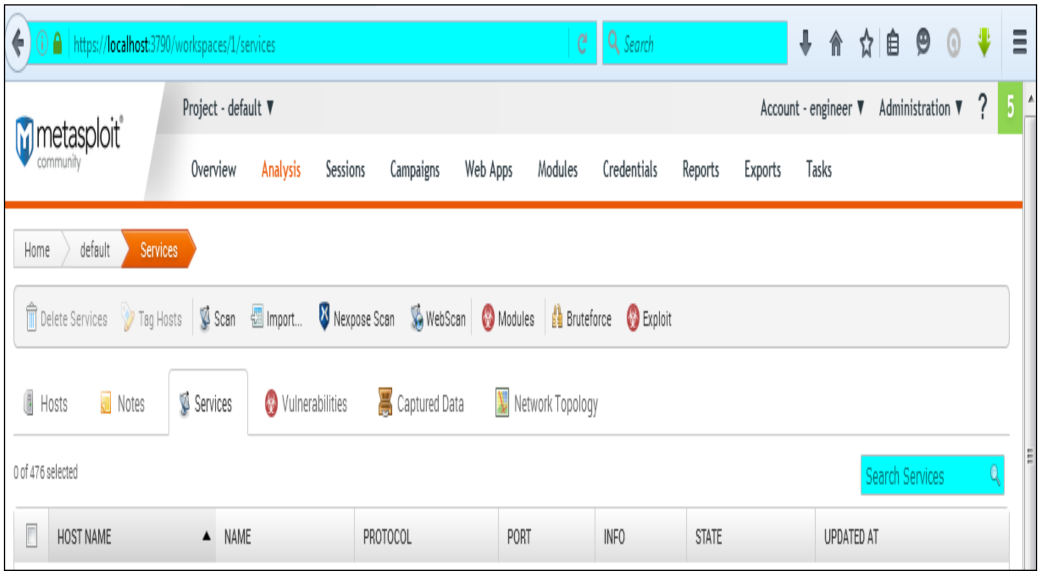
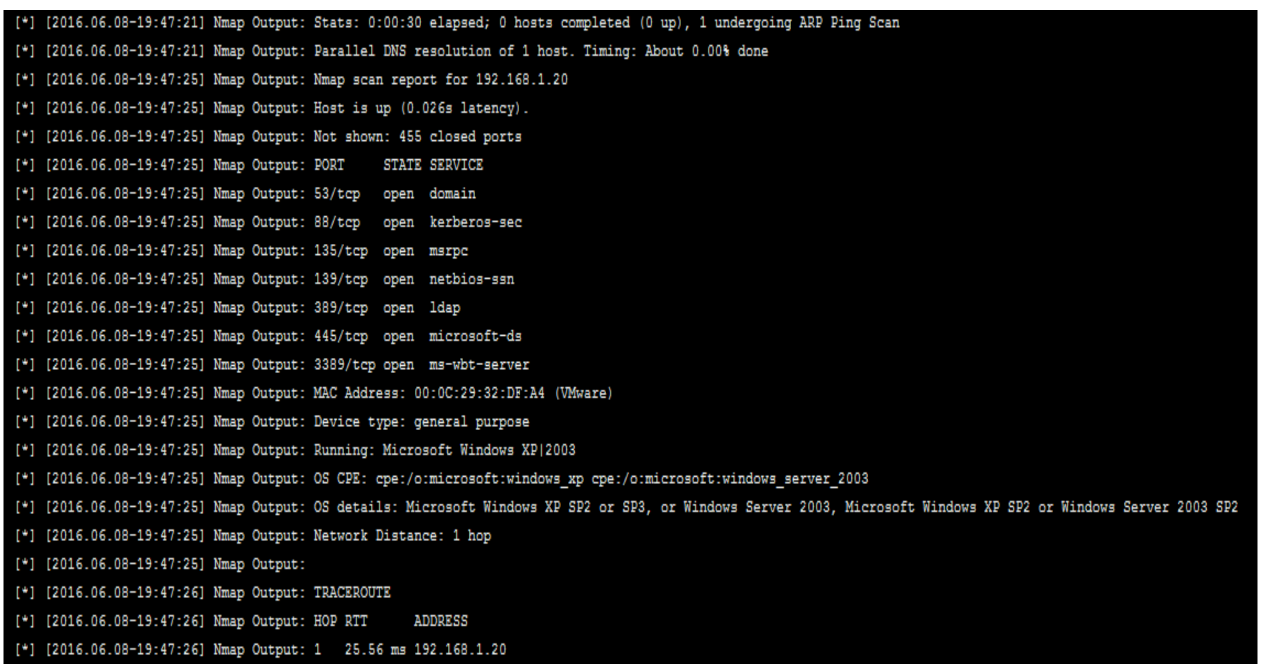


Figure 7: Services Port Scanning

This figure shows that the host name, name of running service, the protocol that the service uses, the port number where the service is running, basic information about the service, state of the service and the time when the service was last updated are some of the information that will be captured from the target.

4. STUDY RESULTS

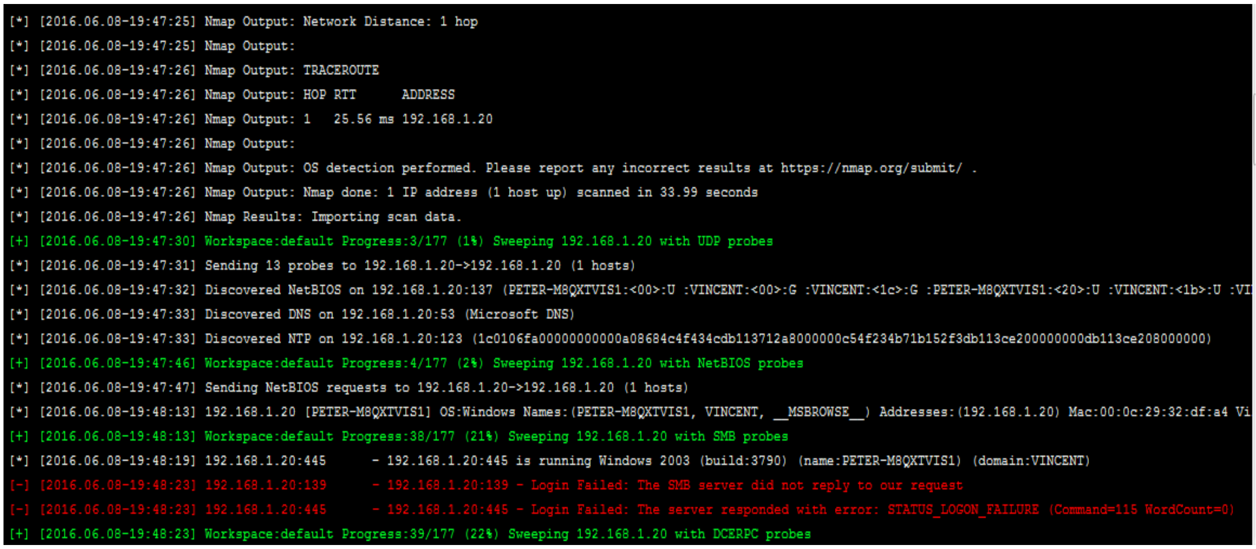
In this section, the results of the study will be presented and the discussions that follow from the observed phenomena will be given. As already stated, port scanning was carried out against the target at IP address 192.168.1.20. Figure 8 shows part of the port scanning output.



```
[*] [2016.06.08-19:47:21] Nmap Output: Stats: 0:00:30 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
[*] [2016.06.08-19:47:21] Nmap Output: Parallel DNS resolution of 1 host. Timing: About 0.00% done
[*] [2016.06.08-19:47:25] Nmap Output: Nmap scan report for 192.168.1.20
[*] [2016.06.08-19:47:25] Nmap Output: Host is up (0.026s latency).
[*] [2016.06.08-19:47:25] Nmap Output: Not shown: 455 closed ports
[*] [2016.06.08-19:47:25] Nmap Output: PORT     STATE SERVICE
[*] [2016.06.08-19:47:25] Nmap Output: 53/tcp   open  domain
[*] [2016.06.08-19:47:25] Nmap Output: 88/tcp   open  kerberos-sec
[*] [2016.06.08-19:47:25] Nmap Output: 135/tcp  open  msrpc
[*] [2016.06.08-19:47:25] Nmap Output: 139/tcp  open  netbios-ssn
[*] [2016.06.08-19:47:25] Nmap Output: 389/tcp  open  ldap
[*] [2016.06.08-19:47:25] Nmap Output: 445/tcp  open  microsoft-ds
[*] [2016.06.08-19:47:25] Nmap Output: 3389/tcp open  ms-wbt-server
[*] [2016.06.08-19:47:25] Nmap Output: MAC Address: 00:0C:29:32:DF:A4 (VMware)
[*] [2016.06.08-19:47:25] Nmap Output: Device type: general purpose
[*] [2016.06.08-19:47:25] Nmap Output: Running: Microsoft Windows XP|2003
[*] [2016.06.08-19:47:25] Nmap Output: OS CPE: cpe:/o:microsoft:windows_xp cpe:/o:microsoft:windows_server_2003
[*] [2016.06.08-19:47:25] Nmap Output: OS details: Microsoft Windows XP SP2 or SP3, or Windows Server 2003, Microsoft Windows XP SP2 or Windows Server 2003 SP2
[*] [2016.06.08-19:47:25] Nmap Output: Network Distance: 1 hop
[*] [2016.06.08-19:47:25] Nmap Output:
[*] [2016.06.08-19:47:26] Nmap Output: TRACEROUTE
[*] [2016.06.08-19:47:26] Nmap Output: HOP RTT      ADDRESS
[*] [2016.06.08-19:47:26] Nmap Output: 1   25.56 ms 192.168.1.20
```

Figure 8: Port Scanning Output- Part 1

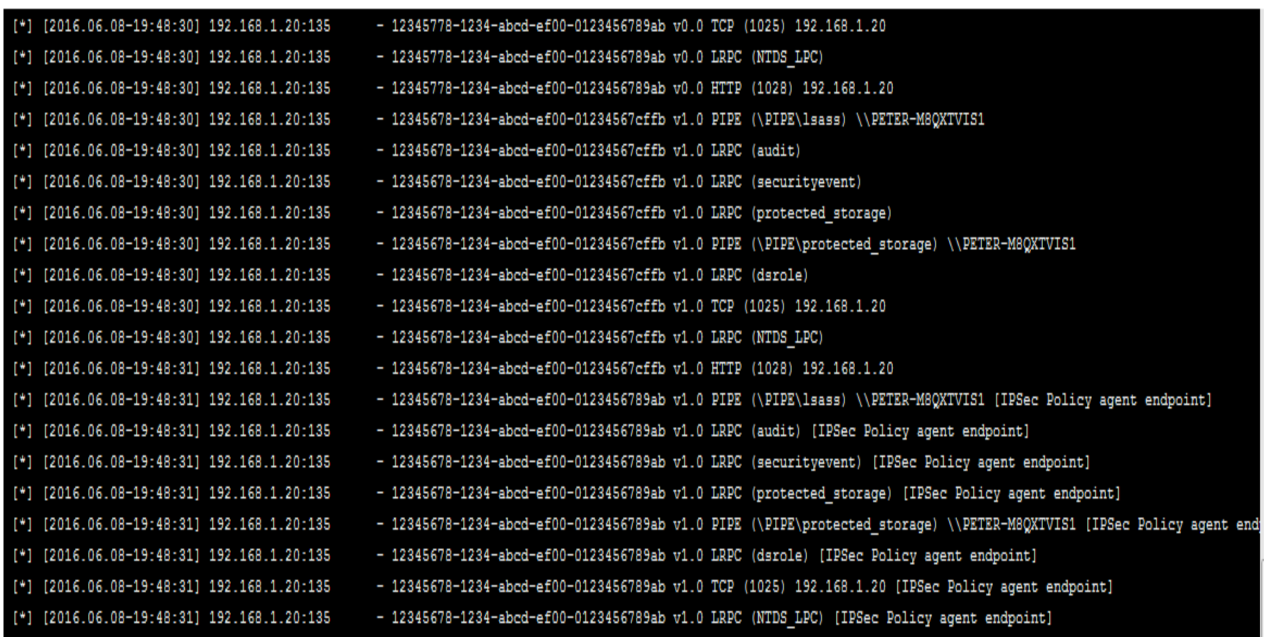
This figure displays interesting statistics about the target machine. To start with, it gives information on the open ports, the MAC address of the target machine and guesses on the probable operating system running on the target. The open ports include port number 53, 88,135,139,389,445 and 3389, all of which are TCP ports. Moreover, the services running on these ports are also provided. The MAC address of the target is also detected (00:0C:29:32:DF:A4) in addition to the type of hypervisor in use (VMware). The guest operating system is also detected to a high accuracy to be Windows Server 2003.



```
[*] [2016.06.08-19:47:25] Nmap Output: Network Distance: 1 hop
[*] [2016.06.08-19:47:25] Nmap Output:
[*] [2016.06.08-19:47:26] Nmap Output: TRACEROUTE
[*] [2016.06.08-19:47:26] Nmap Output: HOP RTT      ADDRESS
[*] [2016.06.08-19:47:26] Nmap Output: 1   25.56 ms 192.168.1.20
[*] [2016.06.08-19:47:26] Nmap Output:
[*] [2016.06.08-19:47:26] Nmap Output: OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] [2016.06.08-19:47:26] Nmap Output: Nmap done: 1 IP address (1 host up) scanned in 33.99 seconds
[*] [2016.06.08-19:47:26] Nmap Results: Importing scan data.
[+] [2016.06.08-19:47:30] Workspace:default Progress:3/177 (1%) Sweeping 192.168.1.20 with UDP probes
[*] [2016.06.08-19:47:31] Sending 13 probes to 192.168.1.20->192.168.1.20 (1 hosts)
[*] [2016.06.08-19:47:32] Discovered NetBIOS on 192.168.1.20:137 (PETER-M8QXTVIS1:<00>:U :VINCENT:<00>:G :VINCENT:<1c>:G :PETER-M8QXTVIS1:<20>:U :VINCENT:<1b>:U :VI
[*] [2016.06.08-19:47:33] Discovered DNS on 192.168.1.20:53 (Microsoft DNS)
[*] [2016.06.08-19:47:33] Discovered NTP on 192.168.1.20:123 (1c0106fa00000000000a08684c4f434cdb113712a8000000c54f234b71b152f3db113ce200000000db113ce208000000)
[+] [2016.06.08-19:47:46] Workspace:default Progress:4/177 (2%) Sweeping 192.168.1.20 with NetBIOS probes
[*] [2016.06.08-19:47:47] Sending NetBIOS requests to 192.168.1.20->192.168.1.20 (1 hosts)
[*] [2016.06.08-19:48:13] 192.168.1.20 [PETER-M8QXTVIS1] OS:Windows Names:(PETER-M8QXTVIS1, VINCENT, __MSBROWSE__) Addresses:(192.168.1.20) Mac:00:0c:29:32:df:a4 Vi
[+] [2016.06.08-19:48:13] Workspace:default Progress:38/177 (21%) Sweeping 192.168.1.20 with SMB probes
[*] [2016.06.08-19:48:19] 192.168.1.20:445      - 192.168.1.20:445 is running Windows 2003 (build:3790) (name:PETER-M8QXTVIS1) (domain:VINCENT)
[-] [2016.06.08-19:48:23] 192.168.1.20:139      - 192.168.1.20:139 - Login Failed: The SMB server did not reply to our request
[-] [2016.06.08-19:48:23] 192.168.1.20:445      - 192.168.1.20:445 - Login Failed: The server responded with error: STATUS_LOGON_FAILURE (Command=115 WordCount=0)
[+] [2016.06.08-19:48:23] Workspace:default Progress:39/177 (22%) Sweeping 192.168.1.20 with DCERPC probes
```

Figure 9: Port Scanning Output- Part 2

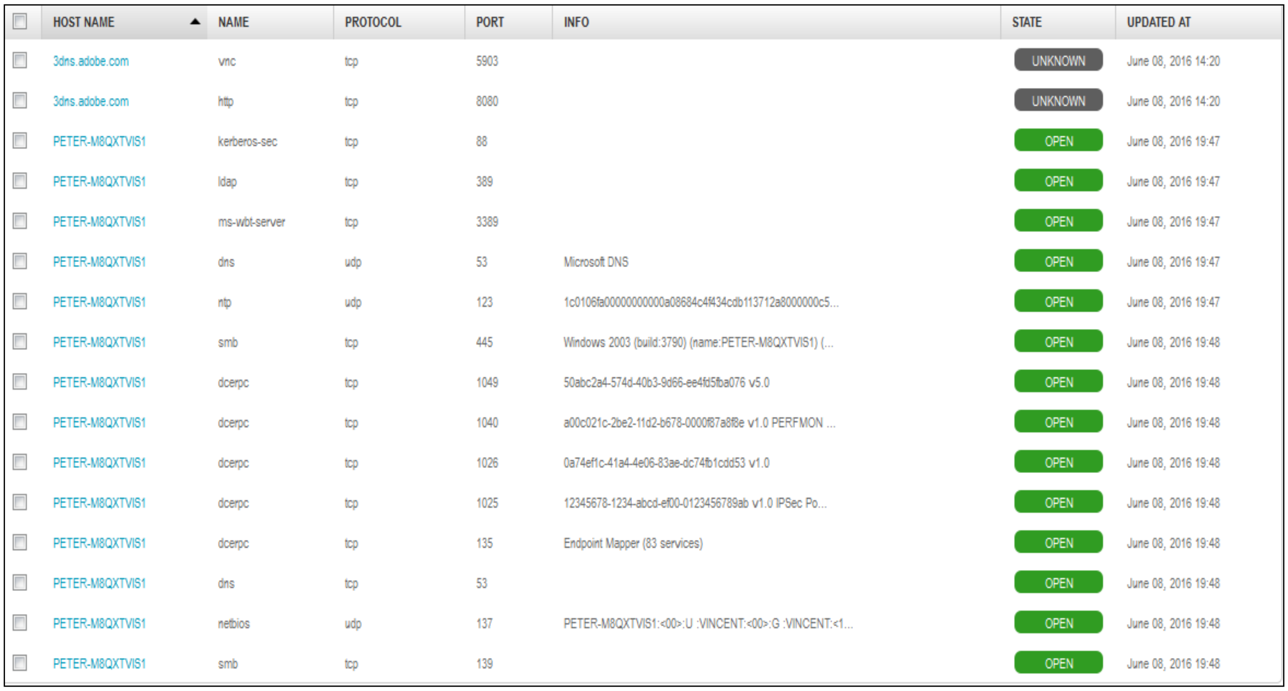
Figure 9 illustrates that the hop distance is detected as 1, and the retransmission timer is 25.56 milliseconds, the NETBIOS name is PETER, the DNS is Microsoft DNS, and the domain is accurately detected as VINCENT.



```
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345778-1234-abcd-ef00-0123456789ab v0.0 TCP (1025) 192.168.1.20
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345778-1234-abcd-ef00-0123456789ab v0.0 LRPC (NTDS_LPC)
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345778-1234-abcd-ef00-0123456789ab v0.0 HTTP (1028) 192.168.1.20
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 PIPE (\PIPE\lsass) \\PETER-M8QXTVIS1
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 LRPC (audit)
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 LRPC (securityevent)
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 LRPC (protected_storage)
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 PIPE (\PIPE\protected_storage) \\PETER-M8QXTVIS1
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 LRPC (dsrole)
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 TCP (1025) 192.168.1.20
[*] [2016.06.08-19:48:30] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 LRPC (NTDS_LPC)
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-01234567cffb v1.0 HTTP (1028) 192.168.1.20
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 PIPE (\PIPE\lsass) \\PETER-M8QXTVIS1 [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (audit) [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (securityevent) [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (protected_storage) [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 PIPE (\PIPE\protected_storage) \\PETER-M8QXTVIS1 [IPSec Policy agent end
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (dsrole) [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 TCP (1025) 192.168.1.20 [IPSec Policy agent endpoint]
[*] [2016.06.08-19:48:31] 192.168.1.20:135      - 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (NTDS_LPC) [IPSec Policy agent endpoint]
```

Figure 10: Port Scanning Output- Part 3

Figure 10 captures some security related data. For example, *protected storage* and *IPSEC* security policy are detected to be employed by the target, Windows Server 2003. When the *Services* option was selected and run, various hostnames were found to be either in unknown states or in open states as demonstrated by Figure 11.



HOST NAME	NAME	PROTOCOL	PORT	INFO	STATE	UPDATED AT
3dns.adobe.com	vnc	tcp	5903		UNKNOWN	June 08, 2016 14:20
3dns.adobe.com	http	tcp	8080		UNKNOWN	June 08, 2016 14:20
PETER-M8QXTVIS1	kerberos-sec	tcp	88		OPEN	June 08, 2016 19:47
PETER-M8QXTVIS1	ldap	tcp	389		OPEN	June 08, 2016 19:47
PETER-M8QXTVIS1	ms-wbt-server	tcp	3389		OPEN	June 08, 2016 19:47
PETER-M8QXTVIS1	dns	udp	53	Microsoft DNS	OPEN	June 08, 2016 19:47
PETER-M8QXTVIS1	ntp	udp	123	1c0106fa0000000000a08684c4f434cdb113712a8000000c5...	OPEN	June 08, 2016 19:47
PETER-M8QXTVIS1	smb	tcp	445	Windows 2003 (build:3790) (name:PETER-M8QXTVIS1) (...	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dcerpc	tcp	1049	50abc2a4-574d-40b3-9d66-ee4fd5fba076 v5.0	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dcerpc	tcp	1040	a00c021c-2be2-11d2-b678-0000f87a8f8e v1.0 PERFMON ...	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dcerpc	tcp	1026	0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53 v1.0	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dcerpc	tcp	1025	12345678-1234-abcd-ef00-0123456789ab v1.0 IPSec Po...	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dcerpc	tcp	135	Endpoint Mapper (83 services)	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	dns	tcp	53		OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	netbios	udp	137	PETER-M8QXTVIS1:<00>:U :VINCENT:<00>:G :VINCENT:<1...	OPEN	June 08, 2016 19:48
PETER-M8QXTVIS1	smb	tcp	139		OPEN	June 08, 2016 19:48

Figure 11: Running Hostnames, Services and Protocols

This figure shows that while some hostnames were in unknown states, some were discovered to be open. Details of the hostnames, the protocols running in these hostnames and the last update time are also displayed.

5. DISCUSSIONS

The experimental results obtained in section (IV) above demonstrate serious VMware hypervisor vulnerabilities. To begin with, the port scanning process was able to discover open ports as well as the services running on them. This is a security challenge as a hacker can terminate these services and establish his own illicit connections to the open ports. Alternatively, since some ports are in unknown state, an attacker may try to establish connections to these ports and if the connections are successful, then he may even take full command of the virtualized guests.

Another security risk that can be identified in this study is the ability to fingerprint both the hypervisor and the guest. The results illustrated that the scanning process detected to a high accuracy the MAC address of the guest (00:0C:29:32:DF:A4), the name of the hypervisor in use for virtualization (VMware), the security policy in place (*IPSEC* and protected *storage*), the protocols running in the discovered ports (such as TCP and UDP) as well as the services running in these ports (such as HTTP, KERBEROS-SEC, DNS and NETBIOS). The obtained information raises questions on the security of the hypervisor because as a virtual machine monitor, it is supposed to offer protection to the virtual machines running in it. However, it has been shown that it does not do so and therefore the guests are exposed to attacks exploiting any of their known vulnerabilities. In fact, the guests are as exposed as when they are not virtualized.

6. CONCLUSIONS AND RECOMMENDATIONS

In a cloud environment, various companies share the same physical resource, such as storage memory, central processing unit, hard disc and networks. This sharing is accomplished through the virtualization process which makes each of the company to feel as if it is actually running on dedicated resources. It then becomes possible for a single hypervisor to support multiple guests, each for the cloud client companies. This study has shown that fingerprinting a hypervisor and the guests is possible. This means that a determined attacker can bring down a whole hypervisor and hence all the guests running in it. Alternatively, he can decide to compromise individual guests using the fingerprinting details obtained. Closing down a port and terminating any services running on it, request flooding through open ports that can lead to denial of service attacks are just but illustrations of the exploits that can be propagated with the help of the obtained data. The researcher therefore suggests that further study be carried out on how to protect hypervisors and hence the guests from fingerprinting and related attacks.

REFERENCES

[1] Chandramouli R. (2014), **"**Security_Recommendations for Hypervisor Deployment", DRAFT NIST Special Publication 800-125-A, 2014.
[2] Foley M. (2014), "Security of the VMware vSphere Hypervisor".
[3] Randell R. (2015), "Virtualization Security and Best Practices"
[4] Kovacs E.(2014) ,"Black Hat: Mind Your Hypervisors", Security week Network ,2014.
[5] Cleary L. (2015), "Attacking the SharePoint Server from the Inside – Part 1, 2015".
[6] Matthias L. (2013)," Analysis of Hypervisor Breakouts, Insinuator, 2013".
[7] Bart A. (2015), "Hack Like a Pro: Metasploit for the Aspiring Hacker, Part 1 (Primer & Overview).
[8] Steven J. (2014)," Hypervisors: The cloud's potential security Achilles heel".
[9] Aneesh M.(2016), "Metasploit_Tutorial - With an example Exploiting the vulnerabilities".
[10] Karpouzas G.(2013), "Metasploit Tutorials – Hacking Exploiting Software Compendium,".

Author

Peter Sungu Nyakomitta Pursuing Msc in Information Technology Security and Audit from Jaramogi Oginga Odinga University of Science and Technology School of Informatics and Innovative System (JOOUST). Received Bsc. Information Technology from JKUAT, Kenya. His research interest is on analysis of VMware Hypervisor Security. He is a career banker with bias in ebanking systems.