NFV and Network Security with Ansible

Conference Paper · April 2019

CITATION
1

READS
937

2 authors:

Alen Šimec
University of applied sciences Zagreb
19 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Antonela Čukurin
Span, Zagreb, Croatia
1 PUBLICATION **1** CITATION
SEE PROFILE

NFV and Network Security with Ansible

Alen Šimec
Zagreb University of Applied Sciences, 10 000 Zagreb, Croatia
alen@tvz.hr

Antonela Ćukurin
*back. ing. comp. * Zagreb University of Applied Sciences, 10 000 Zagreb, Croatia
antonela.cukurin@tvz.hr

***Abstract* – Increase demand for cloud computing technologies led to the new era of Information Technologies (IT). Consequently, Network functions virtualization (NFV) because traditional networks didn't have a capability to satisfy new demands. Networks are faced with large demand for bandwidth, agility, flexibility and scalability. Beside all that requirements, our environment need to be secure. Security in cloud computing has task to secure data between multiple system, hypervisors, virtual machines (VMs) and external/internal networks. Simple and efficient tool like Ansible satisfies all needs and is used for configuration, orchestration and deployment.**

Keywords: NFV, Network security, security, cloud computing, SSH, Ansible

I. INTRODUCTION

Network Function Virtualization (NFV) was created by service providers who published the White Paper in 2012, describing the reasons for NFV. The reason for introducing new services such as NFV, was to respond to the rapid development of the network services, because hardware-oriented networks are limited. With NFV we get cost reduction and accelerated network development that is now scalable, rapid and agile. NFV is a subset of Software Defined Network (SDN) and separates the network functions such as encryption, firewalls or Network Address Translation (NAT), from dedicated hardware and moving them on virtual servers. The main purpose of NFV is reducing the cost of dedicated hardware in the network, by replacing the hereditary hardware functions with software functions that operate on the generic x86 server.

It enables fast and cost-efficient deployment of network functions for better service agility and supporting the agile and flexible deployment of network functions along with their lifecycle management. [1]

With the network that can adjust to new challenges rapidly, security should be in first place. Each component has some level of vulnerability, and it is necessary to have it in mind while planning and have a good set of protection. Small failures can mean significant losses. It is essential to treat virtual networks the same as physical ones, because they are equally vulnerable.

II. Network Function Virtualization - NFV

The NFV architecture consists of four main components: *NFV Infrastructure*, *VNF*, *Management and Network Orchestration (MANO)* and *OSS / BSS Layer*.

A. NFVI

NFVI is a set of hardware and software resources that provide the infrastructure that Virtual Network Functions (VNFs) perform, including servers, switches, virtual switches and virtual machines. NFVI is divided into three parts: hardware, virtualization layer and virtualized resources. Hardware resources within NFVI are computer hardware (servers, hard disk, RAM, etc.), storage resources and network hardware (routers, switches, firewalls, etc.). The second part of NFVI is a virtualization layer that stands just above the hardware. It isolates the hardware part of the software and allows the software part to use

virtual infrastructure irrespective of hardware. This allows the primary layer of hardware to handle one or more virtual machines (VMs), and they share the resources of that physical layer.

An example of NFVI is a hypervisor that provides many benefits, such as the ability to spin multiple VM instances on physical hardware with its operating system and applications, thus fully exploiting the capabilities of a physical server, while traditional physical hardware can spin only one instance. VM's are mobile, meaning they can move from one server to another without having to reinstall the various components because they are independent of the hardware. There are two types of hypervisors. The first one is a bare-metal hypervisor and he is installed directly on physical hardware. Examples are VMware ESX/ESXi, Hyper-V and Oracle VM. Another type of hypervisor is the one that spins on the operating system and provides virtualization services such as memory management. Some examples of another type of hypervisor are VMware Fusion, Oracle VM VirtualBox and KVM.

B. Virtual Network Function (VNF)

The Virtual Network Function (VNF) layer consists of VNFs.

The first generation of NFV system implementations transferred existing monolithic applications to large virtual machine appliances, each representing a single Virtual Network Function (VNF). Multiple VNFs are then chained together using a Service Function Chain, which determines how packets are forwarded from one VNF to another, to constitute a Network Service. [2]

VNFs are software packages that can implement network functions that are operated by dedicated hardware. This leads to reduced network costs, flexibility and better network management. Also, it can help with scalability and network agility. Examples of VNF are virtual firewalls (vFirewall) and virtual routers (vRouter).

C. Management and Network Orchestration (MANO)

MANO includes the management and orchestration of resource functions for compute, network, storage and VM. Provides flexibility and ability to respond to the latest features and requirements of complex network components. Virtualized Infrastructure Manager (VIM) is responsible for controlling and managing NFVI resources. It controls interactions between VNFs and NFVI hardware components and has the required implementation and virtualization layer monitoring tools. Examples of VIMs are OpenStack, OPNFV and Kubernetes. The VNF manager manages the life cycle of VNF instances. It is responsible for initiating, updating, testing and terminating VNF. Therefore, each VNF instance must be linked exclusively to the VNF manager. The third component of MANO is NFV orchestration. NFVO is at the forefront of NFVI resources across multiple VIMs and manages the life cycle of network services, including policy management, performance measurement, scheduling, and tracking.

D. OSS/BSS layer

Operational Support Systems (OSS) are software applications that support activities within the telecommunication network, provide and maintain customer service. OSS and BSS look for constant adaptation, updating, optimization and enhancements, so that all services communicate and work together.

III. SECURITY IN CLOUD COMPUTING

An essential goal of security in cloud computing is to share resources between multiple users or programs safely. With virtualization, we get an architecture where physical hardware can be fully utilized for more distributed logical units of operating systems and applications. Virtualization in one hand protects the system because we have distributed logical units, but on the other hand, that specificity may indicate more significant exposure to attacks. Virtual systems are as vulnerable as physical systems because they contain every segment of the ordinary system. The biggest problem in the cloud computing is security. To be more specific, data security. Today, we have a massive demand for cloud services, which also leads to more frequent attacks. According to Gartner, over the next decade public cloud infrastructure as a service will have at least 60% fewer security incidents than traditional data centers. Subbaiah Venkata, a Google engineer, explained why. With an extensive database of users the public cloud can learn and if an attack on a particular user happens, the solution can be applied to all other users. Security can be achieved

through data tracking, machine learning algorithms, where alarms occur for each oscillation.

IV. NETWORK SECURITY IN CLOUD COMPUTING

A. Virtual network

Virtual networks sit above physical networks and have a logical network structure independent of physical. A significant advantage is their security and flexibility. VPN is used for secure Internet communication and it supports a private network over the public network. Today, VPN is the fundamental component for secure communication in business environments.

B. Virtual Machine Security

VMs are a subject to all attacks like physical hardware. One of the most critical security features is secure communication between virtual machines. Although virtual machines are separate one from another, they share resources such as memory, CPU, RAM, etc. An essential thing in virtualization is isolation. It prevents a virtual machine from affecting another if it runs on the same server. Isolation can be flexibly set due to the different needs within the organization, leading to new problems of VM safety. It is necessary to restrict traffic from one VM to another, the profiles must be limited, and the OS must always be up-to-date. It is also necessary to treat VM as a physical machine so that all security protocols are applied to it as well. No matter how many VMs are running on the same physical hardware, they must be isolated from each other. It is essential to set up a firewall to allow only protocols that are required for using a virtual machine. Best to access a virtual machine is through SSH protocols for Linux environments or RDS (Remote Desktop Services) for Windows.

The security of the SSH protocol is based on the usage of cryptographic methods that allow the protection (confidentiality) of data that move through the insecure network. Besides, these methods can be used for checking the identity of the users involved in the communication and protecting the data from unauthorized modifications or preserving their integrity. [3]

One of the major attacks on VMs is undoubtedly a guest escape. This situation may happen due to poor configuration or designing interactions between host and guest VMs. As the name says, VM leaves the boundaries of its virtual instance. The VM also separates itself from the hypervisor or Virtual Machine Manager (VMM) that manages the VMs or can be taken by the attacker. This brings the attacker access to other VMs that are on the same host and the attacker can leave the VM environment, thus having access to data from other resources and devices. Migration of VMs is a great feature of virtualization, but it can be risky because data, integrity, and VMs are then vulnerable to attacks. [4]

C. Hypervisor security

The central part of virtualization is VMM or hypervisor. The management and isolation of VMs depend on it. One way to protect hypervisors is to implement LDAP (Lightweight Directory Access Protocol). There are two types of hypervisors. We should consider the advantages and disadvantages of each type. According to the CCSP (ISC), bare-metal hypervisor is a better solution because it sits on the hardware while the other type runs on the operating system. There is a lower possibility of an attack. A good practice is to update the hypervisor. Also, a good security solution is the VLAN reservation, for isolation of interface to separate data, application and management traffic. Routing VLANs should not perform a virtual switch and all traffic should go through the firewall. [5]

The hypervisor needs to be protected from any unauthorized access. DoS (Denial of Service) is one of the most famous and most common attacks. Most often, the DoS attack is executed so that the server loads with a large number of requests that the server can no longer fulfill its primary duty. One of the worst attacks on the hypervisor is hyperjacking. This is done by installing a new hypervisor that takes control of the server, and it is difficult to detect this type of attack. If successful, attacker can gain complete control over the virtual environment.

V. ANSIBLE

Ansible is a tool that can be used to automate the most complex environments. It is suitable for the configuration, orchestration, the implementation of the software, updates and other advanced tasks using a

simple and human readable language. It is very adjustable, therefore can be beneficial for small and grand environments. There is a big emphasis on simplicity. However, safety and reliability are an essential part of Ansible. OpenSSH protocol is used for increased security because OpenSSH is one of the most peer-reviewed open source components. Ansible relies on OS credentials to control access to remote computers and, if necessary, can connect with SSH, Kerberos, LDAP or some other protocol for user authentication. Passwords are supported using an SSH key with an SSH agent, which makes it very simple to secure and enable authentication. Root logging is not required, can be connected via user or “sudo” user.

Ansible's architecture consists of inventory, modules, plugins, and playbooks. The Inventory is a simple INI file that contains all the machines that are managed in the groups, defined by us. Essentially it is a host file.

Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default) and removes them when finished. [6] Plugins are part of the code that increases the functionality. There are many useful plugins which you can use or you can write your own.

Playbooks are YAML files. They can be written in any text editor and are a configuration management base on one or more machines. Roles, which contain tasks, can run on an assigned group of servers. They include tasks or commands that we execute by running a playbook. After a playbook is executed, we get a detailed printing of the machines that are covered, the order of the tasks and the print which has not been changed.

Within the playbook we can have handlers that are performed at the end of the script, live in the global namespace, and are run by tasks. The keyword for running the handler is “notify”. No matter how many such lines are in the playbook or task, they will only run once at its end. The main reason for creating a script is to automate a job that will be executed more than once.

A. The script for configuring "sshd_config" file

The script for changing the configuration file "sshd_config" changes and adds lines to increase the security of the SSH protocol. SSH protocol is used for a connection to the remote server. The configuration file is located on the "etc/ssh/sshd_config" path and only root or sudo user can make changes. It contains the location of host key and user authorization key. The script is divided into three tasks: changing the line, adding lines by finding the keyword and changing lines if the condition (true) is satisfied. Three different tasks give us different ways to change a file. Ansible is smart enough to understand what exactly is required. It is not necessary to set special terms if we have a different situation, he will search for keywords. If the word or phrase in module "line" is satisfied, Ansible looks for alternative under "replace" or "lineinfile" modules. The first task "Hardening sshd_config - change", with the "replace" module, will change all instances within a specified string in the file. The second "replace" means a specific string that will be changed with the string defined by the "regexp" keyword. "Regexp" is a regular expression that is searched within a file, and it is necessary within the "replace" module. "With_items" is used if we have more regular expressions to change in a file. The "lineinfile" module is used to manage lines in a file, it is used to change one line in a file or replace it with another, to replace multiple lines, a "replace" module is recommended. This module has many interesting parameters that can be included, for example, "insertafter" or "insertbefore" are used to precisely specified where a new line needs to be inserted. The "Hardening sshd_config - Looks and Change" task uses the "lineinfile" modules but in combination with the "backrefs" parameter. If the expression "regexp" is in the file it will be replaced with a new expression, if there is no expression, nothing changes, and the parameters "insertafter" or "insertbefore" will be ignored. [7]

The first change in the script refers to "PermitRootLogin". The best practice is not to log in with the root user but through the user account and use the sudo user. Protocol X11 is the older network protocol for remote access to applications. It is unsafe, so it is better to shut it down. In order to protect against external attacks, it is best to set the maximum number of user login attempts with "MaxAuthTries 3". The best practice is to take a new version of SSH protocol "Protocol 2" rather than to use the older version 1, a version that is no longer supported. The line with "IgnoreRhosts = yes" refers to circumventing the bad authentication method, we also are changing the line "IgnoreUserKnownHosts yes" ignored "users ~ / .ssh /

known_hosts ", we are sure the hosts will be set up by the administrator rather than any user. The "ClientAliveInterval 50" line goes over the encrypted channel, and the server will send a secure message to the client and expect an answer. A number in the line indicates how many times sshd will send a request before dropping the connection. "ClientAliveCountMax" and previous lines are interconnected, the product of the multiplication of these two numbers indicates the length of the session before termination. It is not recommended to be zero because it means an infinite session. Using “ciphers”, we configure encryption algorithms. The keyword line "LogLevel" comes with the value "info", we should change it to "verbose" to record fingerprint for each SSH key. Possible values for “LogLevel” are quiet, fatal, info, verbose, error, debug. "AllowTcpForwarding no" and "PermitTunnel no" are set for the risk that the user will use SSH tunneling to open the back door and gain access to a VM from a location. [8]

#script for sshd_config file

root@fuelhost1:/etc/ansible/roles/test_sshd/tasks#cat main.yml

```
- name: Hardening sshd_config – change
  replace:
    dest: /etc/ansible/ssh-test/sshd_config
    regexp: "{{ item.regexp }}"
    replace: "{{ item.replace }}"
  with_items:
    - { regexp: 'PermitRootLogin prohibit-password', replace: 'PermitRootLogin no'}
    - { regexp: '#IgnoreUserKnownHosts yes', replace: 'IgnoreUserKnownHosts no'}
    - { regexp: 'X11Forwarding yes', replace: 'X11Forwarding no'}
    - { regexp: 'LogLevel INFO', replace: 'LogLevel VERBOSE'}
    - { regexp: '#Banner /etc/issue.net', replace: 'Banner=/etc/issue.net'}
  notify: restart sshd


- name: Hardening sshd_config – add
  lineinfile:
    dest: /etc/ansible/ssh-test/sshd_config
    line: '{{ item }}'
  with_items:
    - 'AllowAgentForwarding no'
    - 'AllowTcpForwarding no'
    - 'MaxAuthTries 3'
    - 'ClientAliveInterval 900'
    - 'ClientAliveCountMax 2'
    - 'PermitTunnel no'
    - 'Ciphers="aes256-ctr,aes192-ctr,aes128-ctr"'
    -'MACs="hmac-sha2-512-etm@openssh.com,hmac-sha2-512,hmac-sha2-256-etm@openssh.com,hmac-sha2-256,umac-128-etm@openssh.com,umac-128@openssh.com"'
    - 'UseDNS=no'
    - 'Port 22'
    - 'Protocol 2'
    - 'HostbasedAuthentication=no'
    - 'IgnoreRhosts=yes'
    - 'UsePrivilegeSeparation=yes'
    - 'StrictModes=yes'
    - 'TCPKeepAlive=yes'
  notify: restart sshd


- name: Hardening sshd_config – search and alter
#add line module
  lineinfile:
#path to file destination
    dest: /etc/ansible/ssh-test/sshd_config
#searched expression
    regexp: "{{ item.regexp }}"
#replace regexp
    line: "{{ item.lines }}"
```

```
#flag set to yes
   backrefs: yes
#multiple expression parameter
  with_items:
   - { regexp: 'Port', lines: 'Port 22'}
   - { regexp: 'Protocol', lines: 'Protocol 2'}
   - { regexp: 'HostbasedAuthentication', lines: 'HostbasedAuthentication=no'}
   - { regexp: 'IgnoreRhosts', lines: 'IgnoreRhosts=yes'}
   - { regexp: 'UsePrivilegeSeparation', lines: 'UsePrivilegeSeparation=yes'}
   - { regexp: 'StrictModes', lines: 'StrictModes=yes'}
   - { regexp: 'TCPKeepAlive', lines: 'TCPKeepAlive=yes'}
  notify: restart sshd
```

VI. CONCLUSION

The network security is most talked topic in the IT community for a reason. Everybody needs to pay attention, ordinary users (citizens) and large companies in equal measure. Without well-established security system and a well-thought-out design of an environment, problems can multiply very quickly.

In this paper it is presented how NFV changed the game, the importance of network security in the cloud and introduction to Ansible, powerful automatization tool. Combination of a well-thought system and Ansible we can get easily configurable and secure environment.

The SSH protocol provides a high-level of security for all users. It can be used on all Linux operating platforms. The SSH protocol is an extremely important part of securing access to the infrastructure resources and it must be well-configured.

If the configuration goes over the Ansible playbook, it gives us a simple system automation. This article presents how an efficient Ansible can be used for configuration of any environment. Using different modules, plugins and inventories helps building the secure environment on any platform.

VII. REFERENCES

[1] Lal Shankar; Taleb Tarik; Dutta Ashutosh: "NFV: Security Threats And Best Practices", IEEE Communications Magazine; 2017 May; 55(8); 2.

[2] Zarrar Yousaf Faqir; Bredel Michael; Schaller Sibylle; Schneider Fabian: "NFV and SDN—Key Technology Enablers for 5G Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS; 2017 Nov; 35(11); 2470.

[3] Lozić D.; Šimec A.; Pametna komunikacija na Internetu preko REST protokola; 37. international convention on information and communication technology, electronics and microelectronics; Opatija, 2014, 1498-1505; ISBN: 978-953-233-078-6

[4] Carnet, "SSH protocol", 1 11 2018. [Online]. Available: https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2009-08-272.pdf.

[5] Reuben Jenni Susan: „A Survey on Virtual Machine Security", 13 7 2017. [Online]. Available: http://www.cs.umd.edu/class/fall2017/cmsc414/readings/vm-security.pdf.

[6] Yeluri Raghu; Castro-Leon Enrique: „Building the infrastructure for cloud security: A Solutions View"; 1st ed.; Apress, March 2014.; 129.

[7] Ansible, "How Ansible Works", 1 11 2018. [Online]. Available: https://www.ansible.com/overview/how-ansible-works.

[8] Ansible Documentation, "About Ansible", 9 8 2018. [Online]. Available: https://docs.ansible.com/ansible/latest/index.html

[9] SSH, "SSHD_CONFIG – SSH server configuration", 9 8 2018. [Online]. Available: https://www.ssh.com/ssh/sshd_config/.

[10] Pagač Alen; Šimec Alen; Tepeš Golubić Lidija; Primjena Drupal CMS-a u izgradnji web sustava; Polytechnic & Design; Vol. 5, No. 2, 2017.; ISSN: 1849 – 1995

[11] Šimec Alen, Tkalčec Siniša; Postgis kao suvremeni informacijski ekosustav; Polytechnic & Design; Vol. 2, No. 1, 2014.; ISSN 1849 – 1995