GROOT: Infrastructure Security as a Service (ISaaS)

Implementing Security Rules, Safeguards, and IDS tools for Private Cloud Infrastructures

Author
Aleksander Okonski
aleksander.okonski.1656@student.uu.se

Supervisor: Salman Toor
Review: Bjorn Victor

Contents

1 Introduction

The work done in this thesis was done with the intention of facilitating security information in private cloud infrastructure when an organization does not have the resources to deploy sophisticated security solutions. Open source cloud solutions, such as OpenStack, are becoming sought after solutions that allow anyone to create there own private cloud. These implementations lack a clear cut solution to security monitoring. This could lead to weaknesses being exploited in the private cloud allowing unauthorized users access to recourses. To alleviate the threat a three step solution was created. The first was to create a basic outline so that new users could understand the basic steps to securing communicate with created virtual machines (VMs). The second was to create an active monitoring solution for OpenStack. The solution was to be versatile for any type of environment and for administrators to be able to obtain viable information with ease. Lastly a script was created to take a image of a VM once it was found to be compromised. The project is to designed to approach the security of private clouds actively instead of passively. The administrators should be able to detect vulnerabilities early before they are exploited.

2 Problem Description

Servers exposed to the Internet face a myriad of cyber attacks daily(https://www.bleepingcomputer.
server-gets-infected-with-wannacry-ransomware-6-times-in-90-minutes/). For private clouds this is ongoing problem as meny of the users may not know of proper security protocals and leave ports such as SSH open. These types of security vulnerabilities would allow an attacker to gain full control of the virtual machines. To protect the public cloud, such as OpenStack, the mechanisms are thin. The administrators could attempt to monitor network traffic or install specific software on each VM. These solutions are either difficult to set up and maintain or only passively work. Network scanning does not prevent threats but try to catch ongoing attacks (https://www.giac.org/paper/gsec/235/limitations-network-intrusion-detection/100739). This may lead to vulnerabilities present in systems that the administrators do not know about. Its also impractical having to deploy a security solution to every VM that is started on the system. Users may not turn these systems on or may require different OS version that are not compatible with a security system. The cloud is one method of

allowing users to set up and deploy computers with ease, adding additional factors may discourage users. Therefor the solution is to create an active system that can monitor other VMs for potential vulnerabilities.

3 Background

The cloud computing space has grown over the last several years. Business and academia are looking at solutions to migrate their existing infrastructure to the cloud. There are several reasons for this type of business shift: costs, scalability, reliability [13]. The cloud offers some precedented advantages to a standardized computational model. One is able to pay for only the resources used, with more resources added/removed depending on the demand. Another advantage is the ability to spine up/destroy several machines with little overhead. Several companies are fronting the cloud revolution, some examples are Amazon[9], Google[12], Microsoft[10], and Digital Ocean[16]. These companies are providing a public cloud environment, that is to say that anyone can pay for computer resources. A user needs to sign up, provided a credit card, and then start setting up and launching whatever configuration of system they would like.

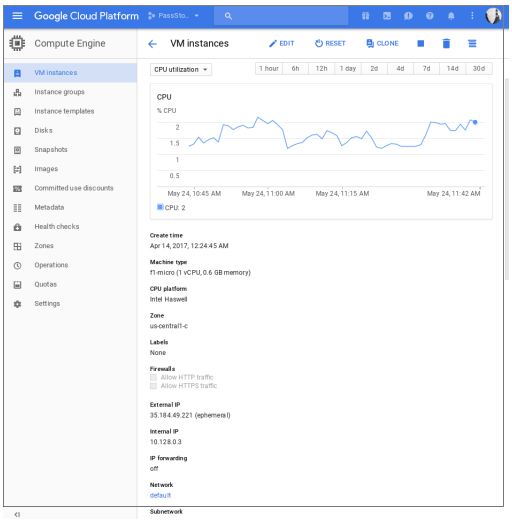


Figure 1: Google cloud interface

The provider will then dedicate the specified hardware and allow the user to login that virtual machine and run whatever tasks the user would like to do.

Many companies that would like to move over to a cloud based model but do not want to migrate there private recourses into public domain. Other companies such as RackSpace, IBM, and VMware are providing private / hybrid cloud resources. This approach has the part or the entire system built on site at the client. This allows the client to have most of the benefits of the cloud but also keep the system within there network.

3.1 Open Source vs Closed Source

An open source project is a type of coding project that has the source code to the project freely available for any one to download, modify, and run. A closed source coding projects is typically found in corporations. The code is not really available and only the end product is distributed to consumers.

3.2 Cloud Models

As mentioned above the cloud space consists of three types of clouds: public, hybrid, and private. The public cloud enables anyone to connect to the host and set up a machine. The host machines that run the cloud environment are located in public data centers throughout the world. The virtual machines are created then run and can be connected to the Internet. The private cloud consist of the host being located in a private hosting environment. The virtual machines that are created are collocated on the internal network. The hybrid environment merges the two, with having some of the machines collocated in private data centers and others located on public data centers. Each of these models have benefits and drawbacks and have to evaluated for each individual project. The these cloud models all share a smiler set of tools employed to provide the necessary computational recourses to users.

In the cloud computing space several computational models exist [1]. These models reflect the different stages of a computer: hardware (Software as a Service), operating system (Platform as a Service), and software (Software as a Service).

- Software as a Service (SaaS) allows for the user to utilize applications (I.E. Email, games, etc.) without the need to set up / worry about the underlying infrastructure. The cloud provider would be responsible with ensuring that the infrastructure and OS are running correctly.

  Software are programs that are run by users, these typically would be run on a personal computer or a server. The software is used by users to perform tasks that a user would like to perform, some examples of software are Microsoft Word, Gmail, and Photoshop. With cloud computing, these are now moved over to the cloud. A user then typically will use a web browser to access and utilize the software that is now run entirely on the cloud. The user has no access the underlying network, computer system, or storage.

- Platform as a Service (PaaS) give the user the ability to create applications (I.E. Web servers, databases, etc.) without the need to create the entire system from the ground up.

  Platform software is a layer of software that enables a user to build their own software from. Some examples of this are the Apache web framework, wordpress, and mysql. A user can install Apache onto their server to create a website. With the cloud model a user is now able to get access to platform software without having to deal with hardware or OS.

- Infrastructure as a Service (IaaS) gives the users a basic virtual machine with the user needing to set up all necessary functionality. This moves the responsibility of management of the hardware, network, and storage from the user to the operator.

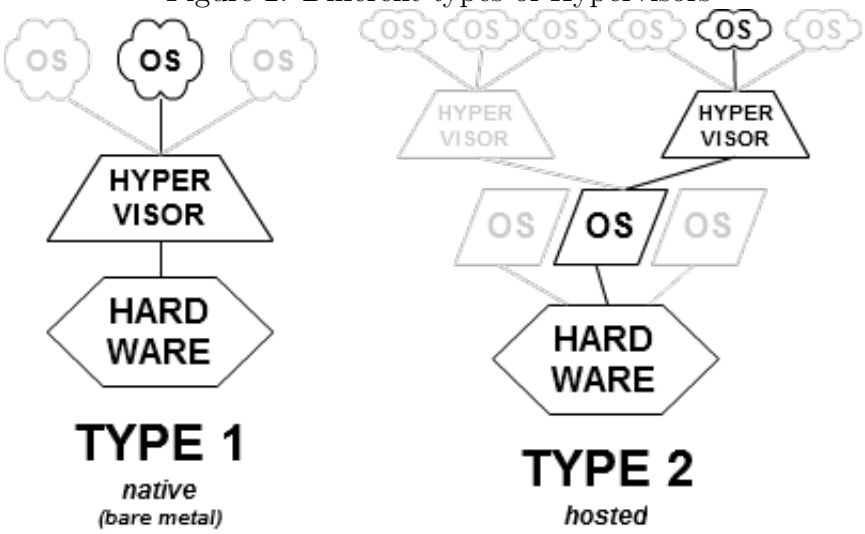
  Infrastructure is the bases of all systems, before cloud computing this was typically server and networking components. Microsoft gives a good explanation bellow: "The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud physical infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components."[https://social.technet.microsoft.com/wiki/contents/articles/4633.what-is-infrastructure-as-a-service.aspx.]

3.3 Cloud Infrastructure

At the most fundamental layer a cloud computer is a server running in a data-center that has a hypervisor which then contains and runs another operating system. These hypervisores are the backbone of cloud computing allowing several virtual environments to use the same hardware. A hypervisor is a layer of code that sits between the computer components (bare metal) and the guest OS. There are two types of hypervisores that are present in systems they are named bare bone and hosted hypervisores. Figure 2 shows these two in example formate. These two hypervisors are very different in the way that they are created. A bare bones hypervisor acts like a small

operating system which only manages the vms while allowing most of the vms to access hardware directly. A hosted hypervisor relies on an existing operating system to function. Although a hosted hypervisor can also directly allow vms to communicate with the hardware without going through the OS. OpenStack uses the hosted hypervisor solution, having the administrators install the OpenStack platform on top of an existing operating system. The hypervisor then translates instructions from the guest system onto the hardware. There are several different hypervisors to choose from (Xen, Oracle VirtualBox, Oracle VM, KVM, VMware ESX/ESXi, or Hyper-V) with each having similar outcomes through different approach to the problem. To control the users and virtual machines many cloud providers (Amazon, Google, etc.) have created proprietary solutions. However, NASA and RackSpace Hosting [5] have created an open source version called OpenStack.

Figure 2: Different types of Hypervisors



3.4 Cloud Roles

When cloud computing first started to take off, the main type of computing resource provided was IaaS in the public cloud [2]. This started to change in the recent years when two new types models for cloud computing emerged. Public and Hybrid clouds allowed for companies to utilize the power of the cloud while still having some or all of there resources located in their own

data centers.

3.5 Cloud Computing vs Standard Models

The cloud computing environment has been growing quickly the last several years [**?**]. However its important to understand the similarities and differences that this new type of computing brings. At the core cloud computing is no different then older client server models. The data is still on hard drives, servers still use OS environments, and networks still connect the servers to the Internet. The main difference between the two models is that in the cloud you pay for what you use and hardware configurations/problems are not a concern. Some of the advantages that cloud computing posses is the fact that there is rapid scalability. It is possible to spin up 100 machines and scale your service according to needs.

3.6 Threats to the cloud

Cloud computing also has a different threat model compared to a normal dedicated server [20, 15, 14]. As cloud servers are, as the name suggests, in the cloud they usually will receive a pubic IP address that allows anyone to directly talk to them. This attach surfaces allows for direct communication to the running system which exposes it to exploitation. The following list shows the attack vector's that a cloud computing systems has [8].

- Data lose or leakage
- Account or service hijacking
- Insecure interface
- Denial of service
- Malicious insider
- Data breaches
- Abuse of cloud services
- Insufficient due diligence

4 Related Work

In this work we will be looking at ways to protect VM clusters by ensuring proper configuration steps are setup and used with the addition of looking at ways to implement an IDS solution into the cloud environment. Before starting the work, we looked into previous work done in this field. Cloud security had been a hot topic in recent times therefor several papers have been written [20, 15, 14]. These particular papers focus on the different aspects and concerns that are present when running in a cloud environment. They are a nice starting point to look at how the threat landscape in the cloud differs from the slandered model. An important distinction of how information is treated differently in a centralized and cloud environments is talked about in "Assessing Cloud Computer Security Issues" [20]. The three staples of information security are confidentiality, integrity, and availability. In the cloud new difficulties come up for each of these classifications as data in the cloud is now accessible to more individuals. The second set of articles looked at was about intrusion detection systems (IDS), more specifically how these can be used within the cloud [11, 17]. These articles did not focus so much on implementation as they were focused more on the theory. An interesting comparison that shows the advantages and disadvantages for different IDS systems is table 2 in [**?**, SurveyOfIDS] This is the basis for the types of IDS solutions that were chosen for this project. As this work was primarily focused on OpenStack, one particular IDS conference talk was used as a starting point for this research [7].

This type of network and system monitoring is not an uncommon. At Cambridge University a set of probes is run on networks to ensure no security holes and ensure administrator's know what is running on the network [3]. In this research a similar type of probing was created for the OpenStack cloud platform.

5 OpenStack

OpenStack is an open source platform for cloud computing [19]. OpenStack is built of many components that are designed to provide a different set of services (Nova, Neutron, etc.) the full list can be found on the open stack website located here. OpenStack is very scalable and diverse system that can be arranged to fit the needs of any cloud environment. For this project we

ran OpenStack newton version 15.0.0.0.rc1. OpenStack is very modular with several core features running as the backbone of the software. The features that were used for this project were: Nova, Neutron, and Swift.

Nova is the platform that provides access to OpenStack compute resources. It is the base platform that a user would interact with the set up, run, configure, and destroy machines. Neutron is the service that controls and configures all of the networking between machines and the external network. Swift is the platform that is used for object and blob storage.

Neutron is the OpenStack layer that provides access to the networking components of the cloud system.

6 Security

The field of computer security is large and diverse [**?**]. The general focus of computer security is to ensure that computer systems follow the CIA model of confidentiality, integrity, and availability. "The design artifacts that describe how the security controls (security countermeasures) are positioned, and how they relate to the overall IT Architecture. These controls serve the purpose to maintain the system's quality attributes, among them confidentiality, integrity, availability, accountability and assurance." [4] The disciplines that are focused on in this paper include: intrusion detection systems, network security, and system security. Intrusion detection systems come in two main forms, host based and network based. A host based system runs on the host and attempts to detect any security threats on the host machine. A network based IDS is connected on the network and inspects network traffic, trying to find threats by inspecting network traffic patterns. Both of these systems have advantages and disadvantages. With a host based system software must be installed and configured on each system. In a network IDS system the packets are monitored for unusual traffic patterns. There are disadvantages to this type of solution also, mainly network traffic can be encrypted and the overall volume of traffic encountered. Some examples of network based IDS tools are Snort. A large portion of the problem for computer security comes from the networks that computers are attached to. The network allows other computers to communicate and attempt to access the particular machine. This is greatly increased if the computer is not placed behind a firewall/rougher and is directly accessible form the Internet. If an Internet facing computer is not updated properly vulnerabilities can be used to run

unintended programs. A unintended program running on a systems can be a concern. These programs can be used to send email spam, steal credentials, or participate in larger network attacks.

6.1 Network Security

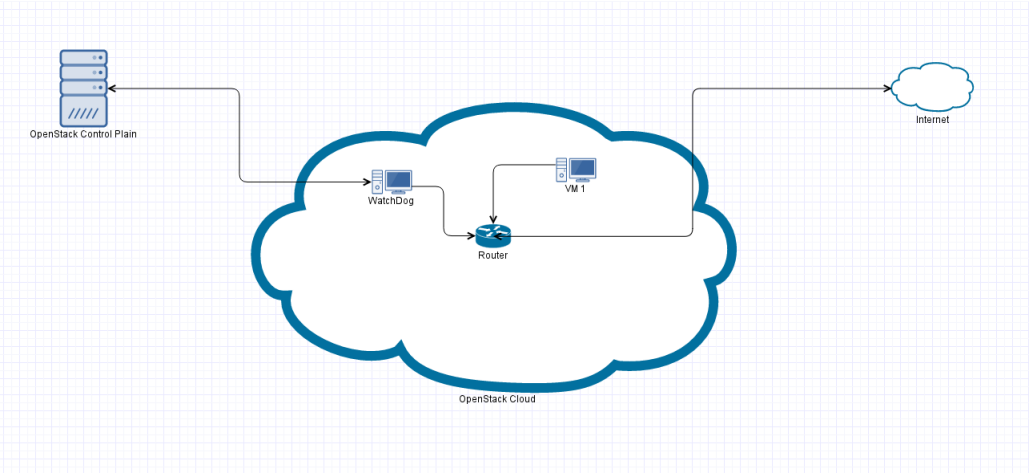
Network security aims to ensure that the underlying communication between hosts is secure. That may be to ensure that messages passed through the network are untampered or intercepted, ensure a device is properly protected from attach, or to divert/stop an attach reaching a machine. As this is a large field in itself we will break it down into smaller sections. Computers need to communicate with each other to relay information, this is done though the networking layers [6]. An attacker could, using different methods, intercept traffic and/or manipulate it. Network security in this context relates to ensure that information is preserved while in transit. For cloud security this type of network security is important to ensure that communication from a user to the machine is kept secure. Network security is also meant to protect machines from external network threats. This usually is done through a network firewall, which will sit between a external and internal network allowing only certain ẗrustedẗraffic through. As most cloud machines are through some way exposed to the wide interest ensuring that traffic to the machines is trusted can be difficult. Most cloud based machines are operated through ssh and may need other ports open to communicate with other services or machines. This posses a network security problem as external open ports will be repeatedly attached by botnets [18]. The last layer of security that the network can provide is protection against flooding or distributed denial of service attacks (DDOS attacks). These attacks try and overwhelm a server and take the service off line.

7 User Recommendation

The first stage of this project involved setting up user recommendations for configuring a secure VM and understanding some security features in the OpenStack platform.

8 Design Implementation

For the first part of the project a modular design was created to enable further types of scanning to be added. An important aspect when designing the watchdog system with as little privileges as possible to prevent any unnecessary security holes. The following is a diagram of an example cloud environment.



The "watchdog" is initiated in the cloud cluster. The once initiated the VM will connect to the OpneStack Control Plain and get the list of servers, IP's, and floating IP's. Once that information is retreated the different modules will be run. In this standard case the two modules created were for nmap and ssh_scan. Nmap is a tool used to scan a local or external network nd view what machines are funning on it. Nmap will be able to scan the machines ports and tell which ones are open and what services are running on them. It will also try and guess heuristically the operating system that is being run on the machine. This tool is therefor used in this system to find machines that have open ports and to see what services are run on them. If a service or port is found to not follow the recommendation then the machines can be terminated. The ssh_scan tool is used to ensure that only a public private key pair are used. It will scan port 22 and alert if any other form of authentication is used. These modules will gather the results from the network and collect them into a central repository. Once the data is gathered it is compared to the config file. If a discrepancy is found between the expected results and the scanned machine is found an alert is sent to the

administrator. For this project the administrator receives notification on a separate slack. This entire process is run every 20 min by a corn job located on the watchdog machine. This type of design allows for a flexible platform that can have each component further customized.

A key part of this project was to create a simplistic interface that would allow an administrator to view changes to the environment and query for more information. Some base goals were created: scalability, ease of use, and intuitiveness. As the system is designed to be deployed over several cloud groups we needed a messaging system that could handle such a task. Email was first considered however it was soon realized that email would not work well due to the frequency of data sent and then inability to interact with the service. The next consideration was Internet Relay Chat (IRC), however as IRC is relatively unused in the professional work environment and users would need to learn how to interact with the protocol that idea was abandoned. Therefor as a final solution it was decided to work to build the messaging solution using Slack. This matched all of our starting criteria and allowed for a clean and simple implementation. Slack is a messaging application that allows groups of users to interact with one another in channels or directly. Once of the really neat features of this is the ability to program bots for users to interact with. Therefor the decision was made to create a bot that would interact with a channel for each cluster of machines. The bot would update the channel with any new information that the scan found ensuring that a constant flow was preserved. Users are also able to query to bot directly and get more information about the environment. This allows seamless integration between getting information from the environment and being able to dive deeper and investigate a machine.

9 Architecture

For the monitoring solution an important architecture design was to ensure that the system was modular to enable it to be adapted to different cloud environments. This would allow individual monitoring tools or logging techniques to be used for configuring to specific infrastructures. As for the proof of concept project Nmap and ssh_scan were implemented.

The main portion of the entire program is the database. As the system is designed in a modular manner all the data is stored and retrieved from the database. The database is a MongoDB instance that has one collection that

hold all the data from different scans. The only data that is needed inside of the database is the server ID. This is the key to which all other data about a particular machine is added to. When data is stored the ID of the VM is passed along with the data. When a user requests data they must also provides the ID for the data they would like.

Apart from the database the other modules are started from the main python program. To do this the system will first connect to the OpenStack backend via credentials provided in a configuration file. This will pull down all of the VM IDs. This is then feed to each scanning model. These modules intern can query the backend for any additional information that they need. To illustrate this the Nmap modules will use the provided IDs to obtain the internal and external IP address for each machine. This will then feed the query to the Nmap program. The Nmap model will then obtain the data and parse out the needed information, such as open ports. Once the data from the model is as wanted it will be sent to the database to be added for each VM.

To add a new model the main function must be edited so that it will call the new model. The output from the new model must follow a set standardized of (ID: ID number; data) this is then sent via the main function to the database model to be interested into the database.

The Slack bot uses the slack API to communicate with the Slack web interface.

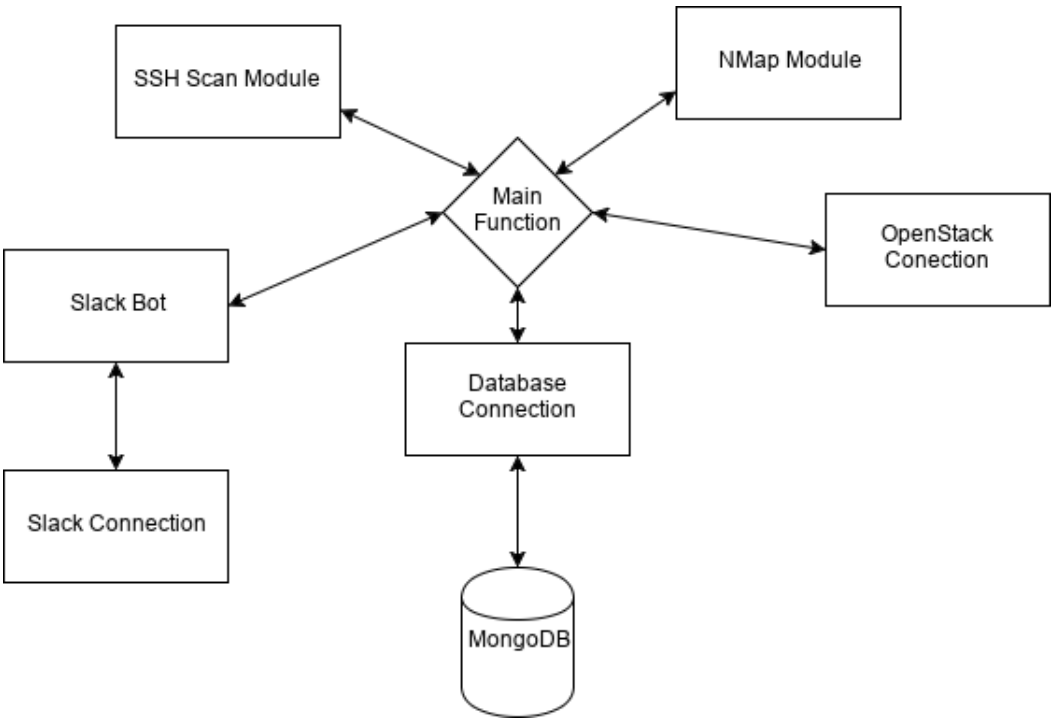


Figure 3: Intrusion detection system architecture, the main component is the "Main Function" and the database. All information is feed through the main function allowing fetures to easly be interchanged and expanded.

9.1 Tools

For this project there were several external tools that were used. These tools were picked to perform specific tasks like collect ssh configuration data.

9.1.1 Nmap

Nmap is a network scanning solution that can be used to scan for open ports, running services, network topography, and others https://nmap.org/. Nmap allows for scanned data to be saved into an ẋml file that can then be filtered and input into the database.

9.1.2 ssh_scan

Ssh_scan is another tool used to gather information about remote ssh configurations. It is able to detect the type of authentication, ssh protocol used, and several others https://github.com/mozilla/ssh_scan. The output is saved to a json file that is then read into the database.

9.1.3 Slack Bot

Slack bot is a small bot built on top of the Slack api to display information form the database and query data in real time. The bot will connect to a channel specifically created for each cloud cluster. There the bot will print relevant information from the database. One of the advantages that Slack provides is the ability to communicate between the users and the bots. Users are also able to query the database via the bot, this allows an administrator to quickly find a problem and find information about that machine with ease.

10 Tests

The system was designed and created in a testing environment. The environment was hosted in an OpenStack cloud environment with several test vms initiated. The watchdog vm was then spun up and the code was deployed. The code was then edited and redeployed via Ansible. New features were able to be tested on a live deployment ensuring that when deployed to other clusters they would function as needed. To ensure that the code functioned with little bugs, tests were run by spinning up new vms and changing there configurations.

11 Results

11.1 User Recommendation

A user recommendation was written to help users properly connect to initialized VMs. The recommendation focuses on how to connect to the VMs using a proper ssh key. The guild first walks thought how to set up a ssh key pair on a users local machine. This ensures that a user creates a proper strong key pair. The user is then instructed on how to add the public key

to the OpenStack interface. Along with the ssh connection other simple recommendations are provided to the user. These are to help a user understand the different security implications no mater what they use the VM for. Some examples of what the recommendations looks at are, open ports and security rules. The user recommendation can be found at the following https://github.com/DarkLog1x/MasterThesis/tree/master/UserRecomendation

11.2 Intrusion Detection System Results

Once the base system was running in our test environment a decision to scale up was made. Our test environment was compatible however a real environment with more machines and users could not have been replicated. Luckily a class was taking place that used the cloud environment. Therefor a watchdog vm was spun up in the classes cloud cluster and started to monitor the vms that were created. Immediately vms started to appear in the Slack channel. After running for a while inconsistencies were started to be found. A small snippet of the Slack channel is shown bellow. In this section 5 machines are shown, 3 of them have inconsistencies that are not in the configuration file.

Servers with the ids of $930ee1bb-e07a-461f-b09f-1e1a9c864a36$ and $81933a08-9451-4d16-aca4-9fb6269b4e0d$ both have 3 ports that are open but should not be. The more concerning vm is $f57b9369-4e15-4151-9b0b-7d405cbdb5e4$. We see that this vm allows for access with password bases authentication. The two machines ($ad74be2b-e452-43df-b307-d8ad3bcd6cb7$ and $6e19b2a1-1dbd-4fe1-83d6-8e57cdd08ff7$) have been created and follow the configuration therefor only the server ID is shown.

```
--Server With ID: 930ee1bb-e07a-461f-b09f-1e1a9c864a36
Server ID: 930ee1bb-e07a-461f-b09f-1e1a9c864a36 | 8088 has been
    found open and is not in the config
Server ID: 930ee1bb-e07a-461f-b09f-1e1a9c864a36 | 8042 has been
    found open and is not in the config
Server ID: 930ee1bb-e07a-461f-b09f-1e1a9c864a36 | 8031 has been
    found open and is not in the config
--Server With ID: f57b9369-4e15-4151-9b0b-7d405cbdb5e4
Server ID: f57b9369-4e15-4151-9b0b-7d405cbdb5e4 |
    ip_external_auth: publickey password keyboard-interactive --
     Should be = publickey
Server ID: f57b9369-4e15-4151-9b0b-7d405cbdb5e4 |
```

```
    ip_internal_auth: publickey password keyboard-interactive --
     Should be = publickey
Server ID: f57b9369-4e15-4151-9b0b-7d405cbdb5e4 | OpenSSH on
    port 22 is version: 7.4 -- Should be = 7.2p2 Ubuntu 4ubuntu2
    .1
--Server With ID: ad74be2b-e452-43df-b307-d8ad3bcd6cb7
--Server With ID: 81933a08-9451-4d16-aca4-9fb6269b4e0d
Server ID: 81933a08-9451-4d16-aca4-9fb6269b4e0d | 8088 has been
    found open and is not in the config
Server ID: 81933a08-9451-4d16-aca4-9fb6269b4e0d | 8042 has been
    found open and is not in the config
Server ID: 81933a08-9451-4d16-aca4-9fb6269b4e0d | 8031 has been
    found open and is not in the config
--Server With ID: 6e19b2a1-1dbd-4fe1-83d6-8e57cdd08ff7
```

Once a potential problem was found the bot can be queried to get a full report of the virtual machine in question.


```
Tenant name: SNIC 2017/13-8

Tenant ID: SNIC 2017/13-8

[
{
    "ID": "f57b9369-4e15-4151-9b0b-7d405cbdb5e4",
    "IP": {
        "ip_external": "130.239.81.28",
        "ip_external_auth": "publickey password keyboard-
            interactive",
        "ip_internal": "192.168.1.9",
        "ip_internal_auth": "publickey password keyboard-
            interactive"
    },
    "OpenStack_info": {
        "created_at": "2017-03-16T08:09:08Z",
        "image_id": "e4752852-c053-4fee-b198-6d990d914e3a",
        "image_name": "CoreOS - latest",
        "updated_at": "2017-03-16T08:09:25Z",
        "user_id": "d152c3dbd72c44af95ac4e6e48ccb0ec"
    },
    "_id": {
```

```
        "$oid": "58db779eb39aa86565e2f1f3"
    },
    "ports": {
        "22": "open"
    },
    "services": {
        "OpenSSH on port 22 is version": "7.4"
    }
}
]
```


The administrate then gets a full view of the machine in question and can then decide what the best possible next action is. As mentioned before the type of data that is present is not fixed therefor each module would add its own collected data for each specific VM. Therefor only the relevant data for the machine is displayed. This type of information condensed information was not present before this system wad developed. Administrators would need to run several programs to gather the data and then would need to synthesis it by hand.

11.3 Hypervisor Information Gathering Results

For the information gathering we focused on getting the memory dump, disk data, and system information.

12 Discussion

Thought the project an attempt was made to ensure that the project was well designed and created. However due to time constraints the code portion of the project will be branded as a prototype. This is because there are several drawbacks that could not have been overcome in this specific time frame. The current implementation utilizes a individual VM for every cluster. This adds up recourses quickly, if a cloud system had 100 clusters then there would be 100 VMs that would only be used for monitoring. It would be beneficial if it was possible to have a smaller number of VMs used to scan multiple clusters. However the current system is unable to perform this type of task. Currently Slack is used as a medium to communicate between the administrators and the scanning mechanism. This has provided a really interesting tool allowing

or bidirectional communication and quick information retrieval. However the reliance on Slack may be jeopardized if Slack decides to change api's, limit bots, or goes out of business. Luckily the system was designed in such a way to allow any communication medium to be added to the system. Therefor if Slack does not continue to meet the standards that an organization needs another tool can be used to replace Slack with ease. An alternative that can be looked into is called HipChat. An issue that was ran into later during the creation of the project was that the Slack channel would get cluttered if a lot of different devices were not following the configuration file. One way to sort out the higher risk targets would be to add symbol's. These symbol's would be able to indicate the severity level / risk that each output would pose. Some ideas to fix this were thought of but were not implemented due to time constraints.

13 Future Work

The project has a large potential. As the code written now is only a prototype there is more work that can be added to expand the tools usefulness. As the code is created with modularity in mind, a user can easily add new scans into the project. If a user wanted, for example, to ensure that a website is correctly configured they could add the nikto /citeNiktoScan vulnerability scanner. The scan can then be run and the data would be added to the database. Another future work that could be expanded on is allowing for different methods of interacting with the system. Currently Slack is used, however administrator's might want email reports or an irc channel instead of Slack. Luckily this can easily be integrated and several systems can be used asynchronously. This allows for administrators to custom configure this system to their desired needs and specifications.

14 Conclusion

tmp

References

[1] Cloud computing. https://en.wikipedia.org/wiki/Cloud_computing.

[2] Cloud history. http://sourcedigit.com/497-timeline-history-of-cloud-computing/.

[3] Friendly probing. http://www.designsequence.net/gkeen/more_html/www-uxsup.csx.cam.ac.uk/security/probing/.

[4] It security architecture. http://www.opensecurityarchitecture.org/cms/definitions/it-security-architecture.

[5] Openstack. https://en.wikipedia.org/wiki/OpenStack.

[6] osi model. https://en.wikipedia.org/wiki/OSI_model.

[7] unobtrusive intrusion detection in openstack. https://www.openstack.org/summit/vancouver-2015/summit-videos/presentation/unobtrusive-intrusion-detection-in-openstack.

[8] A Amini, N Jamil, AR Ahmad, and MR Z'aba. Threat modeling approaches for securing cloud computing. *Journal of Applied Sciences*, 15(7):953, 2015.

[9] Amazon AWS. https://aws.amazon.com/, 2017.

[10] Microsoft Azure. https://azure.microsoft.com/en-us/, 2017.

[11] Bhavesh Borisaniya Hiren Patel Avi Patel Muttukrishnan Rajarajan Chirag Modi, Dhiren Patel. A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, 2013.

[12] Google Cloud Compute. https://cloud.google.com/, 2017.

[13] Tharam Dillon, Chen Wu, and Elizabeth Chang. Cloud computing: Issues and challenges. *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference*.

[14] Ronald L Krutz and Russell Dean Vines. *Cloud security: A comprehensive guide to secure cloud computing*. Wiley Publishing, 2010.

[15] Ankur Mishra, Ruchita Mathur, Shishir Jain, and Jitendra Singh Rathore. Cloud computing security. *International Journal on Recent and Innovation Trends in Computing and Communication*, 1(1):36–39, 2013.

[16] Digital Ocean. `https://www.digitalocean.com/`, 2017.

[17] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari, and Joaquim Celestino JúNior. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of network and computer applications*, 36(1):25–41, 2013.

[18] Wikipedia. Botnet — wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Botnet&oldid=773661657`, 2017. [Online; accessed 6-April-2017].

[19] Wikipedia. Openstack — wikipedia, the free encyclopedia. `"https://en.wikipedia.org/w/index.php?title=OpenStack&oldid=762630785`, 2017. [Online; accessed 30-January-2017].

[20] Dimitrios Zissis and Dimitrios Lekkas. Addressing cloud computing security issues. *Future Generation computer systems*, 28(3):583–592, 2012.