Investigation of Virtual Network Isolation Security in Cloud Computing: Data Leakage Issues

Haifa Al Nasseri
School of Computer Science
University of St Andrews
Email: hmkan@st-andrews.ac.uk

Ishbel Duncan
School of Computer Science
University of St Andrews
Email: Ishbel.Duncan@st-andrews.ac.uk

***Abstract*—Software-Defined Networking (SDN) or Virtual Networks (VNs) are required for cloud tenants to leverage demands. However, multi-tenancy can be compromised without proper isolation. Much research has been conducted into VN Isolation; many researchers are not tackling security aspects or checking if their isolation evaluation is complete. Therefore, data leakage is a major security worry in the cloud in general. This paper uses an OpenStack VN and OpenStack Tenant Network to test multi-tenancy features. We aim to evaluate the relationship between isolation methods used in cloud VN and the amount of data being leaked through using penetration tests. These tests will be used to identify the vulnerabilities causing cloud VN data leakage and to investigate how the vulnerabilities, and the leaked data, can compromise the tenant Virtual Networks.**

I. Introduction

Virtual Networks (VNs) are one of the current hot research topics. VNs helped establish the new concept of the Future Internet [1]. However, new projects and technologies demonstrated the potential benefits of using VNs in Cloud Computing, e.g: Cloud Networking, NaaS, CloudNaas, OpenStack Quantum/Neutron [2]–[4]. This research will utilize OpenStack Neutron to implement the tenants Virtual Network.

Isolation is a main requirement to guarantee privacy and safety of tenant's data as well as providing independence of services or data traffic. However, without proper isolation, the network scalability and performance is compromised [5] [6]. According to the Cloud Security Alliance [7], medium-to-large enterprises flagged that top vulnerabilities were caused by unsuitable data and network isolation. Additionally, researchers [6] clarified that tenants may attack other tenants by DOS or resource consuming if proper isolation has not been used in a VN.

Data leakage is one of the major worries in the cloud due to the multi-tenancy feature [8]. In [9], the authors listed two common concerns arising from a cloud customer survey made by Fujitsu. The two major concerns were: were the information appropriately isolated and what if the cloud operation would lead to data leakage or corruption? This research aims to clarify these concerns, focusing on how appropriately a VN is isolated and if the isolation methods used cause the data leakage. However, the data of concern here are those that could be used to create vulnerabilities in cloud VN Isolation functionality such as in discovering the flow rules or the forwarding policies of the network.

The aims of this research are to detect the VN isolation methods used in OpenStack and to identify the vulnerabilities causing data leakage in a tenant network through penetration testing. An evaluation of the relationship between the amount and quality of VN related data leakage and the VN isolation methods are underway.

In this paper we use certain terminologies to represent specific meaning in this research. The first term is components and is used to refer to ports, virtual switches -Open vSwitch (OVS) or Linux switches-, namespaces, etc. The second term is mechanisms and it implies the use of the combination of components to form the network structure and connections. And finally, the term method indicates the process of how the network isolation is done.

This paper will outline the related work of VN isolation in Section 2, followed by the methodology, the testbed and the implementation status of this research. The expected re- sults are outlined and finally, Section 3 will be the conclusion and the future work.

II. Related Work

Isolation mechanisms and technology by itself carry several challenges needing be solved such as: isolation expressing and verification, network configuration, isolation provision and guaranteeing, performance, scalability and other engi- neering factors [10] [11] [5].

Nevertheless, isolation related security hasn't received proper attention in the research community, although several authors have raised their concerns, for example [12] stated that the multi-tenancy feature in virtualization compromises the client confidentiality without a certain degree of isolation.

Many researchers have investigated VN Isolation using different methods such as: VLAN, GRE [13], EtherIP [14], Flowvisor [15], OVX [16] and OpenFlow [17], etc. The aim of their researches was to define the VN Isolation mechanisms that was adopted in their case studies, or into their proposed projects as VN Isolation solutions. For instance, the Splendid Isolation project [11] was built to solve the VN configuration issues through Isolation.

However, none of those researchers tackled the security of VN Isolation as its main goal. Therefore, our research selects some of the above Isolation methods, those that are applicable

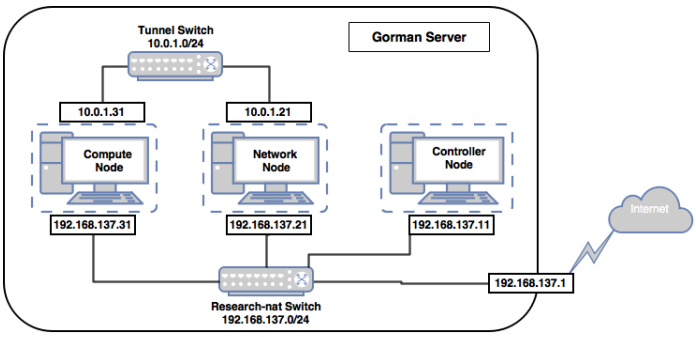

Fig. 1. Three-Nodes OpenStack Cloud Architecture.

to be implemented in OpenStack Neutron, and investigates their vulnerabilities under a controlled experiment.

The Survey of Security in SDNs [18] provided clear security issues related to VNs and openly stated that the least researched security aspect in the field were data leakage and data modification. Consequently, this research contribution will be restricted to enhancing the cloud VN Isolation against data leakage.

The following articles stated theories of the relationship between isolation and data leakage, where the research intended to prove or disprove the theories via empirical work:

1 [18] Conveyed an assumption that if a VN (called logical network here) and their identification/authorization information of a each tenant is securely isolated, then the data that could be used to compromise the network functionality should not be leaked.

2 [19] Stated that unsuccessful resource isolation between the tenants in the cloud may result in a prospective harmful security effect and mentioned side-channel data leakage as one of the security attacks.

3 [20] Declared that operating isolated networks sliced alongside each other, leaking network data from one slice to another should not be achievable.

4 [21] Acknowledged that to guarantee no data leakage between virtual instances in a cloud environment, a well-defined isolation of data and processes is a must.

III. Empirical Work

A. Testbed

The main emphasis of this research is one part of Infrastructure-as-a-Service (IaaS). The OpenStack Cloud environment was selected as a testbed based on the comparison study made by [4]. Moreover, OpenStack contains a module called Neutron, previously called Quantum, which was specifically built for Virtual Networking Services in the cloud [22]. Finally, the [4] comparison encouraged administrators and researchers to deploy OpenStack especially if the intention was to use the existing resources or to use the stack system for experimentation.

The testbed was implemented based on a multi-node architecture with OpenStack Networking (Neutron) using the OpenStack installation guide for Ubuntu 14.04 Kilo [23]. The

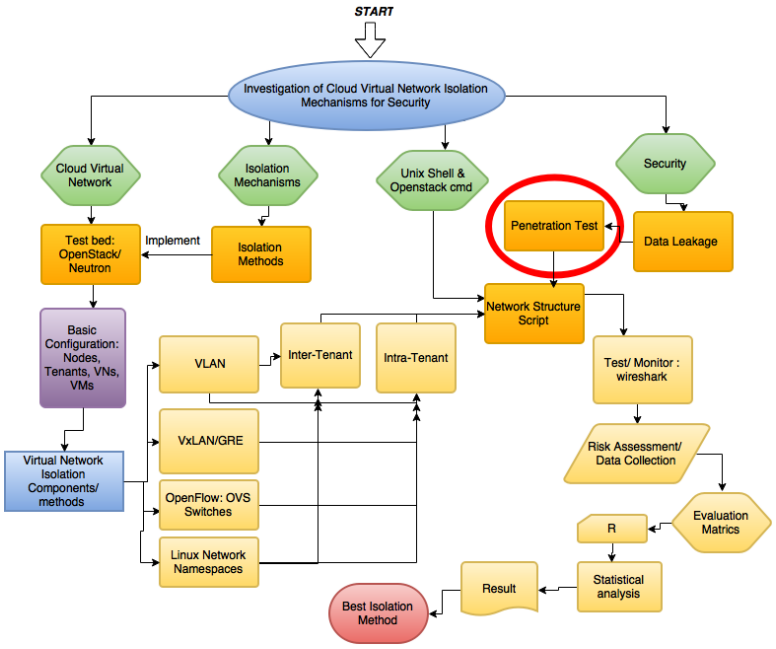

Fig. 2. Test Cloud Virtual Network Isolation with penetration test for data leakage.

testbed shown in Figure 1 consists of three nodes: Controller Node, Compute Node and Network Node. How- ever, the Network and Compute nodes (see Figures 3 & 4) are the focal points in this research as all the VN Isolation components and methods are implemented in both. This will be explained in the next sections.

B. Implementation Status

Figure 2 shows the core of this research, which is to investigate the cloud VN Isolation mechanisms for security. The diagram shows the three main topics of the research: cloud VN, isolation mechanisms and security. The cloud VN represents the experimental testbed, where OpenStack was used as the cloud environment with Neutron to provide the VN implementation (Network as a Service).

In the testbed a scenario of two tenants (intra-tenants) was implemented. Using white box testing to uncover the cloud underlying network structure, it was discovered that OpenStack/Neutron uses four isolation mechanisms/components: VLAN, GRE, OpenFlow (OVS switches) and Linux Network Namespaces.

1) Penetration Tests:: the scope of penetration testing is to focus on exploiting the vulnerability of cloud VN infrastructure, specifically the isolation mechanisms and how secure they are against the data leakage. Two types of penetration test are planned: Network Penetration Test (NIST) and Web Application Penetration Test (OWASP) [24]. OpenStack allows tenants to access their cloud instance in two ways, one way through SSH using the instance floating IP address (Public IP address) and, separately, through a web API called Horizon. Hence, the plan was to use two penetration test methodologies:

NIST and OWASP . The Plan and Discover phases of NIST have already been completed. In the Discover phase of NIST, a white box test was applied by executing several scripts written with shell and OpenStack commands. The scripts executed in the three nodes and collected all possible information about the network structure, components, configuration and databases information. The output of the scripts was used to sketch the internal network components and connections as shown in Figures 3 & 4.

The next two phases of NIST will be the Attack and the Report phases. In the Attack phase we will reuse information from the Discover phase, as it is needed. Moreover, in the Attack Phase we will decide if a component is to be tested and monitored. We currently consider using Wireshark for this phase, but at this stage we can only decide what to monitor or how to test the data leakage after defining the data leakage requirement and parameters.

More penetration testing will be conducted via OWASPs methodology, as this testing is specific for web application testing, as tenants access their cloud assets using the OpenStack Horizon API. In this case, black box testing will be the approach used with the assumption that the tenant wishes to gain extra information of the underlying system with the their current privileges through the web API.

2) Data Collection and Analysis:: Once we successfully collect all the data and identify the risk assessments variables, we can then find the appropriate evaluation metrics that will reflect the number and the type of data leakage. In order to produce the statistical analysis, we will use the R statistical package.

3) Cloud Virtual Network Components and Connection:: as mentioned above, the diagrams of the internal network components of the Compute and Network nodes were drawn (see Figure 3 & 4) after running the scripts and analysing the data collected. Figures 3 & 4 show how the underlying structure of the tenant VN is concentrated in the Compute and Network node.

In Figure 3 the Compute node connects each tenants instance to a unique Linux vSwitch (**qb**) with a collection of virtual ports (**tap & qvb**), however, all tenant instances share the same OVS switch (**br-int**) with each instances **qvb** port paired with **qvo** in an internal bridge, **br-int**, switch. However, all **qvo** belong to a tenant grouped under the same VLAN tag, while other tenant **qvb** ports have their own tenant VLAN tag.

As is noted from Figure 3 the ports (**tap, qb, qvb & qvo**) identified with the same number are automatically generated by OpenStack to represent the connection path of one instance. Moreover, the traffic will be forwarded to the network node through the **br-tun** bridge where all tenants share the bridge and **GRE** port. On the other hand - as Figure 4 demonstrates- the Network node receives the traffic from Compute node through the tunnel switch to the **GRE** port in the **br-tun** bridge before transfering the traffic to the **br-int** bridge. In this scenario, we created one private network and one internal router for each tenant. OpenStack (see Figure 4) created two network namespaces for each tenant, **qdhcp** representing the

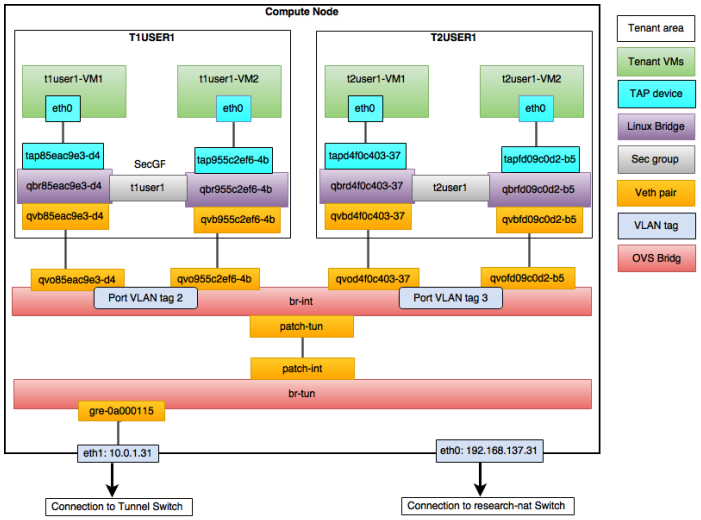

Fig. 3. Compute Node Internal Network Structure of Two Tenants Scenario.

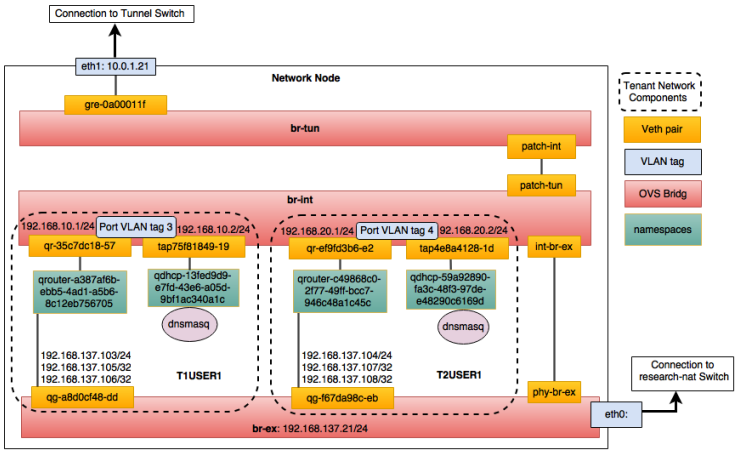

Fig. 4. Network Node Internal Network Structure of Two Tenants Scenario.

DHCP server for the tenants private network and **qrouter** representing the tenants internal router. The bridge **br-int** consist of two types of ports; each **tap** port is related to a specific tenant **qdhcp** namespace, while the **qr** port is connected to the tenants internal router and is considered to be the tenants private network gateway. Both **tap & qr** ports are labeled with a specific tenant VLAN. Moreover, the **qg** port is another port that is connected to the tenant router namespace and is the connection to the external network, configured with floating IP Addresses (public IP addresses for each tenant instance). This physically exists in the **br-ex** OVS bridge. The **qrouter** namespace perform NAT through the iptables within the namespace to allow tenants to connect to their instance remotely using the floating IP addresses.

C. Expected Results

If the virtual network is not securely isolated, a data leakage most likely will occur and this defies the main security requirement of VNs. Our analysis will tackle two categories

and they are: the amount of data leaked (quantity), and the type of data leaked (quality).

Once the data leakage concept has been precisely defined, the testing will lead us to define the criteria for effective isolation methods. To illustrate, if our testing parameters focus on the number of packages leaked per isolation mechanism for each test case, then we can calculate the mean, standard deviation or percentage, and this will indicate the most or least mechanism leaking data. On the other hand, if we categorized the data leaked from each method into types, flow data or forwarding policy data, and distinguish the importance of each, then we will have a reference of which is the most dangerous mechanism to be leaking data used to compromise VN functionality.

We may obtain some statistical analysis from Wireshark, although we are planning to use it for testing and monitoring. After collecting the data in the VN Isolation scenarios, before and after an attack, we will compare the amount of leakage exposed. The outcome data will be analyzed via R.

IV. Conclusion and Future Work

Cloud providers attempt to fulfill a clients requirements by providing more advanced demands in regards to the performance, scalability and traffic isolation. VNs share multi-tenancy features with the cloud and these are successfully satisfied with proper isolation mechanisms. Although security is one of the aims of introducing VNs to the cloud, and despite the fact that VN isolation mechanisms have research attention in the industrial and academic fields in term of the performance, scalability and deployment, the area lacks empirical security research.

This paper outlines an investigation into VN isolation mechanisms used in a cloud and exploits their security vulnerabilities in relation to network data leakage.

The research uses OpenStack as a testbed to investigate a virtual network created and managed by the neutron module. Due to the structure of OpenStack in giving the tenant access their instance either through the network using SSH or through the browser using Horizon web application, two types of penetration tests have been selected for trials. They follow the NIST and OWASP test methodologies.

Currently, two stages (Plan and Discovery) of the NIST method have been completed through white box testing conducted to reveal the internal cloud network structure and expose the isolation mechanisms OpenStack network infrastructure relies on.

The next step will be the attack phase on OpenStack network isolation mechanisms and components to discover any incident of data leakages and the type of data been leaked.

A similar scenario will be adopted to perform the OWASP test on the OpenStack Horizon API and collect any network related data and the type of data collected.

Finally, with statistical analyses of the collected data, a conclusion will be drawn of which VN isolation mechanism and components allow us to compromise the OpenStack Cloud Virtual Network.

References

[1] M. El-Azzab, I. L. Bedhiaf, Y. Lemieux, and O. Cherkaoui, "Slices isolator for a virtualized openflow node," in *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*. IEEE, 2011, pp. 121–126.

[2] J. Carapinha, P. Feil, P. Weissmann, S. E. Thorsteinsson, M. Melo, Ç. Etemoğlu, Ó. Ingórsson, and S. Çiftçi, "Study Report Network Virtualisation – Opportunities and Challenges," *Eurescom*, no. December 2010, 2010.

[3] T. Benson, A. Akella, A. Shaikh, and S. Sahu, "Cloudnaas: A cloud networking platform for enterprise applications," in *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 8:1–8:13. [Online]. Available: http://doi.acm.org/10.1145/2038916.2038924

[4] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "Comparison of multiple iaas cloud platform solutions," in *Proceedings of the 7th WSEAS International Conference on Computer Engineering and Applications,(Milan-CEA 13). ISBN*, 2012, pp. 978–1.

[5] D. Schlosser and M. Jarschel, "Network virtualization: Isolation problems and scalability issues."

[6] H. Moraes, R. V. Nunes, and D. Guedes, "Dcportalsng: Efficient isolation of tenant networks in virtualized datacenters," *Proc. 13th ICN*, 2014.

[7] Y. Mundada, A. Ramachandran, and N. Feamster, "Silverline: Data and network isolation for cloud services." in *HotCloud*, 2011.

[8] C. Alliance, "Security guidance for critical areas of focus in cloud computing v3. 0," *Cloud Security Alliance*, 2011.

[9] M. Okuhara, T. Shiozaki, and T. Suzuki, "Security architecture for cloud computing," *Fujitsu Sci. Tech. J*, vol. 46, no. 4, pp. 397–402, 2010.

[10] F. Hu, *Network Innovation through OpenFlow and SDN: Principles and Design*. CRC Press, 2014.

[11] S. Gutz, A. Story, C. Schlesinger, and N. Foster, "Splendid isolation: A slice abstraction for software-defined networks," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 79–84.

[12] A. Behl and K. Behl, "An analysis of cloud computing security issues," in *Information and Communication Technologies (WICT), 2012 World Congress on*. IEEE, 2012, pp. 109–114.

[13] S. Hanks, D. Meyer, D. Farinacci, and P. Traina, "Generic routing encapsulation (gre)," 2000.

[14] A. Edwards, A. Fischer, and A. Lain, "Diverter: a new approach to networking within virtualized infrastructures," in *Proceedings of the 1st ACM workshop on Research on enterprise networking*. ACM, 2009, pp. 103–110.

[15] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer," *OpenFlow Switch Consortium, Tech. Rep*, pp. 1–13, 2009.

[16] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, and G. Parulkar, "Openvirtex: A network hypervisor," in *Open Networking Summit 2014 (ONS 2014)*, 2014.

[17] R. Kloti, V. Kotronis, and P. Smith, "Openflow: A security analysis," in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[18] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," 2015.

[19] W. Shi, J. Lee, T. Suh, D. H. Woo, and X. Zhang, "Architectural support of multiple hypervisors over single platform for enhancing cloud computing security," in *Proceedings of the 9th conference on Computing Frontiers*. ACM, 2012, pp. 75–84.

[20] C. Schlesinger, A. Story, S. Gutz, N. Foster, and D. Walker, "Splendid isolation: Language-based security for software-defined networks," in *Proc. of Workshop on Hot Topics in Software Defined Networking*, 2012.

[21] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation computer systems*, vol. 28, no. 3, pp. 583–592, 2012.

[22] S. A. Baset, C. Tang, B. C. Tak, and L. Wang, "Dissecting open source cloud evolution: An openstack case study," in *Presented as part of the 5th USENIX Workshop on Hot Topics in Cloud Computing*, 2013.

[23] "OpenStack Installation Guide for Ubuntu 14.04-Kilo." [Online]. Available: http://docs.openstack.org/kilo/install-guide/install/apt/content/

[24] R. Baloch, *Ethical Hacking and Penetration Testing Guide*. CRC Press, 2014.