2016 IEEE 41st Conference on Local Computer Networks Workshops

Towards Vulnerability Assessment as a Service in OpenStack Clouds

Kennedy A Torkura and Christoph Meinel
Chair of Internet Technologies and Systems
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
Email: {kennedy.torkura, christoph.meinel}@hpi.de

***Abstract*—**
Efforts towards improving security in cloud infrastructures recommend regulatory compliance approaches such as HIPAA and PCI DSS. Similarly, vulnerability assessments are imperatives for fulfilling these regulatory compliance requirements. Nevertheless, conducting vulnerability assessments in cloud environments requires approaches different from those found in traditional computing. Factors such as multi-tenancy, elasticity, self-service and cloud-specific vulnerabilities must be considered. Furthermore, the Anything-as-a-Service model of the cloud stimulates security automation and user-intuitive services. In this paper, we tackle the challenge of efficient vulnerability assessments at the system level, in particular for core cloud applications. Within this scope, we focus on the use case of a cloud administrator. We believe the security of the underlying cloud software is crucial to the overall health of a cloud infrastructure since these are the foundations upon which other applications within the cloud function. We demonstrate our approach using OpenStack and through our experiments prove that our prototype implementation is effective at identifying "OpenStack-native" vulnerabilities. We also automate the process of identifying insecure configurations in the cloud and initiate steps for deploying Vulnerability Assessment-as-a-Service in OpenStack.

***Index Terms*—Cloud-security, vulnerability assessment,Security as a Service, cloud-specific vulnerabilities**

I. Introduction

Cloud computing landscape is witnessing a dramatic phase. There is a growing adoption of private Infrastructure as a Service (IaaS) clouds, as the security of this cloud model offers more guarantees than public clouds. Almost every major Cloud Service Provider (CSP) offer a kind of "private cloud", yet there is a preferrence among enterprises for either a hosted private cloud or an "on-premise" private cloud. RightScale [1] asserts that OpenStack and VMware are the leading private cloud vendors. OpenStack [2] offers an open source cloud computing software while VMware [3] provides a proprietary and commercial cloud computing suite. Benefitting from the innovativeness of open-source software, OpenStack has beome a very popular option for private cloud. OpenStack offers almost every service available on commercial clouds such as Amazon Web Services (AWS). However, several challenges hinder OpenStack adoption, accordingly the number of production-level deployments is not commensurate with its popularity. The lack of sufficient security assurances largely contribute to this lapse[4]. Several factors militate against security in OpenStack. It's ecosystem is complex, consisting of over then 15 projects (which could be either installed independentaly/semi-independentaly). Deploying a full stack cloud requires multiple levels of configuration across these services, this introduces misconfigurations. Furthermore, the bi-yearly release circle is a challenge for maintaining stable and current deployments [4]. Every release includes changes in configurations, packages are either deprecated or modified and sometimes new services are introduced. Since there are no provisions for automated upgrades, maintaining current releases introduces security issues. Moreso, there is no integrated vulnerability assessment tool for identifying these security issues.

Existing research in the scope of OpenStack vulnerabilities identify some of these security challenges yet solutions are hardly proffered. The OpenStack security team maintains resources which could be leveraged to improve a good number of these security challenges. However, these resources have low publicity and suffer from lack of automated approaches/tools. But automation is a key requirement in cloud security [5] and it is critical for regulatory compliance. Similarly, several attempts at standardizing the security in the cloud recommend vulnerability assessment as an important actvity [6].

The Cloud Security Alliance (CSA) specifies implementation guidelines for Security-as-a-Service (SecaaS) including its sub-categories such as Vulnerability Assessment-as-a-Service (VAaaS), Monitoring as a Service (MONaaS) and Firewall-as-a-Service (FWaaS). We are more concerned with VAaaS considering its requirement for compliance regulations. Hence, in this work we introduce approaches for designing and employing VAaaS in OpenStack. Our focus is on the system level vulnerabilities from a cloud administrator perspective. We consider vulnerabilities native to OpenStack core software, best practices recommended by the OpenStack security team and how these security provisions can be automated and integrated into a vulnerability assessment system. Hence we designed and implemented Cloud Aware Vulnerability Assessment System (CAVAS), a prototype system that resolves the mentioned security challenges.

Our major contributions are as follows:

- We introduce an approach for integrating vulnerability

DOI 10.1109/LCNW.2016.15

assessment in OpenStack especially at the system level, suitable for cloud administrators.
- Integration of our previously introduced approach [7] for reducing the "window of opportunity" (for vulnerability exploitation) introduced by late release of security patches.
- We present an automated approach for conducting security checks through the use of security policies and security best practices that are recommended by OpenStack security team.
- We briefly survey the state-of-art in cloud vulnerability assessment.

In the next Section, we consider the works that are similar to ours. In Section III, we briefly investigate the current security projects in OpenStack and highlight the challenges in ensuring effective vulnerability assessments. Next, in Section IV we discuss the state-of-the-art in cloud vulnerability assessment by considering current strategies deployed by leading CSPs and third-party SecaaS vendors. Drawing from the aforementioned, we introduce our VAaaS approach in Section V. In Section VI, we evaluate our work and highlight our next steps in Section VII. We conclude the paper in Section VIII.

II. Related Work

We identified two research areas that are related to our work, research in the area of cloud vulnerability assessment and risk management, and research focusing on evolving SecaaS frameworks.

The challenges of risk management and vulnerability assessments in the cloud were investigated in [8]. The authors provided useful research directions for overcoming these challenges, however there is no practical implementation validating their recommendations. We implemented some of these recommendations in our work. Ristov et al [9] conducted a security assessment of the OpenStack cloud from within the cloud and from outside the cloud environment. In their work, they evaluated the vulnerabilities that exist in OpenStack cloud tenants. However, the security of the core OpenStack services and system vulnerabilities was not covered. In this work we focus on vulnerabilities that affect the core OpenStack services. Kamongi et al [10] introduced an approach for cloud vulnerability assessment that employs security ontology knowledge-bases for risk and threat mitigation. The same authors extend their work in the NEMESIS framework [11] however, we opine that their approaches might not be effective in discovering "OpenStack-native" vulnerabilities since they rely on vulnerability scanners that do not adequately include vulnerability information regarding OpenStack.

In [12], the authors introduced "Potassuim", a framework that leverages OpenStack services and APIs for penetration testing. "Project Mirroring" approaches are used to capture the live state of running cloud environments. Third party security tools like the MetaSpoit framework are integrated into "Potassium". The efficiency of this system would be enhanced with OpenStack-native security tools like the one presented in this paper. Almorsy et al's [13] work is closest to ours, formal approaches are employed for specifying vulnerability signatures in web applications. This work differs from ours in that we target system-level vulnerabilities based on disclosed vulnerabilities and best practices. We also apply our framework to a specific cloud environment and demonstrate its feasibility.

On a more general note, the major difference between our work and existing ones is our focus on integrating vulnerability assessment into OpenStack . We evolve novel ways specifically targeted at "OpenStack-native" vulnerabilities to enhance the efficiency of security assessments. This is inline with Grobauer assertion [14] on the need to differentiate between general vulnerabilities and cloud-specific vulnerabilities.

III. State of Security in OpenStack: Challenges and Prospects

OpenStack is an open-source cloud computing software [15] developed and maintained by a large community of developers. In this section, we briefly describe OpenStack's architecture, the current security structure and the challenges to secure deployments.

A. OpenStack Mitaka Architecture

OpenStack Mitaka is the latest release of OpenStack, several new features were introduced with Mitaka but those directly impacting on security include time-based one time password, implied roles and unified identity for multiple authentication sources. OpenStack has a modular structure consisting of several services, the main ones are: swift (object storage), keystone (identity service), horizon (user interface), nova (compute), cinder (Virtual Machine (VM) storage) and glance (VM catalog service).

B. Security and Vulnerability Management in OpenStack

The OpenStack security project consists of an OpenStack Security Team and a Vulnerability Management Team (VMT). The security team's responsibilities include production of security notes and developer guidance documents [16].

The OpenStack VMT is responsible for effective handling of vulnerabilities affecting OpenStack. The VMT has adopted coordinated vulnerability disclosure approach, a middle-ground between full disclosure and non-disclosure approaches. Hence, information about newly discovered vulnerabilities is initially restricted to a small number of OpenStack security developers. Following, a specific vulnerability management procedure [17], the downstream stakeholders are informed through secure channels. This step is critical since OpenStack is an upstream project deployed by several downstream vendors which could be adversely affected by inappropriate disclosure of vulnerabilities. Accordingly, these stakeholders are forewarned of vulnerabilities to ensure smooth patch development and deployment.

C. OpenStack Security Projects

In order to understand the security posture of OpenStack, an insight into existing security projects [16] maintained by

Table I: OpenStack Vendors and Deployment Strategies

Vendor	Supported Host OS	Automation/Orchestration Tool	Security Schemes
OpenStack	Ubuntu, Fedora, RedHat and CentOS	DevStack, Manual installation from OpenStack repository	OpenStack Security Advisories (OSSA), OpenStack Security Notes (OSSN) and OpenStack Security Guide
Mirantis	Ubuntu, Fedora, OpenSuse, MacOS, Windows (with Cygwin)	Fuel and Puppet	Fuel plugins for security monitoring, Firewall-as-a-Service (FWaaS), Bug tracking/fixing
Oracle	Oracle Linux	Docker-based and Ansible	None
RedHat	RHEL	OSP Director (Tripleo), PackStack (RDO)	Monitoring with Nagios ,Bug tracking/-fixing
Suse	Suse Linux Enterprise Server	Crowbar, Chef	None
Ubuntu	Ubuntu Linux	Landscape, MAAS and Juju	Bug tracking/ fixing

the Openstack Security team is imperative. These are: OpenStack Security Notes (OSSN), OpenStack Security Advisories (OSSA), Anchor, and Bandit. OSSN and OSSA are basically security information initiatives aimed at retaining the current information about vulnerabilities. OSSN contains information on third party applications and security best practices such as configurations. On the other hand OSSA like other security advisories, are documents published to disclose information on discovered vulnerabilities. Anchor is a lightweight facility for managing cryptographic certificates, while Bandit is a code linter for Python. The above mentioned security projects are more suited for core OpenStack developers, they do not fit the security needs of normal users or cloud administrators. These projects are neither automated nor modelled after Anything-as-a-Service (XaaS) as requirred for cloud environments. If designed according to the XaaS model such as other OpenStack services e.g. MONaaS [18] and FWaaS, the benefits of these tools could have wider coverage.

D. Lapses and Complexity in Vulnerability Management

Despite its popularity, OpenStack is still considered as a complex software. Firstly, OpenStack has a modular structure consisting of over 15 services. Secondly, almost every Linux distribution supports and maintains distribution-specific OpenStack packages. Thirdly, OpenStack's major releases are unveiled every six months with several changes in the cloud stack including performance improvements and security fixes [19]. While it might be a best practice to upgrade deployments inline with major releases, the upgrade procedure is majorly manual since there is no provision for automated upgrading. This creates opportunities for risks owing to human errors and mis-configurations [4]. Hence, in reality productive deployments hardly maintain the current stable releases. In order to tackle these complexities, vendors employ various automated orchestration strategies as shown in Table 1. Automation is a key feature in cloud environments, however the security implications of adopting these tools ought to be evaluated. For example, automated tools could expose cloud deployments to risks such as changes to roles and privileges and modification of critical files[20]. OpenStack's multi-vendor approach requires a well coordinated vulnerability management strategy, one that properly handles vulnerabilities discovery, vulnerability disclosure and patch release across upstream and downstream vendors. This is not available in the current vulnerability management strategy, for example there is no centralized source of vulnerability information for OpenStack clouds where vulnerabilities discovered by the various vendors is documented.

IV. State of the Art Approaches in Cloud Vulnerability Assessment

Security in the public cloud is a shared responsibility [21], CSPs ensure security at the infrastructure level while cloud customers are responsible for security of their data, applications, OSs and networks. However, cloud customers are limited in the types of security tools they can employ in fulfilling their own part of this responsibility. Owing to multi-tenancy, most CSPs enforce a requirement on customers for prior permission before conducting vulnerability assessments and penetration testing. Alternatively, an evolving approach is the provision of SecaaS by CSPs. Nevertheless, customers are not limited to these services, employment of third-party SecaaS is also permissible. There are currently several flavors of this service especially in the area of web application scanners. SecaaS aims at leveraging on cloud characteristics to empower cloud users to satisfy security requirements at a low cost in terms of finances and human resources. The efficiency of the current SecaaS services is yet to be properly scrutinized, most of the vendors migrated their security approaches to the cloud from traditional computing environments with little or no consideration of the peculiar nature of cloud environments. In the following subsections, we consider some of these vendors and briefly describe their services. We do not aim at comparing the efficiency of these security services but to highlight the current state-of-the art in cloud vulnerability assessment.

1) Amazon Inspector: Amazon Inspector [22] is an automated security assessment service provided on the AWS cloud platform. Just like other services available on AWS cloud, it is provided as a service, following the SecaaS model. Amazon Inspector leverages on a knowledge-base of rules that are mapped to common security issues and best practices. These rules are regularly updated by the AWS security team and include support for popular security metrics like the Common Vulnerability Scoring System (CVSS). Using these rules, users can launch and deploy agents on target instances to scan and identify security lapses in applications, databases and

other resources. Amazon Inspector is accessible via AWS APIs, SDKs, command line tools and AWS management console. The core of Amazon Inspector are the rules which conduct the actual scanning. Each rule is assigned a security level to aid in assessing the severity of a security issue; high, medium, low and informational. These rules are available in four categories;

- Common Vulnerabilities and Exposures
- Centre for Internet Security (CIS) Operating System Security Configuration Benchmarks
- Security Best Practices
- Runtime Behaviour Analysis

2) Google Cloud Security Scanner: Google Cloud Security Scanner [1] is a Web Application Scanner available on the Google Cloud platform. It currently supports only applications hosted on Google App Engine, application developers especially can use it as a service to secure their applications. It is still in beta as at the time of this writing and covers a small variety of web application vulnerabilities such as cross-site scripting and mixed content. Users are therefore advised to compensate this scanner with other scanners to further secure their applications. The Google Security Scanner employs several chrome workers and Google Compute Engines instances to horizontally scale to the required scan load. Just like other web application scanners, the Google security scanner crawls applications, follows links and urls and employs the use of simulated user inputs and event handlers.

3) Third Party Vulnerability Assessment Vendors: There is a large number of third party Vulnerability Assessment Vendors. While some of these vendors evolved their services for traditional security services, some of them are relatively new. They offer various kinds of SecaaS opportunities including vulnerability assessments, monitoring and threat mediation. Some vendors have their images deployed on public clouds, and partner with the respective CSPs for tight cloud security integration. There are several advantages for using this services, for example customers may not be required to request for prior permission to perform vulnerability assessment against resources on AWS if such security operations are conducted via SecaaS partners like AlertLogic and Nessus [23]. Yet, assessments though these providers is limited to some resources. For example, Relational Database Service (RDS) instances cannot be scanned on AWS cloud platform. the other category of cloud vulnerability services offer their services such that they can be directed against resources deployed in public clouds. The key requirements for these vendors is that subscribers provide some form of credentials for "credentialed scans" for example they may create a non admin role on their cloud dedicated to vulnerability scanning. Scanning results can also be saved on a cloud-database such as AWS Simple Storage Service (S3) or exported to a customer preferred location. They are also able to scan across security groups and databases on a scheduled timetable or on a one-off setting.

[1] https://cloud.google.com/security-scanner/

V. Integrating a Vulnerability Assessment Framework into OpenStack

OpenStack does not currently offer an integral vulnerability assessment service. While it is possible for third party assessment tools to be used for vulnerability assessments and auditing, integrated tools offer cloud-native approaches and results. Third-party tools could be limited by CSPs from auditing multi-tenant resources such as hosted databases, the dynamic nature of cloud resources e.g. elastic ip addressing also presents a challenging situation. Such limitations reduce the level of control a CSPs ought to have. Hence, as discussed in Section III, the current approach in cloud vulnerability assessments consists in CSPs offering VAaaS on their platforms. These approaches to vulnerability assessments are designed to be flexible, cheap in cost and easy to use, just as other cloud services. The advantages are similar to those of other cloud services such as ease of use while hiding the complexities of security configurations behind the scenes. VAaaS is also applicable to enterprise private clouds where employees can effectively assess their resources without requiring the normal "technical knowledge". This reduces the burden from the security staff and promotes security in the enterprise. We opine that integrating a VAaaS into OpenStack is an important component for OpenStack security. The basic requirement for a vulnerability assessment tool suitable for integration into OpenStack would be that it is open-source and supports Linux distributions. We considered several open-source vulnerability assessment tools and eventually selected Open Vulnerability Assessment System (OpenVAS). OpenVAS is a fork of the popular Nessus vulnerability scanner. It consists of several security tools suited for vulnerability scanning and assessment, it has features that are commonly found in commercial products. OpenVAS is commonly used for security research. It is suitable for different levels of vulnerability-related tasks such as Local Security Checks and network scanning. We have integrated OpenVAS into OpenStack using our system called CAVAS. Most components of CAVAS are implemented in Java, more details on our implementation is provided in the next sections. Note that we have focused on vulnerabilities affecting the core OpenStack software and applications. We are not concerned with the vulnerabilities that affect the other applications such as web applications and databases. We feel that the existing vulnerability assessment tools are quite mature for the task of traditional applications.

A. Aggregating and Adapting Vulnerability Information for OpenStack

Vulnerability scanners heavily rely on information about existing or discovered vulnerabilities [24], such information is acquired from various sources including the National Vulnerability Database (NVD), Open Source Vulnerability Database (OSVDB) and SecurityFocus [2]. While information from these sources suffice for assessing traditional systems, information

[2] http://www.securityfocus.com/

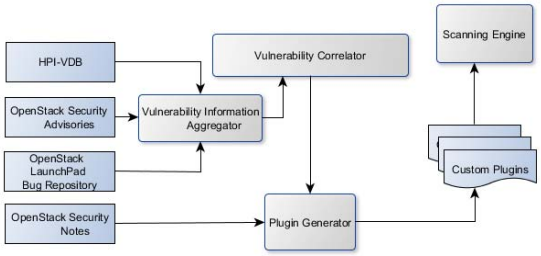

Figure 1: Architecture of CAVAS.

regarding cloud-specific vulnerabilities is requisite for assessing core OpenStack software components such as keystone and swift. In order to acquire these specific information, an approach that targets and captures this information is imperative. Hence, in order to satisfy these requirements, we adopt a two step approach:

1) Vulnerability Information from External Sources: In the first step, we gather information from outside OpenStack. We leverage on Hasso Plattner Institute Vulnerability Database (HPI-VDB) [3], a vulnerability database developed and maintained by the Hasso Plattner Institute (HPI) security research team. It retains over 75000 vulnerabilities extracted from various sources including the NVD, OSVDB and SecurityFocus.

2) Vulnerability Information from Internal Sources: The second step involves aggregation of information from sources within OpenStack, more specifically from the OSSN, OSSA and OpenStack Launchpad Bug-tracker (OLB). OSSN and OSSA are security initiatives maintained by OpenStack security team to keep OpenStack downstream stakeholders and users abreast with security information [25]. OSSN contains updated information about security best practices such as secure configurations. We also observed that several security issues are not mitigated but proposed for implementation in future releases. Also, approaches for mitigating these security issues are described in OSSN. However, the awareness of these information sources is sparse, and OpenStack provides no automated tools for easily consumption of these information. Moreso, the downstream stakeholders selectively use the information according to their requirements i.e. only when it concerns their product. Similar to the security notes, OSSA contains detailed information about vulnerabilities discovered. However, the information at OSSA covers core OpenStack services. The last source of our vulnerability information is the OLB, which is retained on LaunchPad. We include this as an internal information source because it is heavily used for development-related communication within OpenStack development teams and OpenStack vendors. Similar to the previously mentioned sources, the awareness of the information available at OLB is limited, infact most people aware of this source are software developers. However, information derived from OLB is very useful and could be used to improve security. We extend the functionality of our previously

[3]https://hpi-vdb.de/vulndb/

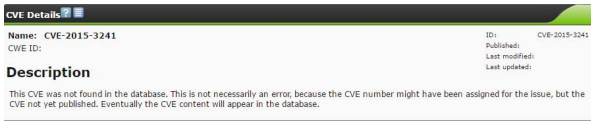

Figure 2: Screenshot of Search Result in OpenVAS Plugins Database Showing CVE-2015-3241 not found.

published paper [7] which detailed a framework for leveraging information from OLB. The framework demonstrated the use of information derived from OLB for reducing the "window of opportunity" for exploiting security holes in software. Accordingly, we have integrated security-related information extracted from OLB into CAVAS.

B. Components of the Our Scheme - Cloud Aware Vulnerability Assessment System (CAVAS)

1) Vulnerability Information Aggregator: The architecture of CAVAS is shown in Figure 1. As shown in the architecture, the Vulnerability Information Aggregator is responsible for collecting information from the sources mentioned in the previous sub-section. We collect information from sources internal and external to OpenStack as described in the previous sub-section. Different approaches are adopted to achieve this objective, depending on the specific source. For example, information from HPI-VDB is collected via a REST interface, while the OSSA git repository is cloned from OpenStack GitHub repository [4] and thereafter parsed into our local MongoDB database. We prefer MongoDB since it supports schema-less database structure, which is suitable for storing vulnerability information [26].

2) Vulnerability Information Processor: The data retrieved from the various sources contains several pieces of information, however we are interested in specific content such as the vulnerability title, vulnerability description, Common Vulnerabilities and Exposures (CVE) identifier, CVSS base score, affected products and availability of a fix. We employ our "custom extraction algorithm" for extracting and collating these pieces of information. For each vulnerability entry, we aim at collating information that represents a comprehensive picture of the specific vulnerability suitable for plugin generation.

3) Vulnerability Correlator: During our previous work on vulnerability life-cycles [27], we realized a gap between when vulnerabilities are publicly announced, when fixes are released by respective vendors and when security tools especially vulnerability scanners develop appropriate plugins to identify these vulnerabilities. We characterized these findings as *Scanner Patch Time* and *Scanner Patch Discovery Time*. These issues are still evident in OpenStack for example, we developed a plugin for CVE-2015-3241 since there was none available in the OpenVAS plugin repository (Figure.2). These plugins are scripts that test target systems for specific vulnerability issues. The Vulnerability Correlator compares the

[4]https://github.com/openstack/ossa

information acquired in the last step with the plugins in the local OpenVAS plugin repository. This step is important to prevent duplicate plugins. Vulnerabilities lacking plugins are then queued for development in the next step.

4) Plugin Generator: The plugin generator automatically constructs the required plugins as determined in the previous step. Vulnerability signature generation has been used in previous security research such as Intrusion Detection Systems (IDS) [28] and anti-virus systems, we gain inspiration from these efforts but apply similar approaches to vulnerability assessment. The plugins are developed in Nessus Attack Scripting Language (NASL), a scripting language used for OpenVAS and Nessus Scanner. Automatic development of these plugins minimizes human intervention and aids timely production of plugins in response to discovered vulnerabilities. We utilize the templating capabilities of Apache Freemarker Templating framework [5] to generate two types of plugins: Local Security Checks and Policy Checks. The Local Security Checks are constructed using information derived from HPI-VDB, OSSA and OLB. These checks test target systems for specified vulnerabilities already discovered and published. On the other hand, Policy Checks are derived based on information extracted from OSSN. These checks may not necessarily search for existing vulnerabilities, rather they verify if best practices recommended by the OpenStack security team are being applied on the target system. These policies also check for configuration errors.

5) Scanner Module: The scanner module is the hub of our system. We leverage on OpenVAS scanner for our framework rather than designing and developing a scanner from the scratch. OpenVAS has a modular structure consisting of OpenVAS scanner, OpenVAS manager, a Command Line Interface (CLI) and a web client. Hence it is feasible to extend the existing functions of any of these modules. Through the use of the the XML-based OpenVAS Management Protocol (OMP), CAVAS executes scan commands to the scanner. OMP supports automation of batch processes which, we have leveraged on this feature in our work.

VI. Evaluation

We have conducted a set of experiments to evaluate the suitability of our prototype implementation. Here, we describe these experiments and provide the results we obtained. Experiments were conducted against a target OpenStack cloud environment. Since the focus of these initial steps is to validate our approach from a cloud administrator perspective, we deployed a basic OpenStack cloud environment as the target. The target environment is an OpenStack Icehouse environment installed on Windows Laptop with the following configuration : Intel i5, dual core CPU and 12GB memory.

We conduct two category of tests, in the first category we aim at identifying vulnerabilities while the second tests are designed to spot mis-configurations based on our security policy checks. Our security policy checks are aimed at identifying

[5]http://freemarker.org/

Vulnerability CVE	CVSS	CA	OpenVAS	Affected Service	CVE
Ubuntu update for gnutls26 USN-2913-4	10.0	X	✓	libgnutls	
Ubuntu update for openssl USN-2913-3	10.0	X	✓	OpenSSL	
Ubuntu update for ca-certificates USN-2913-4	10.0	X	✓	CA-Certificates	
Ubuntu Update for PAM USN-2914-1	10.0	X	✓	OpenSSL	CVE-2016-0702, CVE-2016-0705,
Ubuntu Update for libtasn USN-2957-1	10.0	X	✓	libtasn	CVE-2016-4008
Ubuntu Update for perl USN-2916-1	7.5	X	✓		CVE-2013-7422, CVE-2016-2381
Ubuntu Update for qemu USN-2182-1	7.2	X	✓	qemu-system-x86	CVE-2013-4544, CVE-2014-2894
Resize/delete combo allows to overload nova-compute	6.8	✓	X	OpenStack Nova	CVE-2015-3241
Nova may fail to delete images in resize table	6.8	✓	X	OpenStack Nova	CVE-2015-3280
Glance storage quota bypass when token is expired	6.8	✓	X		CVE-2015-5286
Glance v2 API unrestricted path traversal through filesystem	6.5	✓	X	OpenStack Glance	CVE-2015-1195
PKI Token Revocation Bypass	6.0	✓	X	OpenStack Keystone	CVE-2015-7546
Ubuntu Update for PAM USN-2935-2	5.8	X	✓	libpam-modules	CVE-2015-3238, CVE-2013-7041, CVE-2014-2583
Ubuntu Update for PAM USN-2935-2	5.8	X	✓	libpam-modules	CVE-2015-3238, CVE-2013-7041, CVE-2014-2583
Image status can be changed by passing header 'x-image-meta-status'	5.5	✓	X	OpenStack Glance	CVE-2015-5251
Nova console Cross-Site WebSocket hijacking	5.1	✓	X	OpenStack Nova	CVE-2015-0259
All PUT tempurls leal existence via DLO manifest attack	5.1	✓	X	OpenStack swift	Cve-2015-016
S3 Token incorrect condition expression for SSL insecure	4.3	✓	X	OpenStack Keystone	CVE-2015-1852
Help_text parameter of fields is vulnerable to arbitrary html injection	2.9	✓	X	OpenStack Horizon	CVE-2015-3219
Ubuntu Update for cpio USN-2906-1	4.3	✓	X	Cpio	CVE-2015-1197, CVE-2015-2037
Cinder does not properly track the file format	4.0	✓	X	OpenStack Cinder	CVE-2014-7230, CVE-2014-3641
Format guessing and file disclosure in image convert	reserve	✓	X	OpenStack Cinder	CVE-2015-1850
XSS in Horizon Heat stack creation	4.3	✓	X	OpenStack Horizon	CVE-2015-3219
Another Horizon login page vulnerability to a DoS attack	7.8	✓	X	OpenStack Horizon	CVE-2015-5143
Image data remains in backend after deleting the image created using task api	4.0	✓	X	OpenStack Glance	CVE-2015-1881
Backend_argument containing password leaked in logs	4.0	✓	X	OpenStack Keystone	CVE-2015-3646
Adding 0.0.0.0/0 allowed address pairs breaks L2 agent	4.0	✓	X	OpenStack Neutron	CVE-2015-3221
Image data stays in store if image is deleted after creating image using import task	4.0	✓	X	OpenStack Glance	CVE-2015-3289
Format-guessing and file disclosure via image conversion	3.5	✓	X	OpenStack Glance	CVE-2015-5163
IP, MAC and DHCP spoofing rules can be bypassed by changing device_owner	3.5	✓	X	OpenStack-Neutron	CVE-2015-5240

Figure 3: Comparison table showing vulnerabilities discovered by CAVAS and OpenVAS.

mis-configurations in the deployed targets using information obtained from OSSN. We then conduct credentialed scans against the target cloud using CAVAS. We repeat the same experiments against "vanilla" OpenVAS and the community edition of Nessus scanner. As shown in Figure 3, the number of vulnerabilities discovered via our approach out-numbers those identified with OpenVAS. While CAVAS identifies 23 vulnerabilities, OpenVAS identifies 9 vulnerabilities. Note also that the vulnerabilities identified by OpenVAS are not specific to OpenStack, they are generic Linux vulnerabilities affecting third-party applications e.g. OpenSSL. We do not include the results of Nessus since the scanner only identified host and enumerated running services without identifying vulnerabilities.

We note however that most of the vulnerabilities discovered via with "vanilla" OpenVAS have higher CVSS scores. We also observe that the other vulnerability scanners do not have security policies for scanning OpenStack resources such as configuration files and Database-as-a-Service (DBaaS). Also, security best practices are not implemented in these scanners as done in CAVAS. We observed that several vulnerabilities do not have appropriate plugins in OpenVAS, for example CVE-2015-3241 (Figure.2), but CAVAS identifies this vulnerability (Listing 1) since we implemented appropriate plugin. Our approach is therefore suitable for deployment in a VAaaS where customers with minimum security expertise can employ

it in securing their cloud environments.

```
<get_results_response status="200" status_text="OK
    ">
 <result id="6a4b7fe4-1893-4dfd-9303-3d1e1b18770e"
     >
  <name>OSSA_2015_015</name>
  <creation_time>2016-06-05T08:47:38+02:00</
      creation_time>
  <modification_time>2016-06-05T08:47:38+02:00</
      modification_time>
  <report id="a498a888-6923-448a-8c31-3585ce59b635
      "/>
  <task id="a8f22975-bda3-43fa-9a6f-0bfd0cb4566a">
   <name>Icehouse CAVAS</name>
  </task>
  <host>192.168.77.128</host>
  <port>general/tcp</port>
  <nvt oid="1.3.6.1.4.1.25623.1.0.299938">
   <name>OSSA_2015_015</name>
   <family>OpenStack Security Checks</family>
   <cvss_base>6.8</cvss_base>
   <cve>CVE-2015-3241</cve>
   <bid>NOBID</bid>
   <xref>URL:https://launchpad.net/bugs/1387543</
       xref>
   <tags>check_type=authenticated package
test|cvss_base_vector=AV:N/AC:L/Au:S/C:N/I:N/A:C|
    summary=Check the version of
Nova|vuldetect=Get the installed version with the
    help of detect NVT and check
if the version is vulnerable or not.|affected=
    OpenStack Nova on UBUNTU14.04
LTS|qod_type=package|solution_type=VendorFix</tags
    >
  </nvt>
  <threat>Medium</threat>
  <severity>6.8</severity>
  <description>Package python-novaclient version
      2.17.0-0ubuntu1.2 is installed
which is known to be vulnerable.</description>
 </result>
</get_results_response>
```

Listing 1: Result of Plugin Identifying CVE-2015-3241 Vulnerability.

VII. Future Work

We have applied our framework from the perspective of a cloud administrator. It could be interesting to consider the cloud customer use case. We envisage that in such a scenario, other factors such as scalability and elasticity of the VAaaS would be an important factor, as well as the dynamic nature of cloud instances and resources. Similarly, it is important to extend our approach to include advanced web application scanning capabilities such as fuzzing. This is a useful requirement considering that cloud services are generally accessed through the web interface. Also, integration of other OpenStack services such as swift for storing scanning data such tasks, reports and targets is interesting. Similarly, a useful feature could be integration of threat intelligence frameworks for fast and effective sharing of vulnerability information across OpenStack cloud deployments. This feature has been recommended by CSA [29] and relevant frameworks can be utilized e.g. Security Content Automation Protocol (SCAP), Trusted Automated Exchange of Indicator Information (TAXII) and Cyber Observable Expression (CybOX). Security monitoring and information analytics is useful component for secure environments. OpenStack Telemetry is a project aimed at reliably collecting information on the utilization of physical and virtual resources in an OpenStack cloud. The project is geared towards offering MONaaS in OpenStack. It consists of OpenStack services e.g. Ceilometer, Aodh and Gnochi. An investigation into effective approaches for deploying VAaaS alongside MONaaS is an open research question.

VIII. Conclusion

OpenStack is a popular open-source IaaS cloud computing framework that offers almost every cloud package available on AWS as well as other major CSPs. It is officially supported by over 100 companies however there are few production-ready deployments. This low deployment trend is attributed to security issues, amongst other factors. Several commendable efforts have been made to improve security in recent OpenStack releases. But, these efforts are more beneficial to developers than cloud administrators and normal users. However, a core characteristic of the cloud is self service. One of the major security requirements for cloud services is vulnerability assessment which are essential aspects of any audit and regulatory compliance requirements. Such a service is not available in OpenStack. Accordingly, in this work we have implemented CAVAS, a prototype that provides first steps to integration of VAaaS in OpenStack. We focus on identifying vulnerabilities that are specific to OpenStack core services and software and not those affecting other third party software. We leverage on information provided from public vulnerability information resources such as NVD and OSVDB. Our approach differs from existing solutions in that we also include information published by the OpenStack Security group; OSSA and OSSN. Accordingly, we are able to automate vulnerability assessment and dramatically improve scanning accuracy. We envisage that our approach could be useful for security assessments of OpenStack including vulnerability assessments and security auditing for regulatory compliance.

References

[1] RightScale, "State of the cloud report", *State of the Cloud Report*, 2016. [Online]. Available: http://www.rightscale.com/lp/2016-state-of-the-cloud-report?campaign=701700000015euW.

[2] OpenStack Foundation, *Openstack cloud software*. [Online]. Available: http://www.openstack.org/.

[3] VMWare. [Online]. Available: https://www.vmware.com/cloud-computing/private-cloud.

[4] Talligent, *State of openstack adoption report - industry survey results 2016*, Report, 2016.

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

[6] CSA, *Secaas implementation guidance - category 10 network security*, Cloud Security Alliance, 2012.

[7] Kennedy A Torkura, Feng Cheng, and Christoph Meinel, "A proposed framework for proactive vulnerability assessments in cloud deployments", in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, IEEE, 2015, pp. 51–57.

[8] Burton S Kaliski Jr and Wayne Pauley, "Toward risk assessment as a service in cloud environments.", in *HotCloud*, 2010.

[9] Sasko Ristov, Marjan Gusev, and Aleksandar Donevski, "Openstack cloud security vulnerabilities from inside and outside", *CLOUD COMPUTING*, pp. 101–107, 2013.

[10] P. Kamongi, S. Kotikela, K. Kavi, M. Gomathisankaran, and A. Singhal, "Vulcan: Vulnerability assessment framework for cloud computing", in *Software Security and Reliability (SERE), 2013 IEEE 7th International Conference on*, Jun. 2013, pp. 218–226. DOI: http://dx.doi.org/10.1109/SERE.2013.3110.1109/SERE.2013.31.

[11] Patrick Kamongi, Mahadevan Gomathisankaran, and Krishna Kavi, "Nemesis: Automated architecture for threat modeling and risk assessment for cloud computing", 2015.

[12] Richard Li, Dallin Abendroth, Xing Lin, Yuankai Guo, Hyun-Wook Baek, Eric Eide, Robert Ricci, and Jacobus Van der Merwe, "Potassium: Penetration testing as a service", in *Proceedings of the Sixth ACM Symposium on Cloud Computing*, ACM, 2015, pp. 30–42.

[13] Mohamed Almorsy, John Grundy, and Amani S Ibrahim, "Vam-aas: Online cloud services security vulnerability analysis and mitigation-as-a-service", in *Web Information Systems Engineering-WISE 2012*, Springer, 2012, pp. 411–425.

[14] Bernd Grobauer, Tobias Walloschek, and Elmar Stocker, "Understanding cloud computing vulnerabilities", *IEEE Security & Privacy*, vol. 9, no. 2, pp. 50–57, 2011, ISSN: 1540-7993. DOI: http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/MSP.2010.115http://doi.ieeecomputersociety.org/10.1109/MSP.2010.115.

[15] Openstack Foundation. (Aug. 15, 2016). Openstack cloud software. OpenStack Cloud Software, [Online]. Available: http://www.openstack.org/ (visited on 08/16/2016).

[16] OpenStack, *Openstack security*, Online, Openstack Security. [Online]. Available: https://wiki.openstack.org/wiki/Security.

[17] OpenStack Foundation, *Vulnerability management process*, online. DOI: http://dx.doi.org/10.1016/s1874-5970(05)80009-810.1016/s1874-5970(05)80009-8. [Online]. Available: https://security.openstack.org/vmt-process.html.

[18] Openstack Foundation, *Openstack monitoring*. [Online]. Available: https://wiki.openstack.org/wiki/MONaaS.

[19] OpenStack Foundation, *Openstack releases*, Online. [Online]. Available: http://releases.openstack.org/.

[20] Dave Shackleford, "Virtualization and cloud: Orchestration, automation, and security gaps", [Online]. Available: https://www.rsaconference.com/writable/presentations/file_upload/csv-r02-virtualization-and-cloud-orchestration-automation-and-security_gaps_v2.pdf.

[21] Mohamed Almorsy, John Grundy, Ingo Müller, *et al.*, "An analysis of the cloud computing security problem", in *Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, 30th Nov*, 2010.

[22] *Amazon inspector user guide*. [Online]. Available: http://docs.aws.amazon.com/inspector/latest/userguide/inspector-ug.pdf.

[23] "Amazon web services: Overview of security processes", *Amazon Whitepaper*, Nov. 2014. [Online]. Available: https://media.amazonwebservices.com/pdf/AWS_Security_Whitepaper.pdf.

[24] Eric Whyne Seymour Bosworth Michel E. Kabay, *Computer Security Handbook*, Sixth. John Wiley & Sons, 2014.

[25] OpenStack, *OpenStack Security Guide*, OpenStack, Ed. OpenStack Foundation, 2016.

[26] A. Nakamura, "Towards unified vulnerability assessment with open data", in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*, Jul. 2013, pp. 248–253. DOI: http://dx.doi.org/10.1109/COMPSACW.2013.3410.1109/COMPSACW.2013.34.

[27] Kennedy A Torkura, Feng Cheng, and Christoph Meinel, "Application of quantitative security metrics in cloud computing", in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, IEEE, 2015, pp. 256–262.

[28] Weidong Cui, M. Peinado, H.J. Wang, and M.E. Locasto, "Shieldgen: Automatic data patch generation for unknown vulnerabilities with informed probing", in *Security and Privacy, 2007. SP '07. IEEE Symposium on*, May 2007, pp. 252–266. DOI: http://dx.doi.org/10.1109/SP.2007.3410.1109/SP.2007.34.

[29] CSA, *Secaas implementation guidance category 5 : Security assessments*, Cloud Security Alliance, 2012. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_5_Security_Assessments_Implementation_Guidance.pdf.