Global Journal of Information Technology

Volume 05, Issue 2, (2015) 87-94

http://sproc.org/ojs/index.php/gjit

Penetration testing on cloud---case study with owncloud

Wenjuan Xu *, Frostburg State University, 101 Braddock Rd., Frostburg, MD 21532, USA.
Brian Groves, Frostburg State University, 101 Braddock Rd., Frostburg, MD 21532, USA.
Willson Kwok, Allegany County of Community College, USA.

Suggested Citation:
Xu, W., Groves, B. & Kwok, W. (2015). Penetration testing on cloud---case study with owncloud. *Global Journal of Information Technology*. *5*(2), 87-94. doi: http://dx.doi.org/10.18844/gjit.v5i2.198

Received 25 July, 2014; revised 12 August, 2015; accepted 23 September, 2015.
Selection and peer review under responsibility of Prof. Dr. Adem Karahoca, Bahcesehir University, Turkey

Abstract

The cloud computing techniques bring different security challenges. In this paper, we set up ownCloud as the example cloud computing infrastructure. Then we present our work process and results of a series of penetration testing performed on the ownCloud. We also analyse these results and give key recommendations for addressing the identified vulnerabilities.

Keywords: cloud computing, security, penetration tesing, owncloud

*ADDRESS FOR CORRESPONDENCE: **Wenjuan Xu,** Frostburg State University, 101 Braddock Rd., Frostburg, MD 21532, USA. *E-mail address*: wxu@frostburg.edu

1. Introduction

Cloud computing [1] involves the utilization of a real time communication mediums (e.g. the internet) to connect large numbers of computing devices. Cloud computing includes hardware, networks, storage, services, and interfaces that can be combined to deliver computing as a service. Cloud services involve delivering infrastructure, software, and storage available over the internet based on user demand. Characteristics of modern cloud computing include application programming interfaces (APIs) [2], scalability, and pay-as-you-go models for billing and metering of service. Due to its flexibility, cloud computing technologies have become powerful tools within the technology industry. As such, it is an essential component of modern-day ecommerce.

With an increasingly large number of users and organizations relying on cloud computing technologies for many essential functions, the security of cloud services and environments has the topic of lively discussion. Average users may rely on cloud storage services such as Dropbox [3], Google Drive [4], and content creation apps such as Google Docs [5] to host and share their personal documents. Businesses have increasingly turned to cloud computing technologies to automate essential functions such as sales, human resource management, file sharing, storage, email, and collaboration. With so much data being stored in “clouds” and within cloud services hosted and maintained at offsite locations, cloud computing faces many threats in regards to data security and system vulnerability.

To address these issues, it is necessary to test these security threats and check what the possible risks that these threats may bring. All these testing can provide the foundation for designing and implementing a secure cloud computing infrastructure. In this paper, we use the popular techniques in the penetration testing for the cloud computing infrastructure. We use the own Cloud [6] as the example and apply different hacking technologies on that. Then, we analyse the effects of the hacking to see whether it is the problem of cloud computing server or the problem of cloud itself. Finally, we try to address these security problems.

The paper is organized as follows. We introduce the background knowledge about own Cloud in Section 2. Section 3 explains how we perform the different hacking technologies on the own Cloud that we built, what kind of results we get, and the analysis of these results. Finally, we summarize our work and the future in Section 4.

2. Background

Created by Frank Karlitschek in 2010, ownCloud is an open-source web application designed to allow users to create cloud- based file sharing and data synchronization services. Designed using PHP and JavaScript scripting languages, ownCloud utilizes popular database management systems such as MySQL, SQLite, MariaDB, Oracle Database, and PostgreSQL. Change management is accomplished through the use of SabreDAV, an open source WebDAV server (Web Distributed Authoring and Versioning) designed to facilitate collaboration between user content and web servers. OwnCloud client and server software is available for computers utilizing Windows, Linux, and Macintosh operating systems, with mobile apps available for Android and iOS devices. OwnCloud enjoys full integration with the popular Linux-based Gnome desktop, and has been accepted into the Debian GNU Linux repository. Through a powerful API, ownCloud acts as an extendable platform for applications and plugins. It has features including file storage per conventional directory structures, client synchronization, a task scheduler, music streaming, sharing of content across groups or public URLs, Bookmarking, a URL shortening Suite, a photo gallery, a viewer for ODF type files, cryptography, a calendar, an address book, user and group administration, and an online text editor with syntax highlighting and code folding and a PDF viewer.

3. Penetration testing of ownCloud

3.1. Testing environment

Our testing environment consists of a local network and host computer containing the following components:

- Operating System: Windows 7 Ultimate 64-bit SP1
- CPU: Intel Core i5 2500K @ 4.0GHz (overclock)
- RAM: 16.0GB Dual-Channel DDR3 @ 798MHz (11-11-11-28)
- Motherboard: ASRock Z68 Extreme3 Gen3
- Graphics: 2048MB ATI AMD Radeon HD 7800 Series (Sapphire)
- Hard Drives: 60GB M4-CT064M4SSD2 and 932GB Western Digital WDC WD10EADS-00L5B1 (SATA)
- Firewall: Enabled, Antivirus: Microsoft Security Essentials
- Web Browser: Mozilla Firefox: Funnelcake, 14-1.0
- Network Controller: Realtek PCIe GBE Family Controller
- IP Address: 192.168.1.10, Subnet mask: 255.255.255.0, Gateway server: 192.168.1.1

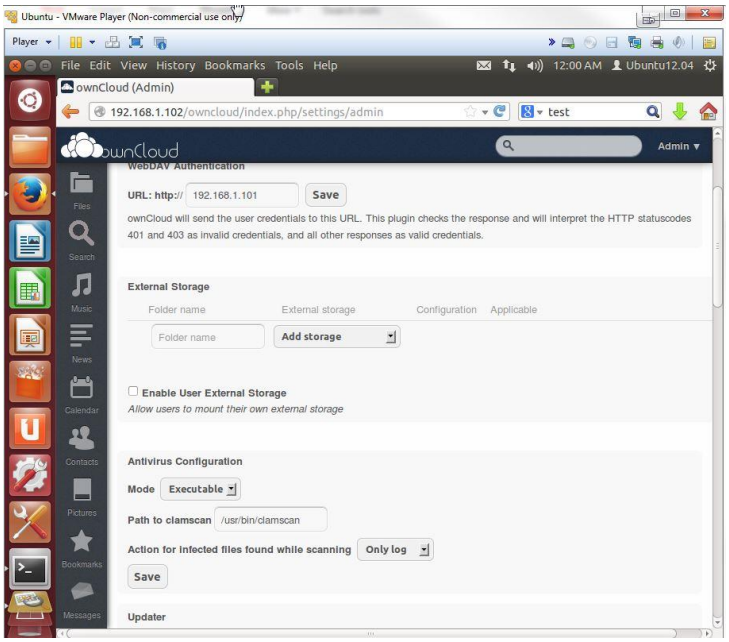


Fig. 1. Installed ownCloud interface

We create three virtual machines running on our host PC. Each virtual machine utilizes a bridged network adapter for communication with the host PC and network, as well as the machines themselves. Utilizing the hardware of the host PC for functionality, the virtual machines consist of an Ubuntu-based Linux server (for hosting cloud service), a Backtrack-based Linux machine (for hacking), and a Windows 7 machine (for additional testing). They are comprised of the following basic components.

- Virtual Machine 1:
- Operating System: Ubuntu 12.04.3 (ownCloud cloud service)
- RAM: 4 GB, Hard Drives: 75 GB (SCSI)
- Notable Software: Ubuntu Server, LAMP server, phpmyadmin, OwnCloud server

software, IP Address: 192.168.1.102

- Virtual Machine 2:
- Operating System: Backtrack R3 (hacking)
- RAM: 1 GB, Hard Drives: 75 GB (SCSI), IP Address: 192.168.1.105
- Virtual Machine 3:
- Operating System: Windows 7(testing)
- RAM: 1 GB, Hard Drives: 60 GB (SCSI), IP Address: 192.168.1.104

OwnCloud server setup involves the creation of an Ubuntu 12.04.3 LTS (Precise Pangolin) virtual machine consisting of 4 GB of memory, and 75 GB of hard drive capacity. Ubuntu server has been installed on top of the existing Ubuntu setup to take advantage of the additional features required for ownCloud installation. Fig.1. shows the example ownCloud web interface with admin, and apps.

3.2. Hacking

In this section we attempt to perform different penetration testings on the build cloud service. We will attempt to carry out some basic experiments within our test environment designed to mimic common security threats which may affect users of similar internet-based cloud services. As shown in Fig.2, we see a representation of our local network housing our ownCloud server and client PCs acquired through Zenmap [7] (a GUI interface for NMAP, a popular open-source network scanner).

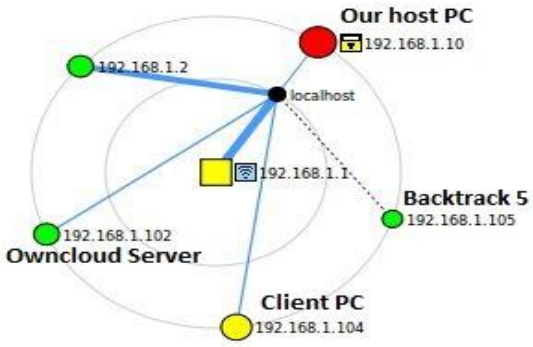


Fig. 2. Our network

3.2.1 The man in the middle attack

A man–in-the-middle attack is a form of computer eavesdropping in which an attacker attempts to intercept the communication between two computers or devices [8]. The attacker makes the victims believe that they are communicating with each other, when in fact the communication is being controlled by the hacker. The attack can be successful if an attacker is able to impersonate endpoints to the satisfaction of the other, and is thus considered a mutual authentication attack. In the steps below, we detail our attempts to intercept the contents of an image file being sent from our Windows 7 virtual machine to our Ubuntu ownCloud Server machine, with the attack being carried out via our Backtrack virtual machine.

We utilize the Ettercap and Driftnet tools on the Backtrack (the penetration testing machine) to monitor the traffic. Then we upload an image from Windows 7(client machine) to the

ownCloud Server. We observe the Driftnet windows and find that we are able to successfully intercept and capture the pictured image (Apollo statue) as it was uploaded from our Windows 7 machine web interface to our ownCloud server (as shown in Fig.3.).

3.2.2 SQL injection

SQL injection [9] is one of the most commonly used to attack data applications in order for an attacker to receive the contents of a database. It exploits security vulnerabilities in an application's software, such as incorrectly used escape characters in SQL statements, or when user input is unexpectedly executed. SQL injection is widely considered as one of the top web application vulnerabilities, with studies suggesting web applications receive several of such threats on a monthly basis.

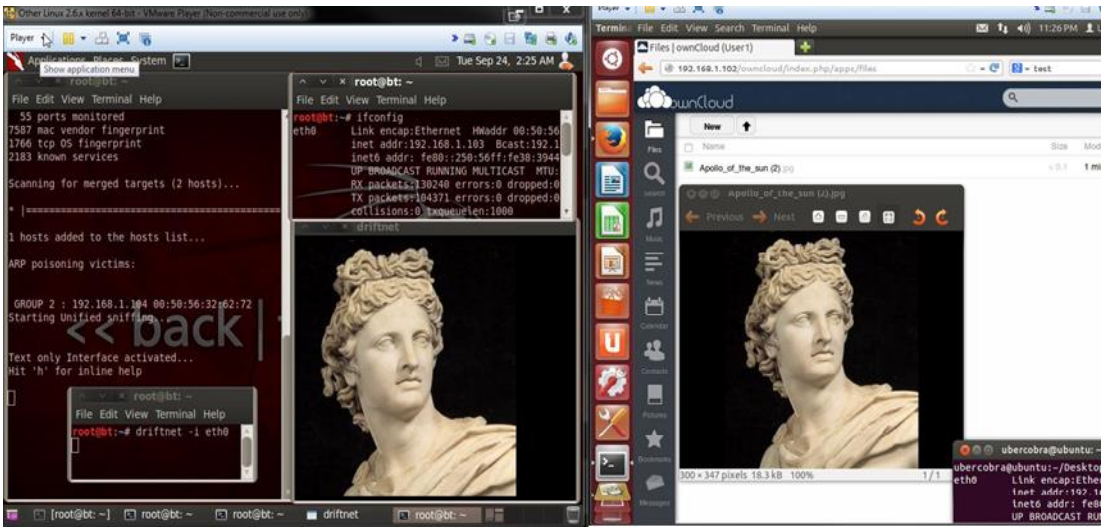


Fig. 3. Man in the middle attack on ownCloud

In the following example, we attempt to perform an SQL injection attack on our ownCloud server database utilizing Backtrack 5 R3, and Metasploit (an open-source penetration testing tool used for exploiting and developing code against target machines). We attempt to utilize msfconsole (Metasploit's user interface, designed to setup and run attacks) to gather information regarding our ownCloud server's MySQL database.

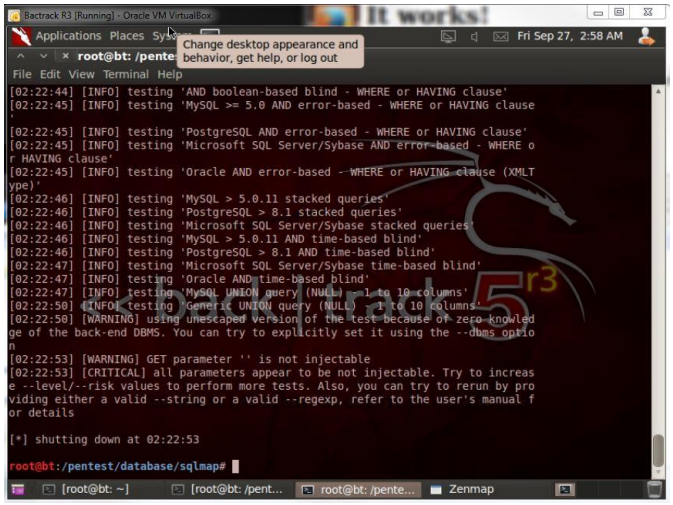


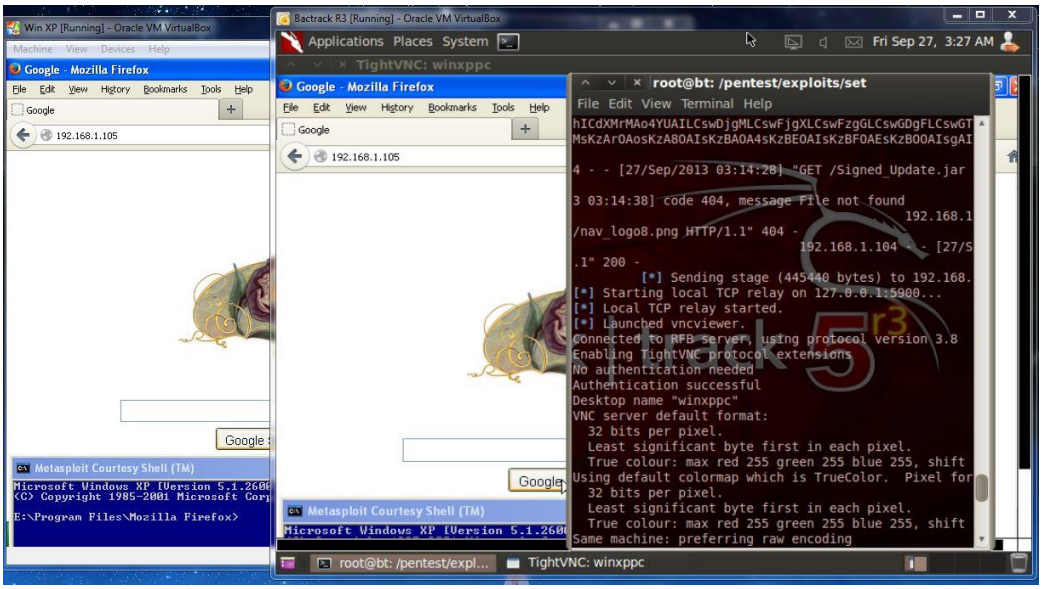
Fig. 4. SQL injection attack on ownCloud

After multiple attempts, we were unable to successfully perform SQL injection techniques to gain further information regarding the ownCloud database in question (as shown in Fig.4.). However, it should be noted that while this novice attempt proved unsuccessful, database servers remain vulnerable to hackers employing a variety of advanced techniques. If successful entry is gained to a services database server, hackers could potentially access database contents, exposing the information of users, and utilizing said information for a variety of nefarious purposes.

3.2.3 Client side ownCloud account attack

In this section, we carry out a series of attacks against a client computer on our local test network. The goal is to gain access to computer resources, then utilize the information in an attempt to gain access to our ownCloud web application administrator account.

In the following, we use the Social Engineering Toolkit (SET) to gain access to our client PC with VNC Viewer (a popular desktop sharing system). SET is designed to perform advanced attacks on computer systems, as has become a favourite tool of penetration testers and hobbyists alike who wish to test the boundaries of system vulnerability.



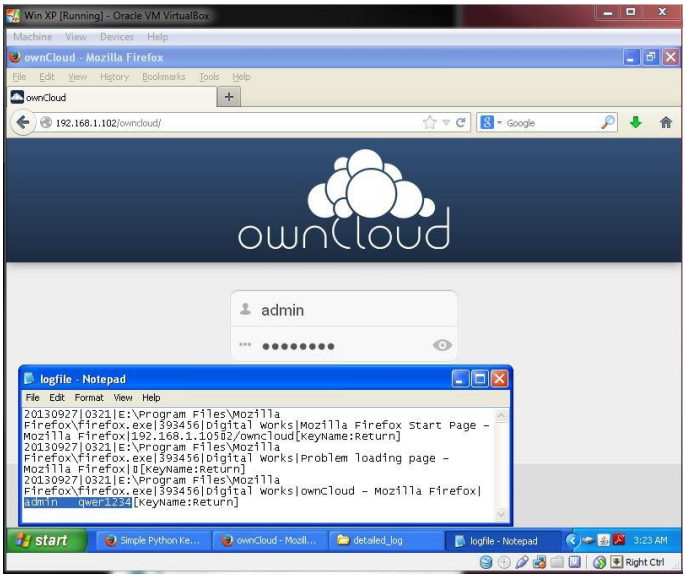


Fig. 5. Client side ownCloud account attack

To start the attack, we first start the SET on Backtrack (192.168.1.103), then we choose Social-Engineering Attacks→Website Attack Vectors->Java Applet Attack Method→Web Templates→Google→Windows Reverse_TCP VNC DLL→Backdoored Executable (BEST)→Terminal: Port of the listener [443]:4444/ .

A web server is then launched from our Backtrack machine which listens for connections to its local IP address (192.168.1.105). In most circumstances, the attacker would want to first alter the DNS settings of the target PC to redirect the victim to the attacking computer when they access a particular website (E.g. Google, Facebook, Twitter, etc.). For our testing purposes, we simply attempt to connect to the IP address of the attacking machine from our client PC (which brings up the Google page we had previously selected in SET options).

As we can see in Fig.5, our Backtrack PC was able to successfully connect to and launch an instance of VNC viewer on our target client PC. Through our Backtrack PC, we then can view and manipulate the client's desktop. Once in control, we proceeded to install key logger software (PyKeylogger- an open-source key logging client designed to record the keystrokes of users) onto the client PC in an effort to record login credentials for the ownCloud web application administrator account hosted by our ownCloud server PC. Figure 5 shows that we can successfully harvest the login credentials for the ownCloud web application administrator account by examining the contents of the key logger logs through our VNC session with Backtrack.

3.3. Result analysis

Based on the testing results, we analyse the ownCloud security as follows.

- *Man in the Middle Attack*: In our experiment, we can successfully perform the man in the middle attack on the built cloud server. Through this attack, we can successfully intercept the image uploaded onto the cloud service. This kind of attack can be solved through implementing encrypted communication such as applying HTTPS on the cloud server side for authenticating the user identity.
- *SQL Injection Attack*: Our SQL injection is not successful and cannot explore further the security vulnerability of the cloud database. Further method to explore the database server security should be investigated later.

- *Client's OwnCloud Account Attack*: This attack is not performed directly in the ownCloud side. However, due to the success attack on the client machine, we finally obtain the user account and password assigned by the ownCloud side. Hence, to enhance ownCloud security, we also need to assure the security in the client side.

4. Conclusion and future work

In our work, we perform three main popular attacks on the cloud network we built. Some attacks are success and some are not success. We disused our result and provide the solutions for solving these vulnerabilities.

There are still a lot of vulnerabilities in the cloud that we need to identify and more experiments that we need to design and work on. Also, the ownCloud that we built is mainly for storage purpose. It is necessary to build a cloud with different services provided and test the different vulnerabilities on top that. In the near future, we will work on these directions, identify different vulnerabilities for different aspects of cloud and provide solutions for better cloud security.

References

[1] Gonzalez, N., Miers, C., Redigolo, F., Simplicio, M., Carvalho, T., Näslund, M., & Pourzandi, M. (2012). A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing, 1(1), 1-18.*

[2] R. Arkia Paul Rajan, & S. Shanmugapriyaa, (2012). Evolution of cloud storage as cloud computing infrastructure service, *IOSR Journal of Computer Engineering*, PP38-45

[3] Drop Box. (2015). Received from: *https://www.dropbox.com/en/*

[4] Google Drive. (2015). Received from: *https://www.google.com/drive/*

[5] Google Docs. (2015). Received from: *https://www.google.com/docs/about/*

[6] OwnCloud. (2015). Received from: *https://owncloud.org/*

[7] ZenMap. (2015). Received from: *https://nmap.org/zenmap/*

[8] Shikha Singh etc. (2014). Cloud computing attacks: a discussion with solutions. *Open Journal of Mobile Computing and Cloud Computing*.

[9] Te-Shun Chou (2014). Security threats on cloud computing vulnerabilities, *International Journal of Computer Science and Information Technology (IJCSIT), 5,* 3.