

Task4 Report



Hashes are cryptographic functions that transform input data (like a password) into a fixed-size string of characters, which appears random. They are crucial in cybersecurity for verifying data integrity and securely storing passwords. Common hash algorithms include MD5, SHA-1, and SHA-256, each producing different hash lengths. Hash functions are designed to be irreversible, meaning it's computationally infeasible to retrieve the original input from the hash. However, they are vulnerable to brute-force attacks and hash collisions, where different inputs produce the same hash. Using strong, unique inputs and modern algorithms enhances security in hash usage.

Table of Contents

1.0 Future Intern Task4 Report	2
1.1 Introduction	2
2.0 Executive Summary	2
2.1 Executive Summary - Recommendations	2
3.0 Password Cracking	3

1.0 Future Intern Task4 Report

1.1 Introduction

The Offensive Security Lab and Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security course. This report should contain all items that were used to pass the overall exam and it will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

2.0 Executive Summary

Conducted penetration test uncovered several security weaknesses present in web applications owned by testphp.vulnweb.com website

When performing the penetration test, there were several alarming vulnerabilities that were identified on testphp.vulnweb.com networks. When performing the attacks, I was able to find different information, primarily due to outdated patches and poor security configurations.

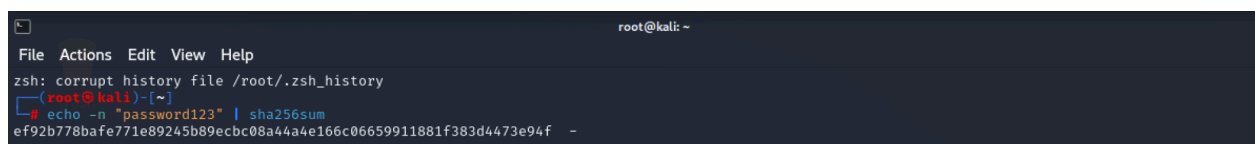
2.1 Executive Summary - Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Information Gathering

We try to create and crack password. There are a lot of ways to create passwords. Now I create sha256 hash in kali linux with commands.

Command: `echo -n "password123" | sha256sum`



```
root@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /root/.zsh_history  
root@kali: ~  
echo -n "password123" | sha256sum  
ef92b778baf771e89245b89ecbc08a44a4e166c06659911881f383d4473e94f -
```

After this i crack this password in a website.Also we can crack with hashcat,john the ripper etc.

SHA256 Hash Cracking | x

https://passwordrecovery.io/sha256/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

GPU based) makes SHA256 a password storage function that is not secure.

An example SHA256 hash of the string password is 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8.

Try to match the example in our online SHA256 password hash tool below.

SHA256 Password Hash Search

Enter a hash below to have it compared against hashes from the **rockyou.txt** password list. These hashes are computed so rapidly that we **test millions of potential passwords** in less than a second.

ef92b778baf771e89245b89ecbc08a44a4e166c06659911881f383d4473e94f

Query

The hash is: password123

And we succeeded in cracking the password.