SQL Injection Attack

Keshia Horton

keshia.horton@students.columbiabasin.edu

CSIA 440 - Cyber Pen Testing

Fall 2018

Abstract

For my Cyber Security Capstone project I chose SQL Injection Attacks. SQL Injection exploit websites that are database driven. These attacks are so prominent that they are number 1 in the Top 10 OWASP Application Security Risks of 2017 (OWASP). DVWA is similar to a healthcare organization’s web application in that it is extremely vulnerable. For my project I used DVWA to simulate a SQL Injection attack and gained access to a “healthcare web application”. I then fixed the vulnerability by enabling a Web Application Firewall.

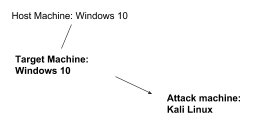
Introduction

SQL is a programming language used to modify, edit, and view information in databases. SQL injection attacks exploit websites that are database driven. An attacker can submit a query to retrieve sensitive information from a website’s database. SQL Injection attacks are common considering there are many websites with a lot of them being database driven. These types of attacks can allow for someone to impersonate another person and bypass authentication, steal sensitive information and delete information. In order for a SQL Injection attack to occur, the website first needs to accept input within a SQL statement, and that the attacker needs to find an input that is also included in the web query (Acunetix)

Problem Statement

A healthcare organization currently has a patient portal on their website where patients can sign up and log in to view their healthcare files and book appointments. This organization claims that their website is immune to SQL injection attacks. For my project I will be recreating their website using the DVWA to test this hypothesis. The reason for choosing the DVWA is because it is a Damn Vulnerable Web Application and healthcare organizations are notoriously known for having poor network security (Finkle). SQL injection attacks are of great concern because they are one of the top security risks according to OWASP and because these databases contain password hashes when can then give the attacker access to a patient’s account that can contain sensitive information such Protected Health Information.

Architecture

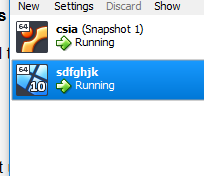


Testing Methodology

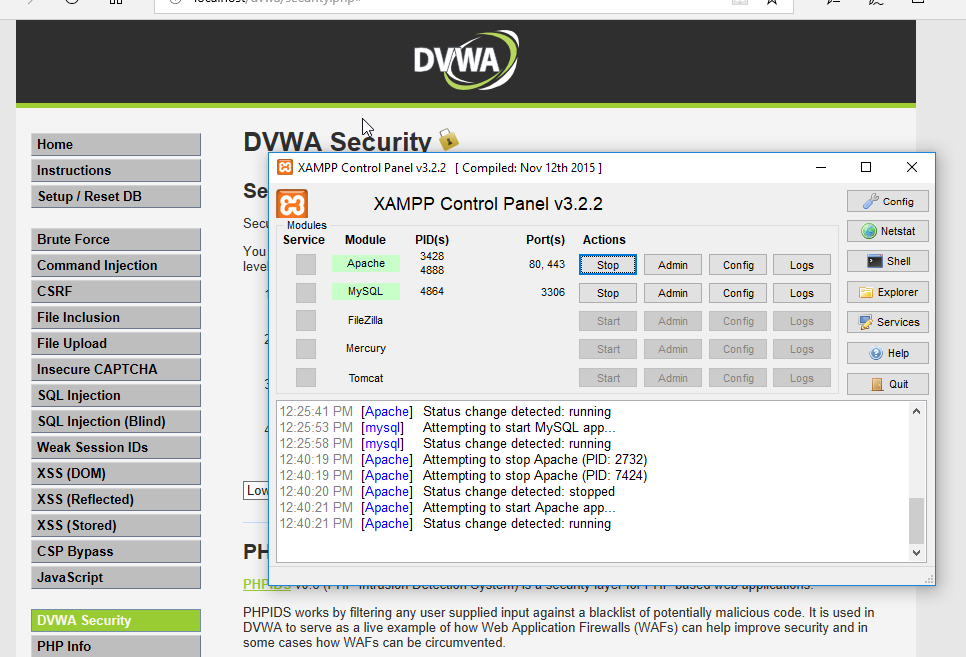
1. I will start by setting up a **Windows 10 machine as the target and a Kali Linux machine as the attacker**
2. I will set up the DVWA with a database consisting of patients as users and **their passwords**
3. I will attempt to use a query to retrieve said **passwords**
4. I will record the data retrieved, and what queries worked
5. Once the data is retrieved, I will fix the vulnerability with a Web Application Firewall and test its efficacy

Implementation

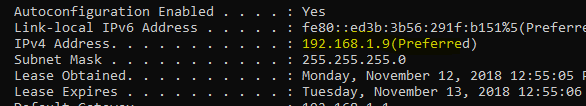
1. Set up both Linux and Windows 10 machines



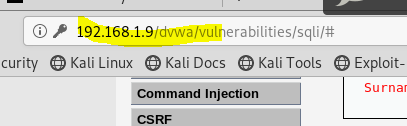
2. Set up DVWA and XAMPP in Windows 10. Reset admin password to password. Security level was set to low.



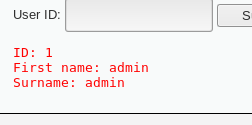
3. Retrieved IP Address of Target machine for Attack machine



4. Logged into Windows 10 DVWA on the Kali machine



5. Tested the database with query 1

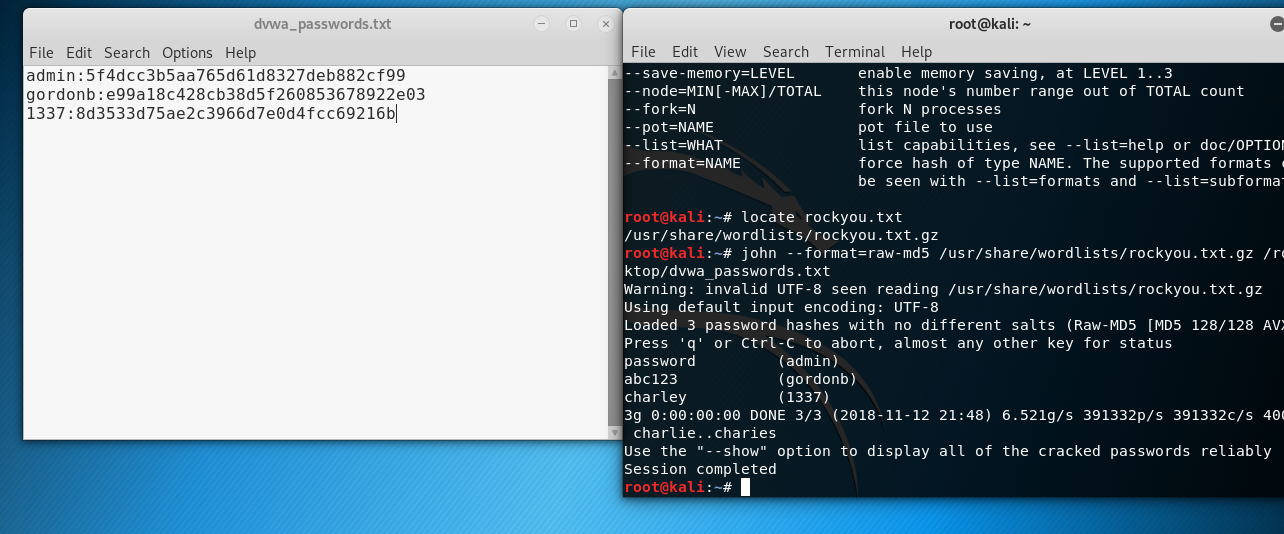


6. To get the version of the mysql database I entered the query %’ or 0=0 union select null, user() #

7. Submitted query to retrieve users and password hashes



8. Then I copied the hashes into a txt file and used John The Ripper to reveal passwords

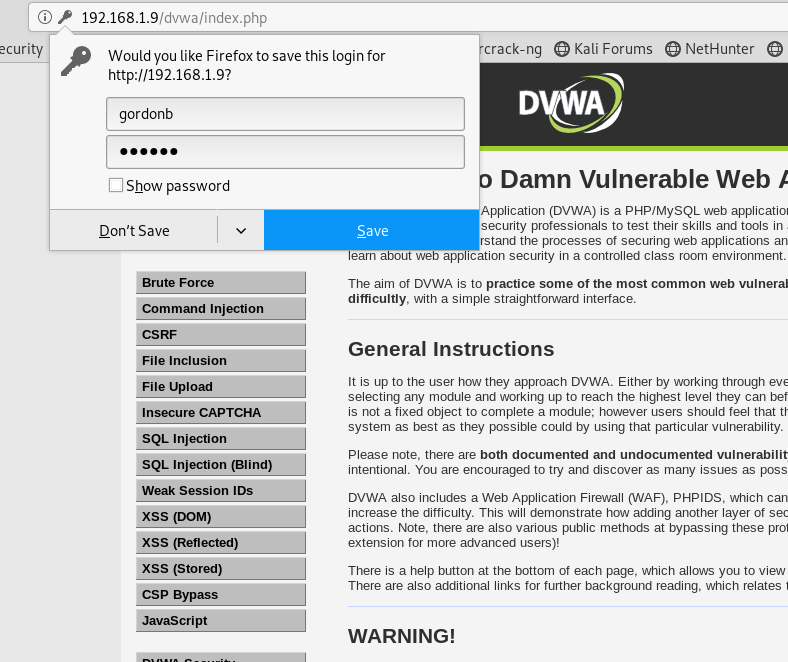


User: Admin Password: Password

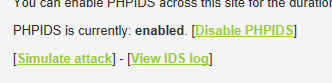
User: gordonb Password: abc123

User: 1337 Password: Charley

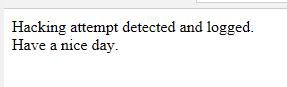
9. Tested the user gordonb’s password and got access to that user’s account



10. To fix this vulnerability I enabled the PHPIDS, the PHP Intrusion Detection System which is the Web Application Firewall in DVWA.

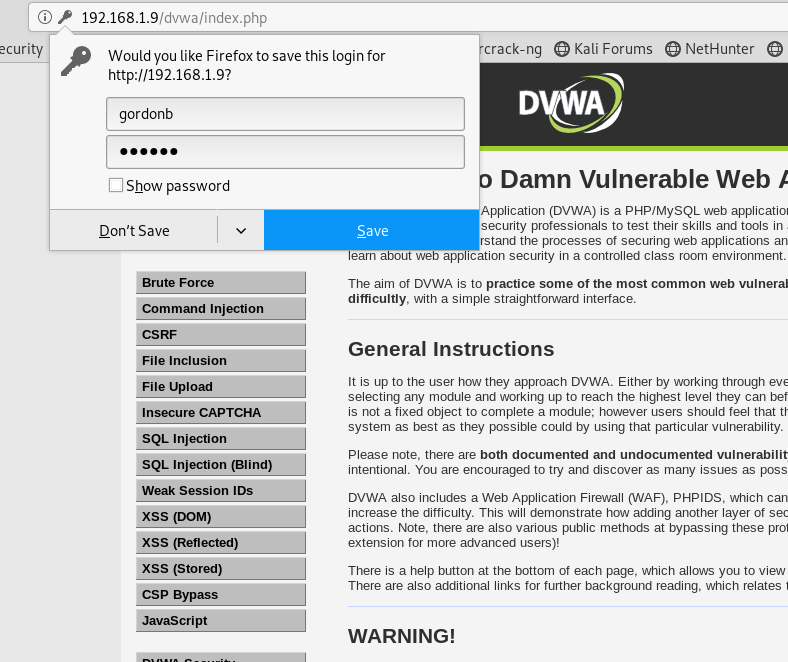
.

11. I then submitted the same query from step 6 and was unable to get in and the attempt was logged.

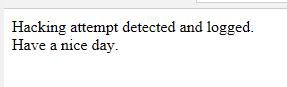


Results

SQL Injection with low security settings in the DVWA resulted in a successful attack as shown with the login with another user using the cracked password from John the Ripper.



The following is the results from the attempted SQL Injection after the WAF was enabled. The attempted attack was logged



Conclusion

The outcomes of this project are that if the healthcare organization left their web application as vulnerable as it is, there will be a security breach and passwords will be obtained. This will result in people’s privacy being violated resulting in potential blackmail or identities being stolen. It is also not smart for hospitals to assume that if they use default settings of a software/hardware or if they use advanced security practices elsewhere, that they are protected from all attacks. Even though using a WAF can protect against SQL Injection attacks, there are many other types of attacks out there that can be used against web applications. Using a basic WAF is not a silver bullet when it comes to web applications.

What I learned working on this project is that it is very easy to cause such a big disruption with basic programming knowledge. Companies can increase their security significantly with little effort as shown in step 10. In my research I learned how prevalent SQL Injections were, in the OWASP Top 10, but also how easy it is to prevent them. DVWA is an easy-to-use tool that lets pen testers simulate several different types of web application attacks and figure out where their own web application is lacking in security.

References

Acunetix. (n.d.). SQL Injection (SQLi). Retrieved from <https://www.acunetix.com/websitesecurity/sql-injection/>

Finkle, J. (2014, August 20). FBI warns Healthcare Firms they are targeted by Hackers. Retrieved from https://www.reuters.com/article/us-cybersecurity-healthcare-fbi/fbi-warns-healthcare-firms-they-are-targeted-by-hackers-idUSKBN0GK24U20140820

OWASP. (2017). Top 10 Security Risks. Retrieved from https://www.owasp.org/images/7/72/OWASP\_Top\_10-2017\_%28en%29.pdf.pdf