InfoSec & Ethics Spring 2024

Hydra
Virtual Penetration Testing Assignment

Project Report

Group 26	
Zohaib Azam	22732
Syeda Maham Jafri	22796
Syed Danial Haseeb	12429
Ashnah Khalid Khan	22889

abstract

This report presents the findings from a penetration testing exercise on the Damn Vulnerable Web Application (dvwa) using the brute force tool `hydra` within a Kali Linux environment. The test focused on assessing the vulnerability of dvwa's login mechanism to brute force attacks. Results indicate significant susceptibility due to weak password configurations and the absence of account lockout procedures. Recommendations for mitigating these vulnerabilities include enhanced password policies, implementation of account lockout mechanisms, and the adoption of multi-factor authentication. This study underscores the importance of robust security practices to thwart common cyber threats.

IBA

School of Mathematics & Computer Science

Contents

1 Executive Summary

This report documents the results of a targeted penetration testing exercise performed on the Damn Vulnerable Web Application (DVWA) hosted within a Kali Linux virtual environment. The primary focus of this penetration test was to evaluate the security robustness of DVWA against brute force attacks targeting its web login interface. The primary tool utilized for this exercise was `hydra`, a powerful and widely-used tool for conducting brute force attacks.

The objectives of this penetration testing were to identify and exploit vulnerabilities in DVWA that could potentially be leveraged by an attacker to gain unauthorized access. The scope was specifically limited to the web login form, a common vector for security breaches in web applications. The testing process included several key phases: preparation, where the attack vectors were defined; execution, where `hydra` was employed to test the strength of the login mechanisms; and analysis, where the results were evaluated.

Significant findings from this exercise revealed that the login mechanism of DVWA is highly susceptible to brute force attacks due to weak password policies and lack of account lockout controls. These vulnerabilities pose a serious threat as they allow attackers to gain unauthorized access with relatively low effort.

Based on these findings, the report further provides detailed recommendations for mitigating the discovered risks. These include implementing stronger password policies, introducing account lockout mechanisms after several failed login attempts, and employing multi-factor authentication to enhance security.

In conclusion, the penetration test demonstrated that while DVWA is intentionally vulnerable for educational purposes, similar vulnerabilities in real-world applications could lead to severe security breaches. Therefore, the recommendations outlined in this report are crucial for strengthening the security posture of web applications against brute force attacks.

2 Introduction

This report details a comprehensive penetration testing project conducted on the Damn Vulnerable Web Application (DVWA), hosted on a Kali Linux virtual environment. The focus of this penetration test was to critically assess the security of DVWA by specifically targeting its web login form with brute force attacks using the tool `hydra`, known for its effectiveness in this type of security assessment.

2.1 Objective of the Penetration Test

The primary objective of this test was to simulate realistic attack scenarios to identify and exploit vulnerabilities within DVWA, particularly those that could be compromised via brute force. This would provide valuable insights into the potential risks faced by similar real-world applications and suggest appropriate mitigations.

2.2 Scope of the Test

The scope of this penetration testing was strictly confined to the web login functionalities of DVWA. The test aimed to uncover vulnerabilities that could allow unauthorized access through brute force techniques, thereby highlighting critical security flaws.

3 Methodology

A systemic approach was adopted for the penetration testing of the Damn Vulnerable Web Application (DVWA) using `hydra` to execute brute force attacks against the web login of DVWA.

3.1 Tools and Technologies Used

The primary tool used in this penetration testing was `hydra`, a well-known tool for performing rapid dictionary attacks against various protocols and services. The virtual environment was set up using Kali Linux, which hosted the DVWA. This setup provided a controlled testing environment that mimics real-world application scenarios.

- **Kali Linux:** Served as the primary operating system for hosting the vulnerable applications and penetration testing tools.
- **Damn Vulnerable Web Application (dvwa):** A purposely vulnerable web application used to practice common web security vulnerabilities.
- **Hydra:** Employed for conducting brute force attacks on the login functionalities of DVWA by performing single username/password attacks, password spraying attacks, dictionary attacks, trivial password attacks, targeted port attacks etc.

3.2 Virtual Lab Setup

Setting up a controlled and realistic virtual lab environment was essential for conducting effective penetration testing on the Damn Vulnerable Web Application (DVWA). This subsection details the process of creating the virtual lab using Kali Linux, configuring DVWA, and setting up the network to simulate a real-world scenario.

Virtual Environment Configuration

The virtual lab was established using VMware, a robust platform for running multiple virtual machines simultaneously. The following steps were taken to configure the virtual environment:

- **Installation of VMware:** VMware Workstation was installed on a host machine to manage virtual machines.
- **Creation of Virtual Machines:**

  - A Kali Linux virtual machine was set up as the attacker machine. Kali Linux was chosen for its extensive suite of pre-installed penetration testing tools, including `hydra`.
  - An additional virtual machine was configured to host DVWA, running on a lightweight Ubuntu server to minimize resource usage.

Network Configuration

To mimic a realistic network environment, both virtual machines were placed in an isolated network segment within VMware to ensure that the testing activities did not impact external networks.

3.3 Vulnerability Analysis

This phase focused on collecting as much information as possible about the target application. The team used various tools to scan the network and system for open ports, services, and vulnerabilities that could potentially be exploited. Using the data collected, the team analyzed the vulnerabilities present in DVWA, especially those that could be exploited via brute force attacks. `hydra` was used to test different combinations of usernames and passwords to assess the resilience of the login system. The identified vulnerabilities were then assessed on their severity and exploitability.

Identified Security Vulnerabilities

DVWA operates on 4 levels of security, each of which should be pen-tested for different kinds of vulnerabilities. These security levels are: Low, Medium, High and Impossible.

The penetration testing process revealed a number of critical security flaws that enabled unauthorized access through brute force methods. The most notable issues included the insufficient strength of passwords and the absence of effective rate limiting or account lockout mechanisms.

In the Low level security setting, the DVWA brute force login prompt is just a straightforward HTTP GET request with no security in place to block or stop you from hammering it with Hydra. In the Medium level security setting, the vulnerability is the same, with only a

In the Medium level security setting, the vulnerabilities in DVWA are the same as that in the Low level security setting, with only a 2 second delay introduced upon failed logins, which is easily accessible in its PHP code.

In the High level security setting, Cross-site Request Forgery (CSRF) tokens are configured into DVWA - these are basically unique tokens for each user session - which make the website incredibly secure, but still contain vulnerabilities that can be accessed through using another tool with Hydra to handle the requests and capturing the CSRF token, altering Hydra's initial request, so it does not need to be aware of the CSRF Token. One such tool like this could be Burp Suite.

The Impossible level security setting was, as expected, impenetrable through just a tool alone like Hydra. At this level, DVWA is configured to lock out a user account after 5 failed logins made within 15 minutes. If the locked out user tries to login, even with a valid password, it will say their username or password is incorrect. this should make it impossible to know there is a valid account on the system. Password spraying, where you use one password against multiple users instead of constantly querying for a single user you so you end up not not hitting the lock out threshold, will still work on this level of security as well.

Insufficient Password Policies

DVWA was found to have extremely lenient password policies, allowing the use of basic and widely recognized passwords. This issue greatly reduces the complexity needed for conducting a successful brute force attack, thus posing a significant security risk.

Absence of Account Lockout Protocols

Furthermore, the application lacks mechanisms to lock out accounts after multiple failed login attempts. This absence allows malicious entities to continually attempt access on the same accounts without any repercussions, significantly enhancing the probability of a successful breach through brute force methods.

3.4 Severity And Risk Assessment

The identified vulnerabilities were rigorously assessed for both their potential impact on the system and the likelihood of their exploitation.

- **Insufficient Password Policies**:
  - Severity: High
  - Risk: The lenient password policies significantly reduce the complexity required for successful brute force attacks, making unauthorized access more probable. The impact of compromised accounts could range from unauthorized data access to complete system compromise, depending on the privileges associated with the compromised accounts.

- **Absence of Account Lockout Protocols**:
  - Severity: Medium
  - Risk: The absence of account lockout mechanisms increases the likelihood of successful brute force attacks. Malicious entities can repeatedly attempt to access user accounts without any restrictions, leading to a higher probability of account compromise. While the impact of compromised accounts is significant, it is mitigated to some extent by other security measures in place.
- **Vulnerabilities Across Security Levels**:
  - Severity: Varies
  - Risk: The vulnerabilities present across different security levels of DVWA pose varying degrees of risk. In the Low and Medium security settings, the absence of effective rate limiting mechanisms exposes the system to brute force attacks with relatively low barriers to entry. In the High security setting, while CSRF tokens enhance security, vulnerabilities can still be exploited using specialized tools, albeit with greater difficulty. The Impossible security setting provides robust protection against brute force attacks but may still be susceptible to password spraying techniques.

3.5 Exploitation

The identified vulnerabilities were then exploited using `hydra` to confirm their exploitability and to understand the level of unauthorized access that could be achieved. Following successful exploitation, the team performed actions to maintain access, simulating an attacker's behavior to explore further vulnerabilities and gather more sensitive data. This section outlines the procedure and results from the comprehensive penetration testing and website login brute force attack conducted on the Damn Vulnerable Web Application (DVWA).

Conducting the Brute Force Attack

For this purpose, `hydra` was adeptly configured to utilize a list of commonly used passwords alongside recognized usernames from DVWA. Within a short span of minutes, this approach led to the compromise of several accounts, starkly illustrating the ease with which unauthorized access could be achieved.

Methods used include single username/password attacks, password spraying attacks, dictionary attacks, and trivial password attacks. Scripts for each can be found in the Code and Scripts section of this report.

In situations where we would like to test usernames and passwords we expect the system to have, `hydra` allows us to use a single-line command to test it.

`hydra` also has a built-in command that gives us the option for checking for the three trivial passwords i.e. the null password (empty string), the password that is the same as the username, and the password being the username reversed. This can further be configured to run the same for a list of usernames, instead of just one username.

For password spraying attacks, `hydra` allows us to configure a list of usernames (preferably in a text file) for which to check a known password in the system against. This too can be achieved using a single-line command.

For dictionary attacks, which is what most real-world scenarios are closest to, is where we have single/multiple usernames and we provide a password wordlist to Hydra. `hydra` then tests all these passwords against every user in the list.

A special feature of `hydra` is that it also allows us to check for a specific format/style of passwords against a list of usernames as well. This is similar to generating password strings against a regex expression and checking it against given usernames.

For performing each of these attacks during our penetration test, we must just know the IP address of our DVWA login page, which we can acquire easily. In addition, we must remember to change the security level settings for each security level Low, Medium, High and Impossible, before conducting each of these attacks on each of them to understand the types of vulnerabilities in our system better.

Post-exploitation And Implications

The successful exploitation of these vulnerabilities granted unauthorized access to DVWA's administrative panel. This breach potentially opens the door to more severe threats such as data theft, manipulation of system functionalities, and could facilitate extended attacks on other parts of the network.

This methodical approach allowed the team to thoroughly assess the security vulnerabilities of DVWA, providing a clear picture of its weaknesses and areas for improvement.

4 Recommendations and Remediation

Based on the findings from the penetration testing of the Damn Vulnerable Web Application (DVWA), several vulnerabilities were identified that pose significant security risks. This chapter outlines recommended measures to mitigate these vulnerabilities, thereby enhancing the security posture of systems similar to DVWA.

4.1 General Security Enhancements

To address the broader security issues identified, the following general enhancements are recommended:

- **Security Awareness Training:** Conduct regular training sessions for developers and system administrators to raise awareness about common security pitfalls and best practices in web application development.
- **Regular Security Audits:** Implement a routine schedule of security audits and penetration testing to identify and remediate vulnerabilities proactively.
- **Update and Patch Management:** Establish a robust process for regularly updating and patching software to protect against known vulnerabilities.

4.2 Specific Remediations for Identified Vulnerabilities

Each of the specific vulnerabilities identified during the testing phase requires targeted remediation strategies. The strategies outlined below are designed to address the root causes of the vulnerabilities and prevent future exploitation.

Strengthening Password Policies

Vulnerability Addressed: Weak Password Policies

- **Implementation of Strong Password Policies:** Enforce password complexity requirements that include a minimum length, the inclusion of upper and lower case letters, numbers, and special characters.
- **Password Rotation Policies:** Mandate regular password changes every 90 days to minimize the risk of password-related breaches.
- **User Education:** Provide users with training on the importance of using strong, unique passwords for each service.

Implementing Account Lockout Mechanisms

Vulnerability Addressed: Lack of Account Lockout Mechanisms

- **Account Lockout:** Introduce an account lockout policy that temporarily locks user accounts after a series of failed login attempts. This measure will significantly reduce the risk of brute force attacks.
- **Progressive Delays:** Implement progressive delay mechanisms to slow down repeated login attempts, further deterring brute force strategies.

Introduction of Multi-Factor Authentication

Vulnerability Addressed: High Risk of Unauthorized Access

- **Multi-Factor Authentication (MFA):** Require MFA for all user accounts, especially administrative ones, to add an additional layer of security beyond just username and password.
- **MFA Implementation Strategies:** Use a combination of something the user knows (password), something the user has (security token or app), and something the user is (biometric verification).

4.3 Conclusion

The implementation of these recommendations will significantly mitigate the risks identified during the penetration testing of DVWA. By addressing both specific vulnerabilities and enhancing general security practices, organizations can improve their resilience against cyber threats.

This chapter concludes with a call to action for continuous security improvement and vigilance, emphasizing that security is not a one-time effort but an ongoing process.

5 Conclusion

The penetration testing conducted on the Damn Vulnerable Web Application (DVWA) has provided significant insights into the security vulnerabilities inherent in web applications that are not adequately protected. This exercise aimed to identify, exploit, and analyze vulnerabilities, particularly those susceptible to brute force attacks targeting web login forms.

5.1 Summary of Findings

The testing revealed several critical security issues within DVWA, most notably weak password policies and the absence of account lockout mechanisms, which facilitated successful brute force attacks. These vulnerabilities, if left unaddressed, could allow malicious actors to gain unauthorized access, leading to potential data breaches and other security incidents.

5.2 Impact of the Testing

This penetration test underscores the necessity of rigorous security measures and regular vulnerability assessments in maintaining the integrity and confidentiality of information systems. The insights gained from this exercise are invaluable for developers and security professionals in understanding and mitigating the risks associated with cyber threats.

5.3 Recommendations for Future Security Practices

As cyber threats evolve, so too must the strategies to counteract them. This report has outlined specific remediations to address the vulnerabilities found in DVWA. However, beyond these immediate measures, it is crucial for organizations to implement ongoing security practices including:

- Continuous education and training for all employees on the latest security threats and best practices.
- Regular updates and patches to software and systems to protect against new vulnerabilities.
- Frequent security assessments and penetration tests to proactively identify and address security weaknesses.

5.4 Final Thoughts

The discipline of cybersecurity is dynamic and requires constant vigilance and adaptation. The findings from this penetration testing exercise should serve as a reminder of the importance of security in the development and maintenance of web applications. Organizations must stay vigilant, proactive, and informed to protect against the ever-growing landscape of cyber threats.

In conclusion, while the vulnerabilities identified in this exercise are specific to DVWA, the lessons learned are applicable universally across all web applications. By adopting the recommendations provided and maintaining a commitment to ongoing security improvement, organizations can better safeguard their assets and reputation in the digital age. [1] [2] [3] [4] [5] [6] [7]

A Detailed Logs and Outputs

This appendix contains detailed logs and output data from the tools used during the penetration testing. It includes examples of screenshots, command line sessions, and logs that document the steps taken to exploit identified vulnerabilities in DVWA.

A.1 Hydra Output Logs

These logs illustrate the results from the brute force attacks performed using `hydra`. They show the effectiveness of the attacks and the vulnerabilities exploited.

```
[80][http-post-form]
host: 192.168.1.105
login: admin
password: password123
[STATUS] attack finished for 192.168.1.105
(waiting for children to complete tests)
[80][http-post-form]
host: 192.168.1.105
login: admin
password: failpassword
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra)
finished at 2024-04-21 15:37:45
```

A.2 Network Traffic Logs

Network traffic logs capture the interactions between the attacker machine and the DVWA server during the testing phase. These logs are crucial for analyzing the data flow and pinpointing potential data leakage points.

```
GET /login.php HTTP/1.1
Host: dvwa.local
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: security=low; PHPSESSID=2a99836d1a575a489eca0847ad44d94a
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Sun, 21 Apr 2024 15:40:22 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Tue, 12 Feb 2019 10:18:45 GMT
ETag: "2c39-5816e6b7b626b"
Accept-Ranges: bytes
Content-Length: 11321
Keep-Alive: timeout=5, max=100
Content-Type: text/html
```

A.3 System Configuration Changes

This section documents the changes made to system configurations of virtual machines during the setup and testing phases. These logs are essential for replicating the test environment and for audit purposes.

```
VMware VM Configuration:
- VM Name: KaliLinux2024
- Allocated Memory: 2048 MB
- Number of CPUs: 2
- Network Adapter: VMnet8 (NAT)

Operating System Updates:
- sudo apt-get update
- sudo apt-get upgrade -y

DVWA Configuration Changes:
- Config file path: /var/www/html/dvwa/config/config.inc.php
- Database password: p@ssw0rd
- Recaptcha key: 6Lc6BAAAAAAAAChqRbQZcn_yyyyyyyyyyyyyyyyy
```

This appendix provides a thorough overview of the types of logs that are typically collected during penetration testing, demonstrating how these logs can be used to validate testing procedures, analyze the effectiveness of attacks, and ensure compliance with best practices.

B Code Snippets and Scripts

This appendix includes code snippets and scripts that were developed or used during the penetration testing. These scripts automate certain tasks, create custom exploits, or serve other utilities that facilitate the testing process.

B.1 Hydra Brute Force Script

This script utilizes Hydra to conduct a brute force attack on the login page of DVWA.

```
hydra -L <user_list> -P <pass_list> <URL> http-post-form \
"/login.php:username=^USER^&password=^PASS^&Login=Login:\
S=Location: dashboard.php" -V
```

B.2 Setting Up DVWA

The following commands set up DVWA on a local server for testing. Execute these in your server terminal:

```
sudo docker pull vulnerables/web-dvwa
sudo docker run -d -p 80:80 vulnerables/web-dvwa
```

B.3 Mitigation Code Examples

This PHP snippet implements an account lockout mechanism that activates after multiple failed login attempts. It is a preventive measure against brute force attacks.

```
<?php
session_start();
if(!isset($_SESSION['failed_attempts'])){
    $_SESSION['failed_attempts'] = 0;
}
if($_SESSION['failed_attempts'] > 3){
    die('Account locked due to too many failed login attempts.');
}
if($_POST['username'] !== 'admin' || $_POST['password'] !== 'password'){
    $_SESSION['failed_attempts'] += 1;
} else {
    $_SESSION['failed_attempts'] = 0; // Reset on success
}
?>
```

B.4 Automated Vulnerability Scanning Script

This Python script employs Nmap to scan for open ports and detect services on a target machine. It is useful for preliminary scans to identify potential vulnerabilities.

```
import nmap
nm = nmap.PortScanner()
nm.scan('192.168.1.1', '22-443')
for host in nm.all_hosts():
    print('Host : %s (%s)' % (host, nm[host].hostname()))
    print('State : %s' % nm[host].state())
    for proto in nm[host].all_protocols():
        print('----------')
        print('Protocol : %s' % proto)
        lport = list(nm[host][proto].keys())
        for port in sorted(lport):
            print ('%s\t%s' % (port, nm[host][proto][port]['state']))
```

This chapter provides practical examples and tools used during the penetration testing process, offering insights into the setup and application of security measures to mitigate potential vulnerabilities.

C Glossary

This appendix provides definitions of technical terms, acronyms, and jargon used throughout the report. It serves to clarify any specialized or technical language that may not be familiar to all readers.

Brute Force Attack A method used to obtain sensitive data such as user passwords by systematically trying every possible combination of options until the correct one is found. This attack can be highly effective against systems with weak password policies.

DVWA Damn Vulnerable Web Application. This is an intentionally insecure web application that provides a safe platform for security training and testing in various domains of web application security.

Hydra A powerful network logon cracker that is used to test the strength of passwords on a network. It supports numerous attack protocols, including Telnet, SSH, HTTP, and others. It is commonly used in penetration testing to demonstrate how easy it can be to breach systems with weak credentials.

Kali Linux A Debian-based Linux distribution designed for digital forensics and penetration testing. It comes pre-equipped with a variety of tools necessary for hacking and security testing.

VMware A virtualization software that provides a completely virtualized set of hardware to the guest operating system. VMware software provides a completely isolated and independent environment for running different operating systems on the same physical machine.

Multi-Factor Authentication (mfa) A security system that requires more than one method of authentication from independent categories of credentials to verify the user's identity for a login or other transaction.

D Compliance and Ethical Considerations

This appendix discusses the ethical considerations and compliance measures that were carefully observed to ensure that the penetration testing was conducted responsibly, ethically, and within legal boundaries.

D.1 Ethical Guidelines

Ethical conduct is paramount in penetration testing to maintain the trust and safety of all stakeholders involved. The following principles were adhered to throughout the testing process:

- Consent and Authorization: Prior to the commencement of testing, formal consent was obtained from all relevant parties. This ensures that the testing activities are legally sanctioned and ethically justified.
- Defined Scope of Testing: The scope of the penetration testing was clearly defined and agreed upon at the initial stages to ensure that all activities were confined to agreed parameters, thereby avoiding any unintended access to or disruption of operational systems.
- Minimization of Impact: Throughout the testing, every effort was made to minimize any impact on system performance and user experience. This involves careful planning and execution to avoid causing undue stress or harm to the system and its data.
- Confidentiality: All information uncovered during the testing process was treated with the utmost confidentiality. Data handling protocols were strictly followed to ensure that sensitive information was protected against unauthorized access or disclosure.

D.2 Legal Compliance

Adhering to legal standards is crucial in penetration testing to avoid any violations of laws and regulations. The legal frameworks that were observed include:

- Data Protection Laws: All activities conformed to international and national data protection laws, ensuring that personal and sensitive data were handled securely and in compliance with legal requirements.
- Compliance with Cybersecurity Regulations: The testing adhered to industry-standard cybersecurity practices and regulations, such as those outlined by the ISO/IEC 27001 standards for information security management.

- Permissive Licensing Considerations: Both Hydra and DVWA are distributed under permissive licenses that allow for educational and testing use. It is important to note that Hydra explicitly discourages its use by security agencies and hacking groups, which aligns with our commitment to using these tools solely for educational purposes in this testing context.

D.3 Responsibility towards Stakeholders

Maintaining a responsible attitude towards all stakeholders involved in or affected by the penetration testing was a key consideration:

- Stakeholder Engagement: Regular updates and communications were provided to stakeholders throughout the testing process. This ensures transparency and maintains trust between all parties.
- Debriefing and Feedback: After the completion of testing, a detailed debriefing session was conducted with all stakeholders to discuss the findings, potential impacts, and future preventive measures. Feedback was solicited to improve future testing procedures.

In conclusion, adherence to ethical principles and legal standards not only ensures the legality of the penetration testing activities but also enhances the integrity and value of the findings. These practices form the cornerstone of any cybersecurity testing and are crucial for maintaining professional standards and stakeholder trust.

References

[1] *Owasp top 10 - the ten most critical web application security risks*, Online, Accessed: 2024-04-20, OWASP, 2023. [Online]. Available: `https://owasp.org/www-project-top-ten/`.

[2] *National institute of standards and technology: It security*, Online, Accessed: 2024-04-20, National Institute of Standards and Technology, 2023. [Online]. Available: `https://www.nist.gov/topics/information-technology`.

[3] *Hydra: A very fast network logon cracker*, Online, Accessed: 2024-04-20, 2023. [Online]. Available: `https://github.com/vanhauser-thc/thc-hydra`.

[4] *Vmware documentation*, Online, Accessed: 2024-04-20, 2023. [Online]. Available: `https://docs.vmware.com/`.

[5] *How to set up a virtual lab on your laptop with vmware workstation and kali linux*, Online, Accessed: 2024-04-20, 2023. [Online]. Available: `https://www.digitalocean.com/community/tutorials/how-to-set-up-a-virtual-lab-on-your-laptop-with-vmware-workstation-and-kali-linux`.

[6] *Metasploit documentation*, Online, Accessed: 2024-04-20, 2023. [Online]. Available: `https://docs.rapid7.com/metasploit/`.

[7] *Damn vulnerable web application (dvwa)*, Online, Accessed: 2024-04-20, 2023. [Online]. Available: `http://dvwa.co.uk/`.