METASPLOITABLE

13th Sep 2022

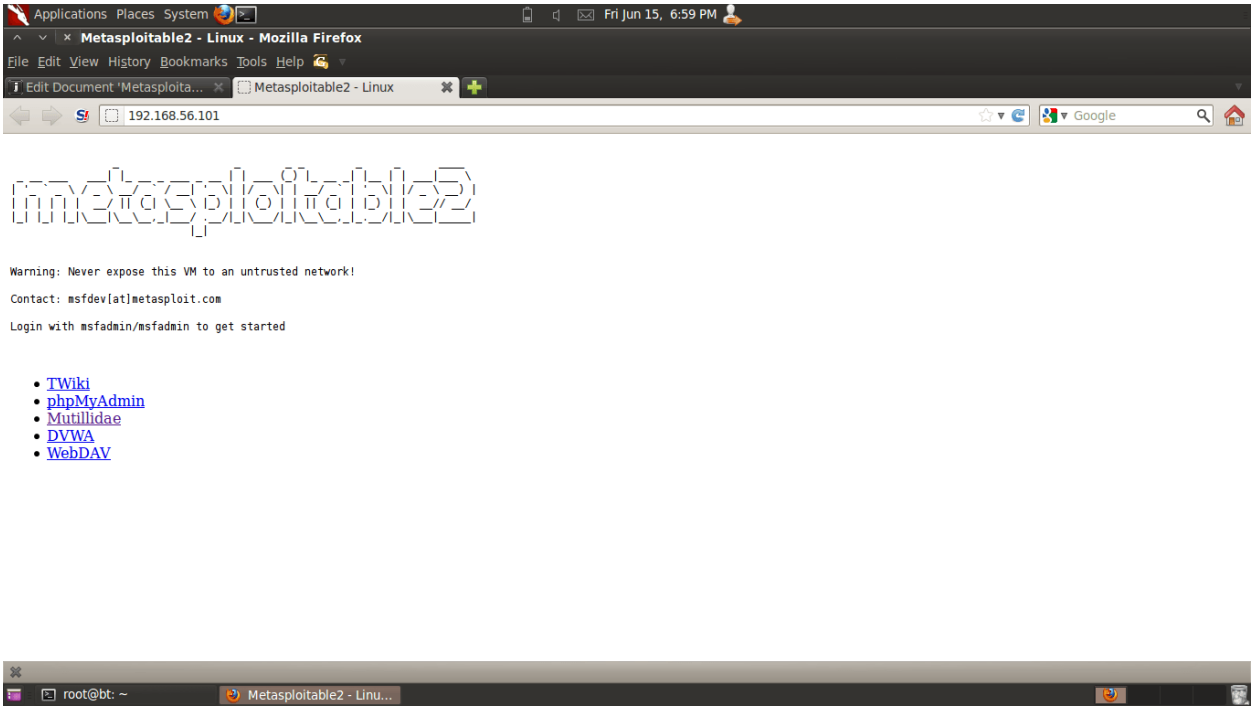
OVERVIEW

A test environment provides a secure place to perform penetration testing and security research. For your test environment, you need a Metasploit instance that can access a vulnerable target. The following sections describe the requirements and instructions for setting up a vulnerable target.

Vulnerable Web Services

Metasploitable 2 has deliberately vulnerable web applications pre-installed. The web server starts automatically when Metasploitable 2 is booted. To access the web applications, open a web browser and enter the URL HTTP://<IP> where <IP> is the IP address of Metasploitable 2. One way to accomplish this is to install Metasploitable 2 as a guest operating system in Virtual Box and change the network interface settings from "NAT" to "Host Only".

Metasploitable 2 is running at IP 192.168.56.101. Browsing to http://192.168.56.101/ shows the web application home page.

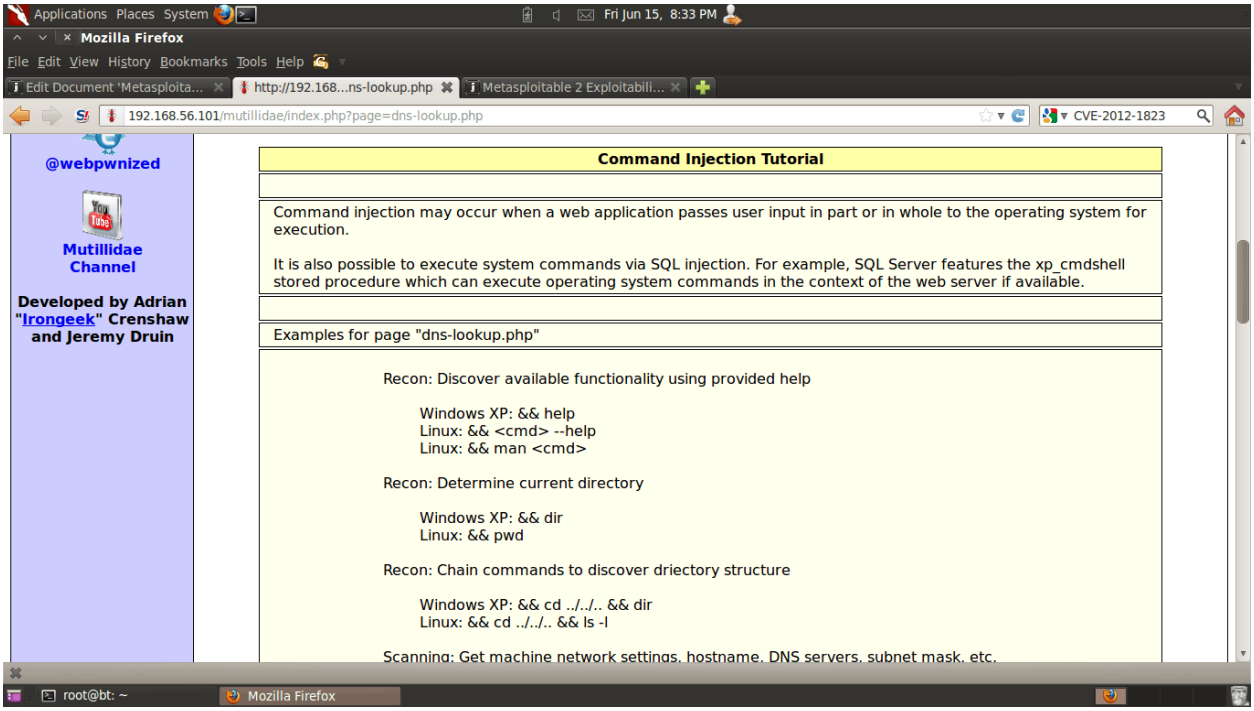


THE APPLICATIONS ARE:

- **mutillidae (NOWASP Mutillidae 2.1.19)**
- **downwardwa (Damn Vulnerable Web Application)**
- **phpMyAdmin**
- **tikiwiki (TWiki)**
- **tikiwiki-old**
- **DAVav (WebDav)**

Mutillidae

The Mutillidae web application (NOWASP (Mutillidae)) contains all of the vulnerabilities from the OWASP Top Ten plus several other vulnerabilities such as HTML-5 web storage, forms caching, and click-jacking. Inspired by DVWA, Mutillidae allows the user to change the "Security Level" from 0 (completely insecure) to 5 (secure). Additionally, three levels of hints are provided ranging from "Level 0 - I try harder" (no hints) to "Level 2 - noob" (Maximum hints). If the application is damaged by user injections and hacks, clicking the "Reset DB" button resets the application to its original state.



The Mutillidae application contains at least the following vulnerabilities on these respective pages:

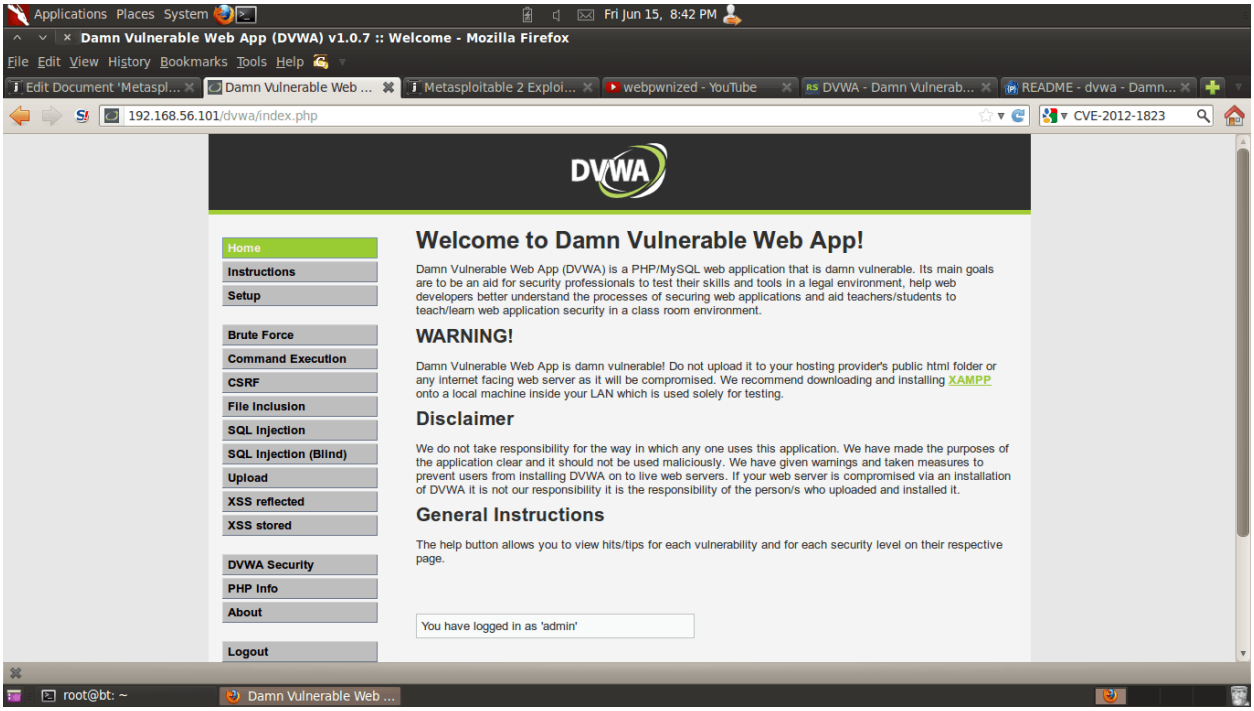
Page	Vulnerabilities
add-to-your-blog.php	SQL Injection on blog entry SQL Injection on logged in user name Cross site scripting on blog entry Cross site scripting on logged in user name Log injection on logged in user name CSRF JavaScript validation bypass XSS in the form title via logged in username The show-hints cookie can be changed by user to enable hints even though they are not supposed to show in secure mode
arbitrary-file-inclusion.php	System file compromise Load any page from any site
browser-info.php	XSS via referer HTTP header JS Injection via referer HTTP header XSS via user-agent string HTTP header
capture-data.php	XSS via any GET, POST, or Cookie
captured-data.php	XSS via any GET, POST, or Cookie
config.inc*	Contains unencrytped database credentials
credits.php	Unvalidated Redirects and Forwards

dns-lookup.php	Cross site scripting on the host/ip field O/S Command injection on the host/ip field This page writes to the log. SQLi and XSS on the log are possible GET for POST is possible because only reading POSTed variables is not enforced.
footer.php*	Cross site scripting via the HTTP_USER_AGENT HTTP header.
framing.php	Click-jacking
header.php*	XSS via logged in user name and signature The Setup/reset the DB menu item can be enabled by setting the uid value of the cookie to 1
html5-storage.php	DOM injection on the add-key error message because the key entered is output into the error message without being encoded
index.php*	You can XSS the hints-enabled output in the menu because it takes input from the hints-enabled cookie value. You can SQL injection the UID cookie value because it is used to do a lookup You can change your rank to admin by altering the UID value HTTP Response Splitting via the logged in user name because it is used to create an HTTP Header This page is responsible for cache-control but fails to do so This page allows the X-Powered-By HTTP header HTML comments There are secret pages that if browsed to will redirect user to the phpinfo.php page. This can be done via brute forcing

DWV

From the DVWA home page: "Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a classroom environment.

- **Default username** - admin
- **Default password** – Password


Applications Places System
Fri Jun 15, 8:42 PM
Damn Vulnerable Web App (DVWA) v1.0.7 :: Welcome - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Edit Document 'Metaspl...
Damn Vulnerable Web ...
Metasploitable 2 Exploi...
webpwnized - YouTube
DVWA - Damn Vulnerab...
README - dvwa - Damn...
192.168.56.101/dvwa/index.php
CVE-2012-1823
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web App!
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.
WARNING!
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.
Disclaimer
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.
General Instructions
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.
You have logged in as 'admin'
root@bt: ~
Damn Vulnerable Web ...