Foundations of computer security

Norbert Oláh

2023.

Content

1. Nessus – Lab
2. Nessus – Advanced scanner

First steps

- Start Parrot linux
- Open Terminal
- Checking the nessus application:
  service nessusd status
- Open the web browser and go to
  https://parrot:8834/
- Log in to Nessus
  UN: nessus PW: nessus

Create your advanced scan 1

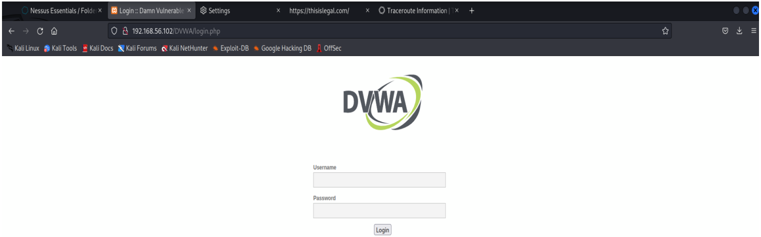
Create and execute an arbitrarily advanced scanner on the target below:

- 172.22.204.195

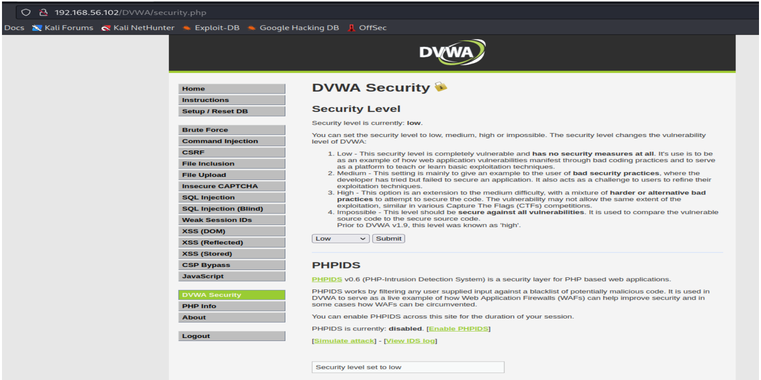
Labour issue

- Visit the DVWA website
  http://172.22.204.188/login.php
  Username: admin
  PW: password

DVWA


Nessus Essentials / Folde
Login :: Damn Vulnerable
Settings
https://thisislegal.com/
Traceroute Information
192.168.56.102/DVWA/login.php
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
DVWA
Username
Password
Login

DVWA



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low | Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

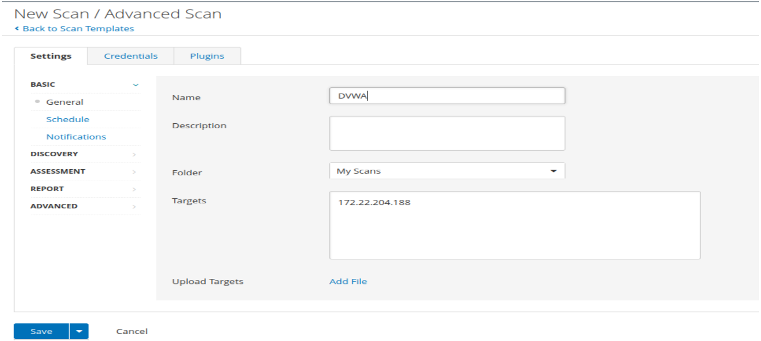
Security level set to low

Create your advanced scan

Create an Advanced scanner with the following settings:

- Assessment/Web application/ Web Crawler /DVWA/
- Assessment/Brute Force Create a username and password file.
- Set the login details in the Credentials tab using the Plaintext Authentication / Automatic authentication.

Create your advanced scan



New Scan / Advanced Scan

‹ Back to Scan Templates

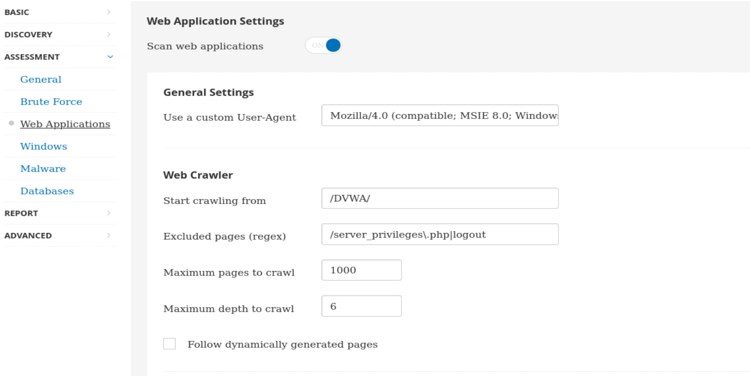
Name: DVWA

Description:

Folder: My Scans

Targets: 172.22.204.188

Create your advanced scan



Web Application Settings

Scan web applications: ON

General Settings

Use a custom User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Window

Web Crawler

Start crawling from: /DVWA/

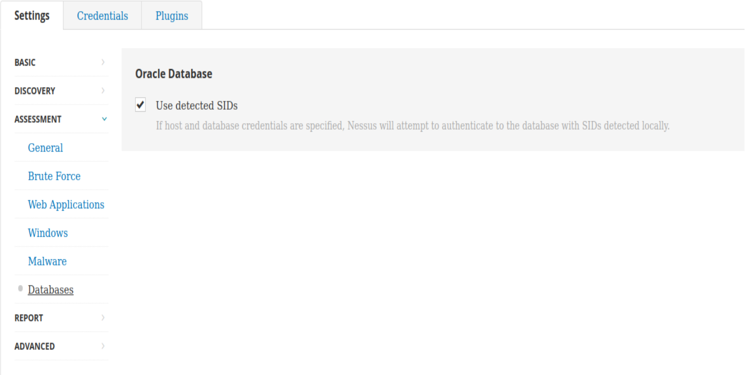
Excluded pages (regex): /server_privileges\.php|logout

Maximum pages to crawl: 1000

Maximum depth to crawl: 6

□ Follow dynamically generated pages

Create your advanced scan

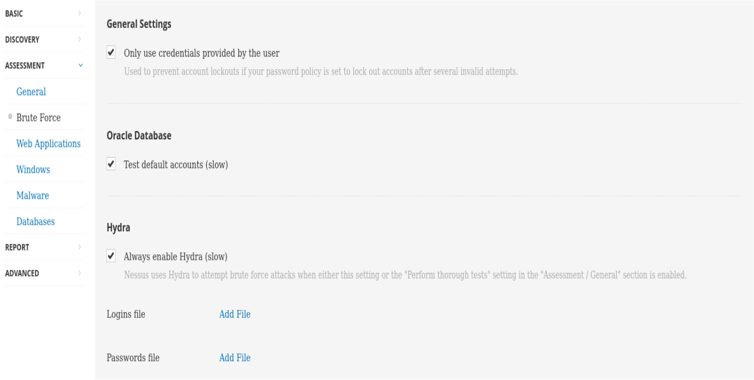


Oracle Database

☑ Use detected SIDs

If host and database credentials are specified, Nessus will attempt to authenticate to the database with SIDs detected locally.

Create your advanced scan



General Settings

☑ Only use credentials provided by the user

Used to prevent account lockouts if your password policy is set to lock out accounts after several invalid attempts.

Oracle Database

☑ Test default accounts (slow)

Hydra

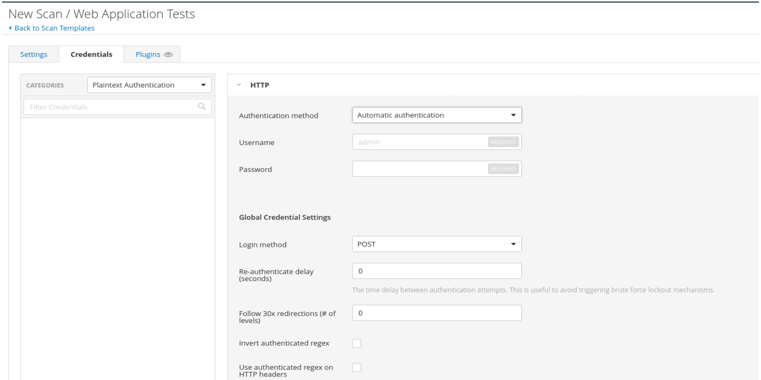
☑ Always enable Hydra (slow)

Nessus uses Hydra to attempt brute force attacks when either this setting or the "Perform thorough tests" setting in the "Assessment / General" section is enabled.

Logins file Add File

Passwords file Add File

Create your advanced scan



New Scan / Web Application Tests

‹ Back to Scan Templates

Plaintext Authentication

HTTP

Authentication method: Automatic authentication

Username: admin REQUIRED

Password: REQUIRED

Global Credential Settings

Login method: POST

Re-authenticate delay (seconds): 0

The time delay between authentication attempts. This is useful to avoid triggering brute force lockout mechanisms.

Follow 30x redirections (# of levels): 0

Invert authenticated regex □

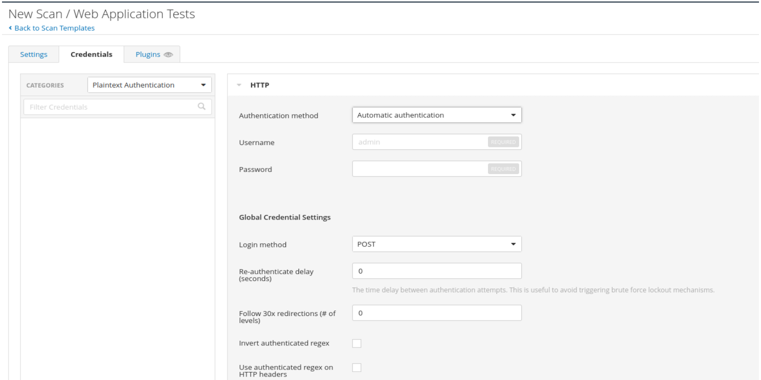
Use authenticated regex on HTTP headers □

Create your web application scanner

Create a Web application scanner with the following settings:

- Name: DVWA, Targets: 172.22.204.188
- Assessment/Web application/ Web Crawler /DVWA/
- Set the login details in the Credentials tab using the the Plaintext Authentication / Automatic authentication.

Create your web application scanner



New Scan / Web Application Tests

‹ Back to Scan Templates

Categories: Plaintext Authentication

Authentication method: Automatic authentication

Username: admin (REQUIRED)

Password: (REQUIRED)

Login method: POST

Re-authenticate delay (seconds): 0

The time delay between authentication attempts. This is useful to avoid triggering brute force lockout mechanisms.

Follow 30x redirections (# of levels): 0

Invert authenticated regex ☐

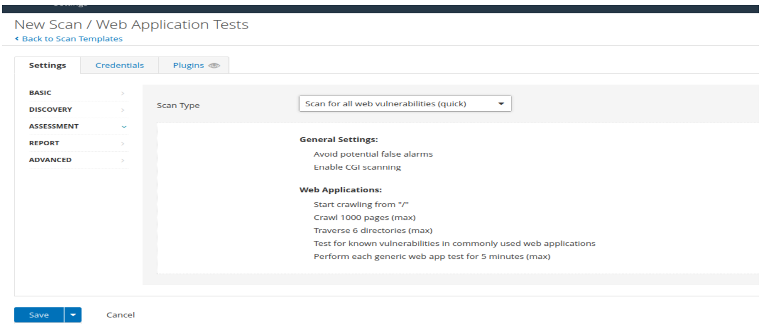
Use authenticated regex on HTTP headers ☐

Készíts el a webalkalmazás szkenned 2

Create a Web application scanner with the following settings:

- Name: OWASP, Targets: 172.22.204.195
- Select any application from OLD (VULNERABLE) VERSIONS OF REAL APPLICATIONS on the page then on the Assessment tab, run first custom and then the complex scan type using this ip address.

Create your web application scanner



New Scan / Web Application Tests

‹ Back to Scan Templates

Scan Type: Scan for all web vulnerabilities (quick)

General Settings:

Avoid potential false alarms
Enable CGI scanning

Web Applications:

Start crawling from "/"
Crawl 1000 pages (max)
Traverse 6 directories (max)
Test for known vulnerabilities in commonly used web applications
Perform each generic web app test for 5 minutes (max)

Final practical test - example

Describe how you would test www.hackthissite.org. Perform the task using Basic scanning. Answer the following questions:

- Target IP Address
- Target ports
- Choose a vulnerability on the list and gather as much information as possible about it (Include a description of the problem, possible solutions, and additional resources about the exposure). If there is no vulnerability select an info type vulnerability and detail the point.

Generate a pdf of the scan results, including all vulnerabilities and related information. Create a separate document to answer the questions. Use the Snipping tool and take screenshots showing the results from your Nessus account.

Thank you for your attention!