LOOKING AT DRUPAL 8 SECURITY

ZIONSECURITY
PREVENT-PROTECT-DETECT-ACT

AUSY
BELGIUM

platform.sh

Table of Contents

1

EXECUTIVE SUMMARY

Executive Summary

Open source software has been popular since the very early days of the Internet, but in the last few years there has also been an increasing trend to use this type for software to build Enterprise web applications. One of the most popular open source content management frameworks for building advanced digital experiences is Drupal, which boasts a very large, dynamic and international community.

Another trend in the Enterprise IT world is to move away from on-premises hosting to cloud hosting, where the customer get access to a growing, integrated set of IT solutions corresponding to different service models, such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Open source software and cloud solutions offer much added business value, but can also raise security concerns. Some users question whether open source software and cloud computing offer the level of safety and security for their applications and data that is required for Enterprise environments.

To create this white paper, three complementary companies, each experts in their own domain, have joined forces to investigate those concerns:

- ONE Agency (AUSY Belgium) has built a website in Drupal 8, the latest and most advanced version released in 2015, in combination with Drupal Commerce and some often-used contributed modules to provide a realistic testing environment. All the best practices with regard to web application security were applied.

- Platform.sh is a continuous deployment high-availability cloud hosting service that helps applications scale effortlessly to serve the most demanding traffic. It can clone a full production cluster, including all its data, in under a minute to deliver up to 20x faster testing, making it ideal for agile development teams. Production environments can propose a 99.99% SLAs thanks to its unique high availability container-based grid & 24/365 follow-the-sun support.

- ZIONSECURITY, an expert on web application security, performed a thorough audit of the website and its hosting infrastructure, identifying only one issue with a Drupal module, which was already being patched by the module maintainer. The conclusion of this audit was that both Drupal and Platform.sh ensure a very high level of IT security that is expected for any Enterprise level application.

What are the conclusions of the audit?

- During the audit we discovered one critical vulnerability and several low vulnerabilities. The critical vulnerability was a bug in the payment module with a high impact, which allowed our team to order products from the store without payment. Due to the nature of the bug someone from our team contacted the maintainers and they patched in a fix immediately.

- Overall Drupal CMS can be seen as a mature framework from an application security point-of-view. The extensive developer and business community behind it makes sure that possible bugs in new modules are rapidly cleaned up with regular patches and updates.

- Attackers or malicious robots cannot easily bypass custom modules if a correct integration with the core has been made.

- A lot of widely used contributed modules are watched by the Drupal security team and regularly tested for vulnerabilities. Stable releases of these modules are flagged as secure after they are tested.

- Hosting the site on a modern cloud-based PaaS provider, Platform.sh, using the Microsoft Azure Germany cloud ("German Sovereign Hosting") added an additional layer of security, both in terms of technical infrastructure and governance, by ensuring that the hosting service was always up-to-date with security patches, properly configured and secured, and resistant to common attack vectors.

- Finding a vulnerability in a Drupal module is not itself a major problem, in fact it is actually a good outcome. All software contains bugs, and it is processes like this which resolve those issues. It is the very large volume of auditing or testing processes (like this audit) which help large open source projects such as Drupal achieve high levels of security, and thus suitability for Enterprise use.

2

INTRODUCTION

Introduction

Drupal is the third most used open-source CMS platform in the world and is used by at least 5% of all websites on the internet. As with all software products and frameworks, security concerns present themselves and Drupal users constantly discover and resolve bugs and vulnerabilities.

Who will enjoy this white paper? Anyone interested in understanding Drupal's security model as part of an evaluation of the framework before using it in important projects.

As Drupal matures, organizations such as NATO, The White House, and the European Commission start using the platform. Many CIO's have questions about how Drupal fits within their existing security policies.

This white paper consists of 3 large sections:

- Read more about the Drupal setup we created for the audit and the way we performed the audit. The Drupal setup is based on Drupal 8, Drupal Commerce modules, Platform.sh, and Microsoft Azure hosting.

- Discover the results of the independent audit.

- Get more context about Drupal's approach to security and you find out everything about the security review process in the Drupal community. Moreover, we share to you our best practices to implement Drupal securely.

At the end of this white paper, you find additional references for further reading.

3

AUDIT PREPARATIONS

Scope & website setup

Audit scope

We chose Drupal in combination with Commerce, to set up a simple webshop to provide a realistic testing environment. Webshops ideally require an extra layer of security to protect any customer-information and order-related information.

We then added some often-used contributed modules because you can't build a real Drupal website without using some contributed functionality. These modules add to the complexity and realism and can also contain possible vulnerabilities when mixed. This is something Drupal agencies should always take into account.

Drupal 8 & commerce modules

We opted for Drupal 8 installation on a Platform.sh infrastructure. We also set-up
basic configuration for the Commerce module and enabled the PayPal payment gateway.

The PayPal payment gateway requires a secret token to be passed to it via the Drupal application, and this in turn was a good opportunity to challenge the security auditors. Their goal was to crack the site and to discover the secret PayPal token.

Find here a complete list of all modules used.

Infrastructure

The site was deployed onto a Platform.sh Standard plan. The websad it is running PHP 7.1 via PHP-FPM. In addition, MariaDB 10.0 and Solr 4.10 were deployed.

The configuration of the project adhered to the reference repository for Drupal 8 that is maintained on the Platform.sh Github account.

For this test, Microsoft Azure's German Cloud provided the underlying
infrastructure. This region was chosen specifically because organizations that are keenly interested in application security may also be interested in the sovereign German hosting offer that the Azure German Cloud represents.

As it is owned and operated by T-Systems, the largest German integrator, it represents a 100% European offering with no legal uncertainties regarding data governance that sometimes arise from the use of non-European Cloud providers.

How we tested

ZIONSECURITY has developed a specific methodology to assess the security level of organizations in a uniform and consistent way. Our methodology is a combination of our know-how obtained by executing security tests for various types of organizations and is based on open standards including:

- The Open Web Application Security Project
- The Open Source Security Testing Methodology Manual
- The ISO 27000 standard

A penetration test or web application security assessment at ZIONSECURITY has a fixed structure in order to ensure a solid methodology and a consistent quality of the reported results. The tooling landscape and technologies used are our expertise and should be considered at all times as state-of-the-art and continuously evaluated towards industry standards, technological innovations, zero-day exploits etc.

Step 1: threat modeling

Threat modeling is a lightweight approach to assessing the risk and exposure of an organization. We need to identify the assets that are important for your business and how these assets are currently protected.
Assets can be:

- Customers stored in a database
- A webmail application
- An Extranet application
- Your wireless network

Step 2: reconnaissance

Passively, we attempt to find out more information about the assets identified in phase 1. Think for example of the screening of public databases, Google, newsgroups and social engineering. This results in possible targets that provide entry points to the assets from phase 1.

Step 3: information gathering

We actively access the targets from phase 2 and learn more about how the assets are protected. We detect all accessible ports, services, and applications running on the targets.

Step 4: vulnerability detection and analysis

We identify possible vulnerabilities focussing on the OWASP top 10. Some of these vulnerabilities are found via automated tools such as web application scanners. Other require a manual approach or a code review. Whenever third party components are used, the version is determined which might contain known vulnerabilities.

Step 5: the attack

We exploit the vulnerabilities from phase 4 without invading the integrity of the targets. The exploitation step allows demonstrating the impact of the discovered vulnerabilities.

Step 6: reporting

Each vulnerability is documented in a vulnerability report with the corresponding impact. This report includes the steps required to reproduce the exploit. We propose countermeasures to solve or to mitigate the vulnerability in a cost-efficient way. A presentation to the project sponsor is foreseen to finalize the security test.

4

AUDIT RESULTS

Scope

This particular security assessment by ZIONSECURITY is a white box penetration test where all components are open source. A code review was not in scope, but some information could be found in the source code and known issues.

Threat modeling and reconnaissance are not relevant for this Drupal 8 case as this is an example application without an organization behind it which would normally be in scope.

Conclusions

This report details one high vulnerability and several low vulnerabilities. Leaving these vulnerabilities unpatched exposes the web application to several business risks.

The most important risk is, of course, the fact that payments can be bypassed. This is a known vulnerability in the PayPal Commerce module.

After investigation, we learned that the Drupal developers put on hold the fixing of the issue due to a dependency with another issue in another module.

The reason for this delay was caused mainly by usability concerns, risking to be forgotten.

However, due to our intervention on drupal.org, the developers now have been convinced to push a preliminary patch due to the criticality of the issue. The patch should be included in the next release to make exploitation impossible.

The other issues are of minor importance and can provide additional lines of defense if implemented correctly.

The administrator account could not be compromised and the PayPal API credentials weren't obtained due to several defense mechanisms which were correctly implemented.

Business risks (A non-exhaustive list)

- A malicious actor can bypass the actual payment and order items from the shop without ever paying for it.
- A captcha measure that is not implemented correctly leaves the application vulnerable to automatic attacks from scanners, bots or scripts written by attackers.
- User enumeration allows an attacker to build a list of users that are active on the application. This information can, in turn, be used to set up a highly-targeted phishing attack that can lead to a user account being compromised.
- A session with a long or indefinite lifetime poses a security risk. If this session is hijacked or compromised in any way, it will take a long time for the session to be invalidated. This leaves a considered timing window for an attacker to abuse the user's account.

Comparison with the OWASP Top 10 Web Security Risks

SECURITY RISK	DESCRIPTION	POSSIBLE
A1 - Injection	Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.	No
A2 - Broken Authentication and Session Management	Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities.	No
A3 - Cross Site Scripting (XSS)	XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.	No
A4 - Insecure Direct Object Reference	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.	No
A5 - Security Misconfiguration	Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, framework, and custom code.	**Yes**

A6 - Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.	No
A7 - Missing Function Level Access Control	Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.	No
A8 - Cross Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim’s browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim’s browser to perform a hostile action to the benefit of the attacker.	No
A9 - Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.	**Yes**
A10 - Unvalidated Redirects and Forwards	Applications frequently redirect users to other pages, or use internal forwards in a similar manner. Sometimes the target page is specified in an unvalidated parameter, allowing attackers to choose the destination page.	No

This OWASP top 10 table represents a broad consensus about what the most critical application security flaws are.
Summary of findings:

RISK	TITLE
High	Payment bypass
Low	Session expiration
Low	reCAPTCHA not validated
Low	Missing Cookie Flags
Low	User Enumeration

5

LOOKING AT DRUPAL SECURITY

Drupal Security out of the box

Drupal 8 implements measures against various types of attacks, such as XSS, CSRF, and SQL Injection. Developers are encouraged to use a set of secure API's, such as the Database Abstraction layer to avoid any mishaps or vulnerabilities, thereby minimizing exposure to attacks. Below you can find a list of measures being taken against some of the more popular attack methods.

XSS attacks or Cross-site scripting attacks abuse the possibility to present malicious client-side scripts to the end-user. By sanitizing and/or escaping the input from forms and such, Drupal avoids the exploitation of these malicious scripts. The sanitation of various data is not done automatically, and is a responsibility of the developers of the Drupal community. Failing to do so leads to bugs and vulnerabilities, for which a patch is to be released as soon as possible.

Executing server side code is practically impossible in a default Drupal 8 setup, and is only exposed by using contributed modules which do not sanitize input properly. Drupal 8 also removed the PHP core module to avoid any misuse.

SQL Injection vulnerabilities are avoided by advocating the use of the Database Abstraction layer, which escapes and sanitizes any variable information which is sent to the Database layer. Attackers won't be able to insert any malicious parts of queries because of this.

Session hijacking or bypassing security is very hard when a Drupal 8 website is served under a HTTPS connection. Also, Session ID's are now stored as a hash in the database, so it's impossible to use the data in case of a breach.

The Twig theme engine avoids bad practices by using server side code in front-end templates, and implements an auto-escaping technique when rendering the output. This ensures security in the front-end layer of the application.

Trusted Host Patterns are used to avoid hijacking of the HTTP Host directive and can be set up very easily. This allows the application only to be accessed using one of the mentioned hostnames.

Improved user passwords by stretching the passwords even more than Drupal 7, making it harder and more time-consuming to brute-force any password.

Mixed mode SSL was removed in Drupal 8 because of the complexities this brought to the table. Instead, people are encouraged to serve their website on an HTTPS-only connection, and the necessary functionality was introduced to the core codebase, therefore removing the need for any contributed module.

Platform.sh security

Service Isolation:

each service for the Drupal 8 site, including PHP, MySQL, and Solr, was deployed in an isolated LXC container. The containers themselves are bound by networking firewalls only to be able to communicate with each other. This provides a level of security against attacks where a breach in one service would automatically lead to an attacker controlling the other services.

Read-only file system:

the application code on Platform.sh is deployed on a read-only file system. This is an effective safeguard against attacks that depend on the attacker adding or modifying PHP files on the server's file system [eg. SA-CORE-2017-001].

Protective Block:

the Drupal application was also protected by Platform.sh's Protective Block, which is able to identify a number of specific application vulnerabilities, such as the infamous Drupalgeddon exploit. It also prevents those vulnerabilities from being exploited on running applications, and blocks the deployment of applications that are deemed vulnerable.

Environmental Variable Management:

the challenge posed to the security analysts, namely the recovery of the PayPal secret token, was made more challenging by the use of Platform.sh environment variable management.

Rather than embedding the PayPal secret token in a file or in code that is managed by Git, the Drupal 8 site received the variable from a vault managed by Platform.sh that is only visible to the PHP-FPM runtime environment.

This frees developers to call on the variable as needed, but for it to be overridden on other environments, such as Development, Test, or Stage environments. There, the variable can be overridden to provide the PayPal sandbox token.

Drupal community security process

The Drupal community has a mature process to resolve any discovered vulnerabilities and is described step-by-step below:

- A vulnerability is discovered by anyone in the Drupal community.
- A separate private issue is reported to the Security team to avoid exposure.
- The issue is reviewed and evaluated on the potential impact on all supported Drupal releases.
- If the issue is valid and poses any security risk, the Security team is mobilized for further analysis of the problem. Maintainers of both Core and Contributed modules are being notified for action.
- The maintainer fixes the issue and receives the necessary support from the Security team.
- The provided fixes are being reviewed and discussed.
- Code patches are created and tested, both manually and automatically by the CI system.
- New, fixed versions of core or contrib are being released and made available on Drupal.org.
- A Security Advisory is written and published via the website, social media, RSS, etc.

The Drupal Security Team

The Drupal Security team is a global group of some of the world's leading web security experts, and is always on-call to assess, evaluate and respond to issues affecting Drupal's security. A great deal of skill, knowledge, and experience goes into making Drupal as secure as possible.

Follow up on the Paypal module security issue

During testing, the modules used by the website were enumerated. Since the platform is set up specifically for this test, we can assume the latest version is installed, so we looked for unfixed and known vulnerabilities which have a security impact.

Security sensitive bugs should not be accessible to the public but only to the security team. However, there was one bug in the PayPal commerce module that had a high security impact. This issue is known and a solution was developed, but the patch was not included in the next release because the solution would lead to a WSOD (White Screen Of Death or blank page) if payment validation would fail.

A dependency with another module existed to allow for the payment module to fail gracefully with a proper message to the user. In practice, this WSOD would only appear if the payment wasn't signed correctly by PayPal or if someone manipulated the request. During normal operation, such a WSOD should not appear.

If an implementation or configuration bug exists or if an attack would manipulate requests, the WSOD would appear and would effectively stop the payment transaction. However, the development team opted for the good-looking and less secure option of disabling the digital signature validation.

The ZIONSECURITY test team reminded the development team of the severeness of the issue. Within hours, the team responded and changed the issue status to Fixed indicating the patch would be included in the next release.

Best practices for securing Drupal 8

The key to security is eternal vigilance. Updating code, both within Drupal and across your hosting infrastructure, is a necessary process to ensure you stay secure. Setting up a secure Drupal web application server and walking away is not sufficient.

Be aware of the update process for your systems (The Drupal Security Team releases Security Updates each Wednesday), and ensure someone is keeping on top of this, with sufficient time allocated to perform updates to Drupal, your web server software, database software, and all other packages installed on your systems.

Below you can find a number of steps that we include in every project:

- We use best-practice, automated infrastructure installations, which are regularly updated. This ensures state-of-the-art (web) hosting platforms (no pun intended), so we can focus on the application rather than the infrastructure.
- Automated deployments with a Continuous Integration pipeline avoid manual interaction with any of the project's environment, mitigating risks of accidental exposure to vulnerabilities.
- We update and maintain our Drupal and modules regularly. This is in most cases the cause of exploitation of a website.
- Our initial, basic installations are all done automatically using security best-practice standards.
- We use the Security Review module to estimate/scan sources of exploitation on our websites.
- All our (website) administration accounts use random generated, strong passwords. If possible we try to use Two-factor auths or key-based verification. We can also add a layer of IP-based or password restrictions.
- We advise and try to enforce usage of HTTPS connections and SFTP connections.
- We implement various security enhancing modules such as:
  - Captcha/Recaptcha
  - Login security (limit number of login attempts)
  - Password policy (to enforce strong passwords)
- We carefully do a review of our web server's filesystem-permissions to avoid accidental write-permissions.
- We use web server directives to protect sensitive files such as update or installation scripts. (.htaccess/.htpasswd)
- We use various HTTP security measures to mitigate attacks and security vulnerabilities on a web server level:
  - Content Security Policy
  - X-XSS Protection
  - HTTP Strict Transport Security
  - X-Frame-Options
  - X-Content-Type
- We can add custom layers of security in the application itself such as Username Enumeration prevention, removing sensitive data in Drupal's TXT files (like Changelog and Readme), and add a layer of antivirus/ malware on server level.

6

ABOUT THE AUTHORS

One Agency

ONE Agency is a Belgian web agency that offers tailor-made solutions to customers who are looking for a partner to realize their web projects, going from websites and multi-sites to web applications, platforms, and digital marketing automation campaigns.

The ONE Agency team consists of more than 130 passionate PHP, Drupal and Java developers and experts in user experience, graphic design, e-marketing and system management. We have over 10 years of experience in realizing web projects and are known as a long-term partner to a lot of our clients.

ONE Agency is part of the full-service ICT company AUSY Belgium, that offers a wide range of innovative and custom-tailored IT services to customers in both the public and private sector. With offices in Leuven, Ghent, and Brussels, AUSY Belgium employs over 580 people and is active for clients such as Proximus, Telenet, the Flemish Government, Doctors Without Borders, the European Commission, Eandis, bpost, ASTRID, and NN Insurance Belgium.

Next to the project-driven approach by ONE Agency, AUSY Belgium offers IT staffing services through the DataFlow brand. As an IT consultancy and staffing company, DataFlow is the AUSY Belgium brand that staffs experienced consultants in-house with customers, to work on different kinds of IT projects and support assignments. DataFlow has been active since 1999 and has built an extensive, international customer portfolio thanks to its qualitative service and human approach to IT.
Some of DataFlow's customers are Proximus and BNP Paribas in Belgium, and Johnson & Johnson, Nike and the European Commission on an international level. DataFlow has over 450 consultants and is active in a wide range of sectors (IT, telecom, finance, industry, pharma, public).

www.one-agency.be
www.dataflow.be

Platform.sh

Platform.sh is a continuous deployment cloud hosting solution for web applications.

It delivers not only screaming fast performance and robust high-availability, but also eliminates testing bottlenecks and provides fail-proof deployments thanks to its unique feature: the full production cluster, containing all its data, can be cloned under a minute into a staging environment for every new feature developed. Every git branch, every pull-request gets an automatically generated ephemeral staging environment that allows for both automated and user-acceptance testing.

Serving the web application market, large e-commerce sites and media sites, it is the official cloud partner of major open source projects such as Symfony and Drupal Commerce, eZ Platform and TYPO3. In 2016, Platform.sh was chosen to operate the flagship Magento Enterprise Cloud Edition as well as Sensio.Cloud by the creators of the widely used Symfony framework. It has also signed exclusive deals with Orange Business Services and Microsoft Germany to offer the first sovereign PaaS solution to European customers.

Platform.sh's fully automated cloud solutions cover the needs of small self-service accounts but can scale to tens of millions of users and power multiple, dedicated cloud regions running tens of thousands of instances over multiple IaaS providers.

24/7 follow-the-sun support combined with a unique, triple-redundant architecture based on a high density grid of Linux micro-containers allow the service to propose 99.99% SLAs on even the most complex use-cases.

Platform.sh is a VC-backed startup headquartered in Paris, with employees across five continents. In its two years of activity it has seen explosive growth acquiring thousands of clients from more than a hundred countries, notably in the United States and Europe. Among its key client accounts, you can find Vivienne Westwood, Reiss, the Canadian Football League, the British Council, Parc Asterix, Seloger.com, Flixbus, and El Universo.

www.platform.sh

ZIONSECURITY
PREVENT-PROTECT-DETECT-ACT

ZIONSECURITY

ZIONSECURITY is market leader and well-known specialist in protecting web applications and web users. We are a young and dynamic team consisting of a mix of web application and network security experts. This mix enables us to deliver high-quality custom projects to small, medium and large organizations.

The vision of ZIONSECURITY is to secure software, either running in the cloud or on-premise. It is our mission to help our customers to stay secure and help them evolve securely with new technologies. We help to protect you from criminals that want to hack your applications and offer corporate users a safe way to use the internet protecting them against malware, ransomware, and botnets.

Next to pure penetration testing by ZIONLABS on infrastructure, (web)applications, mobile apps or IoT devices, we offer Secure Development Life Cycle-tracks with ZIONSDLC. ZIONVERIFIED offers managed services on vulnerability scanning, and with ZIONSECURED we foresee cloud-based DDOS-protection programs or even intrusion detection using real- time monitoring solutions.

Our strategic solutions from ZIONSTRATEGY start from high-level audits and secure workflow assessments up until user awareness programs or even social engineering campaigns. Within ZIONUNIVERSITY we offer a variety of training like CISSP, CSSLP, CSA and Ethical Hacking Workshops (e.g. Troy Hunt).

www.zionsecurity.com

References & Additional Reading

- https://www.keycdn.com/blog/drupal-security
- https://www.keycdn.com/blog/http-security-headers
- http://www.cameronandwilding.com/blog/pablo/10-most-critical-drupal-security-risks
- https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf
- http://2015.pnwdrupalsummit.org/sites/default/files/slides/Site-Security-In-Drupal-8-PNWDS-2015.pdf
- https://www.drupal.org/docs/8/security
- http://drupal.org/writing-secure-code
- Points of contact with the Drupal security team:
  - Releases at http://drupal.org/security
  - Reporting issues: http://drupal.org/node/101494
  - Reporting cracked sites: http://drupal.org/node/213320