How open source helps you prevent the next Drupalgeddon

the best marketing for this talk was SA-CORE-2018-003 and SA-CORE-2018-004

Drupal Hack Camp 2018
Bastian Widmer - @dasrecht | @amazeeio

Bună seara!


Clădire expertizată tehnic
încadrată
în clasa I risc seismic
BÁCÁ
PARCARE DE
REZIDENȚĂ CU
PLATĂ

We will talk about: basics, containers, open source, crypto currencies, attacks and the future

$> whoami bastian

- System Engineer at amazee.io
- Containers in Production 👻🤖
- Zurich, Switzerland
- English, German, Swiss-German and a bit of French
- @dasrecht
- Too many side projects!*
    - TEDxBern
    - DevOpsDays Zurich
    - CommunityRack.org
    - Running TOR nodes for fun
    - Working with real containers

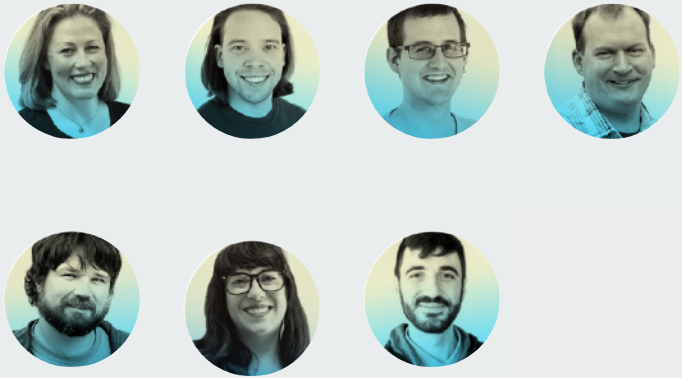
* this list is not complete!

$>


Condecta
Condecta
Condecta
Condecta
1633
Toggenburger


Condecta
Condecta

amazee.io



- Fully Open Sourced Hosting Platform for ~~Drupal~~ Web Projects
- Hosting since 8 years
- We're a remote team of 7
  - Zurich, Switzerland
  - Barcelona, Spain
  - Austin, TX
  - Portland, OR
  - Melbourne
- Hosting in 16 different countries

There are two types of companies: those that have been hacked, and those who don't know they have been hacked.

— John T. Chambers

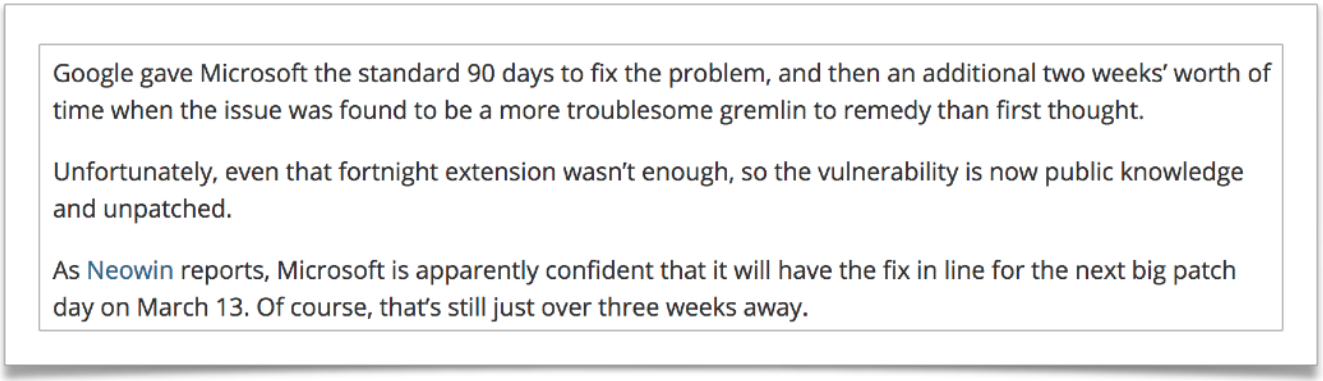
Is open source better compared to closed source?

Opensource

- Auditable by everyone
- The power of many eyes
- Fixes can be found by a bigger team

Closed Source

- You don't know how sustainable a patch is implemented
- you need to trust the vendor completely
- e.g. Microsofts Edge Browser misses to patch a vulnerability after 90 days and 2 weeks



Google gave Microsoft the standard 90 days to fix the problem, and then an additional two weeks' worth of time when the issue was found to be a more troublesome gremlin to remedy than first thought.

Unfortunately, even that fortnight extension wasn't enough, so the vulnerability is now public knowledge and unpatched.

As Neowin reports, Microsoft is apparently confident that it will have the fix in line for the next big patch day on March 13. Of course, that's still just over three weeks away.

That said...

- No evidence that Open source performs better than Closed source
- Transparency of open source is still better
- Nothing is inherently secure

- Heartbleed, Poodle. Shellshock
- CVE-2008-4250 Sasser/Conficker patches were not applied for a long time

Basics: Security Levels

Security Levels

- scores between 0 and 4 are considered Not Critical
- 5 to 9 is considered Less Critical
- 10 to 14 is considered Moderately Critical
- 15 to 19 is considered Critical
- 20 to 25 is considered Highly Critical

https://www.drupal.org/drupal-security-team/security-risk-levels-defined

Risk Metrics

- Access Complexity (AC)
- Authentication (A)
- Confidentiality Impact (CI)
- Integrity Impact (II)
- Exploit (Zero Day Impact) (E)
- Target Distribution (TD)

Basics: Drupal Security Process

How do you feel on Wednesday evenings?

Drupal Security Process

- Releases every Wednesday
- Public Service Announcements (PSA) for high security levels

Drupal Security Process

- Issues are reported to the security team via a hidden issue queue
- If the problem is valid the security team mobilises the maintainer to fix the issue
- New versions are created, reviewed and tested
- New release is created on drupal.org
- Communication channels are used to inform users about the upgrade steps to protect themselves
- If the maintainer fails to fix the issue within the deadline an advisory is issued to disable the module and the module is marked as unsupported.

Disclosure policy

- Coordinated Disclosure policy
- issues are private until there is a fix OR
- until it becomes apparent that the maintainer is not addressing the issue in time

- Public announcements are made after the threat is addressed and a secure version is available

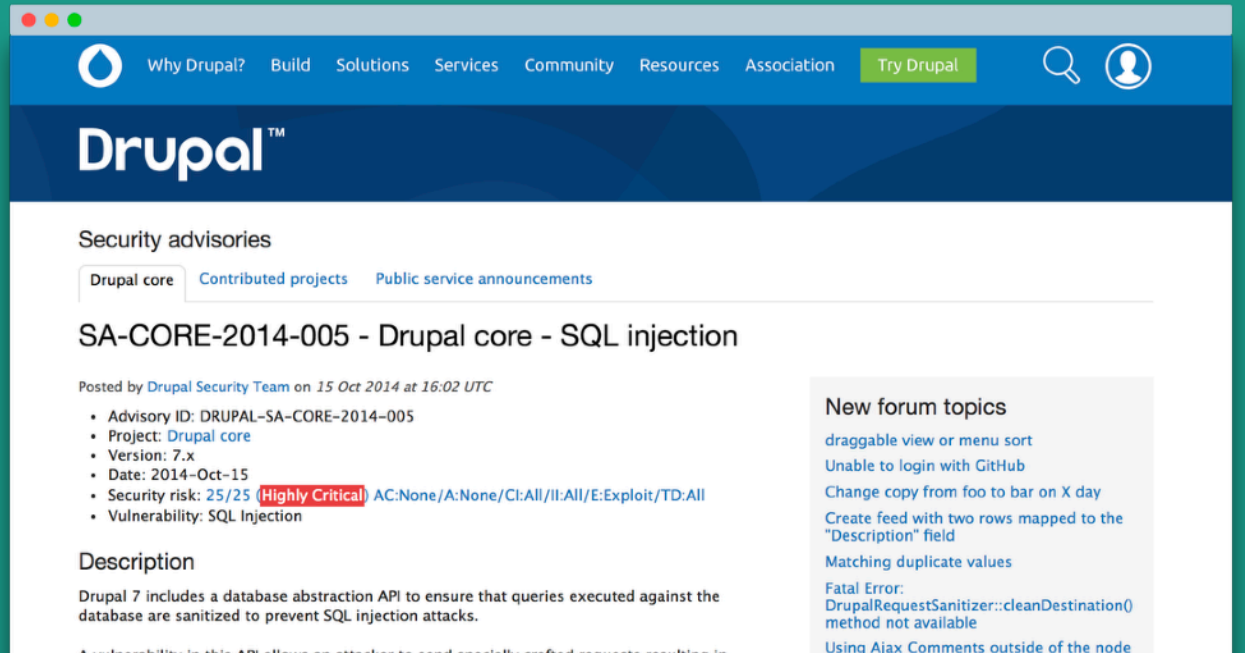
- The same goes for issue reporters

Back in the day™

Back in the day™
aka
2014

DrupalGeddon 1.0

DrupalGeddon 1.0



25/25 ? SHIT!

Drupalgeddon 1.0 - SA-CORE-2014-005

- SQL Injection
- Score 25/25
- 7 Hours from release till attacks were rolling in
- Defacements, Backdoors, Mass Mailing

https://www.drupal.org/forum/newsletters/security-advisories-for-drupal-core/2014-10-15/sa-core-2014-005-drupal-core-sql

DrupalGeddon 2.x

The good news first!

The good news first:
You are not important anymore!

The good news first:
You are not important anymore!
Your Infrastructure is!

The bad news?

The bad news:
You don't get 7 hours anymore

Drupalgeddon 2.0 - SA-CORE-2018-002/004

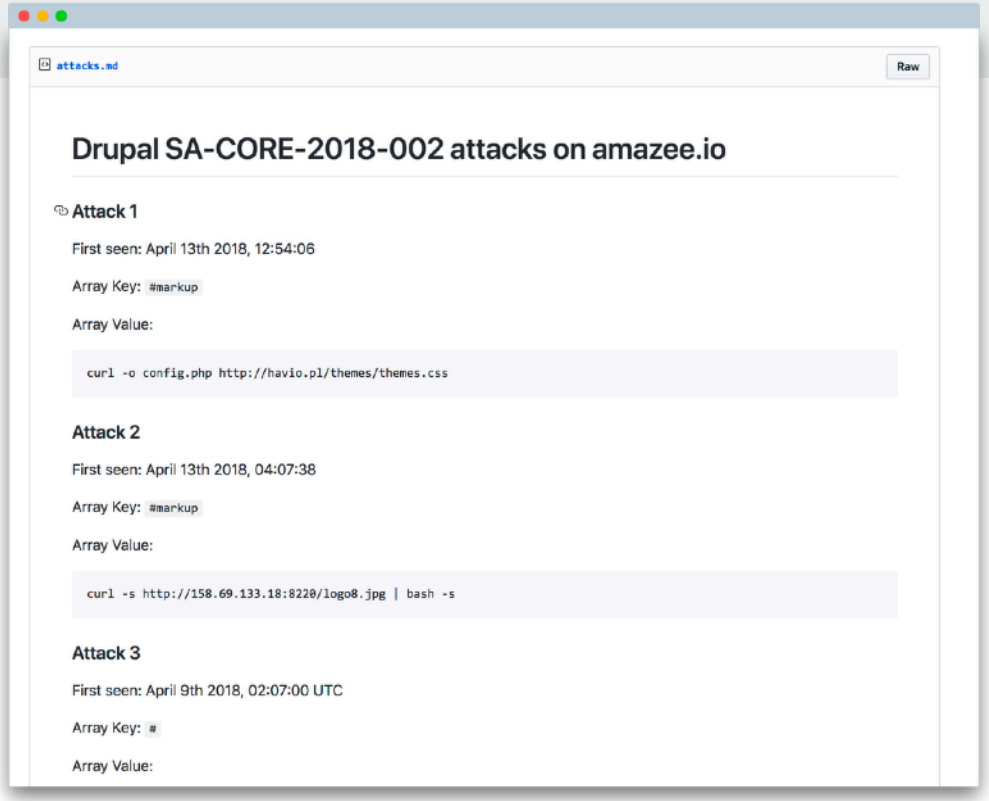
- Non sanitised values
- Score 24/25 / and 20/25
- several hours after exploit was in the wild

Timeline

- SA-CORE-2018-002 released March, 28 2018
- Exploit in the wild: April 12, 2018
- Currently 2000-5000 attempts per day overall
- Other players mitigating over 500’000 attempts per day
- SA-CORE-2018-004 released April, 25 2018

What kind of attacks?

- Nothing „too visible“ for the end user
- Full Stack attack - The user and your server
- Cryptominer JS Inclusions
- Cryptominers on the Server (Cryptojacking)
- Stealing your useraccounts/mail addresses
- Data breaches (GDPR/DSGVO!)



- https://twitter.com/Schnitzel/status/984875838410813440
- https://gist.github.com/Schnitzel/684519cbf268481ac3f9d8cee249efeb

Security is a process not a state

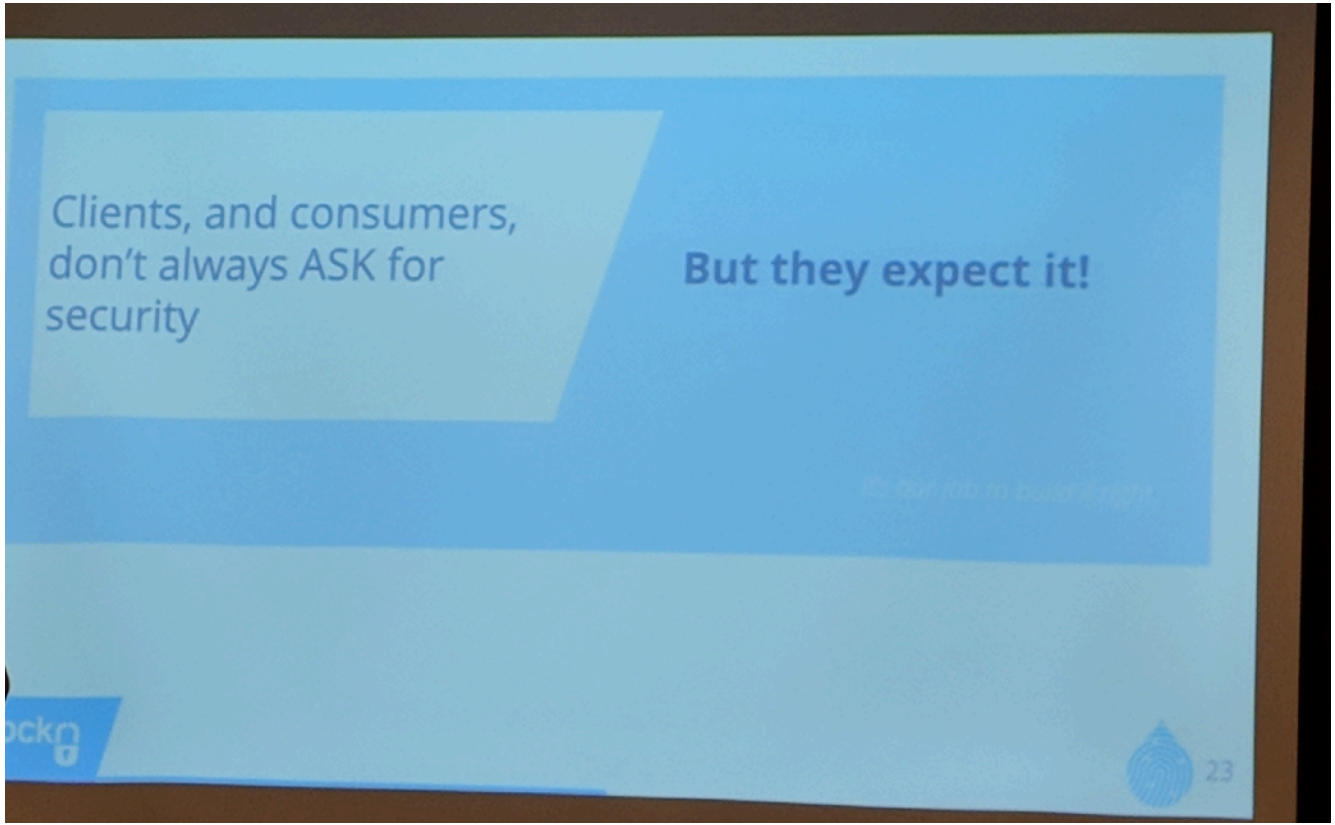
What layers of security do can we deploy?

- Regular Updates
- Drupal Modules
- Web Application Firewall (WAF)
- Hoster / Infrastructure
- Code-level
- Your own measures

Regular Updates

Regular Updates

- Update every week
- at least: Security Related (situative awareness)
- It’s a product - Sell it to your customers
- Unpatched CMS can lead to leaks like:
- Panama Papers - 2.6 TB worth of Data leaked
- Equifax Leak 143 million affected users


Clients, and consumers, don't always ASK for security
But they expect it!
23

BUT I HAVE 100+ SITES!?

Yes! And you're not competing against humans. You are competing against robots!

Security isn't a sprint anymore. It's a marathon (that never ends)

Regular Updates

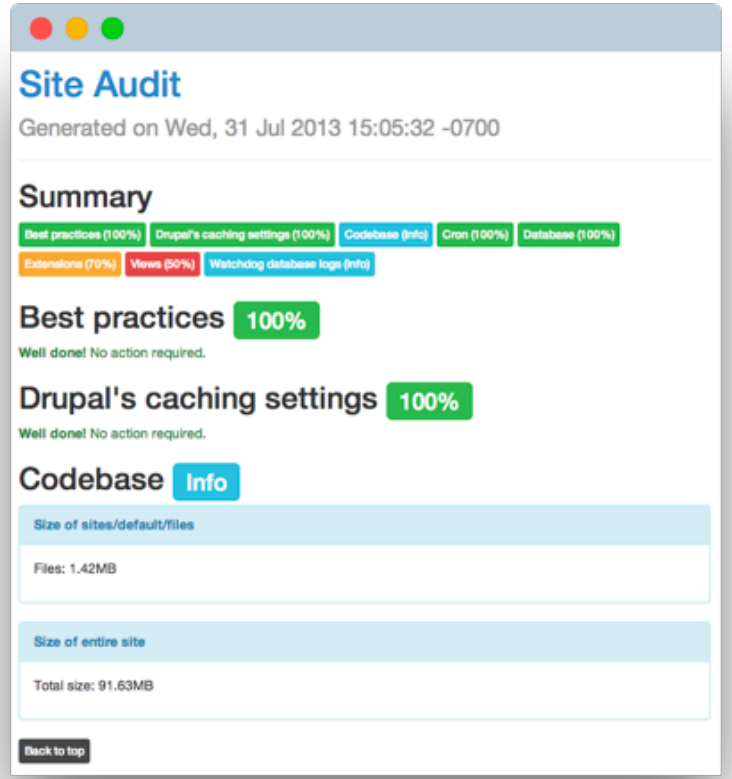
- Automate, Automate, Automate
- DIY - Works but it’s a lot of work
- There should be a fasttrack (just patch and go!)
- Use a „appropriate“ Development workflow: Source Control, Composer
- Dropguard - https://www.drop-guard.net/
- Lumtrio - https://lumturio.com/

Helpful Drupal Modules

Drupal Modules - Site Audit

Site Audit is a Drupal static site analysis platform that generates reports with actionable best practice recommendations.

https://www.drupal.org/project/site_audit



Drupal Modules - Hacked!

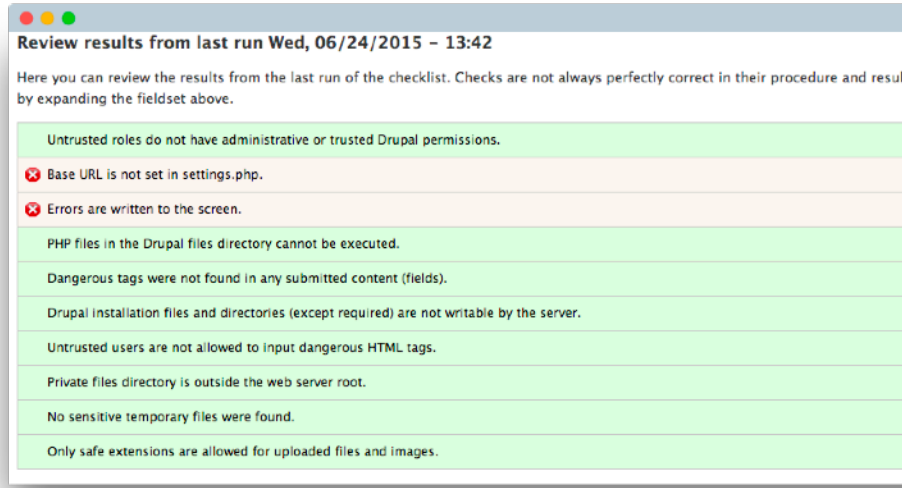
This module scans the currently installed Drupal, contributed modules and themes, re-downloads them and determines if they have been changed.

https://www.drupal.org/project/hacked

Drupal Modules - Security Review

The Security Review module automates testing for many of the easy-to-make mistakes that render your site insecure.

https://www.drupal.org/project/security_review



Drupal Modules - Paranoia

The Paranoia module attempts to identify all the places that a user can evaluate PHP via Drupal's web interface and then block those. It reduces the potential impact of an attacker gaining elevated permission on a Drupal site.

https://www.drupal.org/project/paranoia

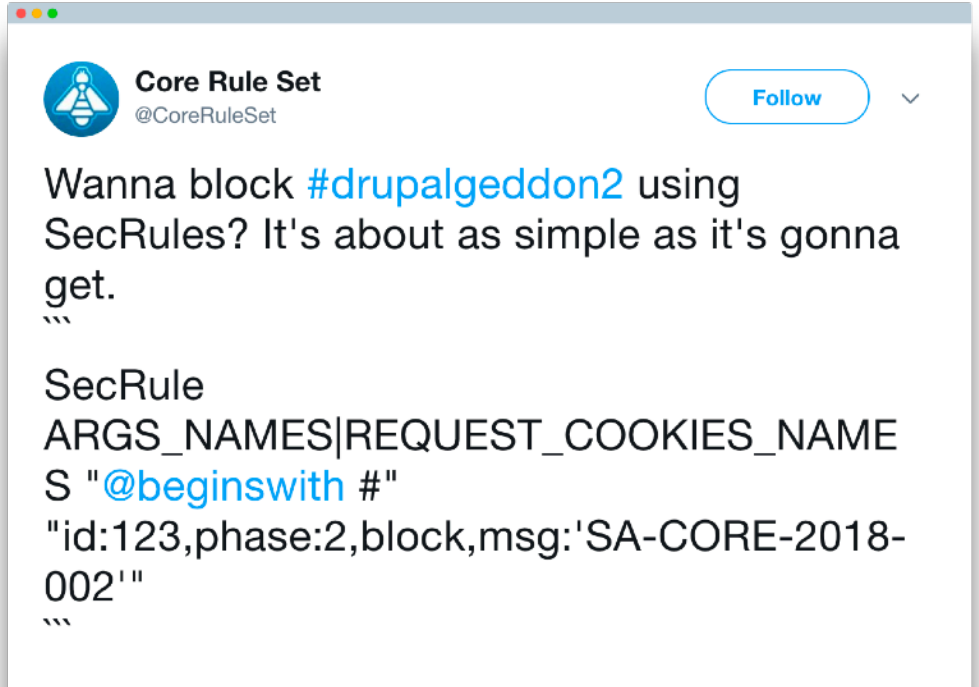
Web Application Firewall (WAF)

Web application firewall (WAF)

- Mod Security (Nginx / Apache)
- Cloudflare (needs Pro Plan)
- Fastly WAF (limited availability release)
- AWS WAF & Trusted Rulesets (F5, Trend Micro, Fortinet)

- Web Application Firewalls **only** buy you time!

Web application firewall (WAF)



Hosting / Infrastructure

Hosting / Infrastructure

- Many providers put mitigations in place to safeguard customers and infrastructure
- Speed is everything!
- Drupalgeddon 2.0 most of the bigger providers implemented infrastructure level mitigations within an hour after the Security release
- This still does not mean that you won't need to patch your site

Hosting / Infrastructure

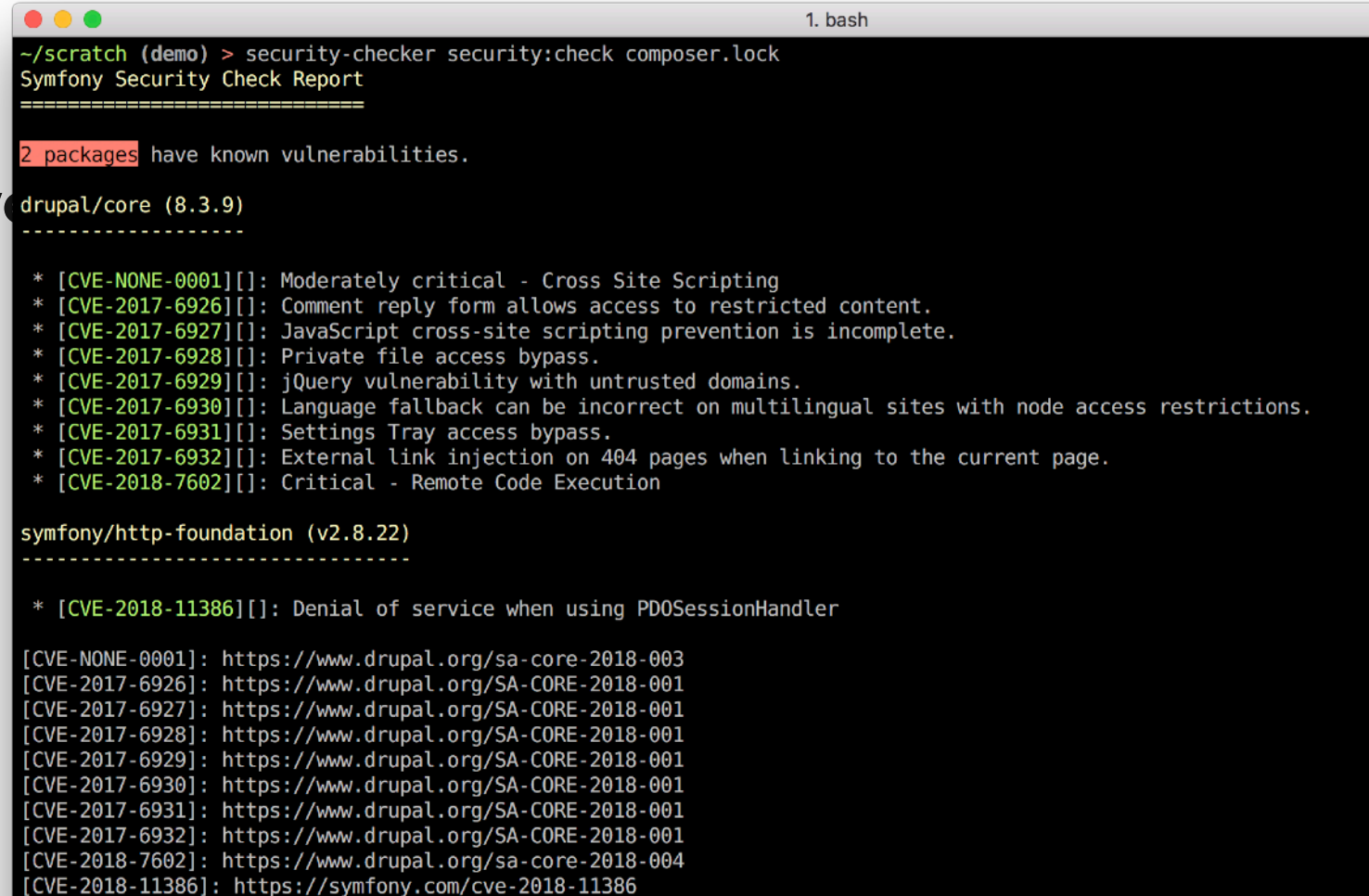
- Environment variables make it easy to rollover the remaining secrets
- Hardening on webserver level - i.e. only allow index.php requests.
- and whitelist where necessary
- Containers / Don’t have any changeable code deployed
- DockerHub Security Scanning https://blog.docker.com/2016/05/docker-security-scanning/

Code-level

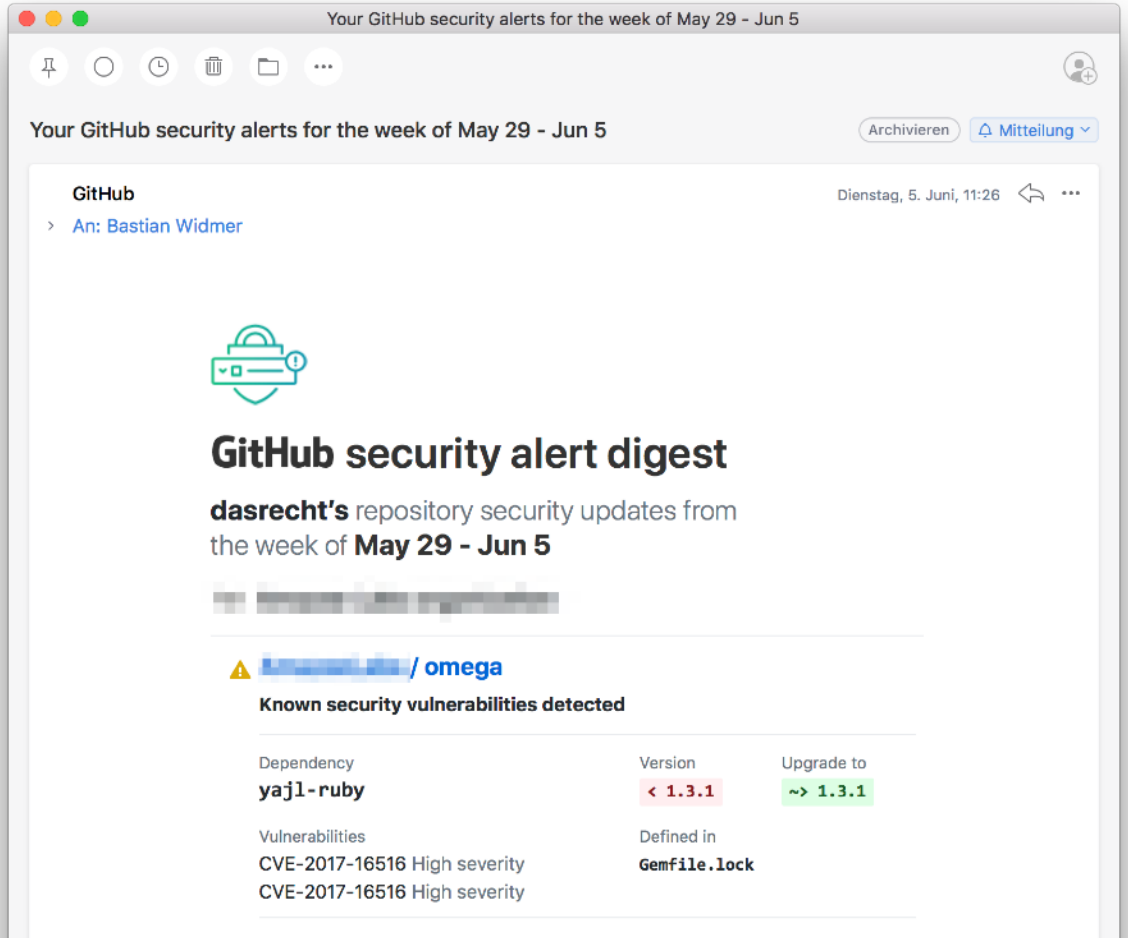
Code-level

- Remove inactive modules - Less attack surface
- Github Security Scans
- Composer Security Scan (https://security.sensiolabs.org/check)

Code-lev


1. bash
~/scratch (demo) > security-checker security:check composer.lock
Symfony Security Check Report
2 packages have known vulnerabilities.
drupal/core (8.3.9)
* [CVE-NONE-0001][]: Moderately critical - Cross Site Scripting
* [CVE-2017-6926][]: Comment reply form allows access to restricted content.
* [CVE-2017-6927][]: JavaScript cross-site scripting prevention is incomplete.
* [CVE-2017-6928][]: Private file access bypass.
* [CVE-2017-6929][]: jQuery vulnerability with untrusted domains.
* [CVE-2017-6930][]: Language fallback can be incorrect on multilingual sites with node access restrictions.
* [CVE-2017-6931][]: Settings Tray access bypass.
* [CVE-2017-6932][]: External link injection on 404 pages when linking to the current page.
* [CVE-2018-7602][]: Critical - Remote Code Execution
symfony/http-foundation (v2.8.22)
* [CVE-2018-11386][]: Denial of service when using PDOSessionHandler
[CVE-NONE-0001]: https://www.drupal.org/sa-core-2018-003
[CVE-2017-6926]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6927]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6928]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6929]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6930]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6931]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2017-6932]: https://www.drupal.org/SA-CORE-2018-001
[CVE-2018-7602]: https://www.drupal.org/sa-core-2018-004
[CVE-2018-11386]: https://symfony.com/cve-2018-11386

Code-level


Your GitHub security alerts for the week of May 29 - Jun 5
Your GitHub security alerts for the week of May 29 - Jun 5
Archivieren
Mitteilung
GitHub
Dienstag, 5. Juni, 11:26
An: Bastian Widmer
GitHub security alert digest
dasrecht's repository security updates from the week of May 29 - Jun 5
[redacted] / omega
Known security vulnerabilities detected
Dependency
yajl-ruby
Version
< 1.3.1
Upgrade to
~> 1.3.1
Vulnerabilities
CVE-2017-16516 High severity
CVE-2017-16516 High severity
Defined in
Gemfile.lock

Your own measures

Your own

- Don’t use passwords for server logins - SSH Keys all the way
- Use Single-Sign-On Services if possible
- Use 2 Factor Authentication
- Restrict login to a certain set of IP addresses (Module: Restrict IP)
  https://www.drupal.org/project/restrict_ip

FUTURE

Future

- Automatic Updates Initiative
https://www.drupal.org/project/ideas/issues/2940731
- Self-Patching Infrastructure (i.e. DockerHub)
- It’s a topic that concerns not just Drupal

Conclusion

The fear of the 0 day exploit

The fear of the 0 day exploit Is not real.

Ask yourself: What would you need to change if a Drupalgeddon style vulnerability hits every week.

Conclusions

- WAF only buys you time - You need to keep your code up to update
- Update regularly - and sell it to your customers
- Automate your processes!
- There is no free lunch - You will need to spend money on security
- Have several layers of security - it will pay out in the long run
- It’s not humans that exploit your site - It’s automated bots

Thank you for your attention!

Bastian Widmer - @dasrecht | @amazeeio

Resources

- https://www.drupal.org/forum/newsletters/security-advisories-for-drupal-core/2014-10-15/sa-core-2014-005-drupal-core-sql
- https://research.checkpoint.com/uncovering-drupalgeddon-2/
- https://www.volexity.com/blog/2018/04/16/drupalgeddon-2-profiting-from-mass-exploitation/
- https://www.fastly.com/blog/recent-drupal-vulnerabilities
- https://twitter.com/CoreRuleSet/status/979198633441681408
- https://help.github.com/articles/about-security-alerts-for-vulnerable-dependencies/