DevSecOps: Drupal Security

Drupal Camp Asheville, 2017

Will Chatham

$> whoami

I Am Will Chatham

Security Analyst for ERT at NOAA's NCEI
Ethical Hacker Type

Twitter - @willc
Drupal - geekamongus
Web - willchatham.com
Email - will@willchatham.com

Outline

- ✓ Perspectives
- ✓ DevSecOps - Idealism vs. Reality
- ✓ DevSecOps and Drupal
  - Awareness
  - Configuration
  - Community
  - Other Stuff

From My Perspective

Full Disclosure: *I am not a Drupal developer*

I work with Drupal developers and system admins to facilitate DevSecOps as much as possible

My background is in security, web development and system administration

From The Federal Perspective

- Slow to adopt new technology
- Resistant to cloud-based services (Saas)
- Bound by a robust Risk Management Framework
- We are embracing Open Source more and more

Starting With The Bird’s Eye View

https://youtu.be/AMq-8Ahttps://youtu.be/AMq-8At1yYAt1yYA

Buzzword + Buzzword...

DevOps

- An agile relationship between Development and IT Operations
- Merge people, processes, and tools
- Provide better service and products by integrating these two business units

SecOps

- An agile relationship between IT Operations and IT Security
- Shared accountability, processes, and tools
- Maintain a commitment to uptime and service without sacrificing security

= a Longer Buzzword: DevSecOps

- Achieve greater efficiency and productivity through team collaboration with *security baked in*
- Shift security to be incorporated early and often rather than at the end
- Everyone is responsible for security
- Emphasis on the Proactive

An Ideal DevSecOps World

The DevSecOps Manifesto

Leaning in over Always Saying “No”

Data & Security Science over Fear, Uncertainty and Doubt

Open Contribution & Collaboration over Security-Only Requirements

Consumable Security Services with APIs over Mandated Security Controls & Paperwork

Business Driven Security Scores over Rubber Stamp Security

Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

24x7 Proactive Security Monitoring over Reacting after being Informed of an Incident

Shared Threat Intelligence over Keeping Info to Ourselves

Compliance Operations over Clipboards & Checklists

http://www.devsecops.org/

The DevSecOps Reality

Do you have a personal Red Team
and a 24x7 Monitoring System?

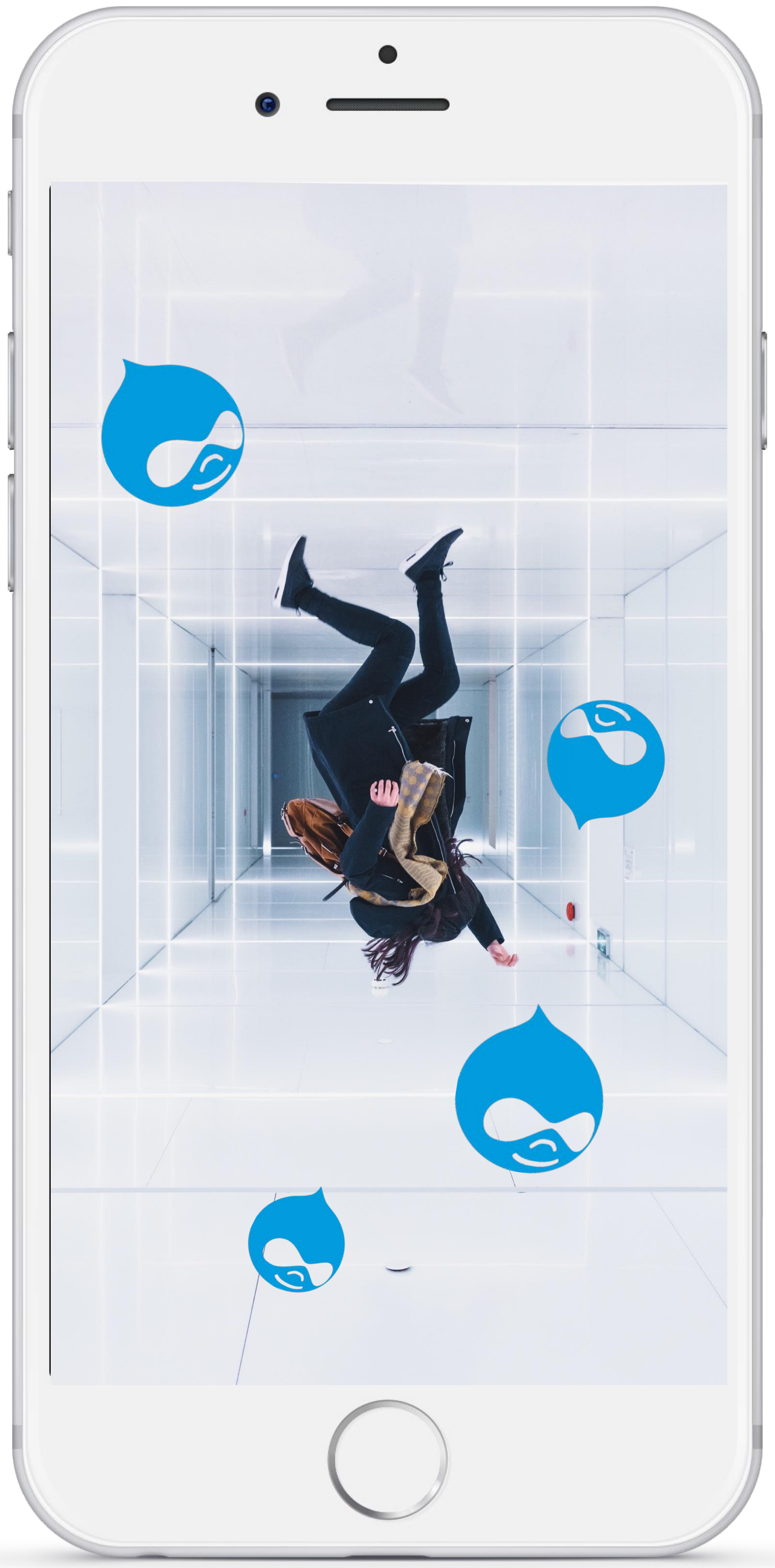
The DevSecOps Reality

Let’s apply what we can where we can...

- **Integrate security early and often - not at the end**
- **Security is Everyone’s Responsibility - developers, sysads, content authors, SEOs, designers, etc**

To reduce the risk where we can...

- **Security is Holistic: from expensive firewalls all the way over to knowing how to spot a dangerous link in an email.**
- **Reduce your Risk Profile**



How does this apply to Drupal?

Three Ways

Awareness

Get to know the risks, the threats, and the ways to mitigate them.

Understand what makes Drupal a target, and what can be done to be a smaller target.

Teach your users and clients all of the above.

Configuration

Know how to configure a secure Drupal website, and make sure it stays that way.

Community

Rely on the mature open source Drupal community to assist you in mitigating risk.

Reduce your risk profile by taking advantage of the Drupal community's expertise, processes, and lessons learned.

Awareness: What Threats?

Awareness: The OWASP Top 10

Application Security Risks - 2017

- ✓ Injection - SQL, XXE, OS, etc
- ✓ Broken Authentication & Session Management
- ✓ XSS - Cross-Site Scripting
- ✓ Broken Access Control
- ✓ Security Misconfiguration
- ✓ Sensitive Data Exposure
- ✓ Insufficient Attack Protection
- ✓ CSRF - Cross-Site Request Forgery
- ✓ Using Components With Known Vulnerabilities
- ✓ Underprotected API’s

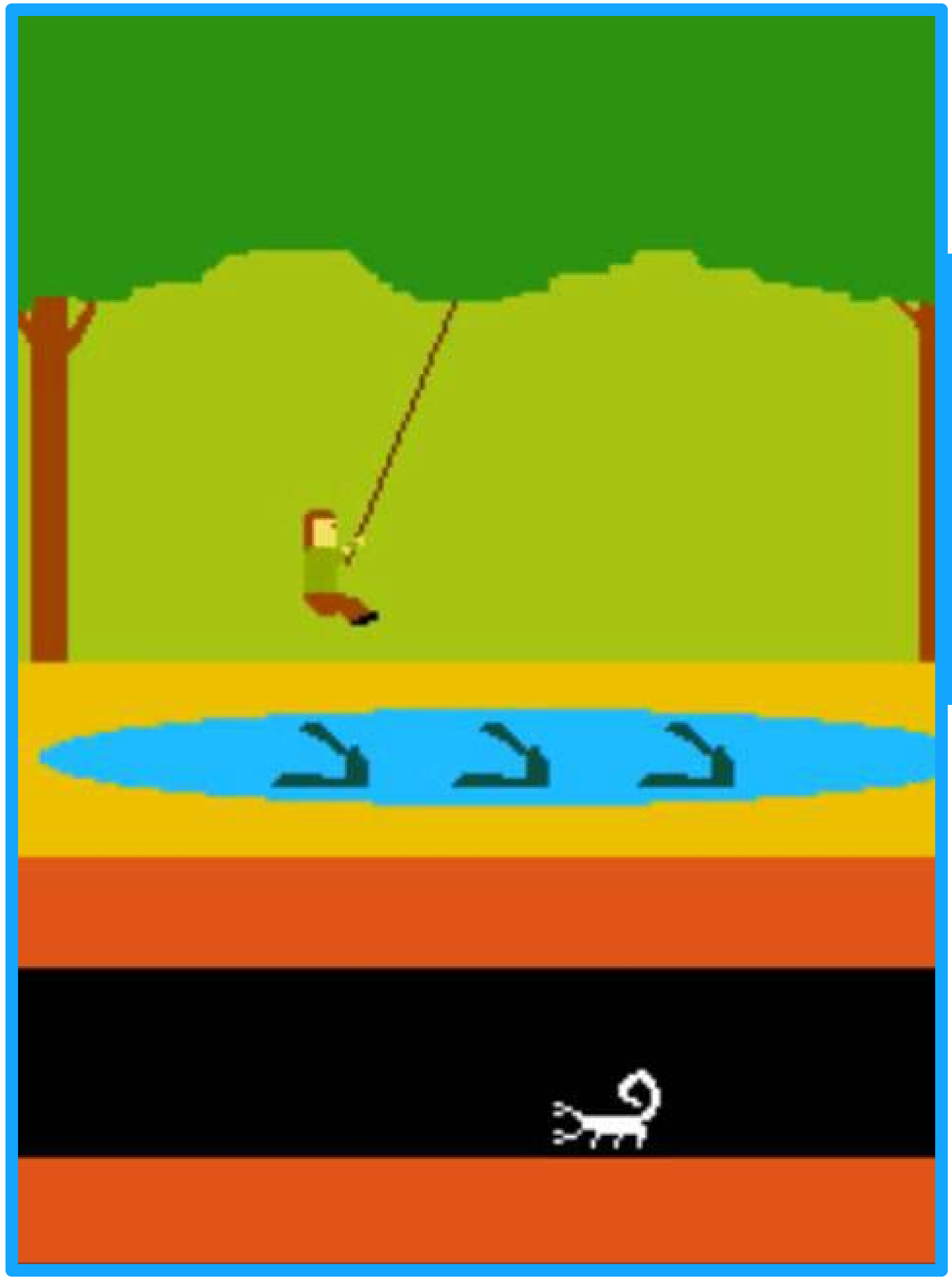
https://www.owasp.org/index.php/Top_10_2017-Top_10

Awareness: The OWASP Top 10

Application Security Risks - 2017

- ✓ Injection
- ✓ Broken Authentication & Session Management
- ✓ XSS - Cross-Site Scripting
- ✓ Broken Access Control
- ✓ Security Misconfiguration
- ✓ Sensitive Data Exposure
- ✓ Insufficient Attack Protection
- ✓ CSRF - Cross-Site Request Forgery
- ✓ Using Components With Known Vulnerabilities
- ✓ Underprotected API’s

OK, great. But what can I do about all this today?



Configuration Pitfalls:

Avoiding Crocodiles and Scorpions

Drupal Security Configuration

Use the Least Privilege Concept
Provide the minimum amount of access necessary to do the job.

General Best Practices

- ✓ Put captchas on forms.
- ✓ Use SSL by default
- ✓ Remove modules and themes you don’t need
- ✓ Update!
- ✓ Subscribe to Drupal Security Advisories

Drupal Security Configuration

Security Modules

Paranoia - scans for and blocks PHP in web interface https://www.drupal.org/project/paranoia

Login Security - limit login attempts, deny by IP
https://www.drupal.org/project/login_security

Password Policy - enforce strong passwords
https://www.drupal.org/project/password_policy

Rename Admin Paths - thwart automated brute force scripts
https://www.drupal.org/project/rename_admin_paths

Security Review - https://www.drupal.org/project/security_review

Drupal Security Configuration

The Security Review Module - (https://www.drupal.org/project/security_review)

Safe file system permissions (protecting against arbitrary code execution)

Text formats don't allow dangerous tags (protecting against XSS)

PHP or Javascript in content (nodes and comments and fields in Drupal 7)

Safe error reporting (avoiding information disclosure)

Secure private files

Only safe upload extensions

Large amount of database errors (could be sign of SQLi attempts)

Large amount of failed logins (could be sign of brute-force attempts)

Responsible Drupal admin permissions (protecting against access misconfiguration)

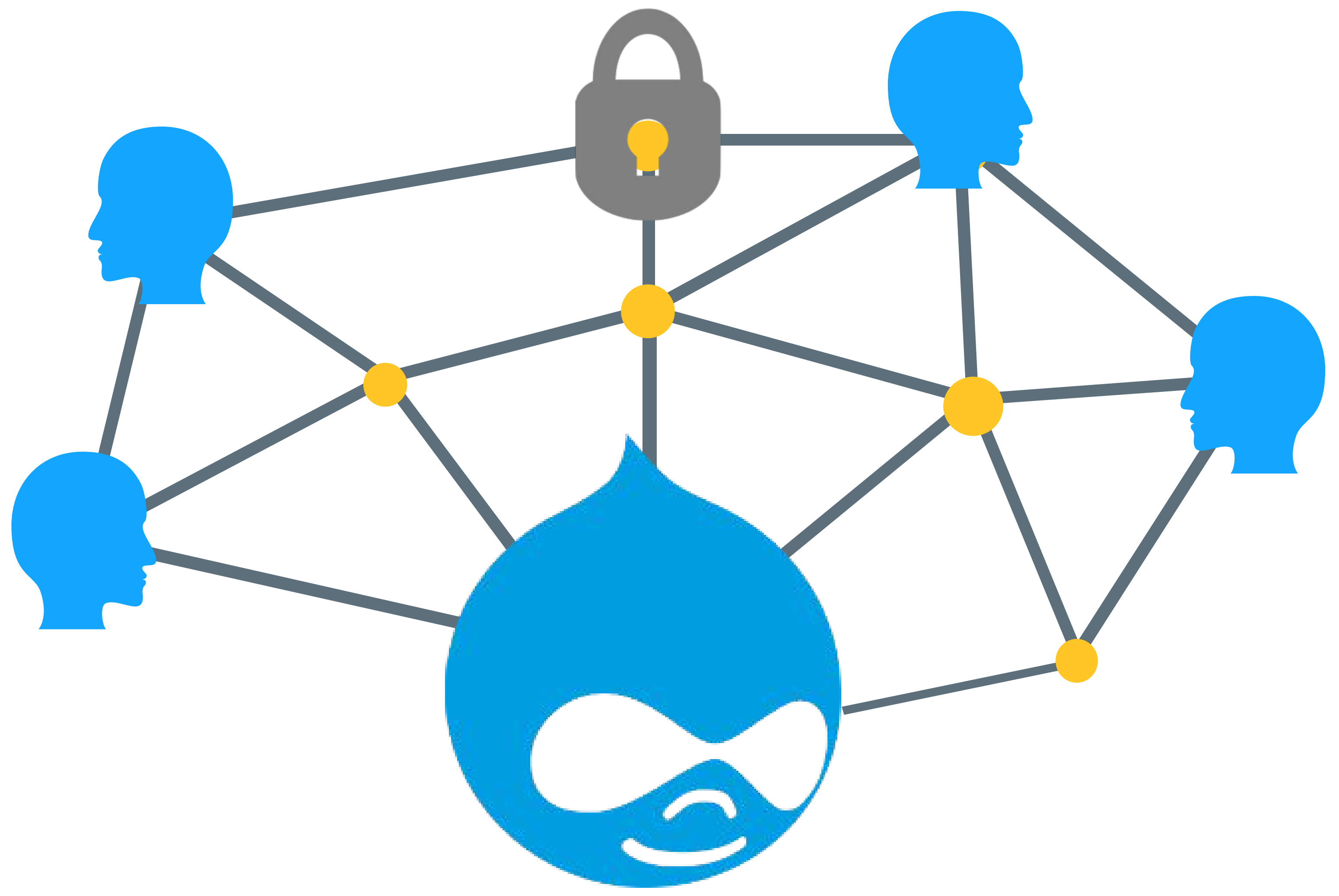
Username as password (protecting against brute-force)

Password included in user emails (avoiding information disclosure)

PHP execution (protecting against arbitrary code execution)

Base URL set / D8 Trusted hosts (protecting against some phishing attempts)

Views access controlled (protecting against information disclosure)



Security Through Community

Use the Drupal community's mature advice and expertise.

Community: Drupal Core Security

- ✓ Traditionally very secure
- ✓ Quick to respond
- ✓ Very communicative
- ✓ Responsible Disclosure helps prevent 0-days

Community: Contrib Modules

What to look out for when choosing modules to use

- ✓ No activity in X days/weeks/months/years
- ✓ Look at the open bugs for red flags
- ✓ Number of user installations
- ✓ Development status: stable, dev, alpha, beta, etc.
- ✓ Is it covered by the **Security Advisory Policy**?

Community: Contrib Modules and SA’s

When modules are covered by the Drupal Security Advisory policy:

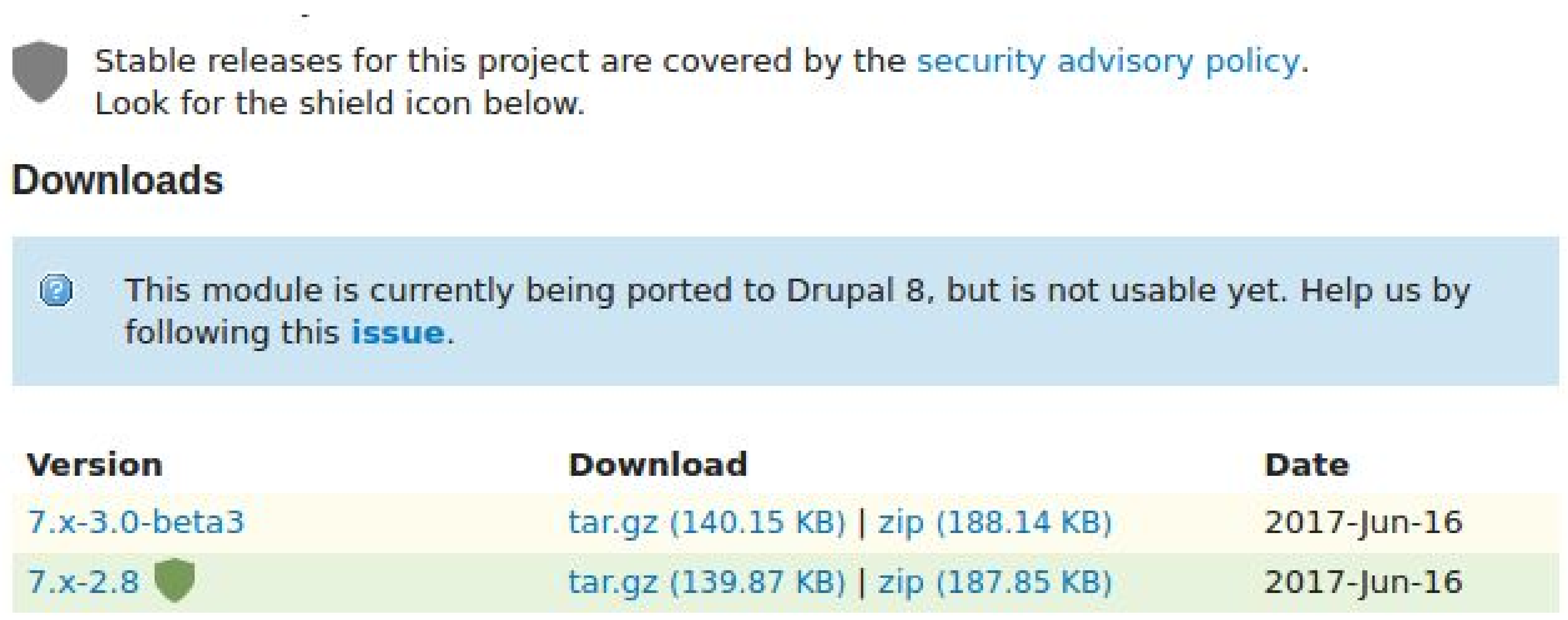
- They come with a responsible disclosure process
- They get the benefit of the Security Team’s help to mitigate problems
- The risk of unpatched, publicly known vulnerabilities is greatly reduced

Keep in mind:

- *Only “stable” modules that have applied for security coverage are covered by SA policy*
- Contrib modules are the biggest threat to a Drupal site
- The Security Team is more reactive than proactive

Community: Security Advisory Policy

Only stable modules get covered by Security Advisories
(but not all stable modules are covered)



Stable releases for this project are covered by the security advisory policy.
Look for the shield icon below.

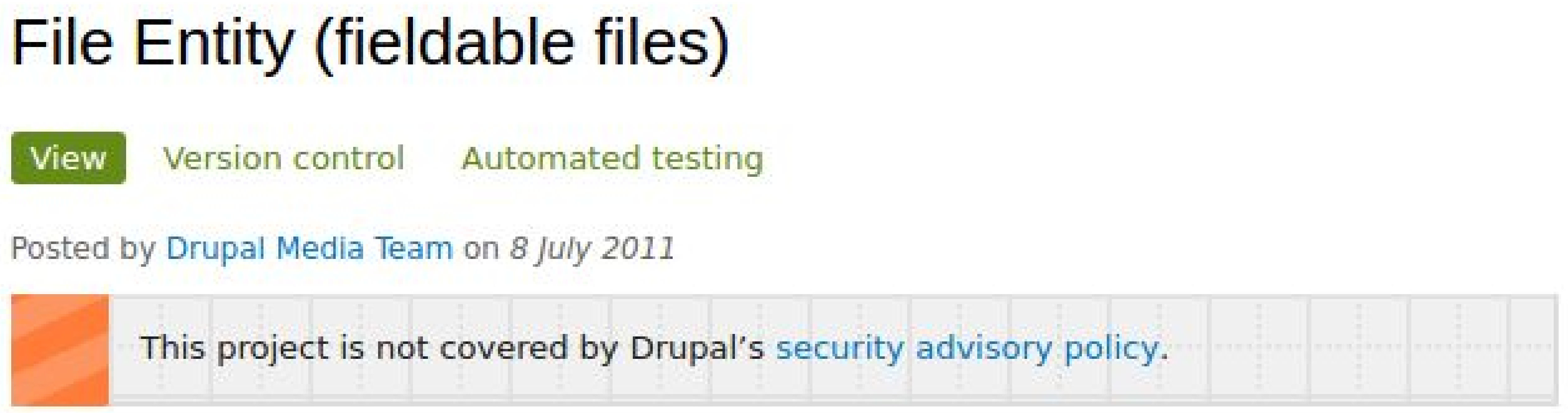
Downloads

This module is currently being ported to Drupal 8, but is not usable yet. Help us by following this **issue**.

Version	Download	Date
7.x-3.0-beta3	tar.gz (140.15 KB) | zip (188.14 KB)	2017-Jun-16
7.x-2.8	tar.gz (139.87 KB) | zip (187.85 KB)	2017-Jun-16

Community: Security Advisory Policy

Alpha, beta, dev, and RC status modules *do not get covered by Security Advisories*


File Entity (fieldable files)

Posted by Drupal Media Team on *8 July 2011*

This project is not covered by Drupal's security advisory policy.

A few other things of note

Bundled Libraries are dangerous -- especially in modules with low activity! (OWASP #9)

Did you know these are officially End of Life?
--jQuery 1.x and 2.x
--Bootstrap 3.x

Did you know they have known vulnerabilities and will not be receiving back-ported security patches?

Recommended Homework

Secure Coding Best Practices (a whole other presentation)

Check your Web Host:

- Who are you sharing space with?
- Backups - are they being done? Can you restore from them?
- Updates - Server, database, services, apps, etc.
- Maintenance
  - Are you keeping Drupal up to date (core and modules)?
  - Are you keeping up on Bugs for non-stable modules?
  - Do you subscribe to Drupal Security Advisories?

Resources

Drupal.org - http://drupal.org/security-team

http://drupal.org/writing-secure-code

http://drupal.org/security/secure-configuration

http://www.devsecops.org/

Drupal Contrib Modules Security Advisories:
https://www.drupal.org/security/contrib

Drupal Core Security Advisories:
https://www.drupal.org/security/
https://www.drupal.org/drupal-security-team/security-advisory-pr
ocess-and-permissions-policy

Thanks!

Any questions?

You can find me at:
@willc
will@willchatham.com
Look for these slides at:
https://www.willchatham.com