Pantheon's Guide to Security in Higher Education and Beyond!

david.needham@pantheon.io

David Needham

Developer Advocate at Pantheon

davidneedham on Drupal, WordPress, GitHub, and Twitter

Some things I enjoy, aside from coding, are

- Biking with my wife and kids
- Board games: is.gd/davidsgames
- Volunteering in the community

gilzow@missouri.edu

Paul Gilzow

Programmer Analyst, Security Analyst at the University of Missouri

gilzow on Twitter, WordPress, Drupal, Facebook, LinkedIn, and GitHub

Web application security and accessibility evangelist. Software instructor. Conference lecturer and presenter. Runs on passion and coffee.

University of Missouri

PANTHEON®

TL;DR

Minimizing Risk

What is “Risk”?

Risk is the intersection of assets, threats, and vulnerabilities

Asset

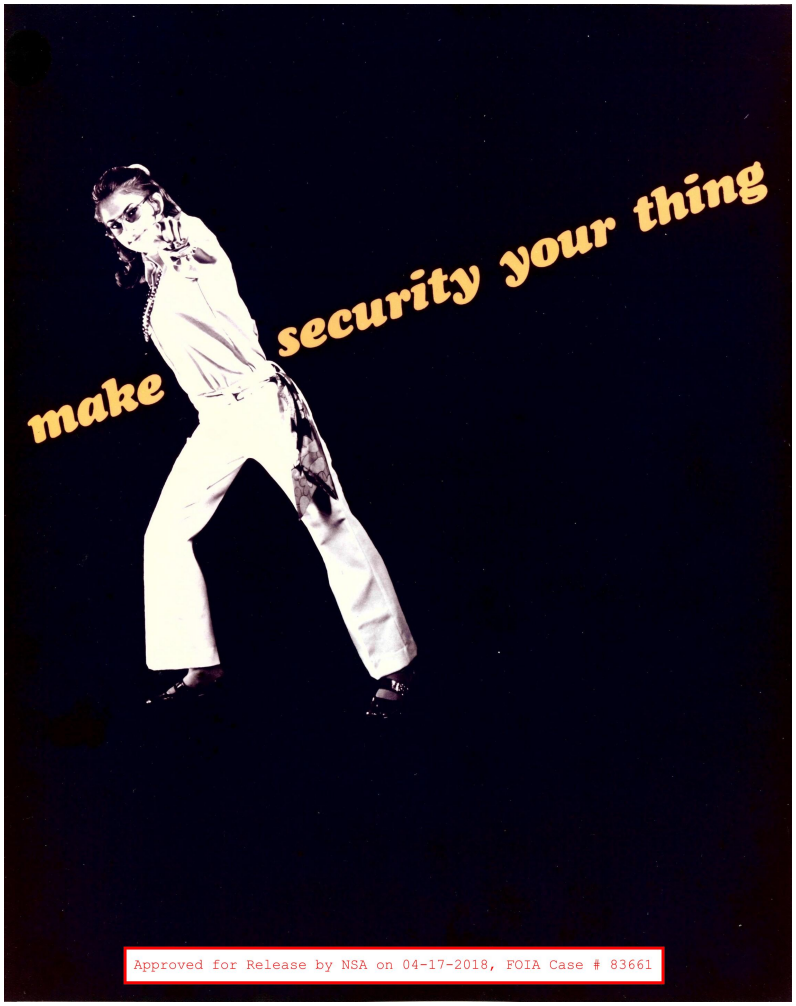
- People
- Property
- Information/Data
- An asset is what we are trying to protect

Threat

- Anything that that represents a potential danger to an asset, whether deliberately or by accident
- A threat is what we're trying to protect against
- *Threat Agent* is a group or individual who exploits a vulnerability to manifest or cause a threat to occur

Vulnerability

- Weakness or holes/ gaps in security procedures or program that can be exploited by a threat to affect assets



What is “Risk”?

- Asset = You
- Threat = Rain
- Vulnerability = Hole in your umbrella
- Risk = you getting wet

What is “Risk”?

The potential for loss, damage or destruction of
an asset(s) as a result of a threat exploiting a vulnerability
multiplied by the impact of the threat occurring

Why Education is an Attractive Target

- Network bandwidth and availability
- Rich in hardware infrastructure
- Poor in human resources
- Resistant to blacklisting
- SEO reputation

Why Education is an Attractive Target

- Personally Identifiable Information (PII) / Sensitive Personal Information (SPI)
- Protected Health Information (PHI)
- Confidential Intellectual Property
- Export Controlled Data
- National Security Interest (NSI)

Backups

- What
  - A snapshot of your
    - files that make up your site
    - database
- Why

Backups

Mizzou
University of Missouri

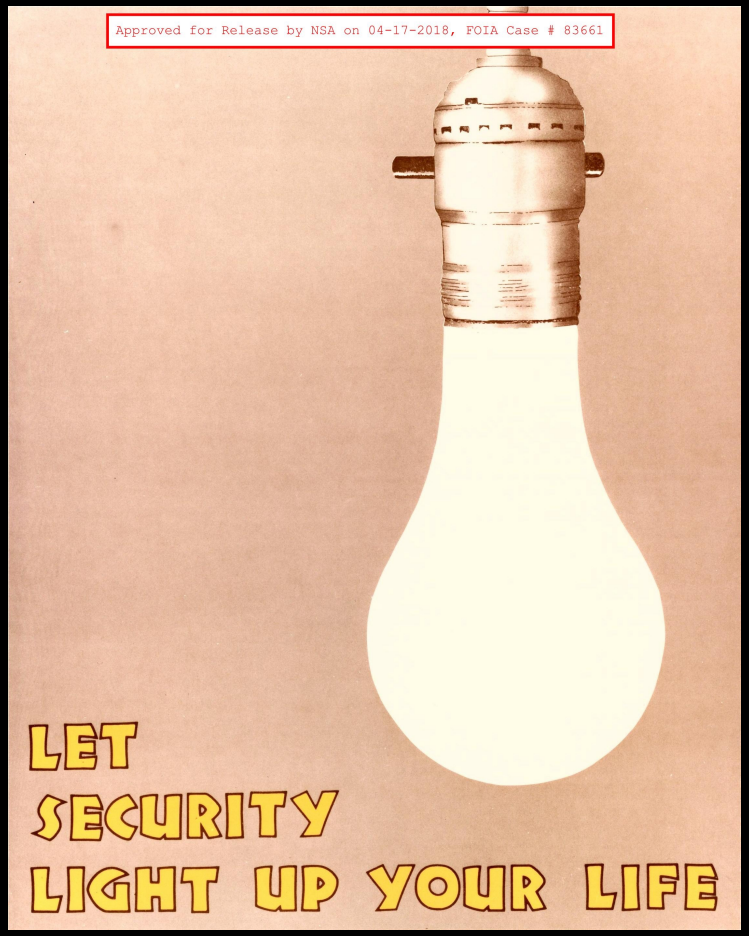
PANTHEON®

Backups

- What
  - A snapshot of your
    - the files that make up your site
    - database
- Addresses two types of threats:
  - Data loss/damage
  - Disruption in service/site downtime
- How does it reduce risk?
  - Lowers Impact

Backups

- Bonus points
  - Protect your backups
  - Don’t keep your back ups in a publicly accessible area
  - Test your backups!



Keep Drupal Up-to-date

- Why/How does it reduce risk?
  - Updates often address security issues
  - Potentially removes an exploitable vulnerability
- How
  - Subscribe to the Security Team newsletter
  - Update and stay on the latest release
  - Automate updates & testing
- Security principle : *Don't use components with known vulnerabilities*

Keep Themes/Modules Up-to-date

- Why/How does it reduce risk?
  - Similar to Drupal core updates, module and theme updates can also contain security fixes, removing potential vulnerabilities
- Bonus points
  - Know what you have installed and ***why*** you have it installed
  - Limit your module/theme use

Hosting Provider

If Drupal is the brain, and your content the heart and soul of your site…

…the hosting provider is the rest of the body

Hosting Provider

- Why
  - Doesn't matter how well you've secured Drupal, if the host is compromised
  - Remains one of the top vectors for compromised sites
- Security principles :
  - *Don't use components with known vulnerabilities*
  - *Establish Secure Defaults / Fail safely*
  - *Separation of Duties/Segmentation*
- Bonus points
  - Know what your host is running and what versions they have installed
  - Engage with the team responsible for hosting and work with them to keep the stack up-to-date

File/Directory Permissions

- What
  - Ensure that files and directories are set to the lowest access necessary
- Why/How does it reduce risk?
  - Improper permissions allow an attacker to access restricted files or directories and potentially modify or delete their contents
- Security principle : *least privilege*

Principle of Least Privilege

- Grant necessary permissions required to perform the intended activities
- For a limited time
- But with the *minimum* rights required for the task(s)
- Removing permissions when no longer needed

File/Directory Permissions

- Bonus points
  - Lock down all area of Drupal to read-only except for those areas that specifically require the ability to write
  - If your environment allows it, set files to only readable (0400) by the owner of the process that php runs under
  - Ideally, only the *files* directory is writable, and then only writeable by the php process
  - Bonus Security Principle: *Minimize Attack Surface*

What is “Attack Surface”?

- The sum of all paths for data/commands into and out of the application
- Plus all of the code that protects those paths
- Plus all of the data used in the application
- Plus all of the code that protects this data

What is “Attack Surface”?

Remove Unused Themes/Modules/Users

- What
  - Remove everything that isn't in active use
- Why/How does it reduce risk?
  - Even if a module/theme is disabled, the files are still there and are publicly accessible
  - A non-active user is one more account that can be compromised
  - Removes potential vulnerabilities
- Bonus points
  - Make module, theme and user audits a routine

Do Your Homework on Theme/Module Selection

- What
  - Research a theme/module before installing
- Why/How does it reduce risk?
  - Every piece of code you add to your system increases your attack surface
  - Every piece of code you add to your system has the potential to introduce new exploitable vulnerabilities

Do Your Homework on Theme/Module Selection

~~Jessica~~ Paul, your paranoia is exhausting

Do Your Homework on Theme/Module Selection

- What
  - Research a theme/module before installing
- Why/How does it reduce risk?
  - Every piece of code you add to your system increases your attack surface
  - Every piece of code you add to your system has the potential to introduce new exploitable vulnerabilities
  - Security principle: Be paranoid, be skeptical

Do Your Homework on Theme/Module Selection

- Security principle: *Treat all third party code/data as tainted and hostile*
- Bonus steps
  - https://security.utexas.edu/dorkbot
  - Run third party code through PHP-CS Security Audit
  - Run local static code analysis

Limit User Roles

- What
  - Give users the lowest possible role that allows them to complete their tasks
  - Only login with a higher privileged account when performing actions that require elevated privileges
- Security principle: Least privilege
- Why/How does it reduce risk?
  - Minimizes the damage if an account is compromised
  - Reduces the opportunity for a rogue user to inflict damage
  - Reduces the opportunity for someone to make a mistake

Limit User Roles

- Bonus points
  - Create or add custom roles that give you the ability to be more granular with permissions
  - Do routine account audits and remove permissions/roles from accounts that don't require them

Protect settings.php

- What
  - Add rules to prevent direct access
  - move the file somewhere not publicly accessible
- Why
  - Contains your database credentials and salts
  - Prevents accidental exposure of those assets
- Bonus points
  - Set file permissions to 0400*
  - See #4 Hosting Provider

Implement SSL

- What
  - Add a SSL/TLS certificate to your site
- Why/How does it reduce risk?
  - Encrypts the data as it is transferred between your site and the end user
- Security principle: *Minimize attack surface*
- Bonus steps
  - Enforce https over the *entire* site, not just login areas

Mizzou
University of Missouri

PANTHEON®

Strong Passwords

- What
  - Use a password that is long and contains randomized alpha characters, numbers and special characters
  - Does not contain common words in the dictionary
- Why/How does it reduce risk?
  - More difficult for attackers (threat agents) to guess, and, historically, brute force
  - Prevent unauthorized access

Strong Passwords

- Even more important than complexity is length

3 to 4 additional characters has the same entropy (number of possible combinations) as passwords using a more complex set of characters

Strong Passwords

Your new password!

MF>E,D4,C!q^m,uSwVh.[2AD+JHsM^6}

Assuming one hundred billion guesses per second will take 6.22 million trillion trillion trillion centuries to brute force

Strong Passwords

- They need to be unique, **for every account, on every site**
  - As of 2017, 7 **billion** credentials have been leaked/exposed
  - Credential stuffing

Strong Passwords

- Use a password manager
- No, really: use a password manager
- Enforce strong, unique passwords for **everyone**
  - Integrate with your institution's single-sign-on system
- Use a strong, *unique,* and **long** password **everywhere**, not just in Drupal

Limit Login Attempts

- What
  - Locks an account or blocks an IP address after so many failed attempts
- Why/How does it reduce risk?
  - Threat is unauthorized access
  - Vulnerability is a weak/common password
  - Reduces the ability of the threat agent to exploit the vulnerability
  - Security principle: *Minimize the attack surface*

Two/Multi-Factor Authentication

- What
  - Adds a secondary (or multiple) step that must be completed in order to authenticate
- Why/How does it reduce risk?
  - Adds an extra layer of defense against authentication attacks
  - Security principle: *Defense-in-depth*

Protect/Limit Access to Login/Admin Areas

- What
  - Add an extra layer of protection to the login area
    - Basic Access Authentication
    - Captcha
    - Password Policy
- Why/How does it reduce risk?
  - Similar to 2FA/MFA in that it adds an extra layer
- Security principle: *Minimize attack surface* and *Defense-in-Depth*

Miscellaneous

- Web Application Firewall
- Move/Hide changelog.txt
- Routine Security Scan
  - Droopescan
- Log actions/activities
- Secure your local machine

Items I believe should be higher in the list

- Block PHP execution (#14t)
- Logging (#11t)
- Segmentation/Isolation (separation of duties)
- Remove software/services that aren't actively used
- Limit the number of modules you use
- Monitor for file changes
- Stay informed!

Summary

- Always be thinking in terms of how you can reduce risk
- Minimize attack surface area
- Principle of least privilege
- Defense in depth
- Don’t use components with known vulnerabilities
- Be paranoid, be skeptical
  - Treat all third party code/data as tainted and hostile
- Security is a continual process; you’re never “finished”

Questions

Give us feedback:

mid.camp/342