Hosting and Web Apps
The Obscurity of Security

Quintin Russ
Mike Jager

OWASP New Zealand Day
Auckland, NZ
July 2010

"Immediately restore the most recent MySQL backup for our site and urgently explain how an attacker was able to compromise your systems and fill our database with rubbish. What steps will you be taking to prevent this sort of thing from happening again? We are not impressed by your inadequate security."

- - Angry Customer #1

"Our site has been defaced yet again, how are your servers so badly configured as to allow a punk-ass kid to deface the front page of our sites? You will be hearing from our attack lawyers with regard to this series of incidents, your security is a joke mate!!!1"

- - Angry Customer #2

Quintin Russ

- Technical Director, SiteHost

  quintin@sitehost.co.nz

- Web Developer in previous life
  - floated divs
  - battled Internet Explorer (on a Mac)
  - developed web applications

SiteHost

Mike Jager

- Senior Network Engineer, Web Drive

  michael@webdrive.co.nz

- Is Not A "Security Guy"
  - wishes everyone would just play nice :-(


CAUTION
THIS SIGN HAS
SHARP EDGES
DO NOT TOUCH THE EDGES OF THIS SIGN
ALSO, THE BRIDGE IS OUT AHEAD

Warning

- We're (mostly) F/OSS-loving LAMP guys
- We hear that people sometimes use Windows, IIS, SQL Server and .NET for their web apps
- Concepts apply to all platforms
  - security is a mindset, not a "solution" or "product"
  - this talk is not about your shiny SOA WAF
  - it is about common mistakes seen almost daily

"What if security isn't part of the spec?"

- - Anon, OWASP New Zealand Day 2009

Make it part of the spec.

trademe.co.nz/jobs

seek.co.nz/it-jobs

Planning/Design

- Read the OWASP website
- Get advice on your project early
- Ensure the development team are "tooled up"
  - training
    - what is the OWASP Top 10?
  - use a framework/standard design
  - identify threats early
    - in the design phase, not testing

Planning/Design

- Attack surface reduction
  - Principle of Least Privilege
  - "Principle of Least Functionality"
- KISS
  - do you need that web-based admin area?
  - CMS for a 5-page website that is entirely static?
- Layers of security
  - avoid single flaws from leaving you vulnerable
    - HTTP 401 and IP-based restrictions on your admin area?

Not all apps are created equal

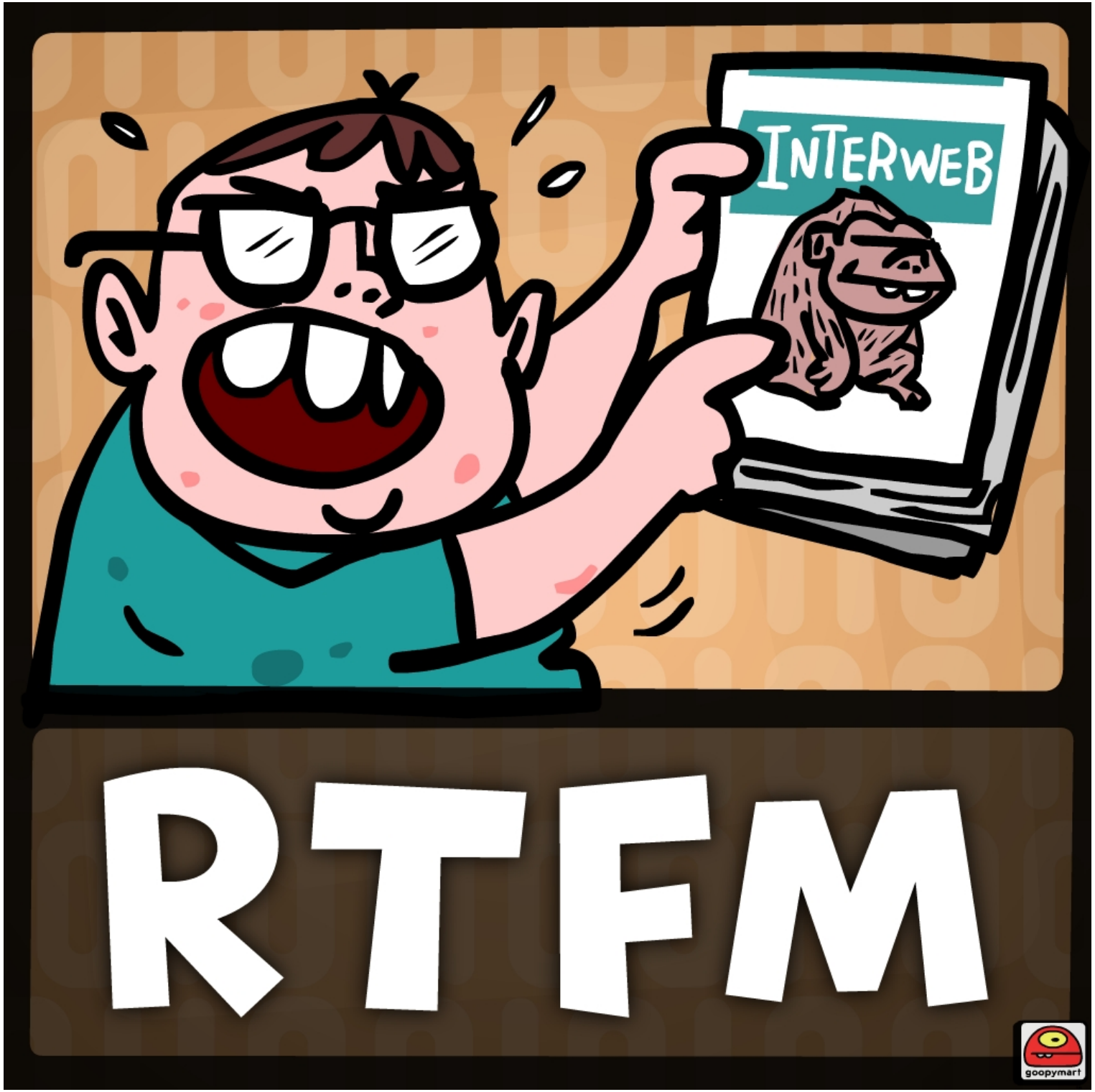
- Sometimes you can't write everything yourself
- So you use that Twitter library on your blog
- Or you heard a software IDS was a good idea
  - surely an IDS would make your app more secure?!
- Maybe you can't use external analytics
  - or you need some shiny shiny for your dashboard
- Build or Buy? - be careful selecting software

Not all apps are created equal

- Choose a "good" software platform
- Everything has some security holes
  - (regardless of what the glossy sales sheets say)
- How does the vendor approach security?
  - disclosure handling - do they advise customers?
  - security team/web page/email?
  - check the application's security history
    - is this application notorious for being Swiss cheese?
    - has the application "never had a reported hole"?

Not all apps are created equal

- How does the vendor operate?
  - release packaging
  - deployment/installation procedures in docs
  - "It's a Feature"
- Beware of plugin architectures
  - Joomla, WordPress, etc
  - don't ruin what may be a good app by picking plugin(s) in a hurry



http://www.flickr.com/photos/goopymart/291194341

Warning:

"The pickle module is not intended to be secure against erroneous or maliciously constructed data. Never unpickle data received from an untrusted or unauthenticated source."

- - Python Documentation

"Many web developers are unaware of how SQL queries can be tampered with... It means that SQL queries are able to circumvent access controls, thereby bypassing standard authentication and authorization checks... These attacks are mainly based on exploiting the code not being written with security in mind. Never trust any kind of input."

- - PHP Documentation

Development

- Attack surface reduction
  - Principle of Least Privilege
  - "Principle of Least Functionality"
- Don't trust your input
  - even from "trusted" sources
- Validate and verify everything
  - cloudy API responses
  - is that response in the format it should be?

Source control

- Use it everywhere
- But understand how it works
  - Meta-data is stored in .svn .git .cvs directories
  - Contains dir lists, usernames, revisions, source code
- "svn checkout" is an invalid installation method
  - F/OSS apps particularly bad for this
    - You can often pull direct from source control
    - .gitignore file in latest zip / tarball release of phpMyAdmin

Data management

- If you don't need it, don't store it
  - Adopt a zero tolerance policy
- If you do need it, how do you need to access it?
  - Passwords
  - credit card data
- Hash (with a salt), don't encrypt
- Keep production data out of development

Data management

- Keep tabs on your data
  - absolute size
  - growth rates
  - which data is used by which code?

Password security

- Re-use of credentials
  - Don't.
- Weak usernames/passwords
  - DB name: wpblog2
  - DB user: wpblog2
  - DB password: wpblog2
- Former Staff / Old Passwords

Filesystem security

- Don't use /tmp and friends
- Permissions
  - understand what that chmod in the docs means
  - most web directories can be 701 or even 101
- Beware log files
  - especially in the DocumentRoot
- Beware test files
  - phpinfo() in php.php

Filesystem security

- Beware accumulated crap
  - session files
  - template caches
- Only production files on production
  - bob.txt
  - bob.phps
  - website.zip
  - database.sql

Deployment best practise

- Remember the filesystem
  - test.php/install.php/setup.php
  - README/INSTALL/ChangeLog/LICENSE
  - many installation guides suggest weak permissions
    - reduces support overhead, "just works"
- Hosting control panels
- SSL
  - if it's a good idea for the authentication, it's a good idea for the content, too

Deployment best practise

- Keep applications separate
  - https://www.example.co.nz/super-secure-store
  - http://www.example.co.nz/badly-coded-blog
  - same trust level?
- Backups!
  - keep your own – no, really!
  - test them – before you need to use them

Clouds

- Remote includes
  - don't, ever, they're madness
- Minimise remote resource use
  - out of your control, regardless of what the SLA says
  - take a local copy of AJAXy Web 2.0 libraries
    - at the wrong end of a wet piece of network
    - good targets for an attacker wanting a high profile
    - what'd we say about remote includes again?
  - do you need 3rd-party analytics on everything?

Clouds

- Outsourced data storage
  - what data are you uploading?
    - do you verify this?
  - where is your data?
    - another country? whose laws is it subject to?
  - who has access to your data?
    - remotely? physically?
  - data retention policy?
    - are there backups that may exist months after deletion?

Software lifecycle management

- Must have a process for being deprovisioned
  - before deployment, and set a date for deprovisioning
  - review constantly, and keep up to date
- Remove unused files and data
- Update, update, update
- Do you monitor upstream for security advisories?
  - who does this?
  - how do they do this?
  - how often do they do this?

Software lifecycle management

- How soon after a release do you plan to patch?
  - How long does it actually take?
  - What happens when a vulnerability is discovered at 5pm on Friday of a long weekend?
    - (are vulnerabilities discovered at 5pm on a Friday of a long weekend?)

Monitoring

- Monitor changes to your website
  - hash production versions of files
  - compare these to hashes stored remotely
- Monitor your website's uptime
- Check external access
  - what does "external" mean for your application?
  - has your whitelist stopped working?
    - Do you have a whitelist?

Monitoring

- DNS
  - have your provider’s DNS servers stopped working
  - has your domain been hijacked?
  - remember that DNS is an external dependency!

Politics

- If you're:
  - a manager, make security part of KPIs
  - not a manager, make sure security is part of your KPIs
- Get buy-in from non-technical personnel
  - Allow business requirements to be overridden
  - Allow staff to delay or abort a release
- Engage OWASP/security communities

Help us help you!

- Talk to your hosting and security support people early, please!
- If you haven't got security experts, find some
- Lean on providers
  - make sure they're doing their part
  - engineers can only build what customers ask for
- But remember...

You cannot outsource the responsibility of ensuring that your apps are secure.

Hosting and Web Apps

The Obscurity of Security

Quintin Russ
quintin@sitehost.co.nz

Mike Jager
michael@webdrive.co.nz