How to secure your dotCMS implementation

Chris McCracken & Daniel Silva

What is “secure”?

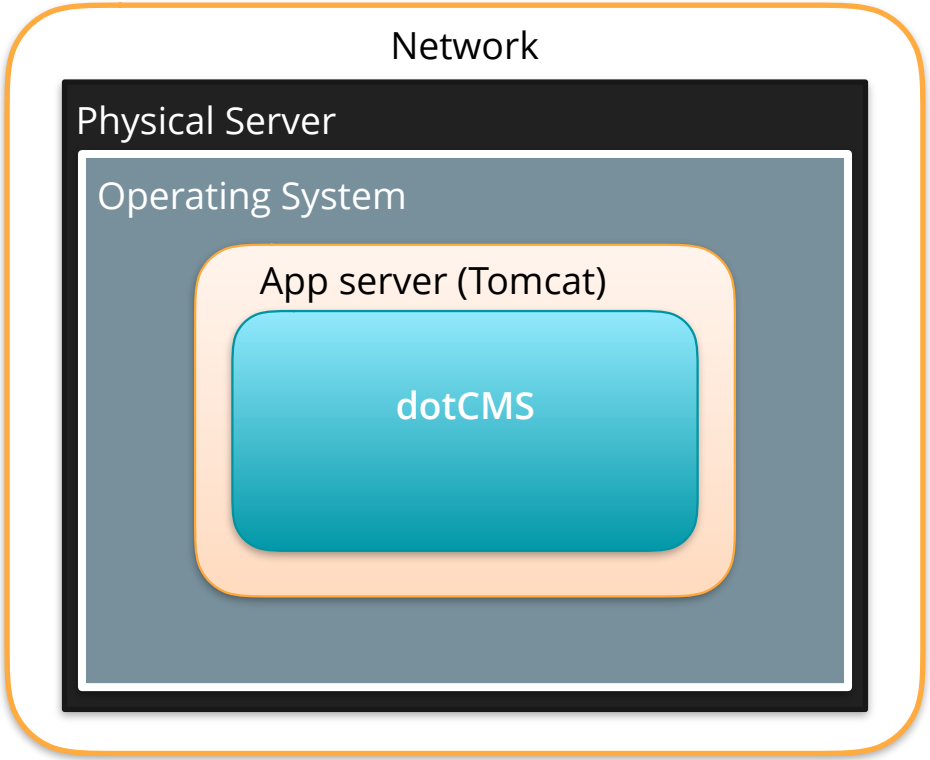
Secure: *“free from danger or attack”*

- Holistic approach
- Not a purchase
- Process, not product
- Compliance
- A matter of perspective

Security Concepts

Concepts

- Know your system
  - No surprises
- Defense in depth
  - Layer defenses like an onion
- Secure by design
  - Think secure before you build
  - Don’t retrofit
- Least privilege
  - Only grant specific access as needed
  - At every layer
- Audit trail
  - If it wasn’t logged, it didn’t happen
  - The log must be trustworthy



Implement security at EVERY layer

Don’t forget…

- Don’t forget the physical layer
- Don’t forget the human layer
    - Train regularly

Securing the OS

Securing the OS

- Deploy dedicated application servers
- Install security updates vigilantly
- Use encryption at rest (full-disk encryption)
- Use a dedicated service accounts (least privilege)
- Minimize network surface area (host firewalling)
- Log thoroughly and reliably

Securing the Data

Securing the Data

- Database
  - Dedicated server, or at least database & user
  - Hardened up-to-date OS
  - Restricted network
- Assets
  - Least-privileged user access

Securing the dotCMS Application

Securing the dotCMS application

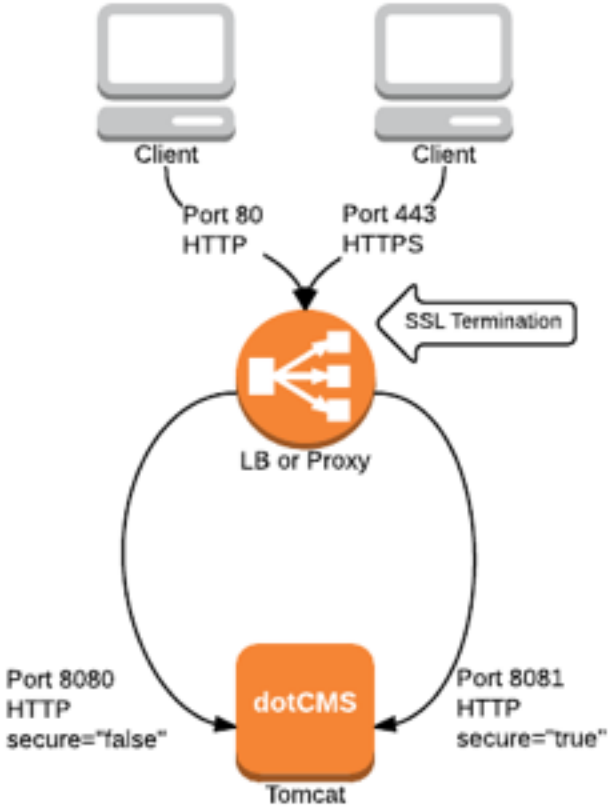
- User & Role management
  - Grant minimal access needed
- Customize login security
  - portal.properties
- External authentication
  - LDAP and SSO
- Scripting & SQL Viewtools
  - With power comes danger
- Denial-of-Service
  - Cache well, perform well
  - Implement upstream web services with care
- Logging
  - Log4j customization & export
- Feature management
  - Disable servlet mappings

Securing the Connection

Securing the Connection

- Encryption in transit (SSL/TLS)
    - Use security constraints or SSL everything
- Proxy or Load Balancer
    - SSL termination
    - Assist service account
- HTTP Firewall
    - WAF or simple ACL
    - dotCMS rewrite rules

SSL connection



SSL Connection

```
    <!-- HTTP Connector from Proxy -->
    <Connector maxThreads="125" port="8080" protocol="HTTP/1.1"
        connectionTimeout="3000" enableLookups="false" redirectPort="443"
        proxyPort="80" scheme="http" />

    <!-- HTTPS (SSL) Connector from Proxy -->
    <Connector maxThreads="75" port="8081" protocol="HTTP/1.1"
        connectionTimeout="3000" enableLookups="false" redirectPort="443"
        proxyPort="443" scheme="https" secure="true" />
```

Q & A