October 2019

APPLICATION AND DATA SECURITY

Makerere University
Faculty of Technology
College of Design, Art and Technology

By Stephen Senkomago Musoke
http://ssmusoke.com

WHY ME?

- ☐ Self taught software tinkerer who loves growing techies
- ☐ Working hard at a normal regular family life
- ☐ My fair share of failed, successful, mind blowing and soul haunting projects
- ☐ Served clients in UK, US, Australia, Europe, South Africa
- ☐ 12 years setting up, growing & running a Ugandan custom software development shop
- ☐ Executive management stint - Worked in and ran a large international custom software service provider in South Africa & Uganda
- ☐ 4 years back to full time software delivery practice

APPLICATION SECURITY

Do not under-estimate the need for security at all levels everyone is out to get you ~Stephen Senkomago Musoke

PRINCIPLES

Security is a measure of quality that has to be baked into software not bolted on

Simplicity is the ultimate sophistication

- Confidentiality – access is to only the data a user needs
- Integrity – data is not altered outside pre-defined protocols
- Availability – systems are accessible and useable to those users who need them, when they need them

APPROACHES

Security is a measure of quality that has to be baked into software not bolted on

Simplicity is the ultimate sophistication

- ❑ First class citizen in requirements gathering, architecture and design
- ❑ Requirements:
  - ❑ Authorization – who can do what, when?
  - ❑ Who can see what when?
- ❑ Architecture – 12factor.net
- ❑ Design
  - ❑ Phoenix servers
  - ❑ Plan for failure – NetFlix Chaos Monkey
  - ❑ Go as simple as you can
- ❑ OWASP - Open Web Application Security Project

12 FACTOR

- ☐ Codebase - One codebase tracked in version control, many deploys- trunk based development
- ☐ II. Dependencies - Explicitly declare and isolate dependencies e.g., composer.json, package.json, pom.xml
- ☐ III. Config – store config in the environment encrypted TRAVIS variables, Hashicorp Vault, AWS Secrets
- ☐ IV. Backing services - Treat backing services as attached resources they are all the same
- ☐ V. Build, release, run - Strictly separate build and run stages and each should be atomic
- ☐ VI. Processes - Execute the app as one or more stateless processes

12 FACTOR

- VII. Port binding – everything is stateless and share nothing
- VIII. Concurrency – just add more workers
- IX. Disposability - Maximize robustness with fast startup and graceful shutdown, do not leak secrets
- X. Dev/prod parity - Keep development, staging, and production as similar as possible
- XI. Logs - Treat logs as event streams (observability)
- XII. Admin processes - Run admin/management tasks as one-off processes e.g., migrations, cleanup scripts (housekeeping)

OWASP GUIDELINES

- Minimize attack surface
- Establish secure defaults – password policy, expiry, access control
- Principle of least privilege –
- Defense in depth – layer the security controls, combine multiple security protocols & approaches
- Fail securely – handle errors gracefully, expose minimal information in errors and stack traces
- Don’ t trust systems & services – validate data inputs, lock down access

OWASP GUIDELINES

- ☐ Separation of duties and responsibilities
- ☐ Avoid security by obscurity – do not keep a key under the carpet because nobody knows its there
- ☐ Keep security simple
- ☐ Fix security issues correctly - carry out a root cause analysis, identify potential changes in design

DEVELOPMENT

- ❑ Write as little code as possible – leverage pre-built libraries
- ❑ Tech stack – use the simplest you can find
- ❑ Testing - automate as much as you can and make them run as fast as you can
- ❑ Deployment – deploy as frequently as you can
- ❑ Validate against best practices in the industry & lessons from others

Good developers write excellent code, great developers write no code, zen developers delete code

PRODUCTION SYSTEM SECURITY

- ❑ Monitor, monitor, monitor – respond to failures they happen (keep the lights on), predict failure
- ❑ Automate credential management
- ❑ Systems fail – bake failure into the process
- ❑ Layer security
  - ❑ Web Application Firewalls
  - ❑ Proxies (for performance)
  - ❑ Anti-virus & Anti-malware protection
  - ❑ DDOS protection – high availability
- ❑ Hire experts to scan your systems and advise

PRODUCTION SYSTEM SECURITY

- ❑ Leverage standards and best practices – NIST, CERT, BS, ISO
- ❑ Upgrade and patch your systems
- ❑ Use the least privilege for any activity – restrict access to root and administrator accounts

DATA SECURITY

Trust but verify

- ❑ Secure your data at rest, in transit and storage
  - ❑ Encrypt what you need
  - ❑ TLS, SSL & HTTPS
  - ❑ Encrypted backups???
- ❑ Datensparsamkeit – only collect and handle the data that you need – do you need that PII, that extra data on days visited or just aggregate
- ❑ Backup your data
- ❑ Verify the backups by restoring them

IN CLOSING

Security is not a one off event but a continuous activity

Security is built in layers – one on top of another

Security is complex and difficult, use experts, standards and best practices for " your" environment & needs

AND MOST OF ALL
Security is every-body' s responsibility

THANK YOU

For questions or suggestions

@ssmusoke

http://ssmusoke.com