Spring Security 4 N00bz

A quick introduction for the terminally insecure

Mark Heckler
Professional Problem Solver, Spring Developer & Advocate
www.thehecklers.com
mark@thehecklers.com
mheckler@pivotal.io
@mkheck

Pivotal

Who am I?

- Author
- Architect & Developer
- Java Champion, Rockstar
- Professional Problem Solver
- Spring Developer & Advocate
- Creador y curador de

SPRING NOTICIAS EN ESPAÑOL

New book!

But you can't buy it yet…

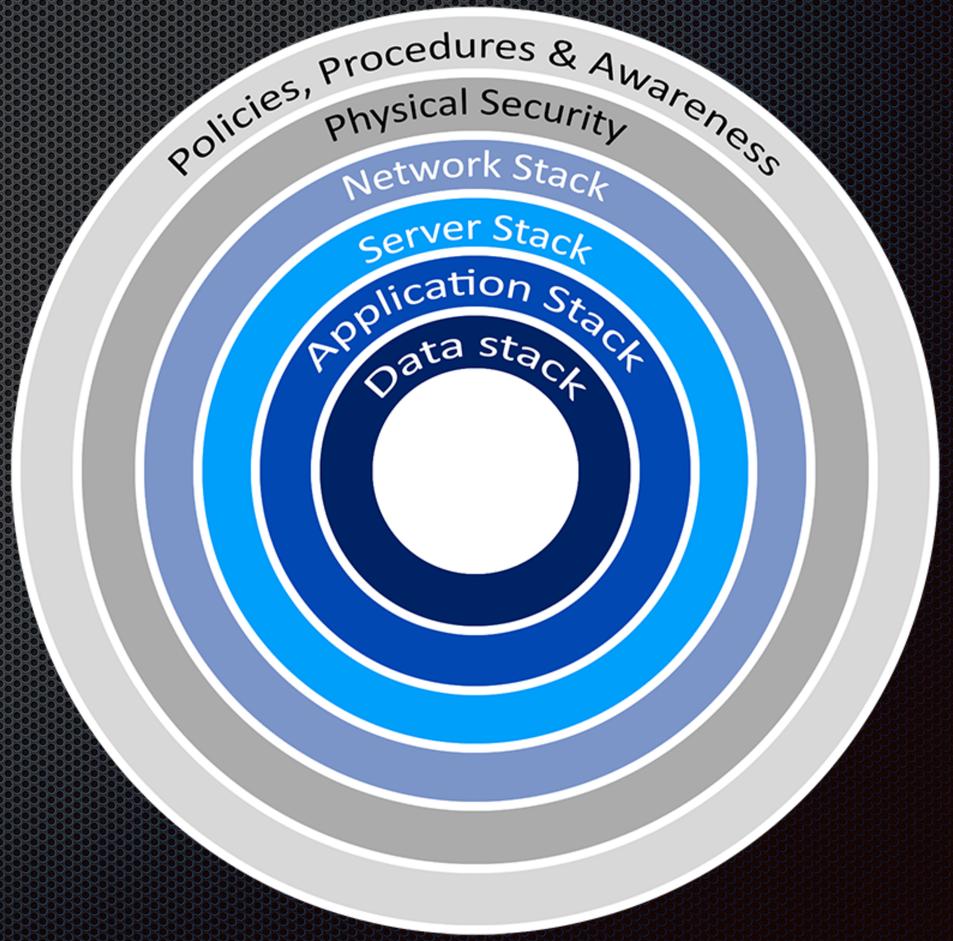
DISCLAIMER: artist's rendition only, not the real cover

Takeaways

- Contextual understanding of outside-in security profile
- System vs. application security
- Authentication & Authorization: who's who in the zoo
- OpenID Connect & OAuth2: what they do & what's the value
- SHOW ME THE CODE

Outside->In security, sort of…

- Cloud deployments have shuffled and/or inverted some of these…
- Obviated others
- General principles apply, if refocused for this **century**



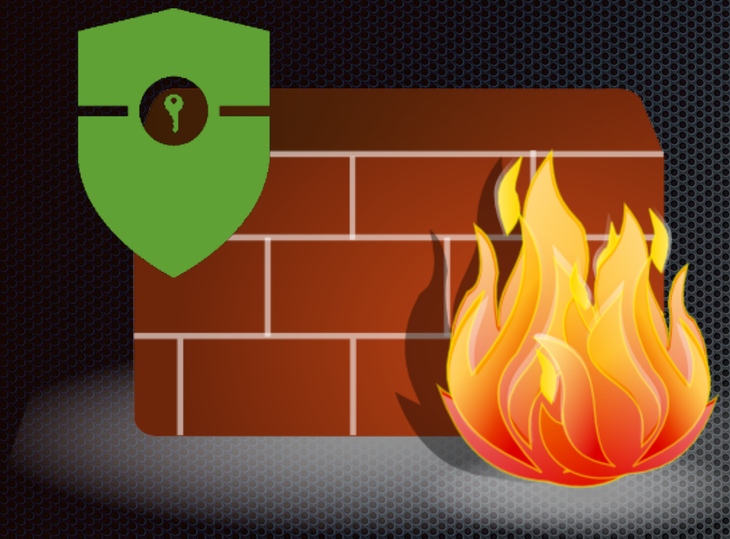
A few thoughts on system security

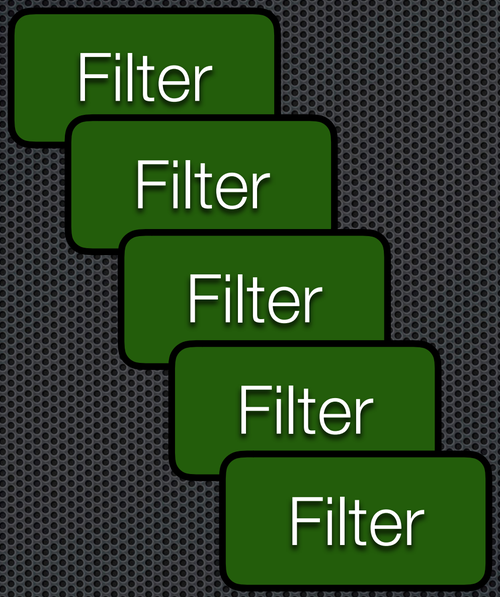
- Password/access hygiene
  - 2FA/MFA
  - Sane authorizations
  - Logging/auditing (with caveats)
- Wire encryption
- Store secrets securely
- Encrypted data at rest

Another time, another talk...

Application security

Spring Security 3000 meter view



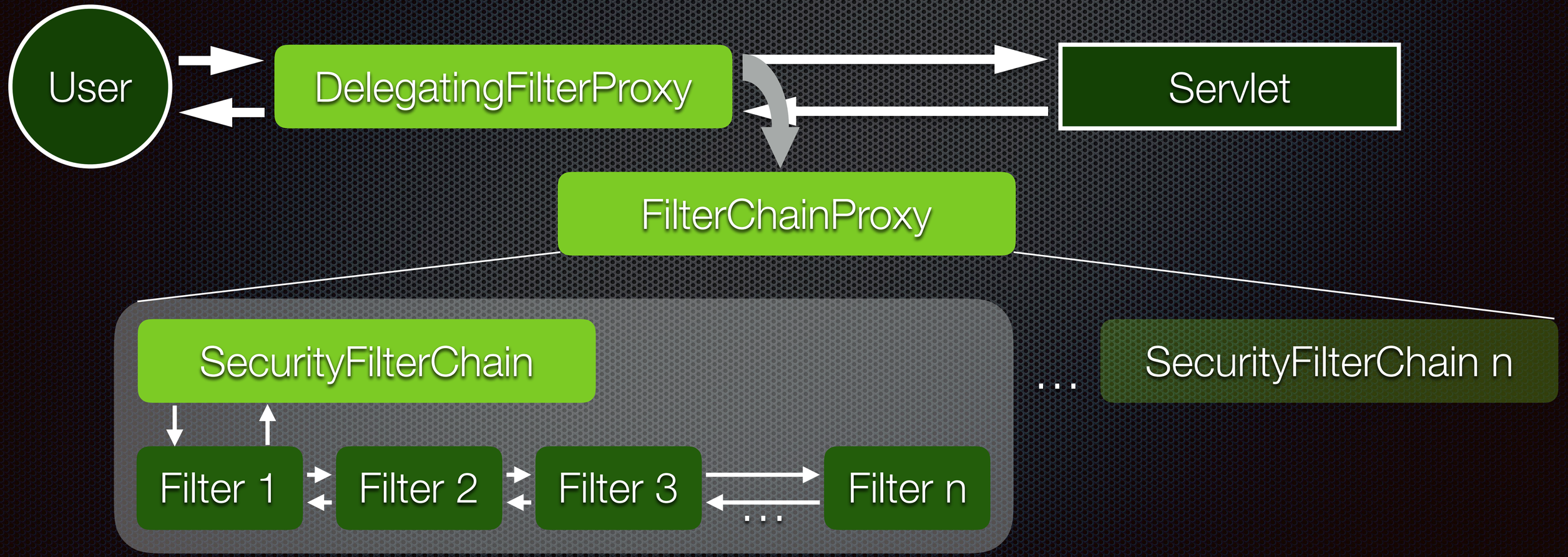



```
HTTP/1.1 200
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Length: 19
Content-Type: text/plain;charset=UTF-8
Date: Sun, 03 Feb 2019 22:19:04 GMT
Expires: 0
Pragma: no-cache
Set-Cookie: JSESSIONID=C9A582CB5B17036FCC7ADB07F799201D; Path=/; HttpOnly
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
```

HttpFirewall SecurityFilterChain Request headers

Of course, there is more...

Spring Security request filtering (simplified)



Let’s code!

@mkheck

www.thehecklers.com

Pivotal

Resources

- https://github.com/mkheck/spring-security-4-n00bz
- https://github.com/jgrandja/oauth2-protocol-patterns
- https://spring.io/projects/spring-security

Thanks for coming, stay in touch (& secure)!