Value Objects a la Domain-Driven Security

A Design Mindset to Avoid
SQL Injection
and
Cross-Site Scripting

#OWASP #AppSecEU #dds

Dan Bergh Johnsson
Omegapoint AB
@danbjson | dearjunior.blogspot.com
dan.bergh.johnsson@omegapoint.se

I finally got his [Mr X] clues. Escaping data for a subsystem is not a part of input validation, because O'Connor is a valid name. Escaping data for a subsystem should be a part of code that communicates with the subsystem.

- Erlend Oftedal
[Security] The security craftsman - Part 4
http://erlend.oftedal.no/blog/?blogid=82

Conclusions

- Domain Driven Design + Application Security
  = *Domain Driven Security*
- Make concepts explicit
  - Username, HtmlText
- Beware of mappings at (sub)system borders
  - iwa.web, java.sql
- Validate upon input
  - if(name.matches(...)) else throw new ...Exception
- Encode upon output
  - String asTextHtml() return .... encodeHtml()
- Use strong type value objects to enforce validation and encoding
  - verifyUser(Username, Password)
  - NewsPage(HtmlText headline, HtmlText content)