Intro to Web App Security

By Hansen Wu

Overview

1. What is a webapp, or, what are we talking about here?
2. Common faults in webapp security
3. Doctrine and mindset of breaking a webapp
4. Mapping a webapp and finding an angle of attack

What is a web app

Defining things

- Provides some service or functionality to end user
- Utilizes client-server model
- Operates in a web browser
- Pretty much anything you do on the web

Common Faults in Web App Security

Common Faults

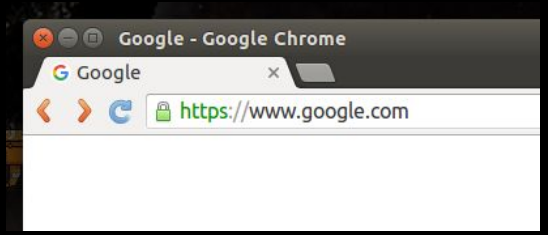
- Crowd mentality, lemmings, or “everyone uses it so it’s secure”
- HTTPS
- Logic failures
- Client-side security

A lot of people use OpenSSL

> OpenSSL Security Advisory [07 Apr 2014]
> ==
>
> TLS heartbeat read overrun (CVE-2014-0160)
> ==
>
> A missing bounds check in the handling of the TLS heartbeat extension can be
> used to reveal up to 64k of memory to a connected client or server.
>
> Only 1.0.1 and 1.0.2-beta releases of OpenSSL are affected including
> 1.0.1f and 1.0.2-beta1.
>
> Thanks for Neel Mehta of Google Security for discovering this bug and to
> Adam Langley <agl@chromium.org> and Bodo Moeller <bmoeller@acm.org> for
> preparing the fix.
>
> Affected users should upgrade to OpenSSL 1.0.1g. Users unable to immediately
> upgrade can alternatively recompile OpenSSL with -DOPENSSL_NO_HEARTBEATS.
>
> 1.0.2 will be fixed in 1.0.2-beta2.

From (OpenSSL.org)[https://www.openssl.org/news/secadv/20140407.txt]

That green lock means you're safe ->



- Common misconception of HTTPS and security
- "Cryptography is typically bypassed, not penetrated"
- Faults in implementation or endpoints are the weak point

The implementation logic is fine

- Authentication
  - Telling user specifically if username/password is incorrect
  - No cooldown between authentication attempts helps bruteforcers
- Logs, errors, debugging help attackers more than users

Conceal your server backend!

AKA, hide your ass!

Error checking only on the client side

- Anything the client can handle, the client can compromise
- Speed/load improvement by handling user info before sending to server
- Needs to be replicated on the server

Doctrine and mindset of breaking a webapp

Doctrine and Mindset

- Forming an idea of the back end
  - File structure, dirs, files, etc.
  - Technologies used
- Thinking as the developer
  - naming convention of files and scripts
    - addUser
    - deleteUser
    - Transactions? Might be addTransaction

Mapping a webapp and finding an angle of attack

Mapping a Web Application

- Manual searching
- Recursive wget
- Techniques:
  - robot.txt
  - common file and dirs
- Spiders
  - Dictionaries

END.