Internal Implementation Disclosure

Troy Hunt
troyhunt.com
@troyhunt

Outline

- How an attacker builds a website risk profile
- Server response header disclosure
- Locating vulnerabilities based on response headers
- HTTP fingerprinting of servers
- Disclosure via robots.txt
- The risks in HTML source
- Internal error message leakage
- Lack of access controls on diagnostic data

Building a website risk profile

- **An attacker wants to understand as much as possible about the risk profile of a website in order to find vulnerabilities**
  - What are the points of untrusted data entry?
  - What sanitisation practices have been employed?
  - What frameworks and libraries is it running?
  - What can be discovered about the structure of the website?
  - Is there anything useful being disclosed in the HTML source?
  - Are there any useful internal error messages bubbling up to the browser?
  - Are there sufficient access controls on diagnostic data?

Understanding robots.txt

```
User-agent: googlebot          # all services
Disallow: /private/            # disallow this directory

User-agent: googlebot-news     # only the news service
Disallow: /                    # on everything

User-agent: *                  # all robots
Disallow: /something/          # on this directory
```

Summary

- **Keep information about frameworks and server versions private**
  - Locating known vulnerabilities based on version is easy
  - Conversely, locating sites on vulnerable versions is also easy
- **Don't *rely* on obfuscating framework and server headers**
  - There are other means of identifying the underlying technology
- **Be conscious of what you're inadvertently disclosing**
  - Is the robots.txt file pointing attackers to "hidden" paths?
  - Does the HTML source contain sensitive information?
- **Internal exceptions should *never* bubble up to end users**
- **Internal logs *must* have proper access controls, their exposure can be absolutely catastrophic**