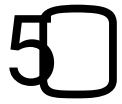

Chapter 5 - Bypassing Client-Side Controls

Bypassing Client-Side Controls

Everything client-side can be changed and modified

Transmitting Data via the Client

If the server knows and specifies a particular item of data, why would the app need to send this value to the client and then read it back?

- It removes the need to keep track of data within the session, reducing per-session data being stored
- if the app uses 2 or more backend servers, it can be difficult to manage sessions with tokens.
- the same as above but with third party servers or components
- it saves development time.

Hidden Form Fields

- Hidden HTML form fields are a common way of transmitting data that is not meant to be modified by the client

URL Parameters

- URL in browser's location bar can be easily modded
- URLs in other areas are not often thought about:
    - Embedded images are loaded using URLs containing parameters
    - where URLs containing parameters are used to load a frame's content

- when a POST form sends info to a URL with params
- when an app uses pop-up windows or other techniques to conceal the browser location bar

The Referer Header

- The referer header is mandatory
- It can be used by devs to check you came from a specific page, eg if you're sending a change password request, did u come from "changePassword.html"?

Opaque Data

- Devs can encrypt or obfuscate data
- used to check if it is a valid request, or can be decrypted and further processed

The ASP.NET ViewState

- ASP.NET uses a opaque data field by default called `ViewState`
- it contains serialized info about the state of the page
- usually base64
- by default adds anti-tampering keyed hash to it (known as MAC protection)
- some devs disable this default protection

Capturing User Data: HTML Forms

There are anti-tampering measures for HTML Forms

Length Limits

- easily circumvented by intercepting the request

Script-Based Validation

- you can disable the javascript however this may break the page
- to overcome this, enter known good, and then change this in the intercept

Disabled Elements

- If an element on a HTML form is flagged as disabled, it is not sent to the server
- if you see one, play around with it as it may have been used in the past and it may still be usable

Capturing User Data: Browser Extensions

Common Browser Extension Technologies

- Often compiled to an intermediate bytecode
- execute within a virtual machine
- may use remoting frameworks

Java

- Java applets run in the JVM
- subject to the Java Security Policy

Flash

- run in the flash virtual machine
- dead since this book was wrote so probly not usefull anymore

Silverlight

- Microsoft's alternate to flash

Approaches to Browser Extensions

- you can intercept it's requests/responses
- you can decompile it's byte code and do static/dynamic analysis

Handling Serialized data

- data is often serialized for transit
- make sure you dont break the format if you are de-serializing it, editing it and re-serializing it
- Java serialized objects can be identified by `Content-Type: application/x-java-serialized-object`

- burp suite has some good serialization tools, eg DSer
- Flash serialized objects can be identified with `Content-Type: application/x-amf`
- Silverlight serialization cab be identified with `Content-Type: application/soap+msbin1`

Obstacles to intercepting traffic from browser extensions

- SSL can mess up the intercept
- browsers may not use the same request path and thus are not captured
- may not use HTTP protocol (rare)

Handling Client-side data securely

Transmitting Data via the client

- there is no need to (and should never) send prices from the client
- if the developer *has* to send critical data via the client, it should be signed and/or encrypted
- when signing/encrypt there are two pitfalls to avoid:
    - some data is vulnerable to a replay attack, eg sending the same encrypted lower price with something of more value
    - if the user knows the plaintext, they can more easily find the encryption / any cryptographic attacks

Validating Client-Generated Data

- Everything is malicious
- Only real validation is done on the server

Logging and alerting

- anomalies should be logged
- ideally alerts should be realtime
- if bad activity is detected the client session should be termianted