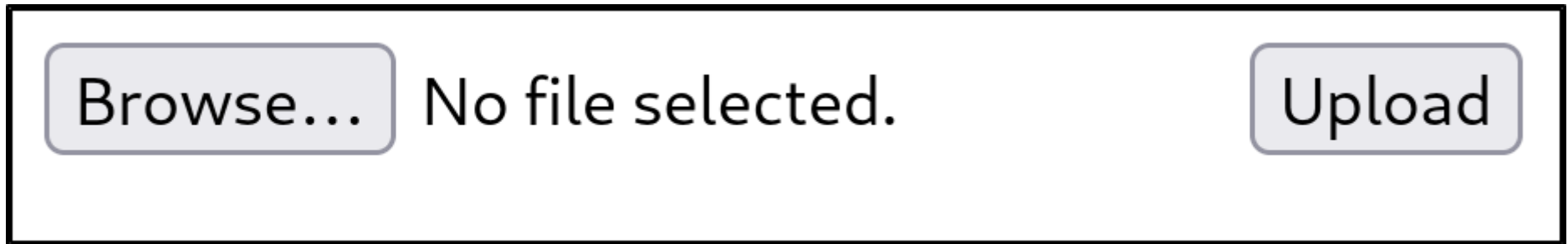


File Upload Attack – Filter Bypass



A rectangular box representing a file upload interface. Inside the box, on the left, is a light gray button with rounded corners labeled "Browse...". To the right of this button is the text "No file selected." in a standard black font. On the far right of the box is another light gray button with rounded corners labeled "Upload".

The web application allows users to upload files. If we are able to identify which directory files are uploaded to, then upload a malicious script file to web app, we could achieve remote code execution on the server through that upload script file

File Upload Attack – Filter Bypass

No file selected.

Invalid File. Please try again

A complication in this case is the fact that we don't know which file extensions are permitted for upload by the web application, so we'll need to find a way to bypass the file upload filter

File Upload Attack – Filter Bypass

Request		Response	
	Pretty	Raw	Hex
1	POST /zoc.aspx HTTP/1.1		
2	Host: 192.168.69.7		
3	Content-Length: 683		

One method of bypassing file extension filters is to record the request in BurpSuite, and then save the request to a file

File Upload Attack – Filter Bypass

```
$ ffuf -request request.txt -w asp_extensions.txt -request-proto http
```



Which is then used by the Ffuf program for fuzzing in an attempt to ID valid file extensions

File Upload Attack – Filter Bypass

```
$ ffuf -request request.txt -w asp_extensions.txt -request-proto http
```



Which is then used by the Ffuf program for fuzzing in an attempt to ID valid file extensions

File Upload Attack – Filter Bypass

vbhtml	[Status: 200, Size: 1333,
shtml	[Status: 200, Size: 1333,
config	[Status: 200, Size: 1328,
vbhtml	[Status: 200, Size: 1333,

If we can identify a valid script file extension for upload, we can then attempt the file upload attack

Privilege Escalation

SeImpersonate Privilege

```
Nombre de privilegio
=====
SeAssignPrimaryTokenPrivilege
SeIncreaseQuotaPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeCreateGlobalPrivilege
```

The SeImpersonate privilege is a feature which allows a user to perform commands in the context of other users

Privilege Escalation

SelImpersonate Privilege

```
c:\windows\system32\inetsrv>whoami  
whoami  
iis apppool\defaultapppool
```

This privilege is typically associated with service accounts, like IIS, SQL Server, and with Administrator accounts

Privilege Escalation

SeImpersonate Privilege

```
Nombre de privilegio
=====
SeAssignPrimaryTokenPrivilege
SeIncreaseQuotaPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeCreateGlobalPrivilege
```

Using SeImpersonate, attackers can elevate privileges to SYSTEM or Administrator level, either through Token Theft and / or Named Pipes

Selmpersonate – Potato Exploit

The Potato-family of Windows exploits all leverage the Selmpersonate privilege in different ways to achieve elevated access on Windows targets



Potato Exploit – JuicyPotato

Which Potato exploit to use on a target largely depends on the version of Windows being used. In this case, we'll be using the Juicy Potato variant

