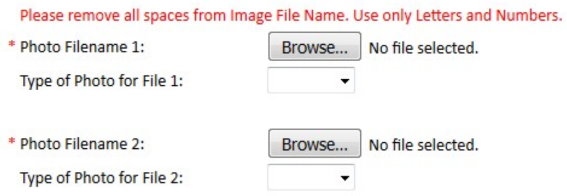
File Uploads and ASM

Dylan _yogibear, 2019-13-05

File Uploads through a WAF

Let’s say we have a web application with a form field that permits the upload of arbitrary files. It would appear to the user similar to the below:


Please remove all spaces from Image File Name. Use only Letters and Numbers.

* Photo Filename 1: Browse... No file selected.

Type of Photo for File 1:

* Photo Filename 2: Browse... No file selected.

Type of Photo for File 2:

Aside from photos, the application may permit users to upload Word documents, Excel spreadsheets, PDF’s, and so forth.

This can cause many false positives when the web application is protected by ASM, because the uploaded files may:

- Contain attack signatures. Image files may be parsed as ASCII, and suspicious-looking strings detected; Word or Excel documents may contain XSS tags or SQL injection strings. After all, Mr. ‘S valuable customers.
- Contain illegal metacharacters, like XSS tags <>
- Be so large that the maximum request size (10MB by default) is exceeded
- Trip other violations

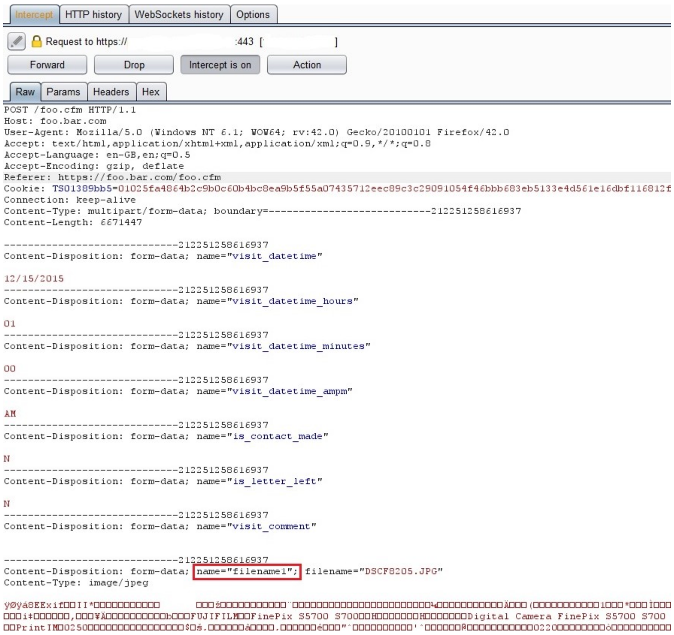
It is therefore necessary to inform ASM that a particular parameter on a form field is one that contains a file upload so that checking for attack signatures and metacharacters can be disabled.

Why not just disable the signature?

Simply, because we do not want to introduce unnecessary exposure into the security policy. Just because a particular signature causes a false positive on the file upload transaction does not mean it sho At the time of writing, ASM permits attack signatures to be selectively disabled on parameters, but not URLs.

Identify the Upload Parameter(s)

Use a HTTP inspection tool such as Fiddler, Burp or Developer Tools to determine the name of the upload parameter and URL. In this case, we are uploading a JPG file named *DSCF8205.JPG;* the para ‘filename1’. The URL is /foo.cfm.

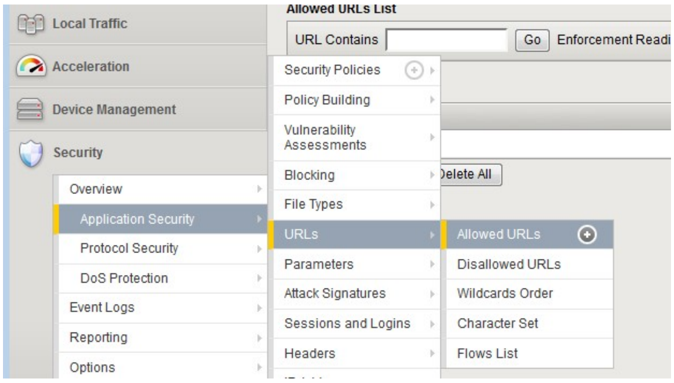


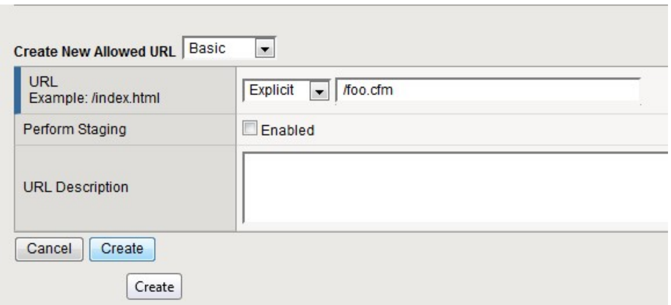
```
POST /foo.cfm HTTP/1.1
Host: foo.bar.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://foo.bar.com/foo.cfm
Cookie: TS01389bb5=01025fa4864b2c9b0c60b4bc8ea9b5f55a07435712eec89c3c29091054f46bbb683eb5133e4d561e16dbf116812f
Connection: keep-alive
Content-Type: multipart/form-data; boundary=---------------------------212251258616937
Content-Length: 6671447

-----------------------------212251258616937
Content-Disposition: form-data; name="visit_datetime"

12/15/2015
-----------------------------212251258616937
Content-Disposition: form-data; name="visit_datetime_hours"

01
-----------------------------212251258616937
Content-Disposition: form-data; name="visit_datetime_minutes"

00
-----------------------------212251258616937
Content-Disposition: form-data; name="visit_datetime_ampm"

AM
-----------------------------212251258616937
Content-Disposition: form-data; name="is_contact_made"

N
-----------------------------212251258616937
Content-Disposition: form-data; name="is_letter_left"

N
-----------------------------212251258616937
Content-Disposition: form-data; name="visit_comment"


-----------------------------212251258616937
Content-Disposition: form-data; name="filename1"; filename="DSCF8205.JPG"
Content-Type: image/jpeg

ÿØÿá8EExif□□II*□□□□□□□□□□□      □□□ž□□□□□□□□□□□¨□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Ä□□□(□□□□□□□□□□1□□□*□□□Ì□□□
□□□i‡□□□□□□,□□□¥Ä□□□□□□□□□□b□□□FUJIFILM□□FinePix S5700 S700□□H□□□□□□□H□□□□□□□Digital Camera FinePix S5700 S700
□□PrintIM□0250□□□□□□□□□□□□□□□□□$□š.□□□□□□â□□□□.□□□□□□ê□□□"ˆ□□□□□□□□□□'ˆ□□□□□□@□□□□□□□□□□□0220□□□□□□□□□ò□□□□□□□□□□□
```

NOTE: This can also be obtained from the ASM request log; however these do sometimes get truncated making it impossible to determine the parameter name if it occurs more than 5KB into the request

Define the Upload Parameter(s)

Assuming the upload is specific to a given URL, create that URL in the ASM policy.



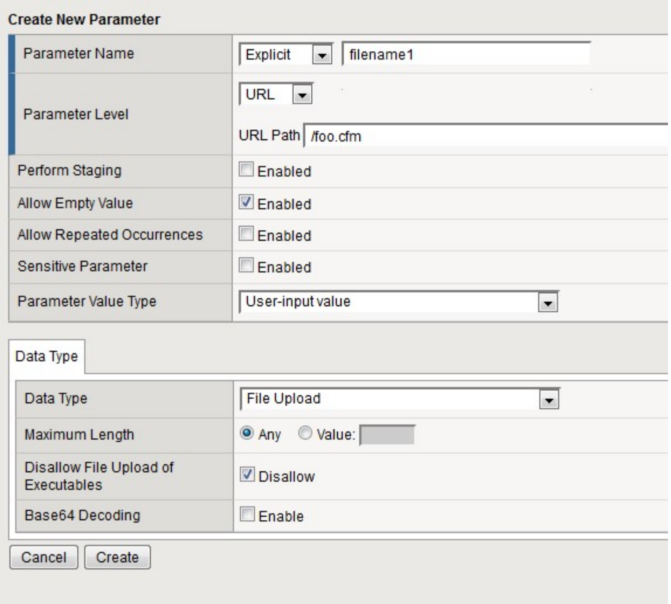

Create New Allowed URL Basic

URL Example: /index.html	Explicit /foo.cfm
Perform Staging	□ Enabled
URL Description	

Cancel Create

Create

Next, create a parameter with the name we discovered earlier, and ensure it is set to type ‘File Upload’.


Create New Parameter

Parameter Name	Explicit filename1
Parameter Level	URL URL Path /foo.cfm
Perform Staging	□ Enabled
Allow Empty Value	☑ Enabled
Allow Repeated Occurrences	□ Enabled
Sensitive Parameter	□ Enabled
Parameter Value Type	User-input value

Data Type

Data Type	File Upload
Maximum Length	● Any ○ Value:
Disallow File Upload of Executables	☑ Disallow
Base64 Decoding	□ Enable

Cancel Create

Alternate Configuration Options

- If file upload is possible in many parts of the site using the same filename, create the parameter globally without defining the URL as we did first here
- If many file upload parameters are present on a single page with a similar name (e.g. filename1, filename2, filename3…), create a wildcard parameter name filename*
- ‘Disallow file upload of executables’ is a desirable feature. It checks the magic number of the uploaded file and blocks the upload if it indicates an executable file.
- As with all ASM configurations, understanding the HTTP fields passed to the application is key

The above procedure should work for most cases, and arbitrary file uploads (except executables) should be allowed. However, there are some cases where additional configuration is required.

Didn’t Work?

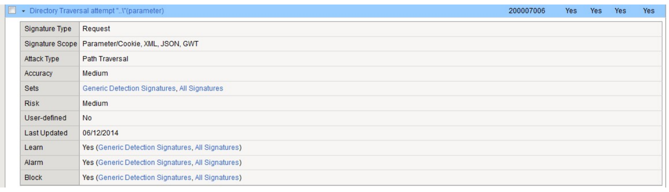
Attack signatures have a defined scope, as seen below:

***Table C.1** Attack signature keywords and usage*

Keyword	**Usage**
content	Match in the full content. See *Using the content rule option*, for syntax information.
uricontent	Match in the URI, including the query string (unless using the**objonly** modifier). See *Using the uricontent rule option*, for syntax information.

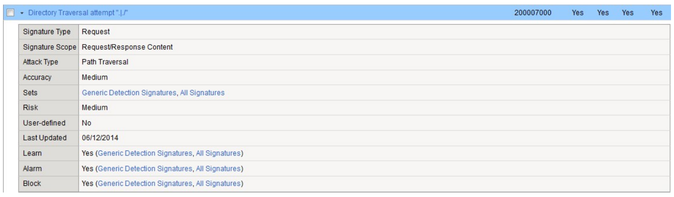
headercontent	Match in the HTTP headers. See *Using the headercontent rule option*, for syntax information.
valuecontent	Matches an alpha-numeric user-input parameter (or an extra-normalized parameter, if using the **norm** modifier); used for parameter values and XML objects. See *Using the value* information, and *Scope modifiers for the pcre rule option*, for more information on scope modifiers. An XML payload is checked for attack signatures when the**valuecontent** keyword is used in the signature. **Note:** The **valuecontent** parameter replaces the **paramcontent** parameter that was used in the Application Security Manager versions earlier than 10.0.
reference	Provides an external link to documentation and other information for the rule. See *Using the reference rule option*, for syntax information.

This information can be found in ASM under “Attack Signatures List”. As an example, search for ‘Path Traversal’ attack types and expand signature id’s 200007006 and 200007000:



□ ▾ Directory Traversal attempt "..\"(parameter) 200007006 Yes Yes Yes Yes

Signature Type	Request
Signature Scope	Parameter/Cookie, XML, JSON, GWT
Attack Type	Path Traversal
Accuracy	Medium
Sets	Generic Detection Signatures, All Signatures
Risk	Medium
User-defined	No
Last Updated	06/12/2014
Learn	Yes (Generic Detection Signatures, All Signatures)
Alarm	Yes (Generic Detection Signatures, All Signatures)
Block	Yes (Generic Detection Signatures, All Signatures)

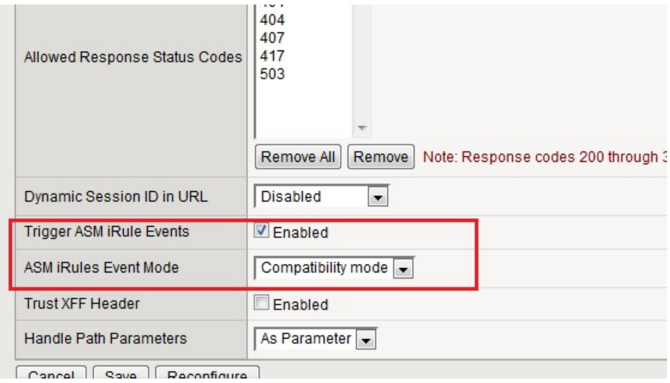


□ ▾ Directory Traversal attempt ".|./" 200007000 Yes Yes Yes Yes

Signature Type	Request
Signature Scope	Request/Response Content
Attack Type	Path Traversal
Accuracy	Medium
Sets	Generic Detection Signatures, All Signatures
Risk	Medium
User-defined	No
Last Updated	06/12/2014
Learn	Yes (Generic Detection Signatures, All Signatures)
Alarm	Yes (Generic Detection Signatures, All Signatures)
Block	Yes (Generic Detection Signatures, All Signatures)

A signature with a ‘Request’ scope does not pay any attention to parameter extraction – it just performs a bitwise comparison of the signature to the entire request as a big flat hex blob. So to prevent this disable it, (b) use an iRule to disable it on these specific requests.

Before we can use iRules on an ASM policy, we need to switch on the ‘Trigger ASM iRule Events’ setting on the main policy configuration page. Further information can be found at: https://techdocs.f5.c ip_asm/manuals/product/asm-implementations-11-5-0/27.html.



Allowed Response Status Codes: 404 407 417 503

Dynamic Session ID in URL: Disabled

Trigger ASM iRule Events: ☑ Enabled

ASM iRules Event Mode: Compatibility mode

Trust XFF Header: □ Enabled

Handle Path Parameters: As Parameter

The below is an iRule that will prevent a request meeting the following characteristics from raising an ASM violation:

- Is a POST
- URI ends with /foo.cfm
- Content-Type is ‘multipart/form-data’
- Attack Signature violation raised with signature ID 200007000

```
when ASM_REQUEST_VIOLATION {
  if {([HTTP::method] equals "POST") and ([string tolower [HTTP::path]] ends_with "/foo.cfm") and ([string tolower [HTTP::header "Content-Type"]] con
    if {([lindex [ASM::violation_data] 0] contains "VIOLATION_ATTACK_SIGNATURE_DETECTED") and ([ASM::violation details] contains "sig_data.sig_id 200
      ASM::unblock
    }
  }
}
```

What if you’re getting a lot of false positives and just want to disable attack signatures with Request scope?

```
when ASM_REQUEST_VIOLATION {
  if {([HTTP::method] equals "POST") and ([string tolower [HTTP::path]] ends_with "/foo.cfm") and ([string tolower [HTTP::header "Content-Type"]] con
    if {([lindex [ASM::violation_data] 0] contains "VIOLATION_ATTACK_SIGNATURE_DETECTED") and ([ASM::violation details] contains "context request") }
      ASM::unblock
    }
  }
}
```

But it’s not an attack signature…

False positives might also be generated by large file uploads exceeding the system-defined maximum size. This value is 10MB by default and can be configured. See https://support.f5.com/csp/article/K

However, this is a system-wide variable, and it may not be desirable to change this globally, nor may it be desirable to disable the violation. Again, we can use an iRule to disable this violation on the file u

```
when ASM_REQUEST_VIOLATION {
  if {([HTTP::method] equals "POST") and ([string tolower [HTTP::path]] ends_with "/foo.cfm") and ([string tolower [HTTP::header "Content-Type"]] con
    if {([lindex [ASM::violation_data] 0] contains "VIOLATION_REQUEST_TOO_LONG") } {
      ASM::unblock
    }
  }
}
```

ASM iRules reference

- https://clouddocs.f5.com/api/irules/ASM__violation_data.html
- https://clouddocs.f5.com/api/irules/ASM__violation.html
- https://clouddocs.f5.com/api/icontrol-soap/ASM__ViolationName.html

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.	F5 Networks	F5 Networks Ltd.	F5 Networks
Corporate Headquarters	Asia-Pacific	Europe/Middle-East/Africa	Japan K.K.
info@f5.com	apacinfo@f5.com	emeainfo@f5.com	f5j-info@f5.com