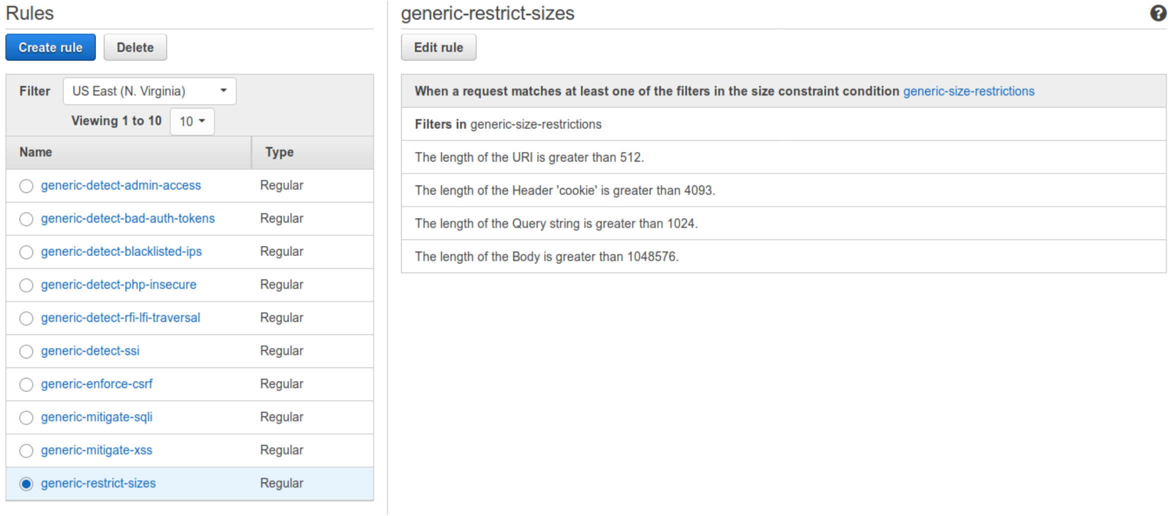
Penetration Testing Report

Attack Vector: Body Size Constraint

A limit needs to be placed on the size of the body of the incoming requests. Larger requests take longer to process. Without any restriction on the size of attachments a user can upload, the application will be vulnerable to uploads of very large size attachments. Large attachments would cause a wastage of storage and network resources.

TEST: Upload an image for a book greater than 1MB. The WAF must prevent this request from reaching the application servers.



Rules

	Name	Type
○	generic-detect-admin-access	Regular
○	generic-detect-bad-auth-tokens	Regular
○	generic-detect-blacklisted-ips	Regular
○	generic-detect-php-insecure	Regular
○	generic-detect-rfi-lfi-traversal	Regular
○	generic-detect-ssi	Regular
○	generic-enforce-csrf	Regular
○	generic-mitigate-sqli	Regular
○	generic-mitigate-xss	Regular
●	generic-restrict-sizes	Regular

generic-restrict-sizes

When a request matches at least one of the filters in the size constraint condition generic-size-restrictions

Filters in generic-size-restrictions

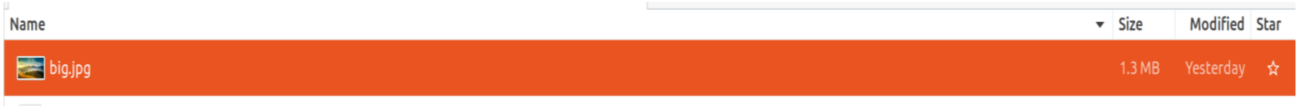
The length of the URI is greater than 512.

The length of the Header 'cookie' is greater than 4093.

The length of the Query string is greater than 1024.

The length of the Body is greater than 1048576.

Screenshot 1: Rule to restrict request body size to 1 MB.

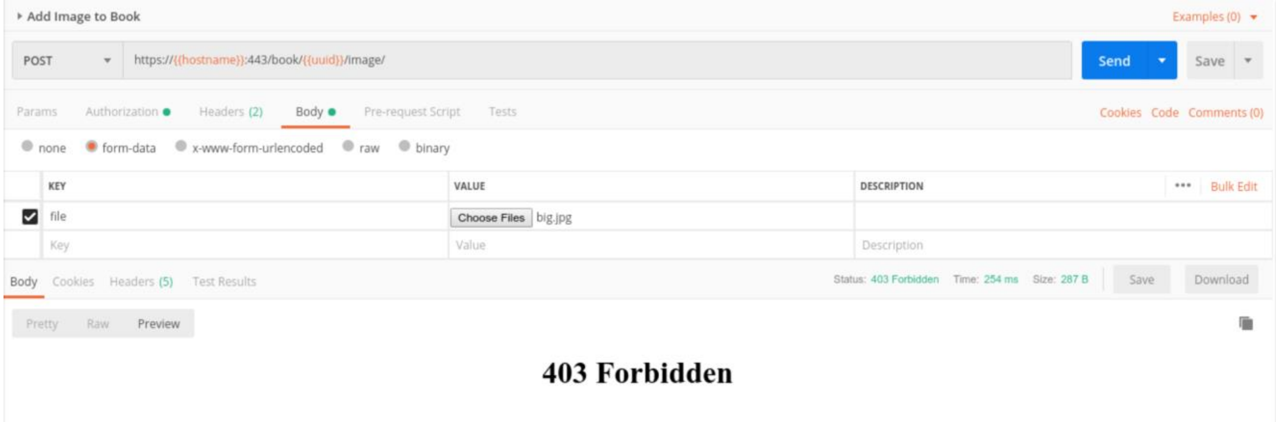


Name	Size	Modified	Star
big.jpg	1.3 MB	Yesterday	☆

Screenshot 2: Uploading file bigger than 1 MB.

RESULT:

With the WAF, the request gets blocked and response code 403 is returned.



POST https://{{hostname}}:443/book/{{uuid}}/image/

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	file	Choose Files big.jpg	
	Key	Value	Description

Status: 403 Forbidden Time: 254 ms Size: 287 B

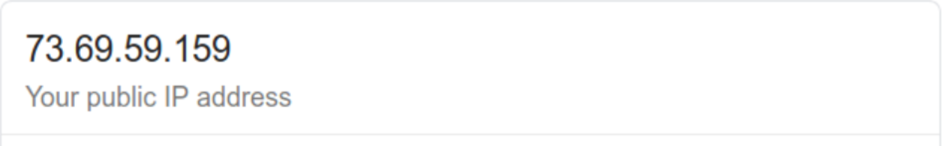
403 Forbidden

Screenshot 3: Request with large attachment getting blocked by WAF.

Attack Vector: Known Attacker Origin Mitigation

Several organizations maintain reputation lists of IP addresses that are operated by known attackers, such as spammers, malware distributors, and botnets. This solution leverages the information in these reputation lists to help block requests from malicious IP addresses.

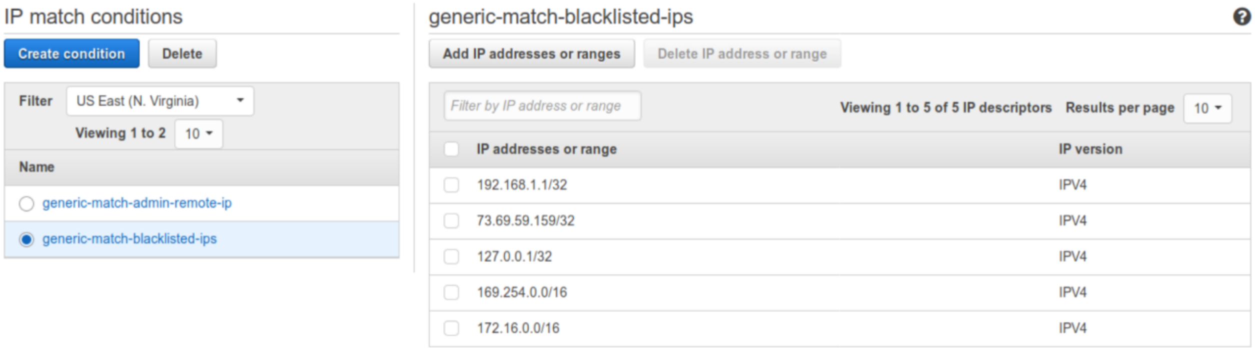
TEST: Configure the IP address of a system as a malicious IP and check if the requests get blocked.



73.69.59.159

Your public IP address

Screenshot 4: My public IP address



Filter: US East (N. Virginia)

Viewing 1 to 2

Name
○ generic-match-admin-remote-ip
● generic-match-blacklisted-ips

generic-match-blacklisted-ips

Filter by IP address or range

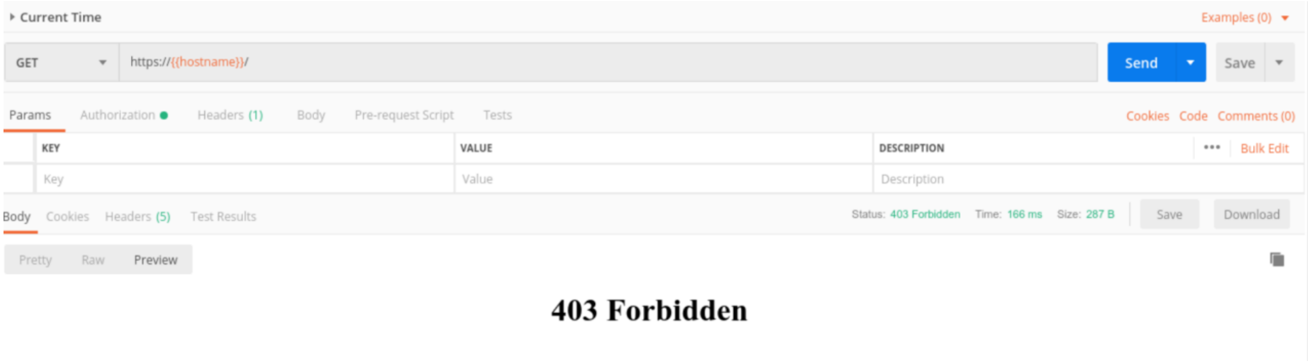
Viewing 1 to 5 of 5 IP descriptors

□	IP addresses or range	IP version
□	192.168.1.1/32	IPV4
□	73.69.59.159/32	IPV4
□	127.0.0.1/32	IPV4
□	169.254.0.0/16	IPV4
□	172.16.0.0/16	IPV4

Screenshot 5: The Public IP address added to the list of blocked IP addresses.

RESULT:

All requests from my system get blocked by the WAF. This demonstrates the ability of the WAF to block requests from known attackers.



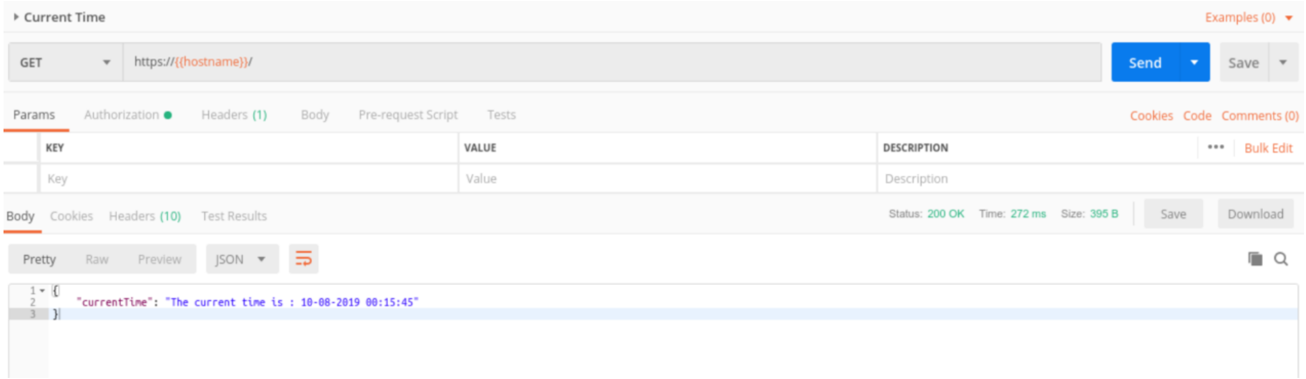
GET https://{{hostname}}/

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 403 Forbidden Time: 166 ms Size: 287 B

403 Forbidden

Screenshot 6: The request is blocked due to IP address being present in the list of blacklisted IPs.



GET https://{{hostname}}/

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 272 ms Size: 395 B

```
{
    "currentTime": "The current time is : 10-08-2019 00:15:45"
}
```

Screenshot 7: The requests from the attackers IP are not blocked without a WAF.