

Penetration Testing

Attack Vector

Used Xsaser an automatic framework to detect, exploit and report XSS vulnerability.

Launched a Kali Linux EC2 Instance, Kali linux comes with preinstalled Xsaser framework.

Command used :

```
xsser --auto -u https://csye6225-su19-sebastianc.me
```

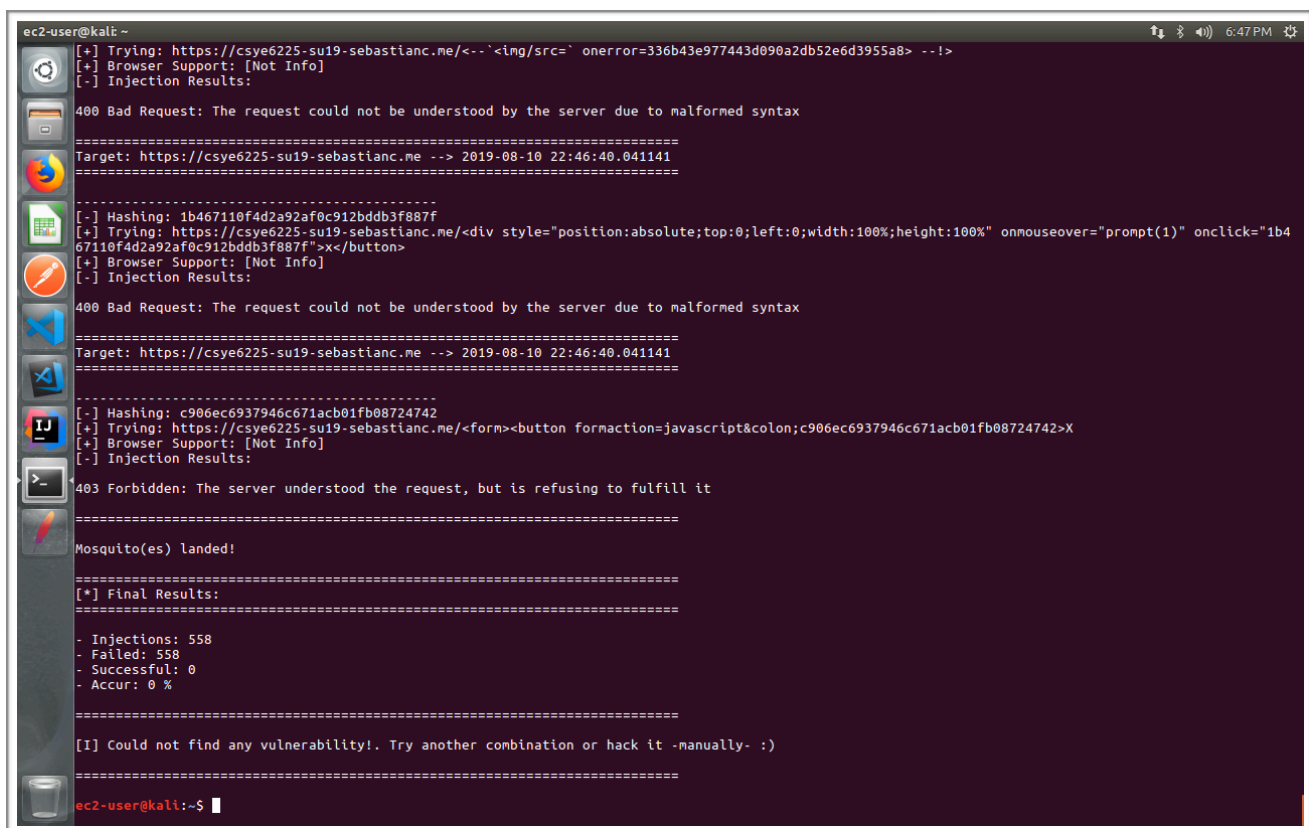
```
xsser --auto -u https://csye6225-su19-sebastianc.me -g "/book"
```

Result

With WAF in place,

WAF was able to show Malicious message in CloudWatch logs. Framework was not able to hack the api even after 558 Injections. WAF created a matrix of all blocked request.

Screenshots:



```
ec2-user@kali ~  
[+] Trying: https://csye6225-su19-sebastianc.me/<--'<img/src=' onerror=336b43e977443d090a2db52e6d3955a8> --!>  
[+] Browser Support: [Not Info]  
[-] Injection Results:  
400 Bad Request: The request could not be understood by the server due to malformed syntax  
=====
```

Target: https://csye6225-su19-sebastianc.me --> 2019-08-10 22:46:40.041141

```
=====
```

[.] Hashing: 1b467110f4d2a92af0c912bddb3f887f
[+] Trying: https://csye6225-su19-sebastianc.me/<div style="position:absolute;top:0;left:0;width:100%;height:100%" onmouseover="prompt(1)" onclick="1b467110f4d2a92af0c912bddb3f887f">x</button>
[+] Browser Support: [Not Info]
[-] Injection Results:
400 Bad Request: The request could not be understood by the server due to malformed syntax
=====

Target: https://csye6225-su19-sebastianc.me --> 2019-08-10 22:46:40.041141

```
=====
```

[.] Hashing: c906ec6937946c671acb01fb08724742
[+] Trying: https://csye6225-su19-sebastianc.me/<form><button formaction=javascript:c906ec6937946c671acb01fb08724742>X
[+] Browser Support: [Not Info]
[-] Injection Results:
403 Forbidden: The server understood the request, but is refusing to fulfill it
=====

Mosquito(es) landed!

```
=====
```

[*] Final Results:
=====

- Injections: 558
- Failed: 558
- Successful: 0
- Accur: 0 %
=====

[I] Could not find any vulnerability!. Try another combination or hack it -manually- :)

```
=====
```

ec2-user@kali:~\$

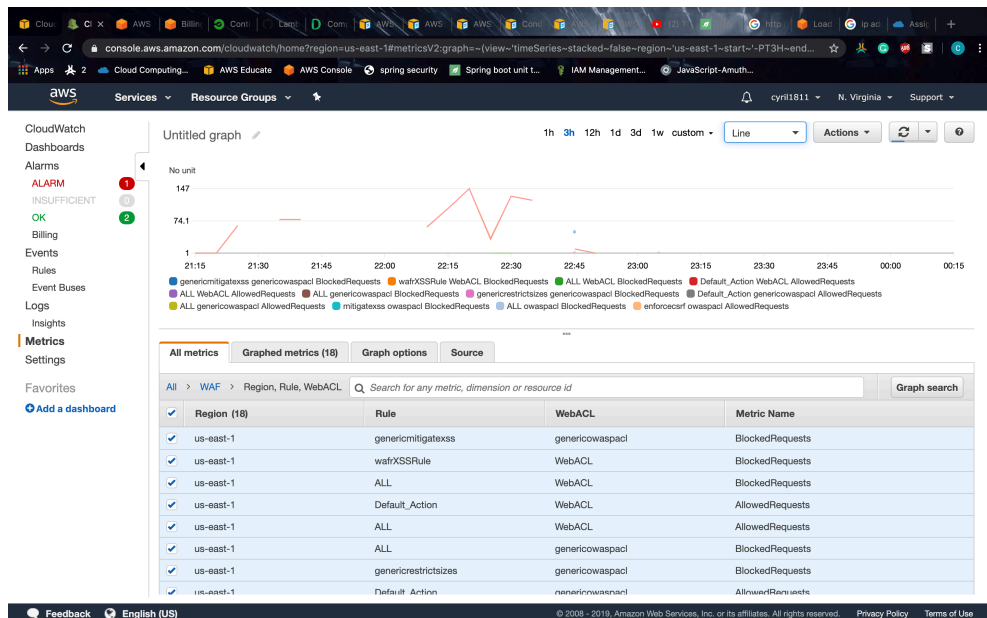
CloudWatch > Log Groups > csy6225_su2019 > webapp

Filter events

all 30s 5m 1h 6h 1d 1w custom

Time (UTC +00:00)	Message
2019-08-10	
22:46:51	2019-08-10 22:46:46.727 WARN 1347 --- [http-nio-8080-exec-4] .w.s.m.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpMediaType...
22:46:55	2019-08-10 22:46:49.850 WARN 1344 --- [http-nio-8080-exec-17] .w.s.m.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpMedia...
22:46:55	2019-08-10 22:46:50.010 WARN 1348 --- [http-nio-8080-exec-57] .w.s.m.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpMedia...
22:46:55	2019-08-10 22:46:49.728 WARN 1371 --- [http-nio-8080-exec-19] .w.s.m.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpMedia...
23:08:23	2019-08-10 23:08:17.859 WARN 1344 --- [http-nio-8080-exec-105] c.n.w.r.UserRestController : User Registration Failed
23:14:08	2019-08-10 23:14:08.439 ERROR 1346 --- [http-nio-8080-exec-6] o.s.b.w.servlet.support.ErrorPageFilter : Forwarding to error page from request [/+ADw-htm...
23:14:08	org.springframework.security.web.firewall.RequestRejectedException: The request was rejected because the URL contained a potentially malicious String ";" a...
	org.springframework.security.web.firewall.StrictHttpFirewall.rejectedBlacklistedUrls(StrictHttpFirewall.java:325) ~[spring-security-web-5.1.5.RELEASE.jar:5.1.5.RELEASE]
	at org.springframework.security.web.firewall.StrictHttpFirewall.getFirewalledRequest(StrictHttpFirewall.java:293) ~[spring-security-web-5.1.5.RELEASE.jar:5.1.5.RELEASE]
	at org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:194) ~[spring-security-web-5.1.5.RELEASE.jar:5.1.5.RELEASE]
	at org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:178) ~[spring-security-web-5.1.5.RELEASE.jar:5.1.5.RELEASE]
	at org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:357) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]
	at org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:270) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]
	at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) ~[catalina.jar:9.0.21]
	at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) ~[catalina.jar:9.0.21]
	at org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestContextFilter.java:99) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]
	at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:107) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]
	at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) ~[catalina.jar:9.0.21]
	at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) ~[catalina.jar:9.0.21]
	at org.springframework.web.filter.FormContentFilter.doFilterInternal(FormContentFilter.java:92) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]
	at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:107) ~[spring-web-5.1.7.RELEASE.jar:5.1.7.RELEASE]

2019-08-10 22:46:46.724 ERROR 1347 --- [http-nio-8080-exec-4] o.s.b.w.servlet.support.ErrorPageFilter : Forwarding to error page from request [/+ADw-htm+AD4APA-body+AD4APA-div+AD4-top secret+ADw-/div+AD4APA-/body+AD4APA-/html+AD4-.toXMLString().match(/.*m),9afffd96db18083de568daa1f2c6050c] due to exception [The request was rejected because the URL contained a potentially malicious String ";"]



Without WAF,

There were error messages in cloud watch which said that spring hibernate rejected the sql injection.

Why did you choose this specific attack vector?

Xsser framework injects api with many attacks like sql injection cross site scripting attack with minimal code of lines. It makes our job easy.

Attack Vector:

IP Block

With WAF,

We blacklisted a particular IP and tried hitting our application using that ip.

Result

We received 403 Forbidden Error. User was not able to access content of api.

Without

We were not able to blacklist a particular IP address

Why we choose this attack?

It helped us to identify that by using WAF we can blacklist a particular API and keep our application safe from hackers.

Screenshot

