

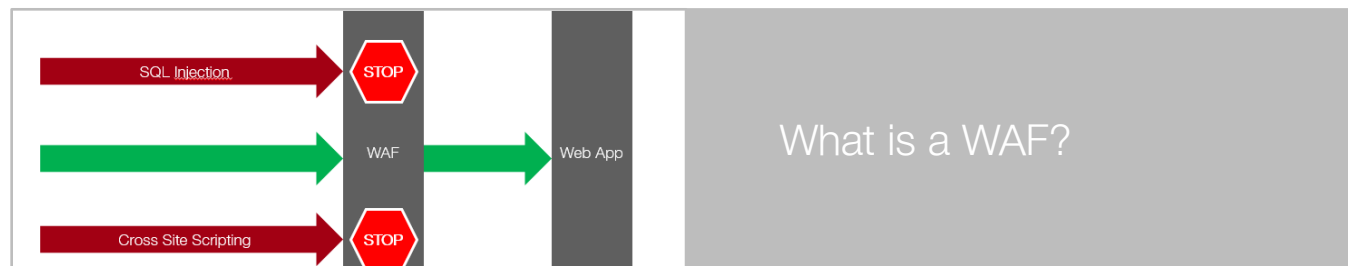
# Securing applications and managing WAFs at a large scale

Fränzi Bühler  
KCD Zurich  
June 13 2024



Story time ...

This talk is about what a WAF is, why we need one and how we manage a large amount of WAFs at SBB.



```
...-WebKitFormBoundaryQvW0UG31cEzp2m57
Content-Disposition: form-data; name="upload"
Content-Type: application/javascript
..../opt/tomcat/webapps/webshell.war
--WebKitFormBoundaryQvW0UG31cEzp2m57-
```

Why a WAF?



WAFs scaled & automated  
@SBB

# Fränzi Bühler

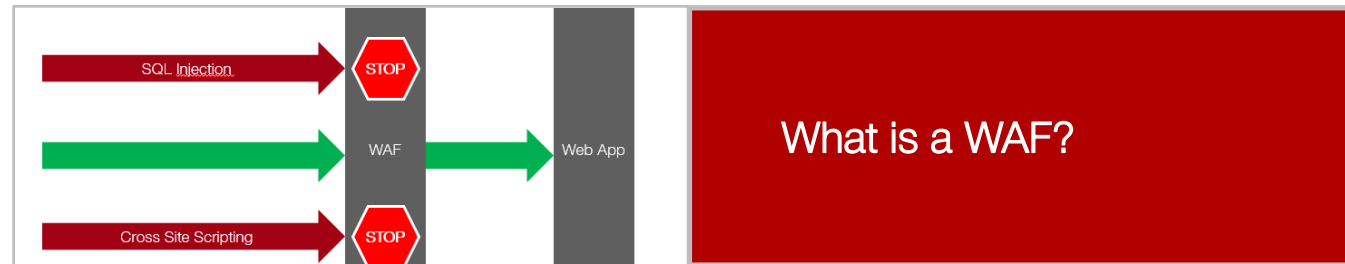
I'm a huge fan of WAFs.

Security Engineer SBB  
OWASP CRS Developer





# A WAF shields our web applications from attacks.



What is a WAF?

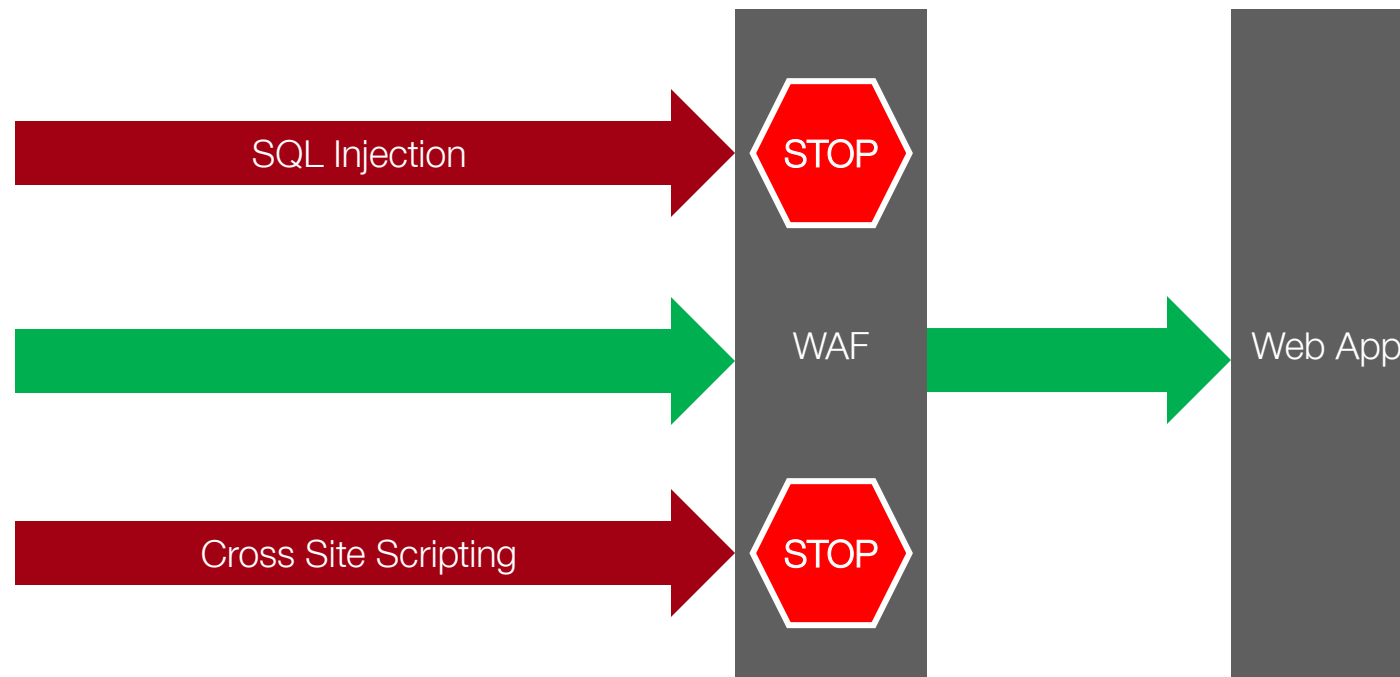
```
...-WebKitFormBoundaryQvW0UG31cEzp2m57-
Content-Disposition: form-data; name="upl
.../opt/tomcat/webapps/webshell.war
...-WebKitFormBoundaryQvW0UG31cEzp2m57-
```

Why a WAF?



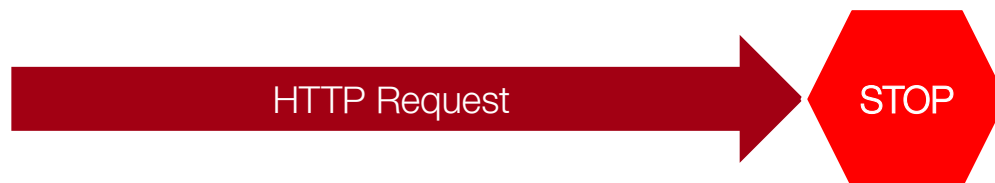
WAFs scaled & automated  
@SBB

A WAF is the first line of defense and it blocks attacks before they reach the application.



A WAF detects and blocks attacks by identifying specific patterns.

https://example.com/get-file?file=../..../etc/passwd



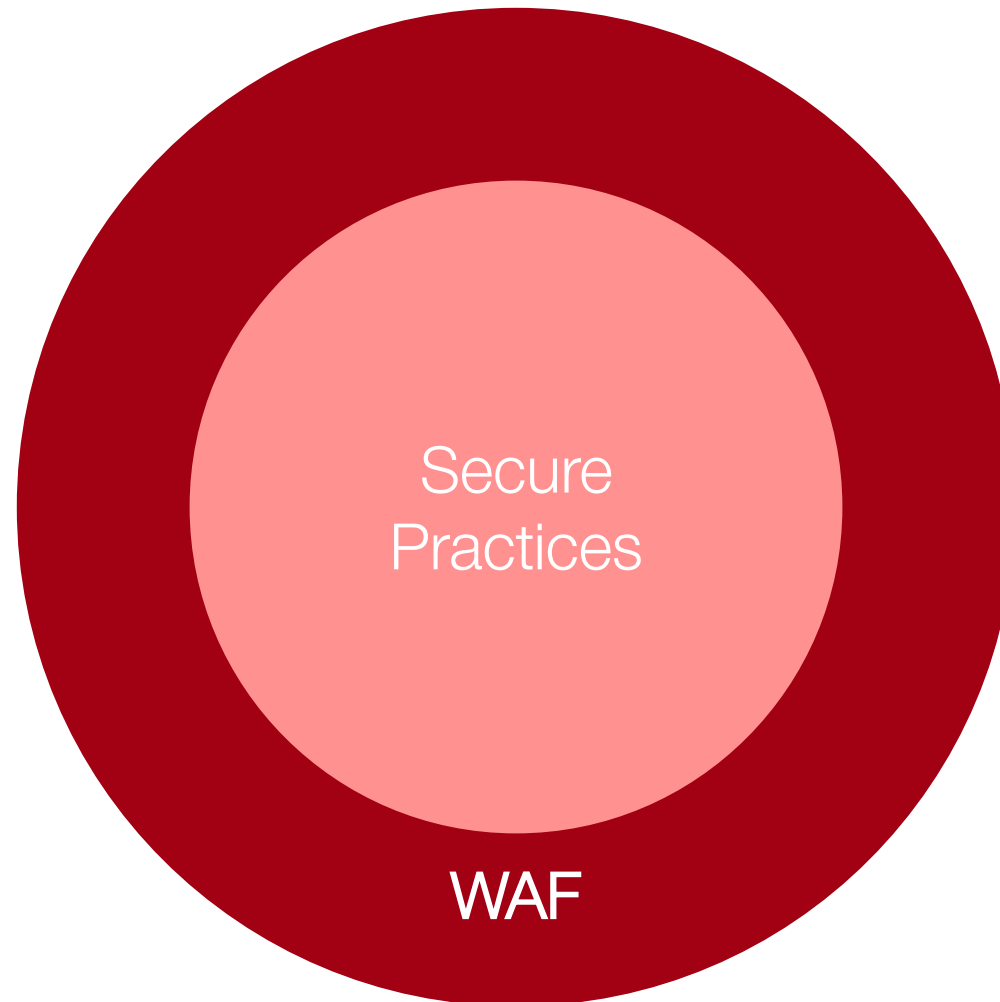
→ Path Traversal Attack (../)

A WAF detects many common web application attacks.

Data-Leakage  
Remote-Command-Execution  
Remote-File-Inclusion  
Multipart-Attacks  
SQL-Injection  
PHP HTTP-Method Java  
Cross-Site-Scripting  
Scanners  
Local-File-Inclusion  
Session-Attacks  
Web-Shells



A WAF is an additional layer of defense if other measures fail.



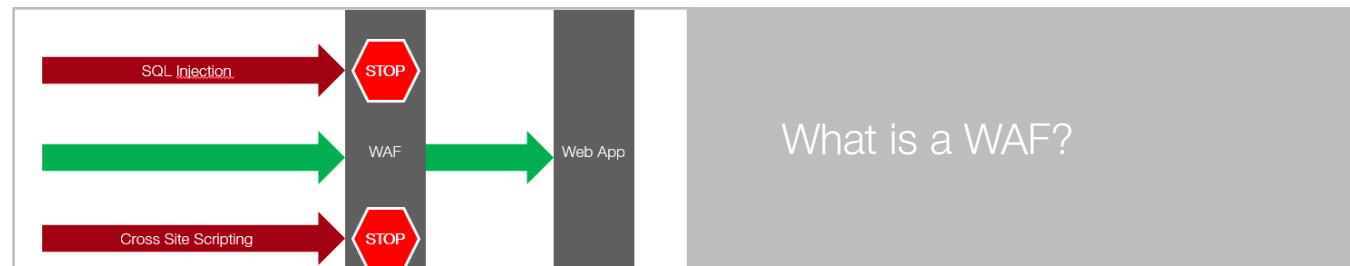
A false positive is a legitimate request that looks like an attack but actually isn't.

`https://example.com/person?name=Axel+Schmidt`



→ Remote Command Execution: Direct Unix Command Execution  
«axel» is a CLI tool to download files

Now that we know what a WAF is, let's understand why we need one by using an example.



```

--WebKitFormBoundaryQvW0UG31cEzp2m57
Content-Disposition: form-data; name="upl
../opt/tomcat/webapps/webshell.war
--WebKitFormBoundaryQvW0UG31cEzp2m57-

```

Why a WAF?



WAFs scaled & automated  
@SBB

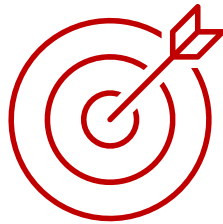
The vulnerability behind the story: CVE-2023-50164 uses a path traversal attack to upload a malicious file outside of web root.

```
-----WebKitFormBoundaryQvW0UG31cEzp2m57
Content-Disposition: form-data; name="uploadFileName"

../../../../opt/tomcat/webapps/webshell.war
-----WebKitFormBoundaryQvW0UG31cEzp2m57--
```

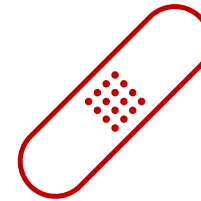
→ Path Traversal Attack (../)

This is exactly the use case of a WAF: a general rule that protects by default.



## Path Traversal Attack

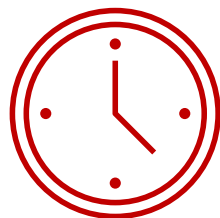
../.. is a standard WAF rule



## Virtual Patching

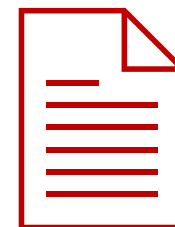
If no rule applies, I can write my own patterns

It gives us time to analyze the vulnerability and properly plan a software update.



It buys us time!

But we still patch or  
update the application

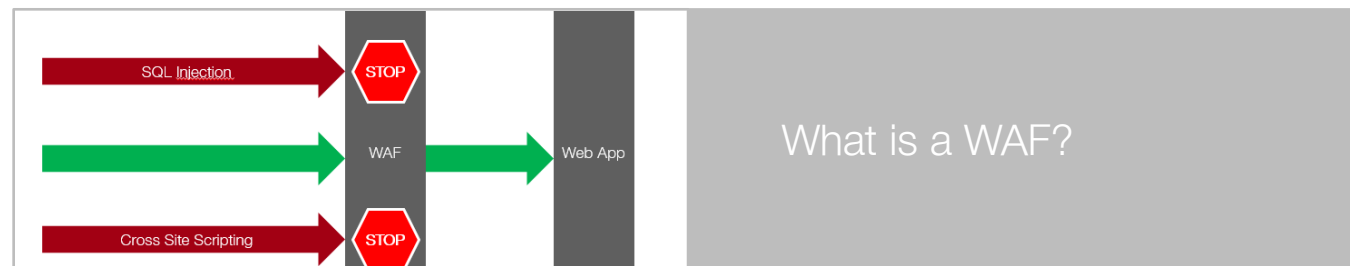


Logging & Monitoring

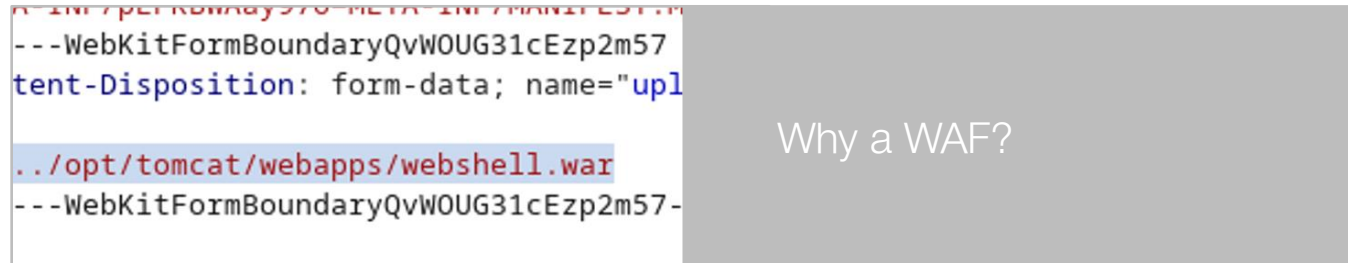
Deep insights into HTTP  
requests and responses



# SBB also uses WAFs to enhance its security posture.



What is a WAF?



The screenshot shows a web browser displaying a malicious payload. The text is partially obscured by a blue highlight. The visible text includes: `--WebKitFormBoundaryQvW0UG31cEzp2m57`, `tent-Disposition: form-data; name="upl`, `../opt/tomcat/webapps/webshell.war`, and `--WebKitFormBoundaryQvW0UG31cEzp2m57-`.

Why a WAF?



WAFs scaled & automated  
@SBB



At SBB, a dedicated team takes care of WAFs.

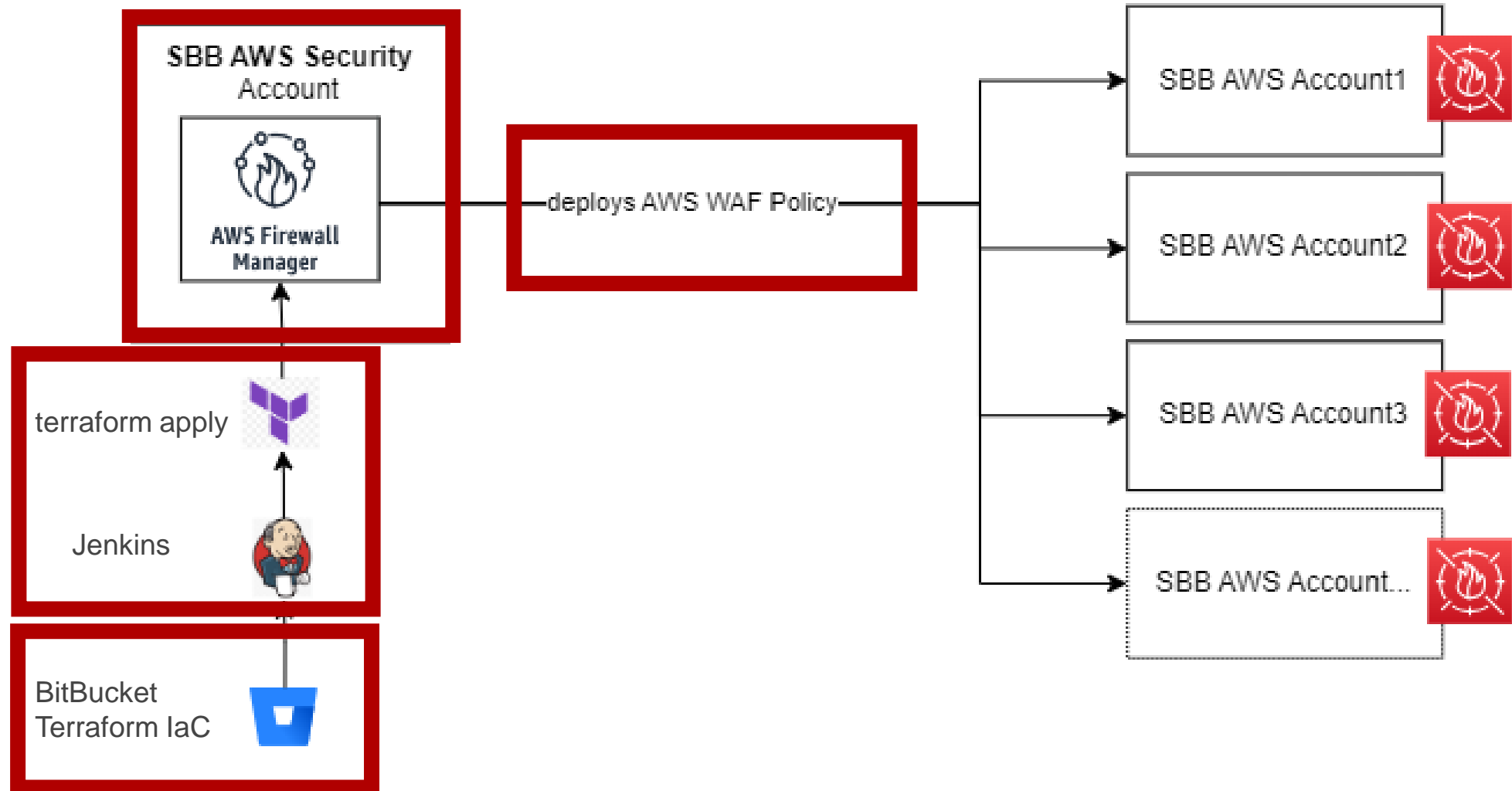




When many applications require many WAFs, a high level of automation is required.



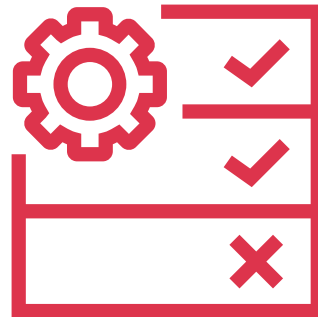
We store WAF policies centrally and automatically deploy them to the various customer accounts.



Some customers tune themselves and send us their pull request.

```
241 241         key    = "aws-waf:managed:aws:core-rule-set:CrossSiteScripting_URIPath"
242 242     }
243 243 }
244 244
245 245 # Check if URI Path is IN Regex String, if yes -> allow (Block Negated)
246 246 statement {
247 247     not_statement {
248 248         statement {
249 249             regex_match_statement {
250 250 -             regex_string = "^(\\/plugins\\/.*|\\demo0\\/.*)$"
250 250 +             regex_string = "^(\\/plugins\\/.*|\\api\\v5\\stuff.*)$"
251 251         field_to_match {
252 252             uri_path {}
253 253         }
254 254         text_transformation {
255 255             priority = 0
256 256             type      = "NONE"
257 257         }
```

AWS regularly releases rule set updates to protect against the latest attacks.



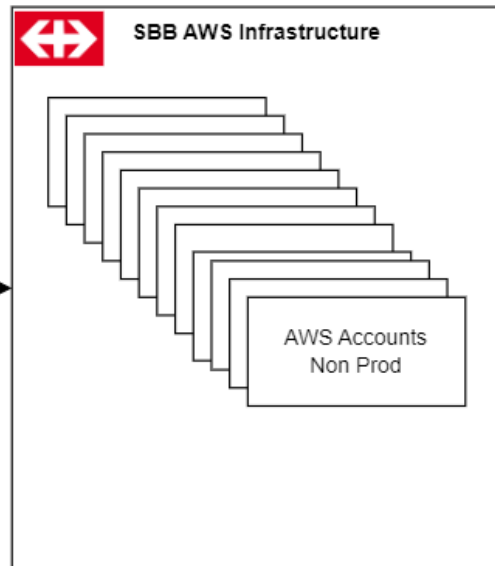
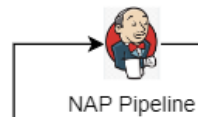
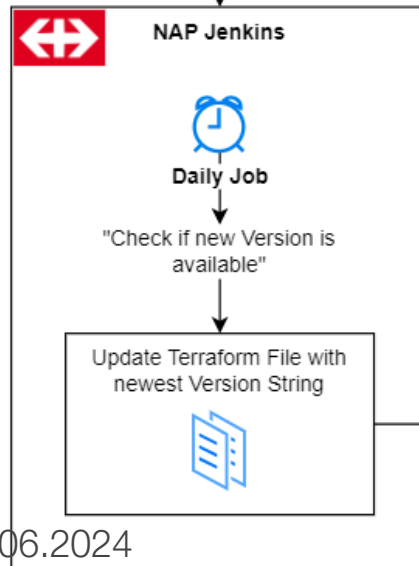
AWS Managed rules



# We daily check for rule sets updates and adopt them.



```
variable "var_nonprod_AWSManagedRulesCommonRuleSet" {  
  description = "Variable for Rule Version:AWSManagedRulesCommonRuleSet"  
  type        = string  
  default     = "Version_1.7"  
}
```



We also have some challenges to solve and can always improve.



work in progress ...

We have learned a lot and I would like to share our findings and key takeaways.



Use a WAF!



Establish Security **BEFORE** Production -> Get involved early



Block by default and build exceptions

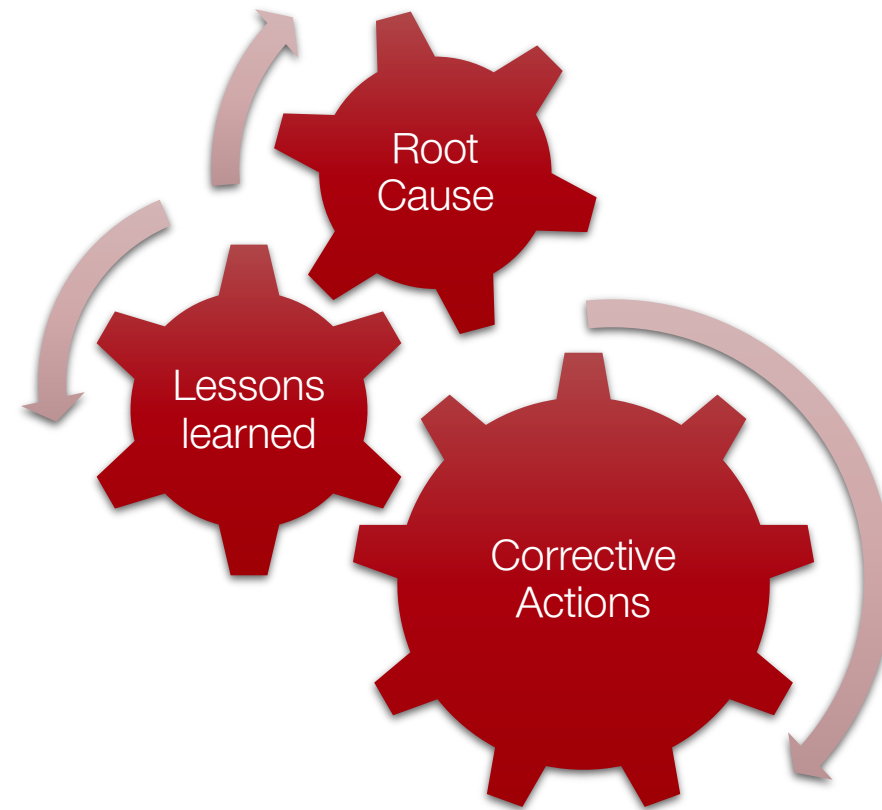


Centralize WAF Policies



Automate WAF Deployments

One more thing: If I could give one more piece of advice:  
Live a postmortem culture.



If you want to play with and experience a WAF, here are some OWASP CRS links:



<https://coreruleset.org/docs>



<http://sandbox.coreruleset.org/get-files?file=../etc/passwd>

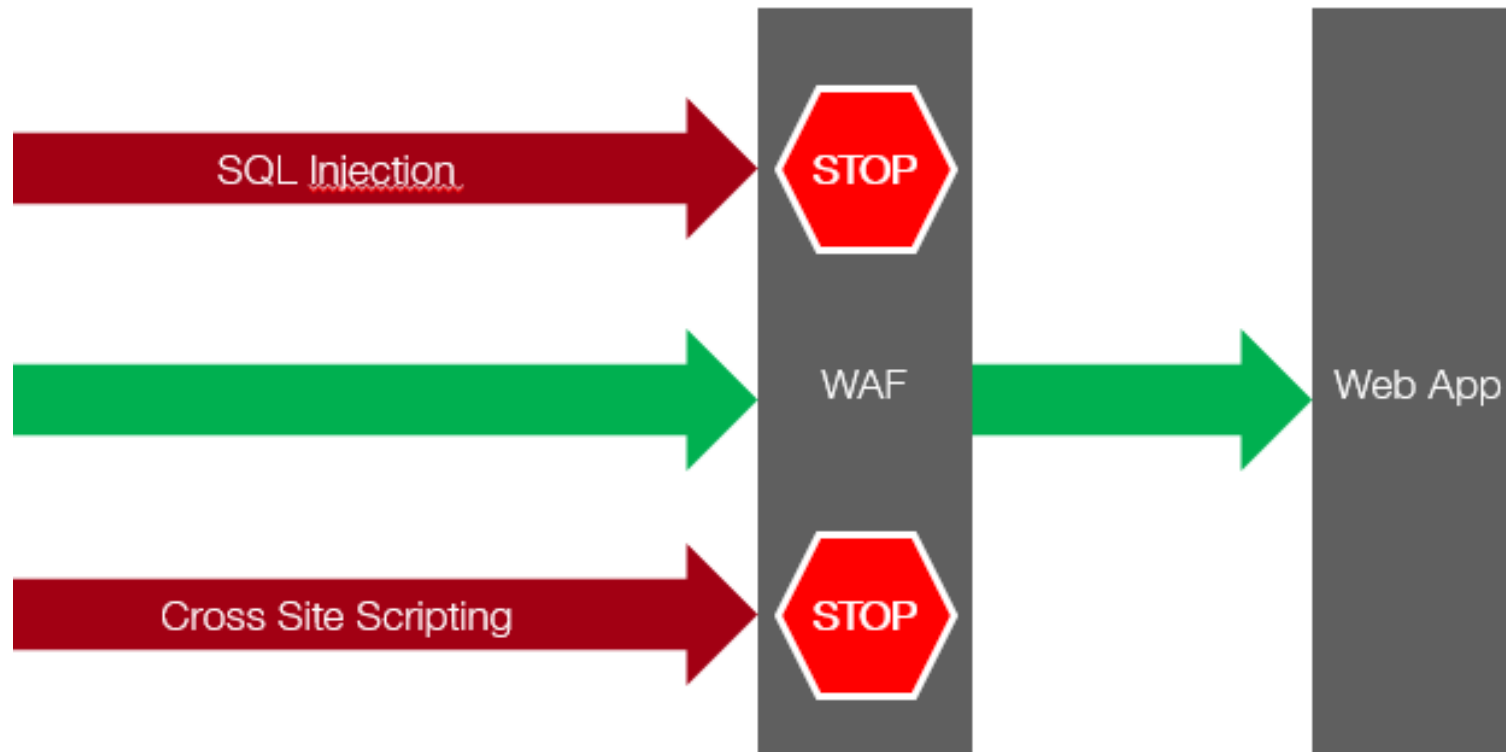


```
curl -H "x-format-output: txt-matched-rules" \  
http://sandbox.coreruleset.org/get-files?file=../etc/passwd
```



```
curl -H "x-format-output: txt-matched-rules" \  
'https://sandbox.coreruleset.org/person?name=Axel+Schmidt'
```

In summary, a WAF helps you protect your applications from common web application attacks and enhance your application security posture.



Contact:

Franziska Bühler  
Security Engineer

[franziska.buehler@sbb.ch](mailto:franziska.buehler@sbb.ch)

Questions?