Secure Your Code

Injections and Logging

Philipp Krenn @xeraa

elastic

Let's talk about security...


THIS IS FINE.

A1:2017-Injection

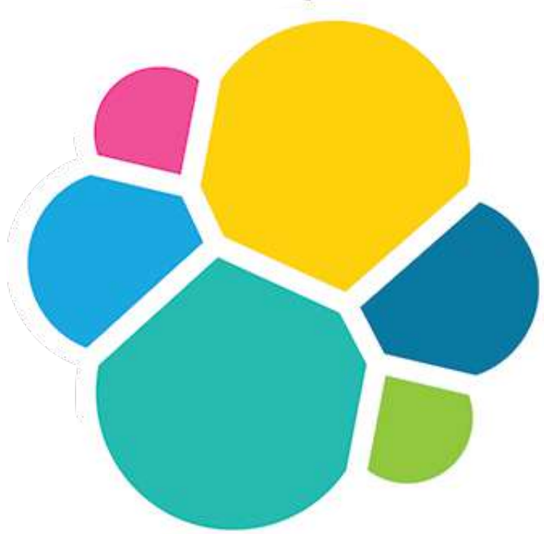
https://www.owasp.org/index.php/Top_10-2017_Top_10


THIS IS FINE.

A10:2017-Insufficient Logging & Monitoring

https://www.owasp.org/index.php/Top_10-2017_Top_10


TH-
THIS IS NOT FINE!!
OH MY GOD EVERY THINGS ON FIRE
WHAT THE HELL IS MY PROBLEM



elastic

Developer 🥑

Disclaimer

I build **highly** monitored Hello World apps

Hello World of SQL Injection:
https://xeraa.wtf

https://xeraa.wtf/login.php 🤔

Hello World of SQL Injection

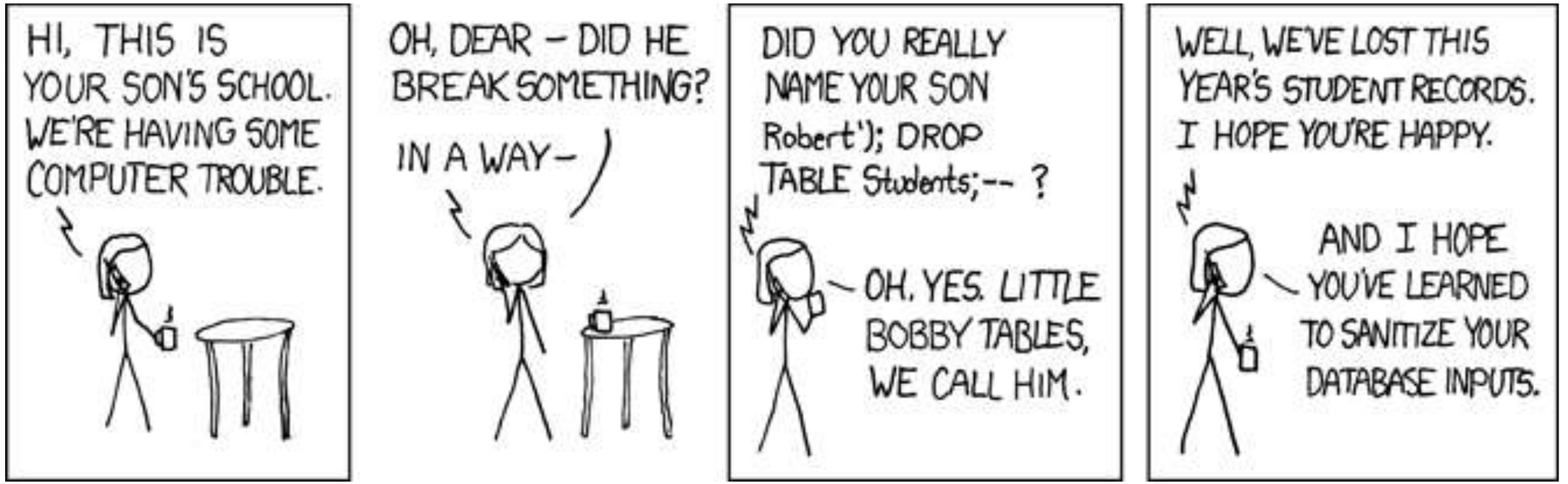
```
$sql = "SELECT *
        FROM `employees`
        WHERE name='$name' AND password=SHA1('$password')";
```

Hello World of SQL Injection

```
' or true --
```

SQL injection, or as I like to call it, accidental GraphQL.

— *https://twitter.com/markdalgleish/status/1186413157276762112*


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

https://xeraa.wtf/read.php?id=1 🤔


```
sqlmap --url "https://xeraa.wtf/read.php?id=1" --purge
```

More SQL Injections

```
$sql = "SELECT * FROM employees WHERE id = " . trim($_GET["id"]);
error_log("SQL query [read.php]: " . $sql . "\n", 3, "/var/log/app.log");

mysqli_multi_query($link, $sql);
if($result = mysqli_use_result($link)){
    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
```

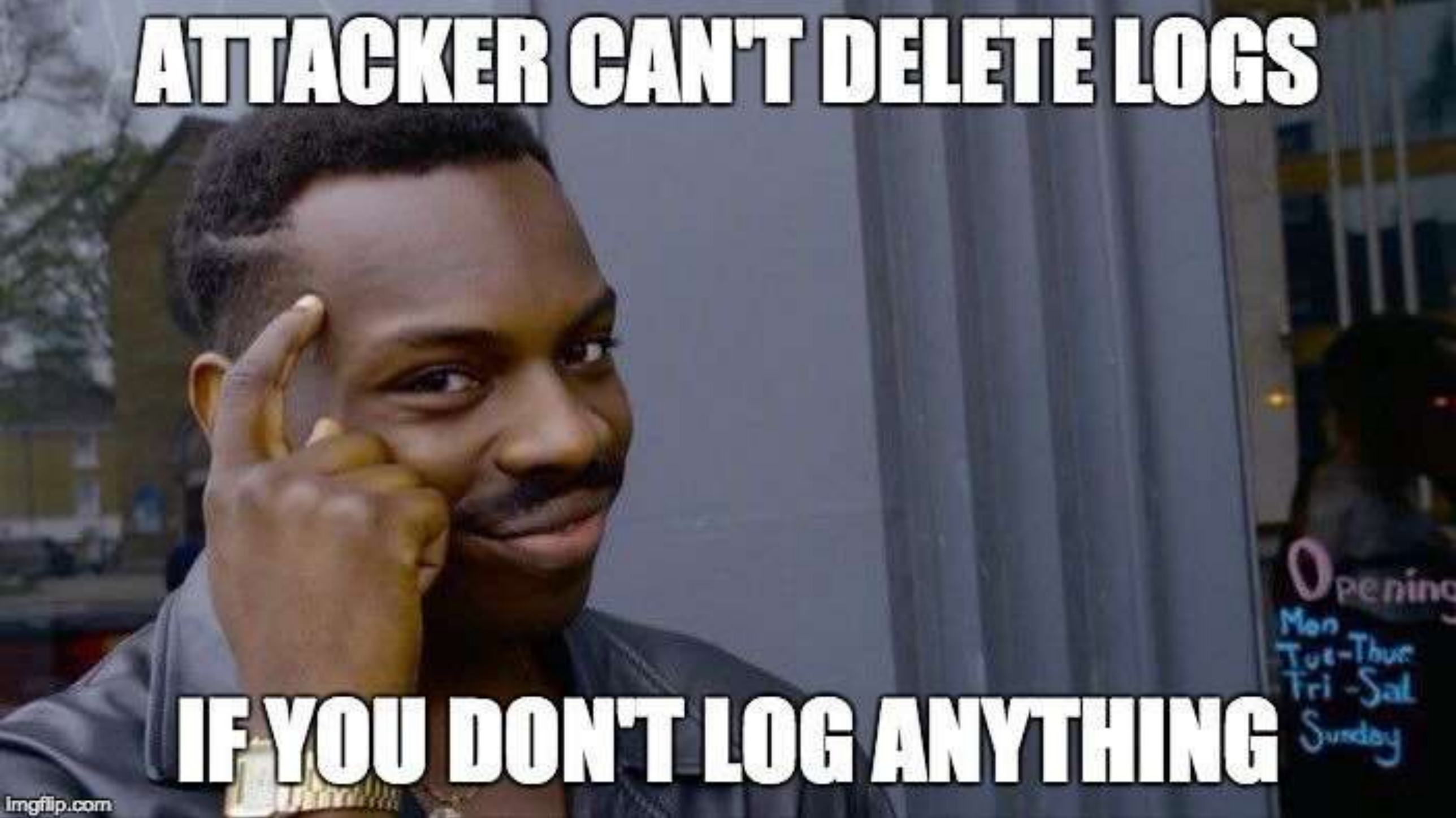
More SQL Injections

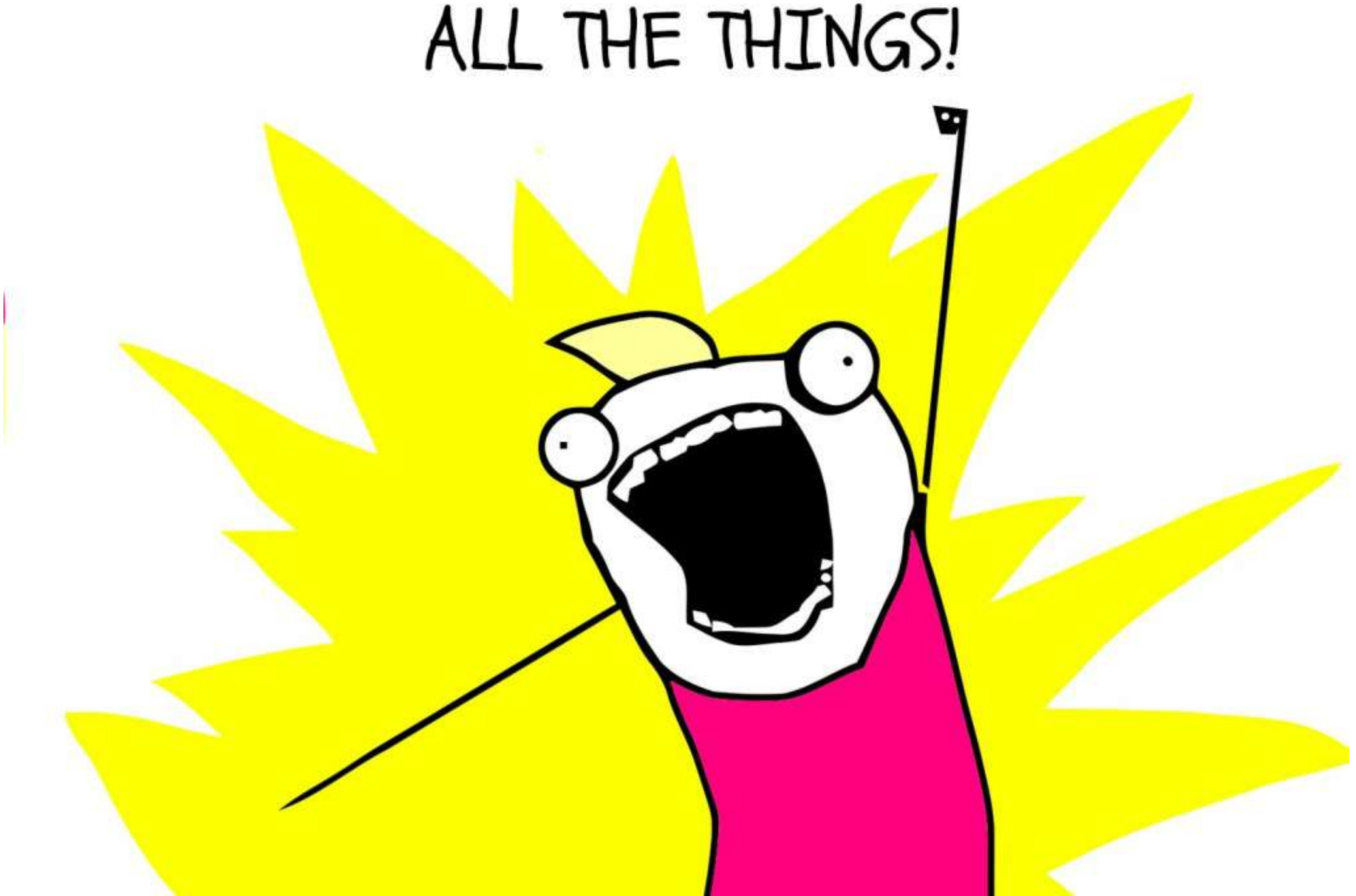
```
;INSERT INTO employees (name) VALUES ('Bad Actor')
```

No Escaping Either

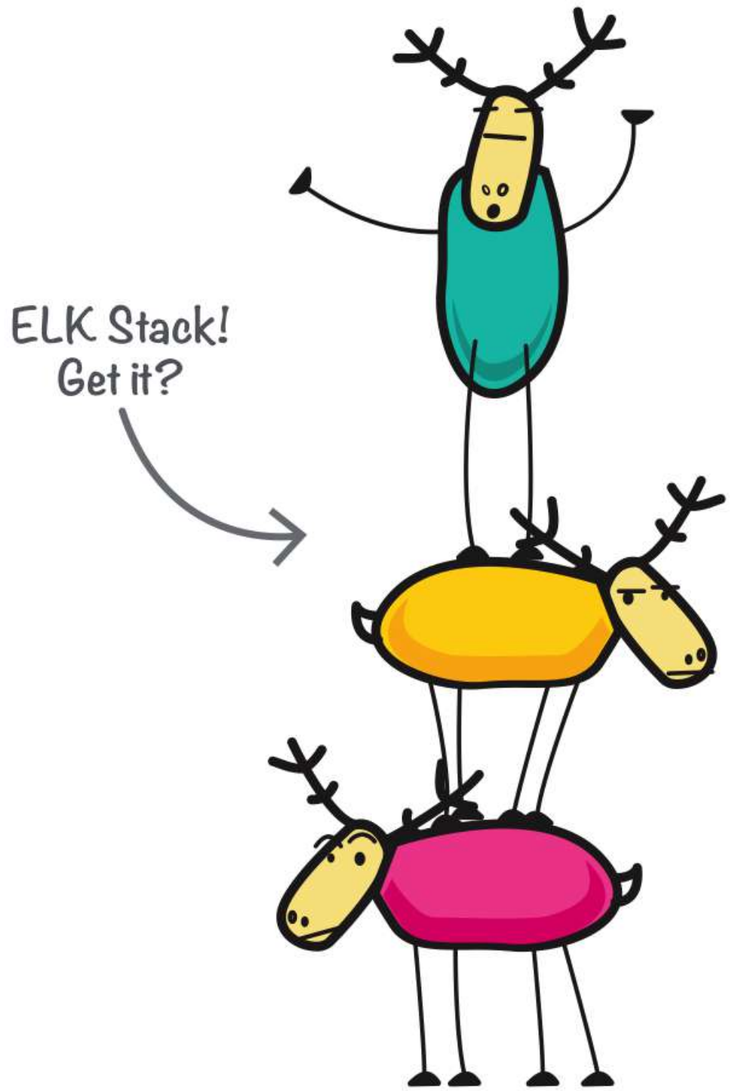
```
;INSERT INTO employees (name) VALUES ('<script>alert("Hello Friend")</script>')
```

What's Going on in Our App?


ATTACKER CAN'T DELETE LOGS
IF YOU DON'T LOG ANYTHING
Mon
Tue-Thur
Fri -Sat
Sunday
imgflip.com


ALL THE THINGS!

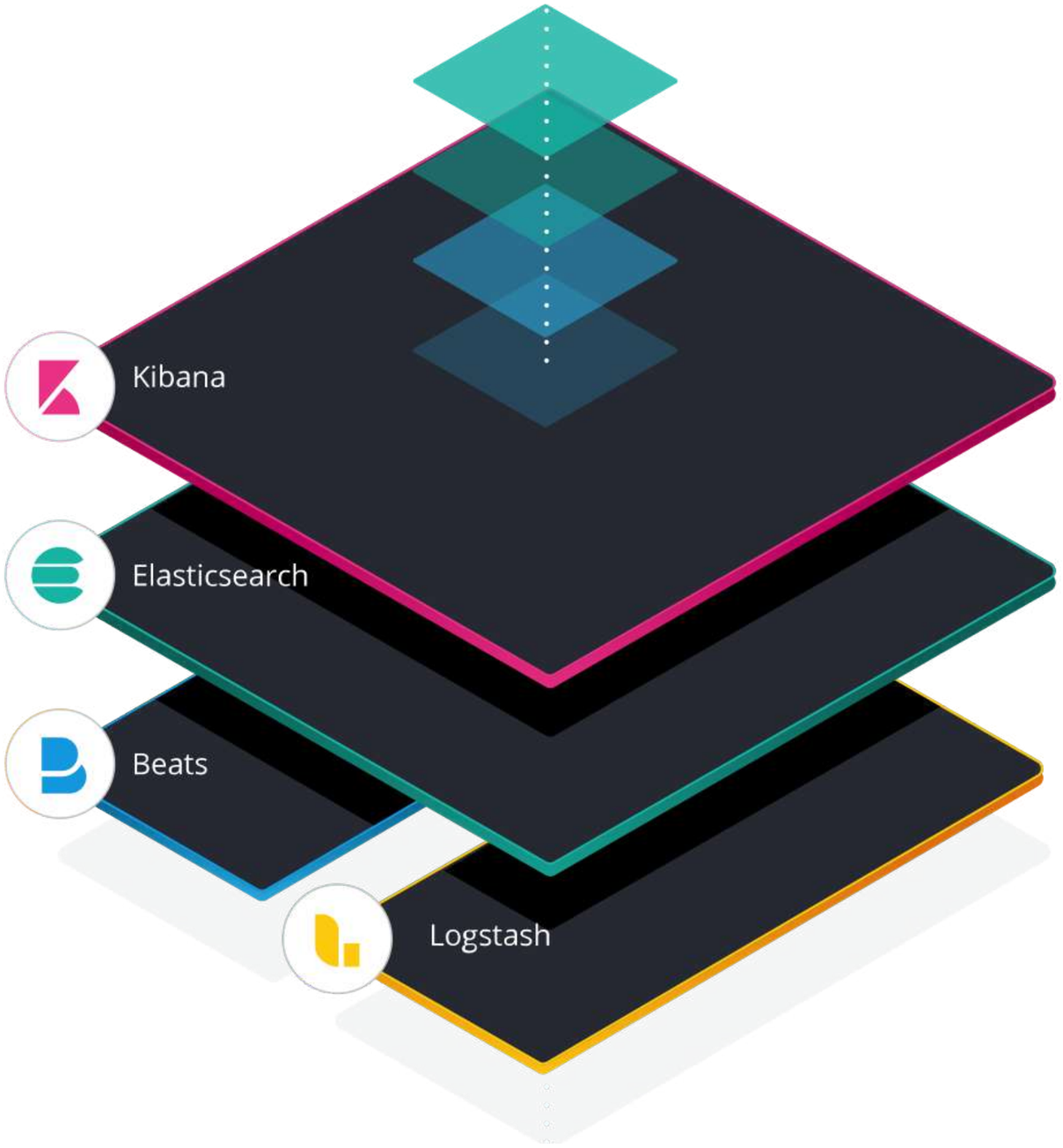

me looking
for the bug
7.2 GB
of log file



E Elasticsearch

L Logstash

K Kibana


Kibana
Elasticsearch
Beats
Logstash

Stream Settings

application : "app"

Customize Highlights 10/12/2019 3:39:26 AM Stream live

read.php

Timestamp	Message
	56a74,0x7176786a71),NULL-- OeFT
Oct 12, 2019 @ 03:37:56.983	SQL query [read.php]: SELECT * FROM employees WHERE id = -2655 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a767171,(CASE WHEN (6534=6534) THEN 1 ELSE 0 END),0x7176786a71),NULL-- rwEP
Oct 12, 2019 @ 03:39:01.988	SQL query [read.php]: SELECT * FROM employees WHERE id = id=-4156 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a767171,0x6e6566704b7445664d4574464e6b6258795976777868 4b6c704875446b4c415453504645 51756a74,0x7176786a71),NULL-- OeFT
Oct 12, 2019 @ 03:39:26.989	SQL query [read.php]: SELECT * FROM employees WHERE id = 4
Oct 12, 2019 @ 03:39:26.990	SQL query [read.php]: SELECT * FROM employees WHERE id = id=-4156 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a767171,0x6e6566704b7445664d4574464e6b6258795976777868 4b6c704875446b4c415453504645 51756a74,0x7176786a71),NULL-- OeFT
Oct 12, 2019 @ 08:02:34.686	SQL query [read.php]: SELECT * FROM employees WHERE id = 1
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 2
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 1
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 2
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 3
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 4
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 3
Oct 12, 2019 @ 08:02:34.687	SQL query [read.php]: SELECT * FROM employees WHERE id = 4

06 PM 09 PM Sat 12 03 AM 06 AM 09 AM 12 PM 03 PM

No additional entries found Load again

DELETE or DROP?

OWASP ModSecurity Core Rule Set

THE 1ST LINE OF DEFENSE

elastic

Open source

Cross-platform web application firewall (WAF)

Visibility into HTTP(S) traffic

Rules to implement protections

OWASP ModSecurity Core Rule Set (CRS) Version 3

- HTTP Protocol Protection
- Real-time Blacklist Lookups
- HTTP Denial of Service Protections
- Generic Web Attack Protection
- Error Detection and Hiding

Commercial Rules from Trustwave SpiderLabs

- Virtual Patching
- IP Reputation
- Web-based Malware Detection
- Webshell / Backdoor Detection
- Botnet Attack Detection
- HTTP Denial of Service Detection

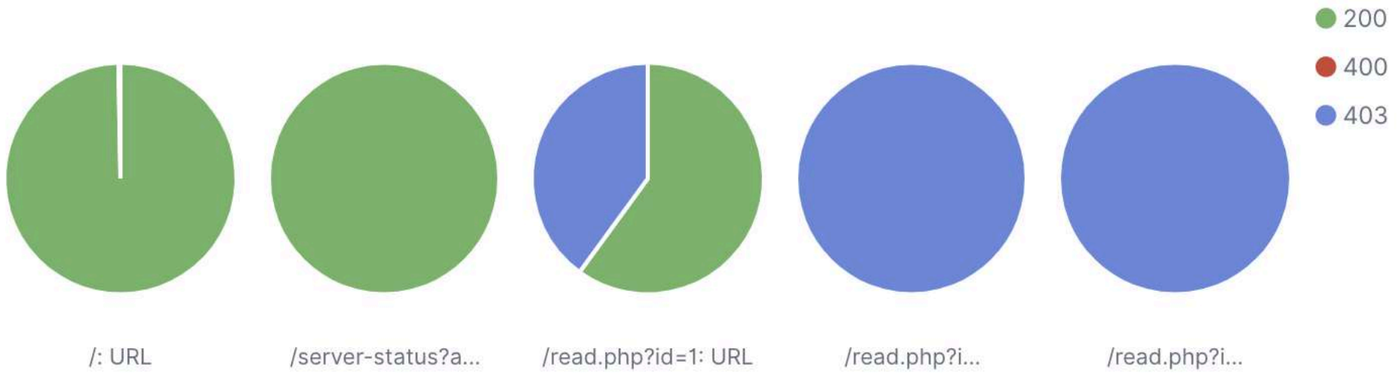
Rerun sqlmap

```
sqlmap --url "https://xeraa.wtf:8080/read.php?id=1" --purge
```

Rerun sqlmap

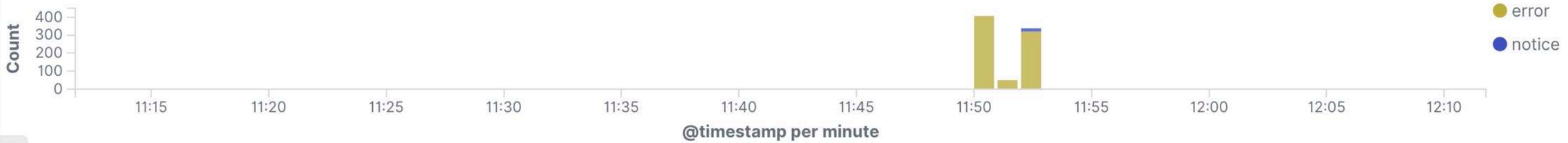
```
sqlmap --url "https://xeraa.wtf:8080/read.php?id=1" --purge
       --random-agent --tamper=space2comment
```

Top URLs by response code [Filebeat Apache] ECS



Browsers breakdown [Filebeat Apache] ECS

Error logs over time [Filebeat Apache] ECS



Apache errors log [Filebeat Apache] ECS

1–50 of 795

	Time	source.address	log.level	apache2.error.module	message
>	Oct 12, 2019 @ 11:52:36.097	141.138.9.226	error	-	[client 141.138.9.226] ModSecurity: Warning. detected SQLi using libinjection with fingerprint 'sB1c' [file "/usr/share/modsecurity-crs/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "68"] [id "942100"] [rev "1"] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: sB1c fo

/etc/modsecurity/modsecurity.conf

```
SecRuleEngine On | DetectionOnly | Off

SecRequestBodyLimit 13107200

SecRequestBodyNoFilesLimit 131072
```

/etc/modsecurity/modsecurity.conf

`SecAuditLogFormat JSON`

https://www.cryptobells.com/mod_security-json-audit-logs-revisited/

Custom Rule

```
SecRule REQUEST_FILENAME "create.php" "id:'400001',chain,deny,log,msg:'Fake Shay detected'"
SecRule REQUEST_METHOD "POST" chain
SecRule REQUEST_BODY "@rx (?i:(shay|banon))"
```

Log Output

```
["[file \"apache2_util.c\"]
[line 273] [level 3]
[client 141.138.9.226]
ModSecurity: Access denied with code
403 (phase 2).
Pattern match \"(?i:(shay|banon))\"
at REQUEST_BODY.
[file \"/etc/modsecurity/
modsecurity_custom_rules.conf\"]
[line \"1\"] [id \"400001\"]
[msg \"Fake Shay detected\"]
[hostname \"xeraa.wtf\"]
[uri \"/create.php\"]
[unique_id
\"XaHedrDrUdohwCZwrBiJlQAAAAI\"]"]
```

Not Fool Proof

login.php: `'(0r)1=1()`


WHAT THE FUCK WAS
I EVEN THINKING

Conclusion

Security incidents come in three levels:
FYI, **WTF**, and **OMG**

Write Better Code

Use Libraries / Frameworks

Is Anybody Using This?

GitLab now adds the `modsecurity` Web Application Firewall (WAF) plug-in to your cluster when you install the Ingress app in your Kubernetes cluster.

https://about.gitlab.com/blog/2019/09/22/gitlab-12-3-released/#web-application-firewall-for-kubernetes-ingress

Elastic Stack for Security*

mozilla

* Logging. New SIEM & Endpoint Security products.

ModSecurity ❤️ Logging

elastic

Examples

https://github.com/xeraa/mod_security-log

https://dashboard.xeraa.wtf

Secure Your Code

Injections and Logging

Philipp Krenn @xeraa

elastic