Sinkholing

A highway to a sinkhole

What is Showmax

- Online Video Streaming Service
  - Built upon microservice architecture (we love containers)
  - We are distributed (EU + Africa)
  - We log everything!
  - Engineering in Prague and Beroun

https://kinkyrhino.co.za/product/showmax-and-chill

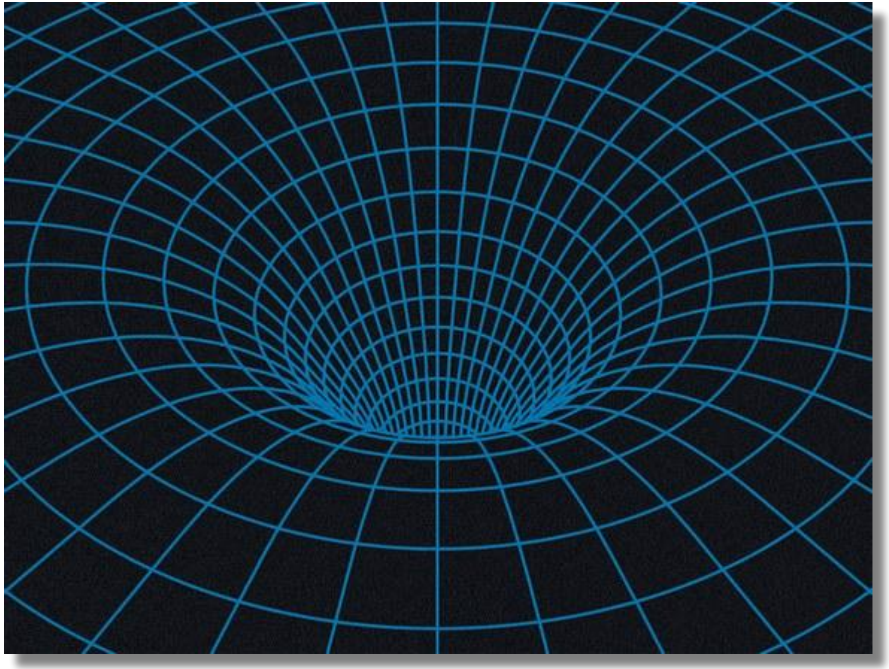
What is Sinkholing

- Distributed solution to protect the platform against common attacks
  - Password guessing
  - Security flaws exploitation
  - Our services misuse
  - … etc.

https://www.smh.com.au/world/the-black-hole-of-guatemala-city-20100601-wv8z.html

How to dig a sinkhole?

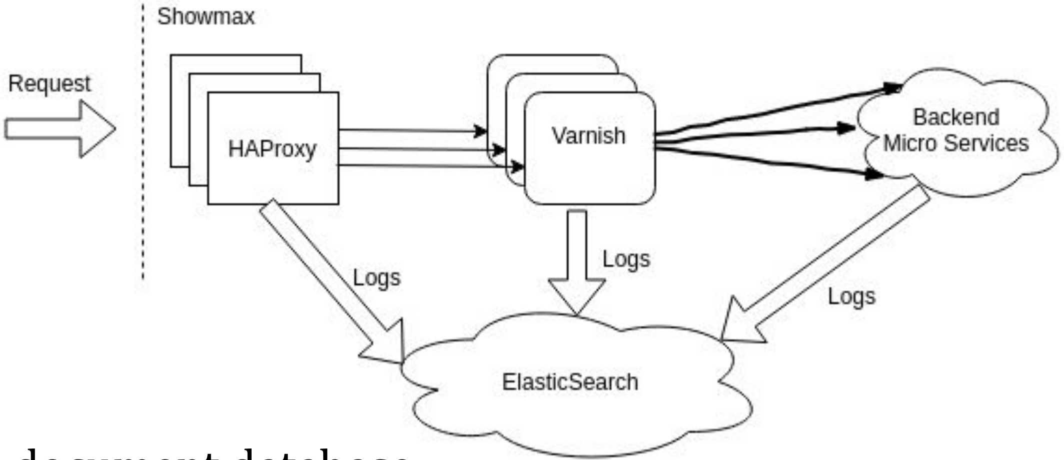
- Three main parts
  - Decision making mechanism
  - Persistent storage for bans
  - Tool that actually blocks requests



https://www.wired.com/story/what-is-sinkholing

HTTP request path @ Showmax

- HAProxy on frontends
  - Basic preprocessing and validation
  - Header manipulation
- Varnish
  - Caching && routing
- Microservices
  - Containerized
- Every step is logged
  - ElasticSearch - distributed document database



Where to find the truth?

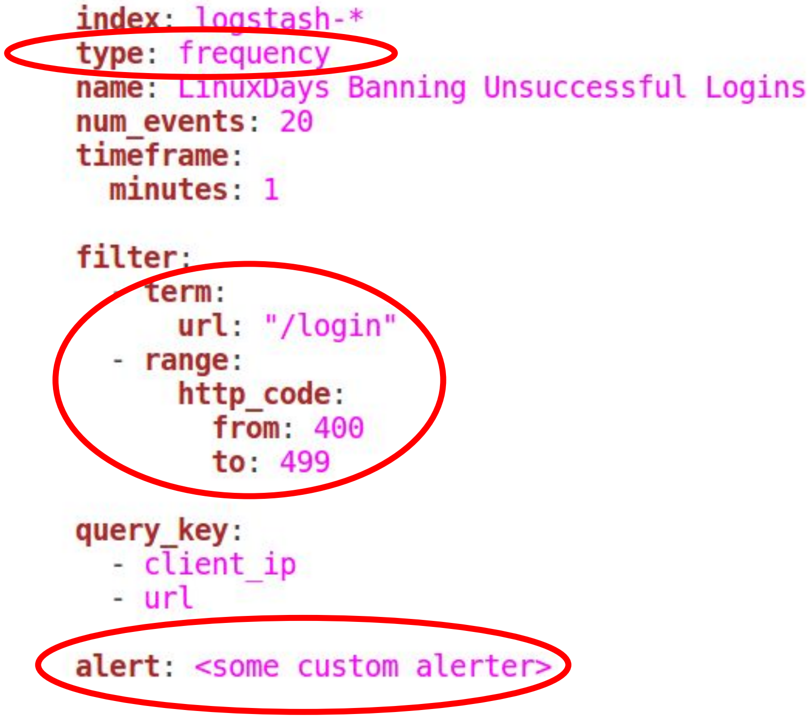
- ElasticSearch
  - Each step of request path is logged in real-time
  - All the data is there
  - Powerful tool for identifying attackers
  - -> Source of all the truth for Sinkholing
- Easy to search
  - Looking for password guessers?

```
url: "/login" AND http_code: [400 TO 499]
```

http://blog.deerwalk.com/the-single-source-of-truth-in-population-health-analytics

ElastAlert (by Yelp©) - search and alert effectively

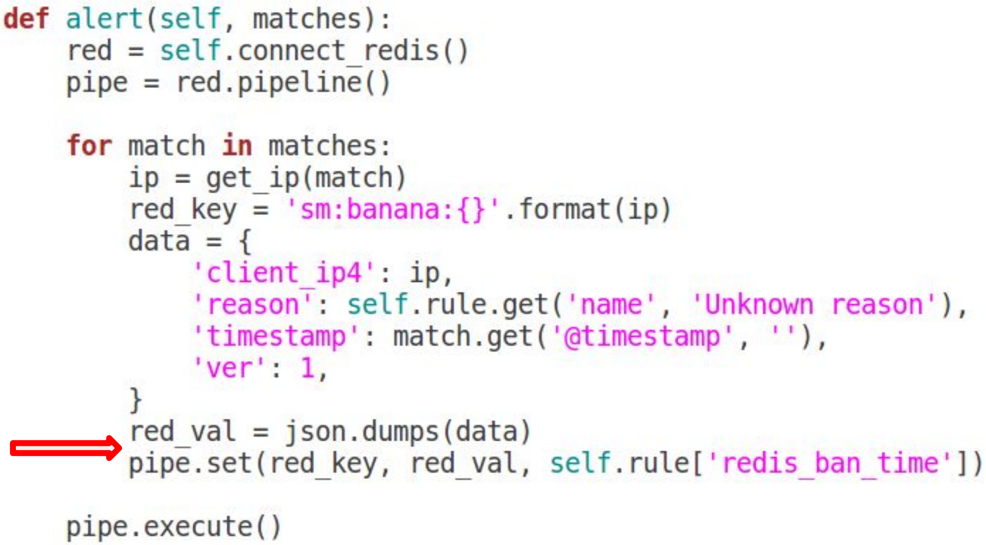
- Periodically queries ElasticSearch
- Alert raised when match is found
  - Numerous built-in alerts (email, messaging clients, HTTP POST, …)
  - Possibility to write custom alerter
- Highly customizable



```
index: logstash-*
type: frequency
name: LinuxDays Banning Unsuccessful Logins
num_events: 20
timeframe:
  minutes: 1

filter:
  - term:
      url: "/login"
  - range:
      http_code:
        from: 400
        to: 499

query_key:
  - client_ip
  - url

alert: <some custom alerter>
```

Where to store ban info?

- Shared source of the truth of bans
- Redis
    - Fast key-value store
    - Key expiration solves automatic unban
- Custom ElastAlert alerter
    - Subclass Alerter class
    - Implement alert function
    - Store json-encoded info to Redis



```python
def alert(self, matches):
    red = self.connect_redis()
    pipe = red.pipeline()

    for match in matches:
        ip = get_ip(match)
        red_key = 'sm:banana:{}'.format(ip)
        data = {
            'client_ip4': ip,
            'reason': self.rule.get('name', 'Unknown reason'),
            'timestamp': match.get('@timestamp', ''),
            'ver': 1,
        }
        red_val = json.dumps(data)
        pipe.set(red_key, red_val, self.rule['redis_ban_time'])

    pipe.execute()
```

HAProxy - the Great Sinkholing Barrier

- Hard (?) decision - HAProxy (robust) or iptables (light & fast)?
- Why HAProxy
  - Better capabilities for intruder detection
    - IP, UserAgent, Request content, etc
    - Extendable via LUA
  - Synthetic response “You have been banned”
  - Already the first segment in the Showmax processing pipeline

https://lanecorley.com/2016/04/06/breaking-the-50-barrier-in-church-planting

HAProxy configuration - IP based ban example

- `frontend` and `backend` configs are the most important
- Use ACLs

```
acl <aclname> <criterion> [flags] [operator] [<value>]
```

  - Detect request from IP
  - Send to Sinkholing backend
- Static IP list

```
acl ip_ban src -m ip -n 123.123.123.123 192.192.192.192
```

- Dynamic IP - control via socket

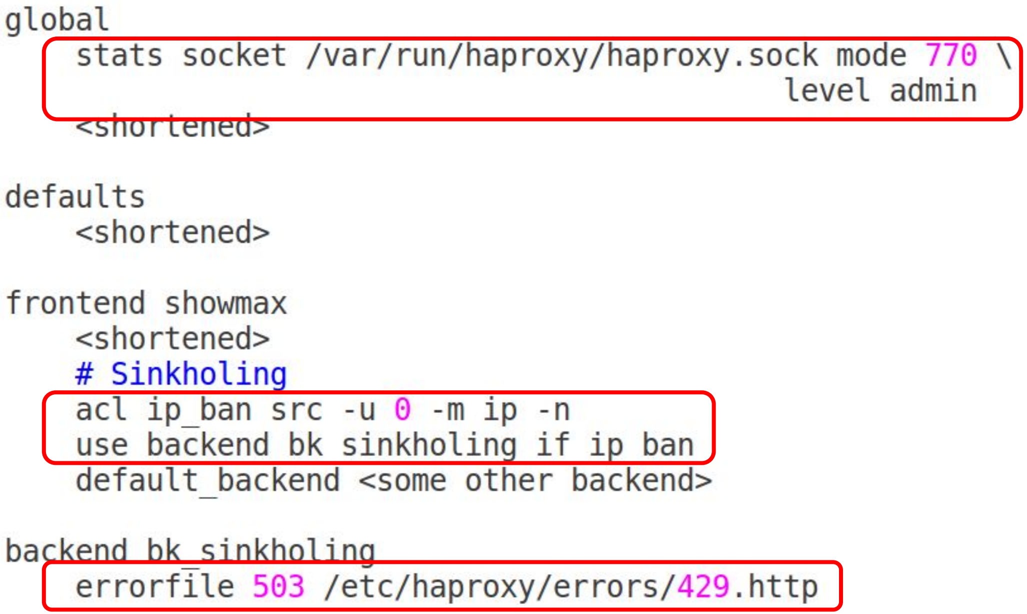
```
acl ip_ban src -u 0 -m ip -n
```

HAProxy configuration - IP based ban example

- Socket commands

```
add acl #<ID> <value>
show acl #<ID>
clear acl #<ID>
```

```
echo "show acl #0" | socat
/run/haproxy/admin.sock stdio
```

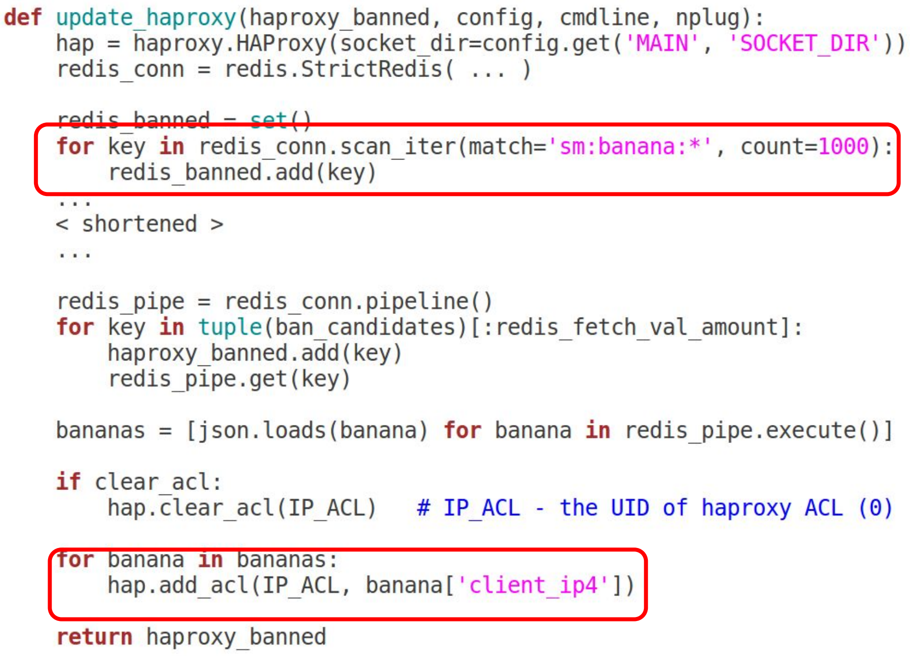


```
global
    stats socket /var/run/haproxy/haproxy.sock mode 770 \
                                            level admin
    <shortened>

defaults
    <shortened>

frontend showmax
    <shortened>
    # Sinkholing
    acl ip_ban src -u 0 -m ip -n
    use_backend bk_sinkholing if ip_ban
    default_backend <some other backend>

backend bk_sinkholing
    errorfile 503 /etc/haproxy/errors/429.http
```

The last missing piece ...

... is to fill HAProxy's ACL

- Python script that
  - Regularly fetches ban info from redis
  - Downloads full info about new bans
  - Fills HAProxy
  - Uses `'haproxyadmin'` python lib



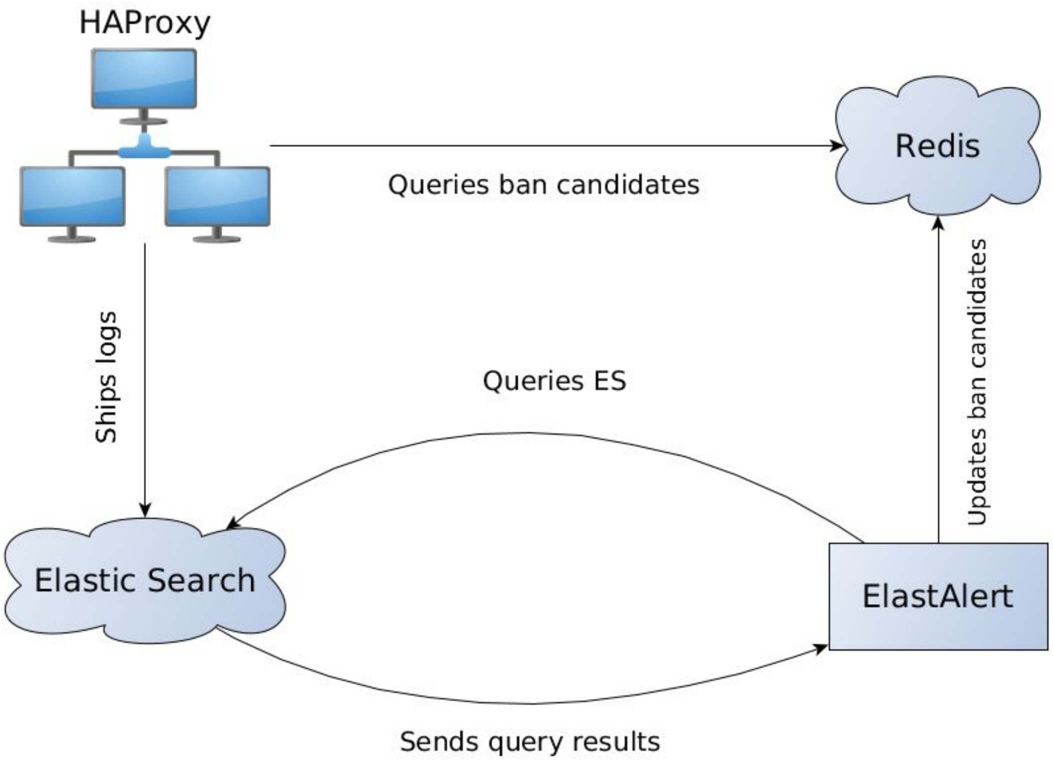
```
def update_haproxy(haproxy_banned, config, cmdline, nplug):
    hap = haproxy.HAProxy(socket_dir=config.get('MAIN', 'SOCKET_DIR'))
    redis_conn = redis.StrictRedis( ... )

    redis_banned = set()
    for key in redis_conn.scan_iter(match='sm:banana:*', count=1000):
        redis_banned.add(key)
    ...
    < shortened >
    ...

    redis_pipe = redis_conn.pipeline()
    for key in tuple(ban_candidates)[:redis_fetch_val_amount]:
        haproxy_banned.add(key)
        redis_pipe.get(key)

    bananas = [json.loads(banana) for banana in redis_pipe.execute()]

    if clear_acl:
        hap.clear_acl(IP_ACL)    # IP_ACL - the UID of haproxy ACL (0)

    for banana in bananas:
        hap.add_acl(IP_ACL, banana['client_ip4'])

    return haproxy_banned
```

Lesson learned:

- Africa is far - high latency ~170ms
- handshake is too expensive

To sum it up!

- The path of request is logged to elastic
- ElastAlert
  - Checks logs
  - Pushes bans to Redis
- HAProxy
  - Fetches bans from Redis
  - Blocks request



18 684

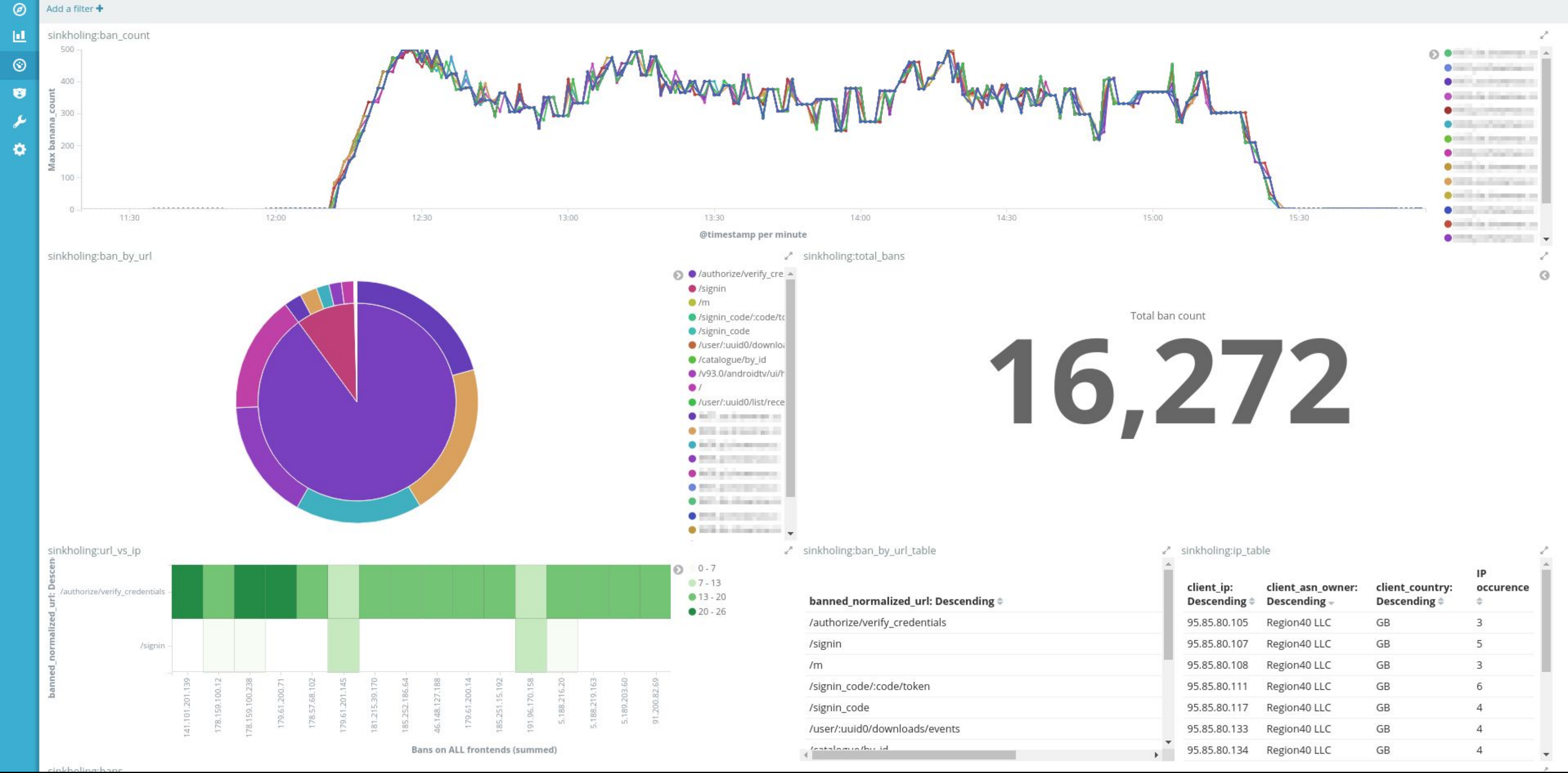
Bans in the last month

1 519 699

Requests blocked in the last month

86

Origin countries of blocked HTTP requests



banned_normalized_url: Descending
/authorize/verify_credentials
/signin
/m
/signin_code/:code/token
/signin_code
/user/:uuid0/downloads/events
/catalogue/by_id

client_ip: Descending	client_asn_owner: Descending	client_country: Descending	IP occurence
95.85.80.105	Region40 LLC	GB	3
95.85.80.107	Region40 LLC	GB	5
95.85.80.108	Region40 LLC	GB	3
95.85.80.111	Region40 LLC	GB	6
95.85.80.117	Region40 LLC	GB	4
95.85.80.133	Region40 LLC	GB	4
95.85.80.134	Region40 LLC	GB	4

Questions?

Find out more on our blog

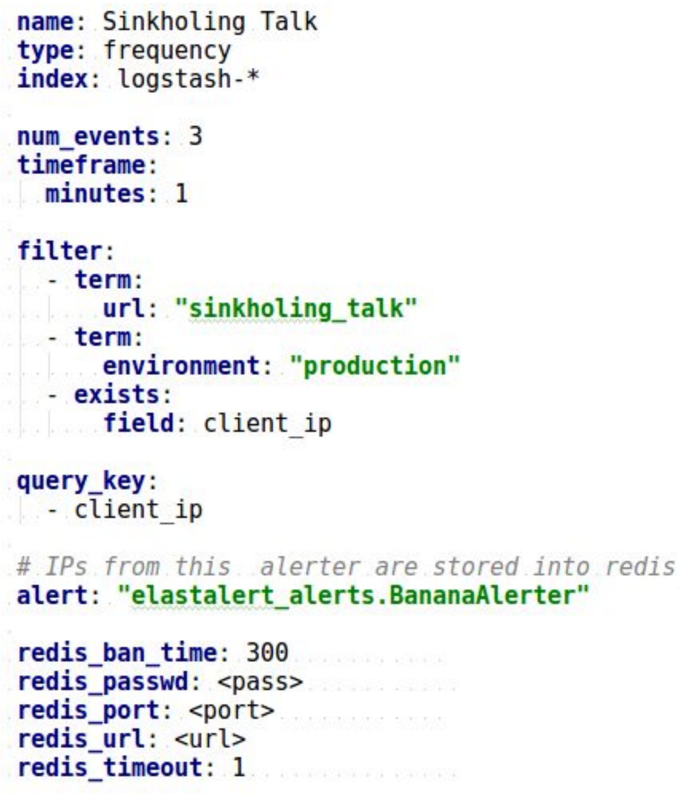
https://tech.showmax.com/

429 HTTP error file content

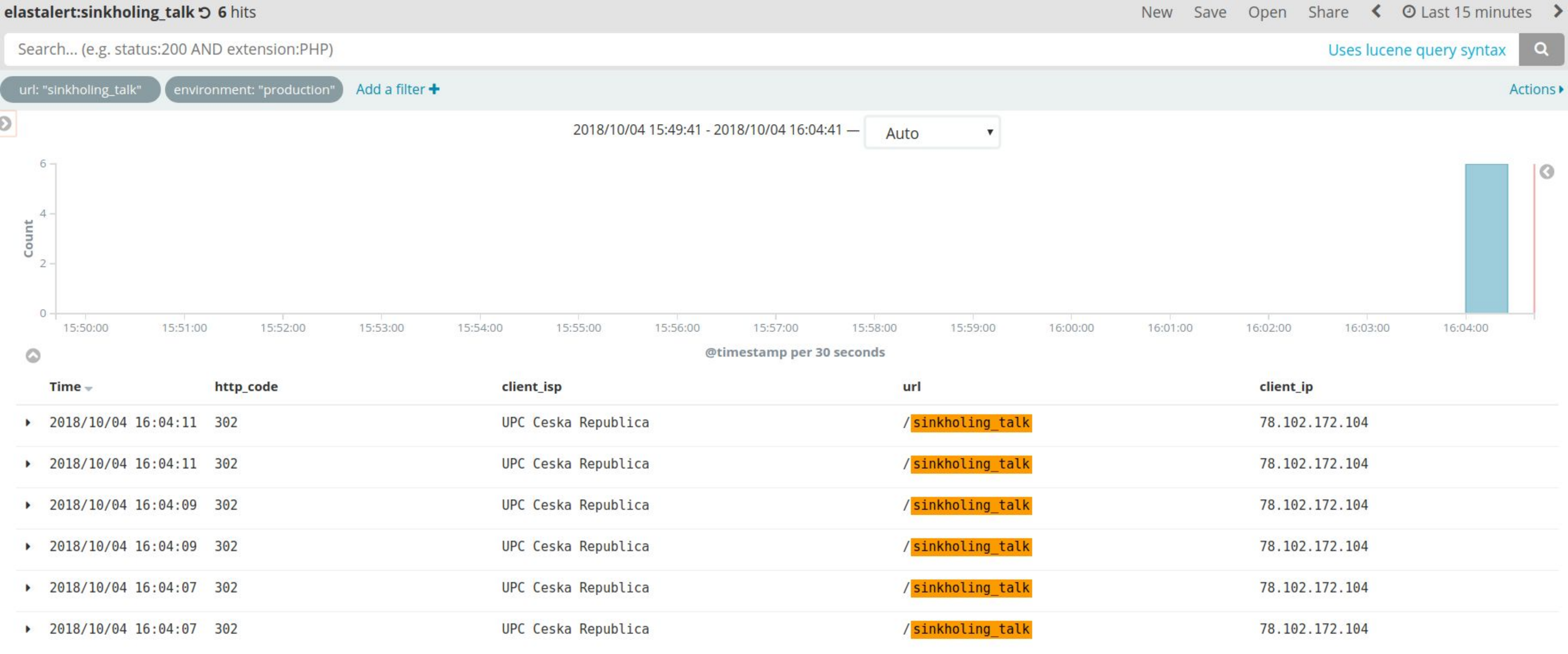
```
HTTP/1.0 429 Too Many Requests
Cache-Control: no-cache
Connection: close
Content-Type: application/json

{
    "error_code": "HAP1007",
    "lang": "eng",
    "message": "Too many requests. You have been banned.
                Please slow down a bit..."
}
```

Live demo

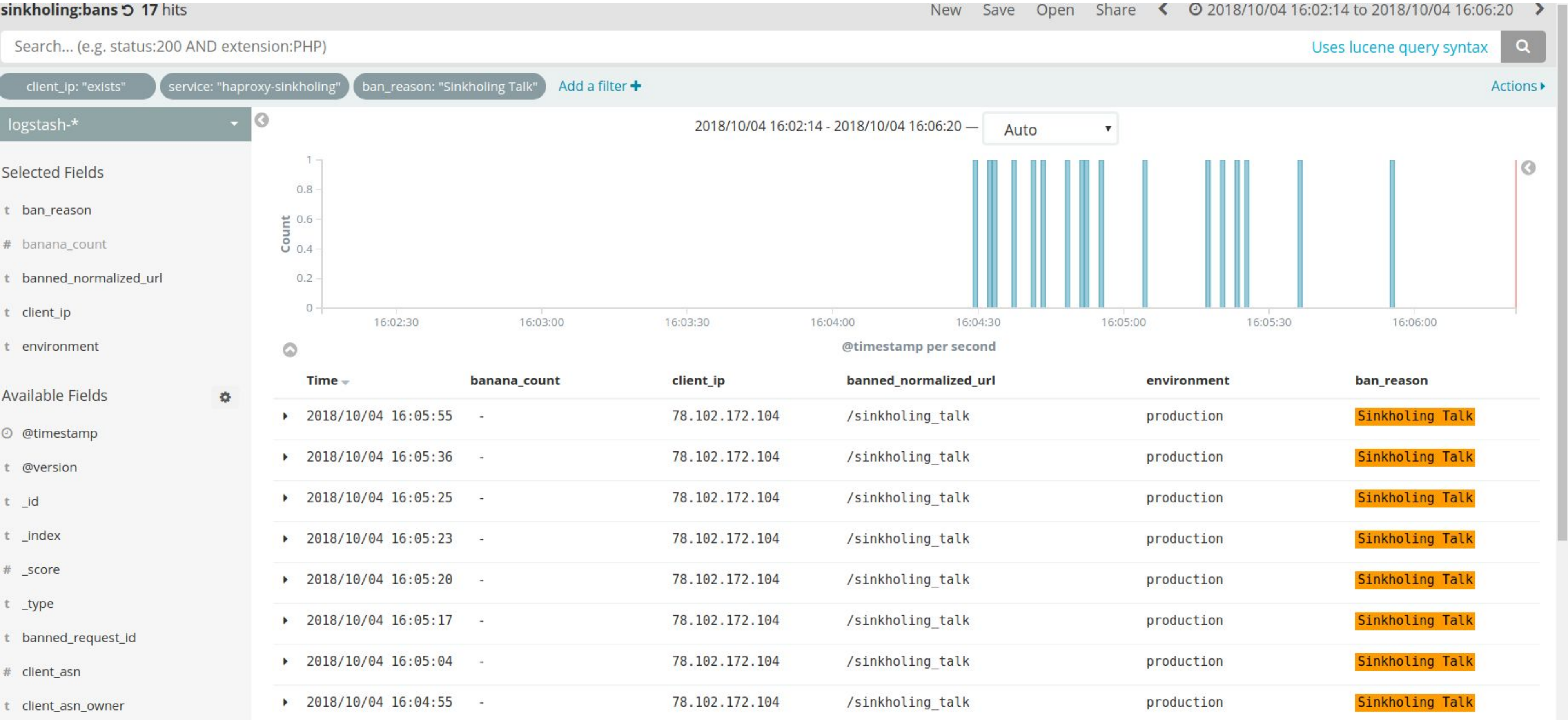
```
curl https://www.showmax.com/sinkholing_talk -v -L
```



```
name: Sinkholing Talk
type: frequency
index: logstash-*

num_events: 3
timeframe:
  minutes: 1

filter:
  - term:
      url: "sinkholing_talk"
  - term:
      environment: "production"
  - exists:
      field: client_ip

query_key:
  - client_ip

# IPs from this  alerter are stored into redis
alert: "elastalert_alerts.BananaAlerter"

redis_ban_time: 300
redis_passwd: <pass>
redis_port: <port>
redis_url: <url>
redis_timeout: 1
```



Time	http_code	client_isp	url	client_ip
2018/10/04 16:04:11	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104
2018/10/04 16:04:11	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104
2018/10/04 16:04:09	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104
2018/10/04 16:04:09	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104
2018/10/04 16:04:07	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104
2018/10/04 16:04:07	302	UPC Ceska Republica	/sinkholing_talk	78.102.172.104



client_ip: "exists"

service: "haproxy-sinkholing"

ban_reason: "Sinkholing Talk"

2018/10/04 16:02:14 - 2018/10/04 16:06:20 — Auto

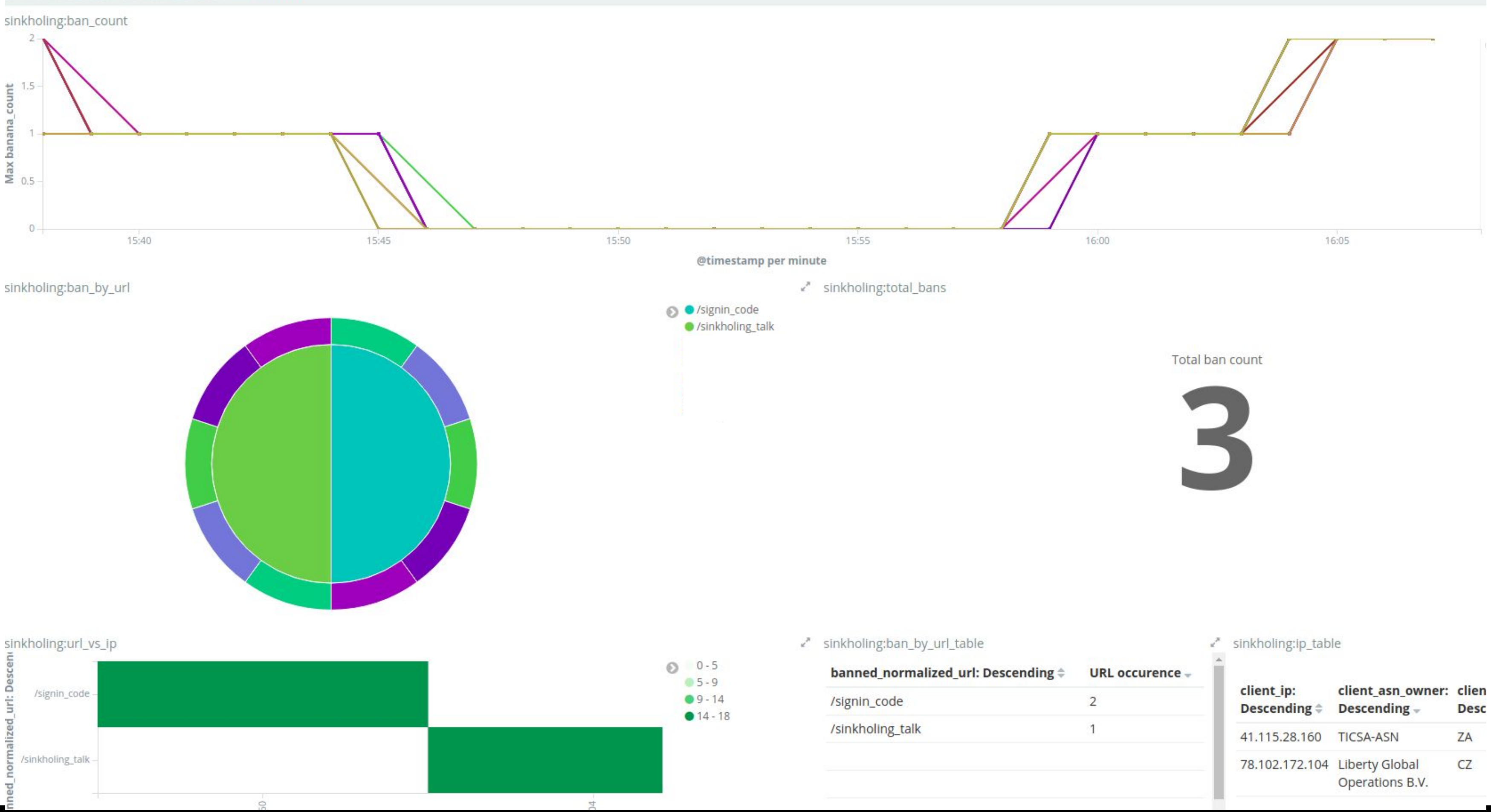
Selected Fields

- t ban_reason
- # banana_count
- t banned_normalized_url
- t client_ip
- t environment

Available Fields

- @timestamp
- t @version
- t _id
- t _index
- # _score
- t _type
- t banned_request_id
- # client_asn
- t client_asn_owner

Time	banana_count	client_ip	banned_normalized_url	environment	ban_reason
2018/10/04 16:05:55	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:36	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:25	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:23	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:20	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:17	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:05:04	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk
2018/10/04 16:04:55	-	78.102.172.104	/sinkholing_talk	production	Sinkholing Talk

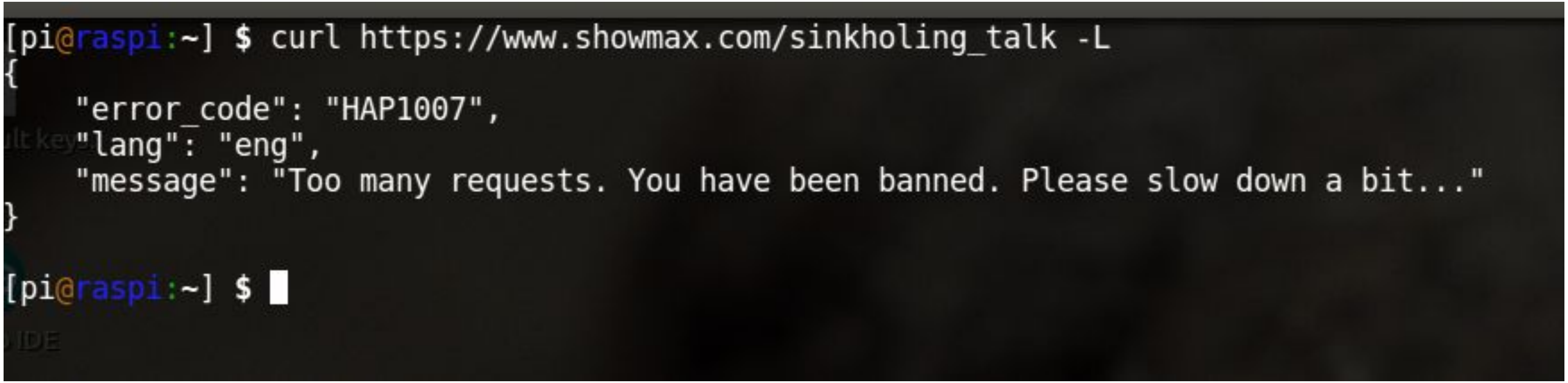


sinkholing:ban_by_url_table

banned_normalized_url: Descending	URL occurence
/signin_code	2
/sinkholing_talk	1

sinkholing:ip_table

client_ip: Descending	client_asn_owner: Descending	clien Desc
41.115.28.160	TICSA-ASN	ZA
78.102.172.104	Liberty Global Operations B.V.	CZ



```
[pi@raspi:~] $ curl https://www.showmax.com/sinkholing_talk -L
{
    "error_code": "HAP1007",
    "lang": "eng",
    "message": "Too many requests. You have been banned. Please slow down a bit..."
}

[pi@raspi:~] $ █
```