Security Risks Hiding In Plain Sight

Detectify Hacker School Online - Part 10

Sebastian Neef - @gehaxelt

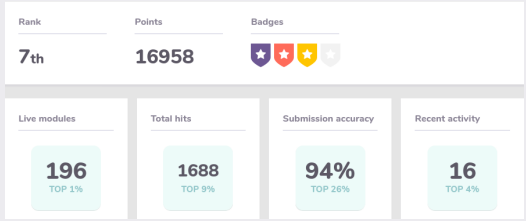
August 12, 2020

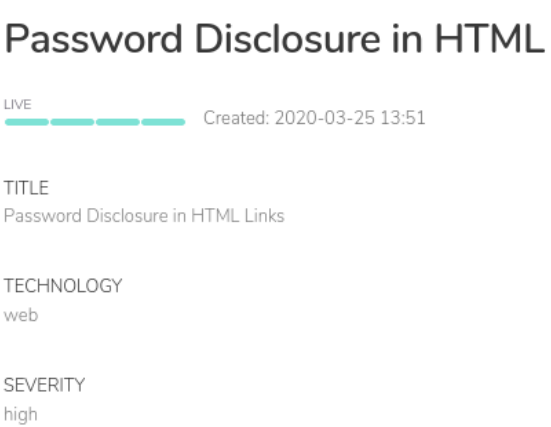
$>whoami

About me

- MSc CS student at TU Berlin
- Freelancer in IT-Sec since A-Levels
- Co-Founder of `Internetwache.org`
- Bug Bounties since 2013

Detectify





Module information

- Submitted in March 2020
- Live since April 2020
- Hits: 1 (so far)

Module information

- Submitted in March 2020
- Live since April 2020
- Hits: 1 (so far)

Idea: A module that

- is easy to implement
- has high severity
- discovers misconfiguration / lack of attention issues

$>research –init

Question:

How many websites have credentials in their HTML code?

$>research –init

Question:

How many websites have credentials in their HTML code?

HTTP Authentication

- Defined in RFC 7617[a]
- A `WWW-Authenticate` header leads to a login prompt
- Browsers understand `[protocol]://[username]:[password]@[host]` (RFC3986[b])
- Usable in any HTML element with URL-attributes, i.e.
  - `<a href="..."`
  - `<script src="..."`

[a] `https://tools.ietf.org/html/rfc7617`
[b] `https://tools.ietf.org/html/rfc3986`

$>research –limitations

Only XPATH and regular expressions

gehaxelt/Sebastian 10:38 AM

@Detectify-Kristian Is it possible for a module to iterate over all elements of the source code? I have an idea, but that would require to extract information from attributes of the HTML source code. It might also work with regexes, but iterating over all elements would be nicer.

Detectify-Berg 3:02 PM

We do support xpath which should be able to do that.

Detectify-Kristian 3:18 PM

That would work!

$>research –limitations

Only XPATH and regular expressions

gehaxelt/Sebastian 10:38 AM

@Detectify-Kristian Is it possible for a module to iterate over all elements of the source code? I have an idea, but that would require to extract information from attributes of the HTML source code. It might also work with regexes, but iterating over all elements would be nicer.

Detectify-Berg 3:02 PM

We do support xpath which should be able to do that.

Detectify-Kristian 3:18 PM

That would work!

Datasets

- Scans.io moved to Censys.io
- Censys.io 'academic research account' still waiting for approval :-/

$>research –datasets

Project Sonar by Rapid7

- HTTP[a] and HTTPS[b] GET responses to various ports against the whole IPv4 range
- ~250 GB of gzip compressed data
- 161m JSON-formatted responses

[a] `https://opendata.rapid7.com/sonar.http/`
[b] `https://opendata.rapid7.com/sonar.https/`

$>research –datasets

Project Sonar by Rapid7

- HTTP[a] and HTTPS[b] GET responses to various ports against the whole IPv4 range
- ~250 GB of gzip compressed data
- 161m JSON-formatted responses

[a] `https://opendata.rapid7.com/sonar.http/`
[b] `https://opendata.rapid7.com/sonar.https/`

Caveats

- Requests against IP-addresses without vhosts → Might miss actual websites
- But it is better than nothing ;-)

$>research –implementation-step=brainstorming

Brainstorming

1. Parallize using GNU parallel[a]
2. Parse JSON-based HTTP responses using Python
3. Use XPATH to extract suitable HTTP elements
4. Use regex to check for credentials
5. Output results

[a] `https://www.gnu.org/software/parallel/`

$>research –implementation-step=brainstorming

Brainstorming

1. Parallize using GNU parallel[a]
2. Parse JSON-based HTTP responses using Python
3. Use XPATH to extract suitable HTTP elements
4. Use regex to check for credentials
5. Output results

[a] `https://www.gnu.org/software/parallel/`

The easy part:

- Step 1: RTFM ;-) (`cat scan-files.txt| parallel -j6 'python scan.py {}'`)
- Step 2: Use my existing dataset processing `scan.py`

$>research –implementation-step=xpath

XPATH

- Select elements to analyze
- Query: `//a[contains(@href, '@') and contains(@href, ':') and not(contains(@href, "mailto:"))]`
- Some URI-protocols create false positives :-/

$>research –implementation-step=xpath

XPATH

- Select elements to analyze
- Query: `//a[contains(@href, '@') and contains(@href, ':') and not(contains(@href, "mailto:"))]`
- Some URI-protocols create false positives :-/

Other elements

- link/href
- iframe/src
- img/scr
- embed/src
- audio/src
- video/src
- script/src
- source/src

$>research –implementation-step=regex

Userinfo aka the 'username:password' part

- Defined in RFC3986[a] as

```
userinfo      = *( unreserved / pct-encoded / sub-delims / ":" )
pct-encoded   = "%" HEXDIG HEXDIG
sub-delims    = "!" / "$" / "&" / "'" / "(" / ")"
/ "*" / "+" / "," / ";" / "="
unreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~"
```

[a] https://tools.ietf.org/html/rfc3986

$>research –implementation-step=regex

Userinfo aka the 'username:password' part

- Defined in RFC3986[a] as

```
userinfo      = *( unreserved / pct-encoded / sub-delims / ":" )
pct-encoded   = "%" HEXDIG HEXDIG
sub-delims    = "!" / "$" / "&" / "'" / "(" / ")"
/ "*" / "+" / "," / ";" / "="
unreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~"
```

[a] https://tools.ietf.org/html/rfc3986

Regex v1

- Regex to only match `[protocol]://[username:password]@[host]/path`
- First attempt: `((?:https?:)?(?:\/\/)?[^\/?]+:[^\/?]+@.+)(:|\/|\.)`[a]

[a] https://regexr.com/513ih

$>research –implementation-step=regex

Regex v2

- Avoid false positives (i.e. matches inside URL parameters)
- Second attempt: `^((http|ftp|rtsp)s?:)?\/\/[^\/?]+:[^\/?]+@[^\.\/]*`[a]
- Compared against `urllib.parse`'s `urlparse` → same results

[a] `https://regexr.com/59rov`

Examples

Does match

- http://foo:bar@example.com/
- //foo:bar@example.com/

Does not match

- foo:bar@example.com/
- mailto:bar@example.com
- http://example.com/?foo=my:bar@example.com

<a> tags

- 40982 URLs / 24190 unique
- 1166 credentials / 636 unique

procotol	occurrences
ftp://	553
http://	58
https://	21
rtsp://	4

$>research –results

<a> tags

- 40982 URLs / 24190 unique
- 1166 credentials / 636 unique

procotol	occurrences
ftp://	553
http://	58
https://	21
rtsp://	4

all tags

- 188671 URLs / 55457 unique
- 1350 credentials / 881 unique

procotol	occurrences
ftp://	672
http://	93
https://	21
rtsp://	14

$>research –results

<a> tags

- 40982 URLs / 24190 unique
- 1166 credentials / 636 unique

procotol	occurrences
ftp://	553
http://	58
https://	21
rtsp://	4

all tags

- 188671 URLs / 55457 unique
- 1350 credentials / 881 unique

procotol	occurrences
ftp://	672
http://	93
https://	21
rtsp://	14

```
http://admin:123456@192.168.1.74:8074/mjpeg/snap.cgi?chn=1
http://admin:admin@10.0.0.44/web/mobile.html
```

Thank you for your attention!

References:

- https://0day.work/credentials-hiding-in-plain-sight-or-how-i-pwned-your-http-auth/
- contact [a/t] 0day [d.o.t] work