



UNIVERSITÄT **BONN**

# Version Detection

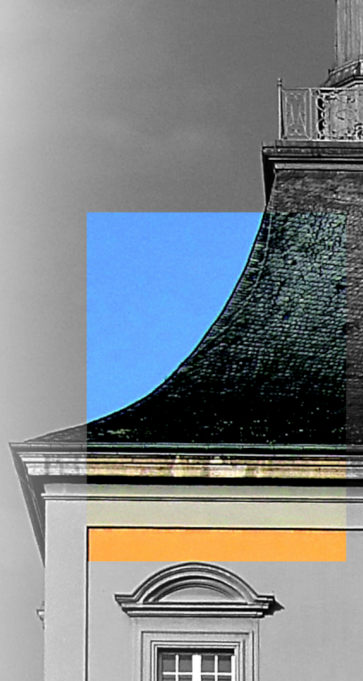
For Software and Libraries

Ben Swierzy

[swierzy@cs.uni-bonn.de](mailto:swierzy@cs.uni-bonn.de)

University of Bonn | Institute of Computer Science 4

Lecture IT Security | Uni Bonn | WT 2024/25



- What is Version Detection?
- Why should you want to detect versions?
- Types of Version Detection
- Banner Grabbing
  - Lots of Strings
  - Commercial Services
- Structural Analysis
  - Static Analysis
  - Dynamic Analysis

# What is Version Detection?

---

## A definition for Version Detection:

- ” Version Detection refers to the process of identifying software or library versions given a static artifact or dynamic system. It operates from a position where the target has no explicit interest in announcing its versions. ”



A software version is a unique identifier which maps to a unique state of a software. It usually consists of numbers or letters which are often assigned in ascending order to generate an order.

Examples:

- Linux 6.10.6
- Firefox 129.0.2
- intel-ucode 20240813-2
- pdfTeX 3.141592653-2.6-1.40.26
- Flavius 32501e228e1e865e397ccb437712066bae9ccdef

# Semantic Versioning

Systematic software versioning helps your dependency management.

Version **X.Y.Z**

**X** Major Version

increment when you make incompatible API changes

**Y** Minor Version

increment when you add functionality in a backward compatible manner

**Z** Patch Version

increment when you make backward compatible bug fixes

**Why should you want to detect versions?**

---

# Motivation: CVE & CVSS

## CVE Description Components

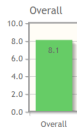
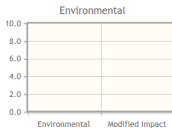
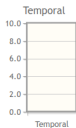
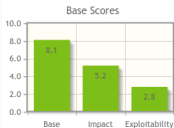
- Software
- Version range
- Vulnerability

## CVSS

- Measures CVE severity
- Multiple categories form a score between 0 and 10
- Multiple versions with different categories



# Motivation: CVE & CVSS



**CVSS Base Score:** 8.1  
 Impact Subscore: 5.2  
 Exploitability Subscore: 2.8  
**CVSS Temporal Score:** NA  
 CVSS Environmental Score: NA  
 Modified Impact Subscore: NA  
**Overall CVSS Score:** 8.1

Show Equations

## CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

## Base Score Metrics

### Exploitability Metrics

#### Attack Vector (AV)\*

Network (AV:N) Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

#### Attack Complexity (AC)\*

Low (AC:L) High (AC:H)

#### Privileges Required (PR)\*

None (PR:N) Low (PR:L) High (PR:H)

#### User Interaction (UI)\*

None (UI:N) Required (UI:R)

### Scope (S)\*

Unchanged (S:U) Changed (S:C)

### Impact Metrics

#### Confidentiality Impact (C)\*

None (C:N) Low (C:L) High (C:H)

#### Integrity Impact (I)\*

None (I:N) Low (I:L) High (I:H)

#### Availability Impact (A)\*

None (A:N) Low (A:L) High (A:H)

<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>

## Be careful with CVEs

**CVE:** CVE-2020-19909 Integer Overflow in Curl

**First CVSSv3 Scoring:**

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H - 9.8

**Problematic parameter:**

```
1 curl --retry-delay 18446744073709552
```

**Issue:**

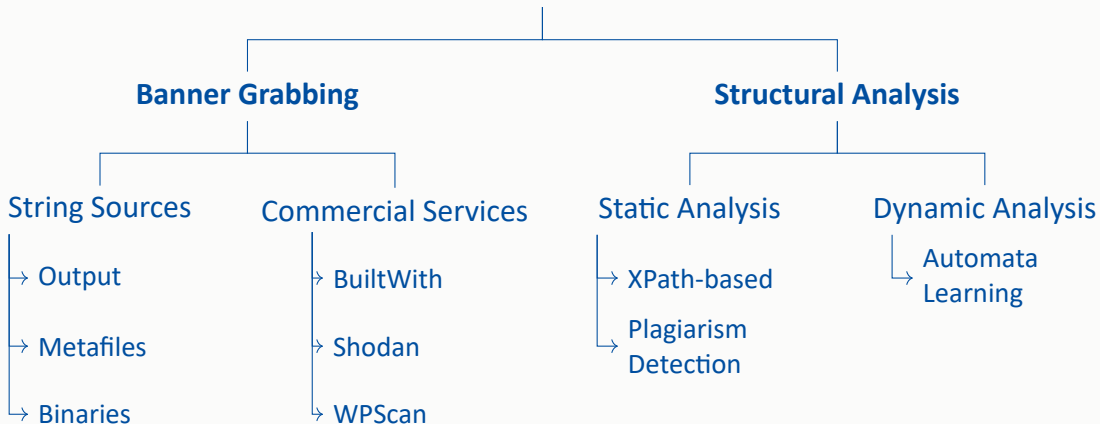
Few actors control the CVE ecosystem (Mitre, NVD) and have policies with implications that many do not consider.

<https://daniel.haxx.se/blog/2023/08/26/cve-2020-19909-is-everything-that-is-wrong-with-cves/>

# Types of Version Detection

---

## Version Detection



# Banner Grabbing

---

# Banner Grabbing

Most systems include functionality to announce their version.

For this, systems include a special string called **banner**.

**Banner grabbing** denotes the process of reading this banner.

Banners are usually unstructured and their location varies.

**Main advantages:** Easy to fetch, able to identify unknown versions

**Main disadvantages:** Easy to hide and spoof

# Banner Grabbing

---

Lots of Strings

# Lots of strings

We will look at different sources for banners.

- **Output**

Banners contained in the direct output of a system

- **Metafiles**

Accidental files containing version information

- **Binaries**

Banners in BLOBs and ELF's



## Banners from Output

If you have local access and the system and you analyze a locally installed package, the task is easy.

### Example for Ubuntu:

```
$ apt-cache policy gcc
```

```
gcc:
```

```
  Installed: 4:13.2.0-7ubuntu1
```

```
  Candidate: 4:13.2.0-7ubuntu1
```

```
  Version table:
```

```
*** 4:13.2.0-7ubuntu1 500
```

```
500 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages
```

```
100 /var/lib/dpkg/status
```

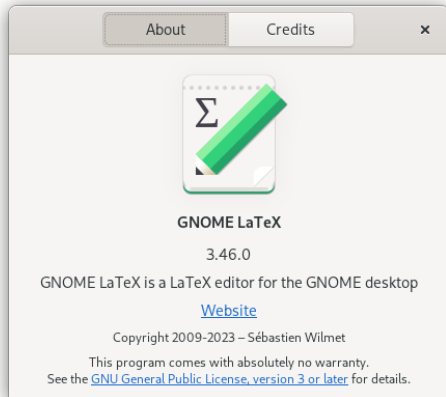
## Banners from Output

Most command line utilities offer an option similar to `--version`.

```
$ gcc --version  
gcc (GCC) 14.2.1 20240805  
Copyright (C) 2023 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

## Banners from Output

Most GUI Tools offer version information through their menu, often at Help > About.



## Banners from Output

Most web servers announce their version in the Server HTTP Header:

```
$ curl -I https://itsec.cs.uni-bonn.de/feedback/  
HTTP/2 200  
strict-transport-security: max-age=63072000; includeSubdomains; preload  
accept-ranges: bytes  
content-length: 509  
content-type: text/html  
etag: 66a26059-1fd  
server: nginx/1.27.0
```

## Banners from Output

Many web applications announce their version in `<meta>` tags:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="generator" content="WordPress 6.4.1" />
    [...]
```

... and we can find even more version information in HTML.

# Metadata in the Output

The HTML output usually contains many more versions. They are most often classified as **metadata** and not strictly as banners.

- **CDN script URLs**

```
<script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js">
```

- **Web App Plugins**

```
<link rel='stylesheet' href='https://wordpress.org/wp-content/plugins/gutenberg/.../style.css?ver=18.8.0' />
```

- **Inline Data**

```
window._wpemojiSettings = {baseUrl:"https://s.w.org/images/core/emoji/15.0.3/72x72/"}
```

Metafiles are served on a web server, but not required for functionality. Usually, they are accessible through misconfiguration or accidentally copied.

Some types of metafiles include

- Readmes
- Files from package managers
- Files from version control
- Artifacts from the build process

Readme files are often used to detect versions of WordPress extensions.

```
$ curl https://wordpress.org/wp-content/plugins/gutenberg/readme.txt | head  
=== Gutenberg ===  
Contributors: matveb, joen, karmatosed  
Tested up to: 6.5  
Stable tag: 18.8.0  
License: GPLv2 or later  
License URI: http://www.gnu.org/licenses/gpl-2.0.html
```

The Gutenberg plugin adds editing, customization, and site building to WordPress.  
Use it to test beta features before their official release.



Files from package manager exist if a project is served directly. In the web, we can find many metafiles from composer (PHP) and npm (JavaScript).

```
$ curl https://wordpress.org/composer.json
{
  "name": "wordpress/wordpress.org",
  "description": "wordpress.org multi-network install",
  "license": "GPLv2+",

  "require-dev": {
    "dealerdirect/phpcodesniffer-composer-installer": "^0.7.0",
    "phpunit/phpunit": "^9.4",
    "spatie/phpunit-watcher": "^1.23.2",
    "wp-coding-standards/wpcs": "2.*"
  },

  "scripts": {
    "format": "phpcbf -p",
    "lint": "phpcs",
    "test": "php -d xdebug.mode=off ./vendor/bin/phpunit",
    "test:watch": "phpunit-watcher watch"
  }
}
```

Files from package manager exist if a project is served directly. In the web, we can find many metafiles from composer (PHP) and npm (JavaScript).

```
$ curl http://149.202.74.137/package.json
{
  "private": true,
  "devDependencies": {
    "autoprefixer": "^10.3.7",
    "axios": "^0.21",
    "browser-sync": "^2.27.7",
    "browser-sync-webpack-plugin": "2.3.0",
    "laravel-mix": "^6.0.6",
    "lodash": "^4.17.19",
    "postcss": "^8.3.11",
    "tailwindcss": "^2.2.17"
  },
  "dependencies": {
    "@rateyo/jquery": "^3.0.0-alpha.2",
    "jquery": "^3.6.0",
    "rateyo": "^3.0.0-alpha.2",
    "slick-carousel": "^1.8.1"
  }
}
```

Files from **version control systems** can be very sensitive as they may contain the complete history of the repository. Besides allowing **access to source files**, these can include **credentials** that have been pushed into the repository at any point in time.

```
$ curl https://flux-cdn.com/.git/HEAD  
ref: refs/heads/masters
```

**Source maps** are artifacts from JavaScript bundling process and can reveal file system paths of the source code. The package manager pnpm writes the package versions into directory names for symlinking.

```
$ curl https://assets-cdn.getbento.com/static/analytics/js/snowplow-3.1.6.js.map | jq .sources | grep .pnpm
"../../../../common/temp/node_modules/.pnpm/tslib@2.3.0/node_modules/tslib/tslib.es6.js",
"../../../../common/temp/node_modules/.pnpm/uuid@3.4.0/node_modules/uuid/lib/bytesToUuid.js",
"../../../../common/temp/node_modules/.pnpm/uuid@3.4.0/node_modules/uuid/lib/rng-browser.js",
"../../../../common/temp/node_modules/.pnpm/uuid@3.4.0/node_modules/uuid/v1.js",
"../../../../common/temp/node_modules/.pnpm/uuid@3.4.0/node_modules/uuid/v4.js",
"../../../../common/temp/node_modules/.pnpm/uuid@3.4.0/node_modules/uuid/index.js",
"../../../../common/temp/node_modules/.pnpm/crypt@0.0.2/node_modules/crypt/crypt.js",
"../../../../common/temp/node_modules/.pnpm/charenc@0.0.2/node_modules/charenc/charenc.js",
"../../../../common/temp/node_modules/.pnpm/sha1@1.1.1/node_modules/sha1/sha1.js",
"../../../../common/temp/node_modules/.pnpm/jstimezonedetect@1.0.7/node_modules/jstimezonedetect/dist/jstz.min.js",
```

Banners are usually stored directly in the binary. If we know how versions look, we can obtain them quite easily.

```
$ strings /boot/vmlinuz-linux | grep -E '[0-9]+\.[0-9]+\.[0-9]+'
6.10.7-arch1-1 (linux@archlinux) #1 SMP PREEMPT_DYNAMIC Thu, 29 Aug 2024 16:48:57 +0000
6.10.7-arch1-1 (linux@archlinux) (gcc (GCC) 14.2.1 20240805, GNU ld (GNU Binutils) 2.43.0) #1 SMP PREEMPT_DYNAMIC Thu, 29 Aug 2024
16:48:57 +0000
1.5.2
```

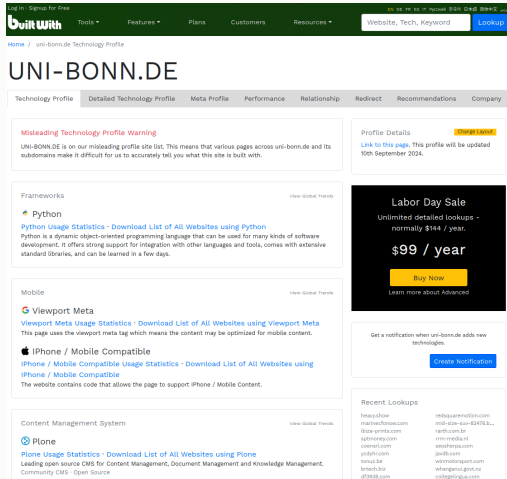
```
$ strings /usr/bin/gcc | grep -E '[0-9]+\.[0-9]+\.[0-9]+'
GLIBC_2.3.2
GLIBC_2.3.4
GLIBC_2.2.5
14.2.1
14.2.1 20240805
[...]
```

# Banner Grabbing

---

Commercial Services

Builtwith is a closed source product analyzing **technologies on the web.**



The screenshot shows the Builtwith website profile for **uni-bonn.de**. The page has a dark green header with the Builtwith logo, navigation links (Tools, Features, Plans, Customers, Resources), a search bar, and a 'Logup' button. Below the header, the profile title 'UNI-BONN.DE' is displayed. A 'Misleading Technology Profile Warning' states that the site is on a misleading profile list. The 'Frameworks' section highlights Python, with a link to download a list of websites using Python. The 'Mobile' section highlights Viewport Meta and iPhone / Mobile Compatible. The 'Content Management System' section highlights Plone. A 'Labor Day Sale' banner offers unlimited detailed lookups for \$99/year. A 'Recent Lookups' section lists various domains analyzed.

**Log In / Signup for Free** | Website, Tech, Keyword | **Logup**

Home / uni-bonn.de Technology Profile

## UNI-BONN.DE

Technology Profile | Detailed Technology Profile | Meta Profile | Performance | Relationship | Redirect | Recommendations | Company

**Misleading Technology Profile Warning**

UNI-BONN.DE is on our misleading profile site list. This means that various pages across uni-bonn.de and its subdomains make it difficult for us to accurately tell you what this site is built with.

**Profile Details** | **Change Layout**

[Link to this page](#). This profile will be updated 10th September 2024.

**Labor Day Sale**  
Unlimited detailed lookups - normally \$144 / year.  
**\$99 / year**  
**Buy Now**  
Learn more about Advanced

Get a notification when uni-bonn.de adds new technologies.  
**Create Notification**

**Recent Lookups**

- heqychoe
- marinacforow.com
- liza-printa.com
- spdmoney.com
- coenarl.com
- yedhy.com
- tonuz.be
- lurech.biz
- df9338.com
- redsquaremotion.com
- mid-size-euu-82476.b...
- ranth.com.br
- mm-media.nl
- seosherpa.com
- jeffib.com
- winnomortport.com
- whangpus.gov.no
- collagetingua.com

Builtwith is a closed source product analyzing technologies on the web.

Log In / Sign up for Free

**builtwith** Tools • Features • Plans • Customers • Resources • Website, Tech, Keyword [Lookup](#)

[Home](#) / [Plans & Pricing](#)

## Plans & Pricing

BuiltWith is free to use for individual site lookups forever. All plans can be canceled at any time.

Per Month

Per Year Discounted

### Basic

Suits fixed scope requirements.

**\$295**  
= €267

Technologies	2
Keywords	2
Retail Reports	2
System Logins	1

[Upgrade to Basic](#)

### Pro

Fully featured most popular plan.

**\$495**  
= €448

Technologies	Unlimited
Keywords	Unlimited
Retail Reports	Unlimited
System Logins	1

[Upgrade to Pro](#)

### Team

Unlimited Enterprise Access.

**\$995**  
= €900

Technologies	Unlimited
Keywords	Unlimited
Retail Reports	Unlimited
System Logins	Unlimited

[Upgrade to Team](#)

**All Plans Include** [See All Features](#)

Data from 673 million+ websites ✓  
And billions more websites from third parties

Unlimited Report Download ✓  
You can download reports you create

Excel and CSV Export ✓  
Bulk export reports you create to Excel



Shodan offers a closed-source **search engine for servers**. Thus, it is suitable to find IoT devices or many other interesting services.

The screenshot displays the Shodan search engine interface. At the top, there is a search bar with a magnifying glass icon. Below the search bar, there are tabs for 'View Report', 'View on Map', and 'Advanced Search'. A 'Product Spotlight' banner is visible, encouraging users to keep track of their internet connections using 'Shodan Monitor'.

The main content area shows three search results, each for an IP address from the University of Bonn (131.220.15.221, 131.220.15.113, and 131.220.220.29). Each result includes a 'Startseite' link, a 'SSL Certificate' section, and a list of associated domains and organizations.

**Result 1: 131.220.15.221**

- Startseite**: [Startseite — int-de](#)
- SSL Certificate**:
  - Issued By: **GEANT OV RSA CA 4**
  - Issued To: **mail.uni-bonn.de**
  - Organization: **Rheinische Friedrich-Wilhelms-Universität Bonn**
  - Supported SSL Versions: **TLShv1, TLShv1.1, TLShv1.2**
- Associated Domains**:
  - pop.uni-bonn.de
  - imap.uni-bonn.de
  - smtp.uni-bonn.de
  - mail.uni-bonn.de
- Associated Organizations**:
  - GEANT Networking
  - Rheinische Friedrich-Wilhelms-Universität Bonn
- Additional Info**:
  - \* OK IMAP Server 6.3.13 at **uni-bonn.de** ready
  - \* CAPABILITY [IMAP4 IMAPREV] ACL NAMESPACE SOSPLUS IDLE LITERAL+ QUOTA ID MULTIAPPEND

**Result 2: 131.220.15.113**

- Startseite**: [Startseite — int-de](#)
- SSL Certificate**:
  - Issued By: **GEANT OV RSA CA 4**
  - Issued To: **mail.uni-bonn.de**
  - Organization: **Rheinische Friedrich-Wilhelms-Universität Bonn**
  - Supported SSL Versions: **TLShv1, TLShv1.1, TLShv1.2**
- Associated Domains**:
  - pop.uni-bonn.de
  - imap.uni-bonn.de
  - smtp.uni-bonn.de
  - mail.uni-bonn.de
- Associated Organizations**:
  - GEANT Networking
  - Rheinische Friedrich-Wilhelms-Universität Bonn
- Additional Info**:
  - \* OK IMAP Server 6.3.13 at **uni-bonn.de** ready
  - \* CAPABILITY [IMAP4 IMAPREV] ACL NAMESPACE SOSPLUS IDLE LITERAL+ QUOTA ID MULTIAPPEND

**Result 3: 131.220.220.29**

- Startseite**: [Startseite — int-de](#)
- SSL Certificate**:
  - Issued By: **Let's Encrypt**
  - Issued To: **www.uni-bonn.de**
  - Organization: **Let's Encrypt**
  - Supported SSL Versions: **TLShv1, TLShv1.1, TLShv1.2**
- Associated Domains**:
  - www.uni-bonn.de
  - www2.uni-bonn.de
  - www3.uni-bonn.de
- Associated Organizations**:
  - Let's Encrypt
  - Rheinische Friedrich-Wilhelms-Universität Bonn
- Additional Info**:
  - HTTP/1.1 200 OK
  - Date: Mon, 02 Sep 2024 18:46:25 GMT
  - Server: Apache
  - Strict-Transport-Security: max-age=15768000; preload
  - X-Apache: p0m05-0027-locked load **uni-bonn.de**
  - Cache-Control: max-age=0, s-maxage=604800, must-revalidate
  - Content-Language: de
  - Content-Type: text/html; charset=utf-8

Shodan offers a closed-source **search engine for servers**. Thus, it is suitable to find IoT devices or many other interesting services.

Choose Your Plan

No contracts. No setup fees. Cancel anytime.

Freelancer

\$69/month

LOGIN TO SUBSCRIBE

- ✓ Up to 1 million results per month \*
- ✓ Scan up to 5,120 IPs per month
- ✓ Network Monitoring for 5,120 IPs

- ✓ Access to most filters
- ✓ Allows paging through search results
- ✓ Basic access to the Streaming API
- ✓ Commercial Use

- ✓ E-Mail support

Small Business

\$359/month

LOGIN TO SUBSCRIBE

- ✓ Up to 20 million results per month \*
- ✓ Scan up to 65,536 IPs per month
- ✓ Network Monitoring for 65,536 IPs

- ✓ Access to most filters
- ✓ Allows paging through search results
- ✓ Basic access to the Streaming API
- ✓ Commercial Use

- ✓ E-Mail support
- ✓ Vulnerability search filter

Corporate

\$1099/month

LOGIN TO SUBSCRIBE

- ✓ **Unlimited** results per month \*
- ✓ Scan up to 327,680 IPs per month
- ✓ Network Monitoring for 327,680 IPs

- ✓ Access to all filters
- ✓ Allows paging through search results
- ✓ Basic access to the Streaming API
- ✓ Commercial Use

- ✓ Premium Support
- ✓ Vulnerability search filter
- ✓ Batch IP Lookups

Wappalyzer is an open source SaaS **web technology scanner**. It works with an extensive set of regular expressions.

[wappalyzer](#) / [src](#) / [technologies](#) / **p.json**

Code	Blame	3105 lines (3105 loc) · 78.9 KB
------	-------	---------------------------------

```

1723     "website": "http://pligg.com"
1724 },
1725 "PPlone": {
1726     "cats": [
1727         1
1728     ],
1729     "cpe": "cpe:2.3:a:plone:plone:*:*:*:*:*:*:*:",
1730     "icon": "PPlone.svg",
1731     "implies": "Python",
1732     "meta": {
1733         "generator": "PPlone"
1734     },
1735     "website": "http://plone.org"
1736 },
1737 "Plotly": {
1738     "cats": [
1739         25
1740     ],
1741     "icon": "Plotly.png",
1742     "implies": "D3",
1743     "js": {
1744         "Plotly.version": "([\\d.]\\.\\.version:\\\\1"
1745     },
1746     "scriptSrc": "https?://cdn\\.plot\\.ly/plotly",
1747     "website": "https://plot.ly/javascript/"

```

WPScan is an open source **WordPress security scanner**. It retrieves versions through banner grabbing to search for known vulnerabilities.

## WPScan CLI Scanner

The WPScan CLI tool is a black box WordPress security scanner written for security professionals and WordPress site maintainers to test the security of their sites. The WPScan CLI tool uses our database of **43,472** WordPress vulnerabilities.

Install now by running: `gem install wpscan`

[View on GitHub →](#)

## What does WPScan check for?

- The version of WordPress installed and any associated vulnerabilities
- What plugins are installed and any associated vulnerabilities
- What themes are installed and any associated vulnerabilities
- Username enumeration
- Users with weak passwords via password brute forcing
- Backed up and publicly accessible wp-config.php files
- Media file enumeration
- Vulnerable Timthumb files
- If the WordPress readme file is present
- If WP-Cron is enabled
- If user registration is enabled
- Full Path Disclose
- Unload directory listing

# Structural Analysis

---

# Structural Analysis

---

Static Analysis

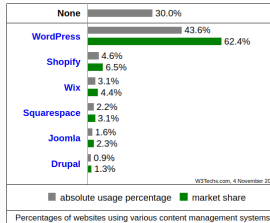
Structural static analysis compares features from statically analyzed assets to those supplied by a reference system.

We consider two use cases with different solutions:

- 1 Web Application Version Detection with XPath-based detection
- 2 Detection of library versions in JavaScript Bundle using plagiarism detection

## XPath-based detection

Web Applications power are an important foundation of the modern world wide web infrastructure. The following analysis technique focuses on the special class of **Content Management Systems (CMS)**.





## XPath-based detection

Web Applications power are an important foundation of the modern world wide web infrastructure. The following analysis technique focuses on the special class of **Content Management Systems (CMS)**.

Version Detection is highly useful for web applications for several reasons

- Internet exposure
- Widely deployed
- High risk of attacks

## XPath-based detection

Banner grabbing has several disadvantages:

- Application specific
- Often relies on aggressive scanning
- Banners can be disabled

XPath-based detection improves these by being

- Generic
- Robust
- Passive

## Foundations: XPath Basics

```
1 <html>
2   <head>
3     <title>Example</title>
4   </head>
5   <body>
6     <h1 class="example">
7       Hello World!
8     </h1>
9   </body>
10 </html>
```

- `//title`  
`<title>Example</title>`
- `/html/body/h1/text()`  
`Hello World!`
- `//h1[@class='example']`  
`<h1 ...> ... </h1>`
- `//body[1]`  
`<body> ... </body>`

The main features in this scenario are derived from HTML files.

```
1 <html>
2   <head>
3     <title>Example</title>
4   </head>
5   <body>
6     <h1 class="example">
7       Hello World!
8     </h1>
9   </body>
10 </html>
```

- `/html`
- `/html/head`
- `/html/head/title`
- `/html/body`
- `/html/body/h1`
- `/html/body/h1[@class=example]`

## Asset Hashing

Additional features are derived from hashing (static) assets (CSS, JS) with SHA256.

**Main advantages:** Performance increase, no equivalents for other formats

## Feature Extraction Process

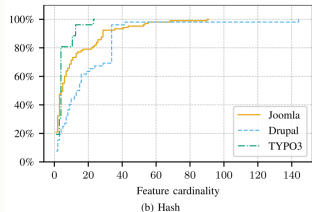
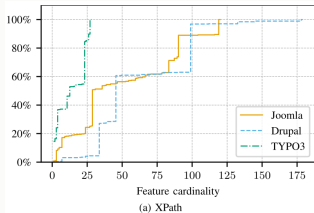
- 1 Generate reference instances
- 2 Extract features
- 3 Store fingerprints
- 4 Feature pruning (take only fingerprints with highest feature cardinality)

## Version Detection Process

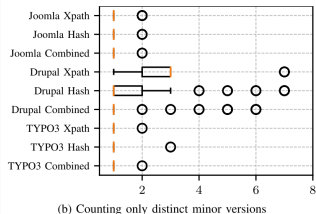
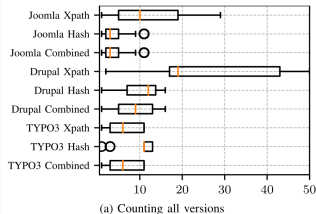
- 1 Extract features
- 2 Compare with fingerprint database
- 3 Select version(s) with highest amount of matching features

# XPath-based detection: Results

## CDF of feature cardinality

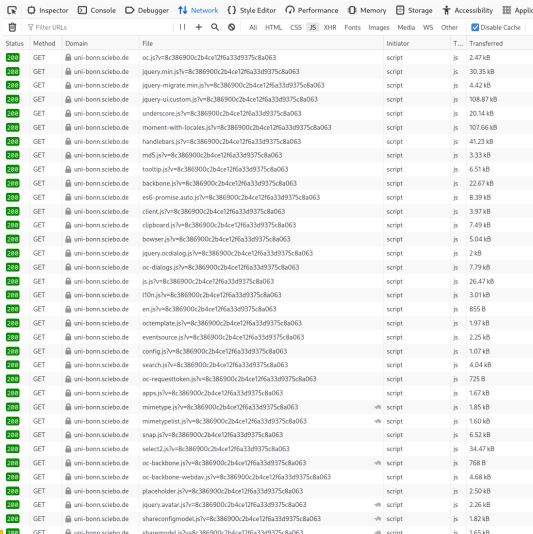


## Amount of candidates



## Why JavaScript Bundling

- Less HTTP requests
- Dependency ordering
- Scoping through IIFEs







Status	Method	Domain	File	Initiator	Transferred
200	GET	uni-bonn.sciebo.de	oc.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	2.47 kB
200	GET	uni-bonn.sciebo.de	jquery.min.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	30.35 kB
200	GET	uni-bonn.sciebo.de	jquery-migrate.min.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	4.42 kB
200	GET	uni-bonn.sciebo.de	jquery-ui.custom.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	108.87 kB
200	GET	uni-bonn.sciebo.de	underscore.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	20.14 kB
200	GET	uni-bonn.sciebo.de	moment-with-locales.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	107.66 kB
200	GET	uni-bonn.sciebo.de	handlebars.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	41.23 kB
200	GET	uni-bonn.sciebo.de	m5.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	3.33 kB
200	GET	uni-bonn.sciebo.de	tooltip.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	6.51 kB
200	GET	uni-bonn.sciebo.de	backbone.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	22.67 kB
200	GET	uni-bonn.sciebo.de	es6-promise.auto.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	8.39 kB
200	GET	uni-bonn.sciebo.de	client.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	3.97 kB
200	GET	uni-bonn.sciebo.de	clipboard.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	7.49 kB
200	GET	uni-bonn.sciebo.de	browser.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	5.04 kB
200	GET	uni-bonn.sciebo.de	jquery.ocdialog.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	2 kB
200	GET	uni-bonn.sciebo.de	oc-dialogs.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	7.79 kB
200	GET	uni-bonn.sciebo.de	js.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	26.47 kB
200	GET	uni-bonn.sciebo.de	l10n.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	3.01 kB
200	GET	uni-bonn.sciebo.de	en.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	855 B
200	GET	uni-bonn.sciebo.de	octemplate.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.97 kB
200	GET	uni-bonn.sciebo.de	eventsource.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	2.25 kB
200	GET	uni-bonn.sciebo.de	config.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.07 kB
200	GET	uni-bonn.sciebo.de	search.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	4.04 kB
200	GET	uni-bonn.sciebo.de	oc-requesttoken.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	725 B
200	GET	uni-bonn.sciebo.de	apps.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.67 kB
200	GET	uni-bonn.sciebo.de	mimetype.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.85 kB
200	GET	uni-bonn.sciebo.de	mimepolylist.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.60 kB
200	GET	uni-bonn.sciebo.de	snap.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	6.52 kB
200	GET	uni-bonn.sciebo.de	select2.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	34.47 kB
200	GET	uni-bonn.sciebo.de	oc-backbone.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	768 B
200	GET	uni-bonn.sciebo.de	oc-backbone-webdav.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	4.68 kB
200	GET	uni-bonn.sciebo.de	placeholder.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	2.50 kB
200	GET	uni-bonn.sciebo.de	jquery.avatar.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	2.26 kB
200	GET	uni-bonn.sciebo.de	shareconfigmodel.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.82 kB
200	GET	uni-bonn.sciebo.de	sharemodel.js?v=8c386900c2b4ce12f6a33d9375c8a063	script	1.65 kB



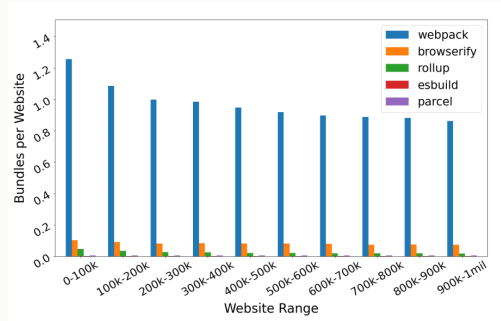
## Foundations: JavaScript Bundling

The bundling process consists of multiple steps:

-  **Tree Shaking**  
Select source files to be bundled
-  **Code Split**  
Detect logical boundaries through dynamic imports between modules
-  **Packaging**  
Wrap all components together
-  **Minification**  
Shrink source code size

## Foundations: JavaScript Bundlers in practice

Several JavaScript bundlers are used in practice. They employ different algorithms and architectures but **share the same approach**.



J. Rack and C. Staicu, "Jack-in-the-box: An Empirical Study of JavaScript Bundling on the Web and its Security Implications," CCS 2023

## Foundations: JavaScript Bundles with Webpack

```

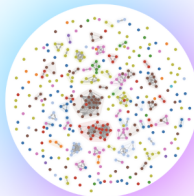
1 (window.webpackJsonp=window.webpackJsonp||[]).push([[80],{
2   maj8:function(e,t,n){"use strict";/* object-assign, (c) Sindre Sorhus, @license MIT */ var r=Object.
    getOwnPropertySymbols,i=Object.prototype.hasOwnProperty,o=Object.prototype.propertyIsEnumerable;function a(e){if(
    null==e)throw new TypeError("Object.assign cannot be called with null or undefined");return Object(e)}...},
3   ZK3j:function(e,t,n){"use strict";var r=n("Y4pH"),i=n("qWlw");function o(e,t){return 55296==(64512&e.charCodeAt(t))&&!(
    t<0||t+1>=e.length)&&56320==(64512&e.charCodeAt(t+1)))}function a(e){return(e>>>24|e>>>8&65280|e<<<8&16711680|(255&e
    <<24)>>>0)}function s(e){return 1===e.length?"0"+e:e.toHex=function(e){for(var t="",n=0;n<e.length;n++)t+=s(e[n].
    toString(16));return t},t.rot32=function(e,t){return e>>>t|e<<<32-t}},
4   xy68:function(e,t,n){"use strict";var r=n("ZK3j"),rot32=function i(e,t,n){return e&t~e&n}function o(e,t,n){return e&t^
    e&n^t&n}function a(e,t,n){return e^t^n}t.ft_1=function(e,t,n,r){return 0===e?i(t,n,r):1===e||3===e?a(t,n,r):2===e?o
    (t,n,r):void 0},t.s1_256=function(e){return r(e,6)^r(e,11)^r(e,25)},t.g0_256=function(e){return r(e,7)^r(e,18)^e
    >>>3},t.g1_256=function(e){return r(e,17)^r(e,19)^e>>>10}}
5 });
  
```

Figure 1: A simplified version of a real-world bundle from nytimes.com. With dashed lines we highlight the compartments and with the arrow we show a direct dependency between the last two compartments.

## Dolos

### Source code plagiarism detection

Quick and easy plagiarism detection for a wide range of programming languages.

[Documentation](#)[Examples](#)[Use Dolos →](#)

#### Free web application

No installation required. Secure, private and fast. Just upload your files and get a report.

[Try Dolos →](#)

#### Multilingual

Dolos supports many programming languages by leveraging the tree-sitter parser library.

[Supported languages →](#)

#### CLI & Library

Run Dolos from the command line or use it as a library in your own project. For advanced users.

[Installation instructions →](#)

#### Open source

View, use and contribute to the source code. Licensed under the MIT license.



#### Fueled by research

Dolos is the result of active research in the field of source code plagiarism detection.



#### Advanced algorithms

Using state-of-the-art algorithms, Dolos helps you discover plagiarism.

# Plagiarism Detection

**General Objective:** Given  $N$  inputs, find similar fragments between them

**Dolos Input:**  $N$  source code files

**Dolos Output:** Pairwise similarity score

Algorithm:

- 1 Tokenization
- 2 Fingerprinting
- 3 Indexing
- 4 Reporting

## Step 1: Tokenization

Immunity against simple modifications is achieved by tokenizing the input into an abstract syntax tree (AST).

```
function sum(a, b) {  
    return a + b;  
}
```

```
program ([1, 0] - [4, 0])  
  function ([1, 0] - [3, 1])  
    identifier ([1, 9] - [1, 12])  
    formal_parameters ([1, 12] - [1, 18])  
      identifier ([1, 13] - [1, 14])  
      identifier ([1, 16] - [1, 17])  
    statement_block ([1, 19] - [3, 1])  
      return_statement ([2, 2] - [2, 15])  
        binary_expression ([2, 9] - [2, 14])  
          identifier ([2, 9] - [2, 10])  
          identifier ([2, 13] - [2, 14])
```

## Step 2: Fingerprinting

Dolos finds common sequences of successive tokens with the following algorithm:

- 1 Split tokens into  $k$ -grams
- 2 Use a fast hashing function
- 3 Select hashes with a windowed rolling hash function  
(Winnowing algorithm, window size  $w$ )

## Step 3: Indexing

For efficient search, the fingerprints are converted into an associative container.

```
index = {  
  hash1: SharedFingerprint {  
    file1: [ occ1, occ2 ],  
    file2: [ occ1 ]  
  },  
  hash2: SharedFingerprint {  
    file1: [ occ3 ],  
    file3: [ occ1 ]  
  }  
}
```



## Step 4: Reporting

There are multiple metrics of comparing source files with each other:

**Similarity:**  $\text{sim}(a, b) = \frac{S_a + S_b}{T_a + T_b}$

**Total Overlap:**  $S_a + S_b$

**Longest fragment:** Longest run of consecutively shared fingerprints

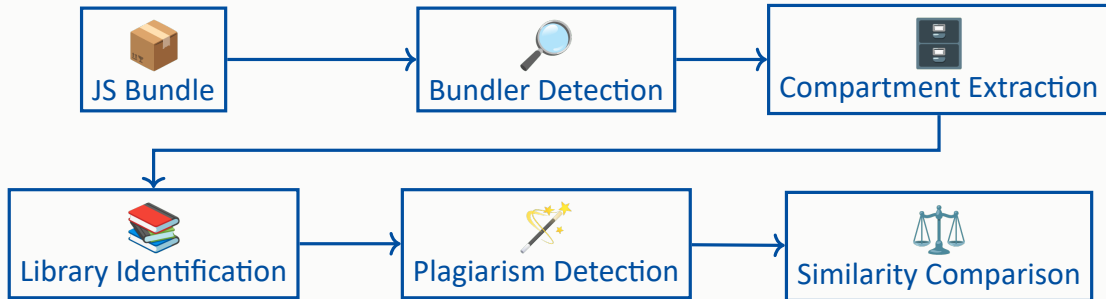
## Plagiarism Detection: Dolos Example

### Comparing chunk-vendors.34601dc3.js with index.js

The compare view matches code fragments & differences between 2 files

[illegible]

## JavaScript Bundle Analysis Pipeline



## Plagiarism Detection

The implementation of Dolos needs some **adaptations, fixes and improvements** to work well with our pipeline.

As Dolos works on single source files, we need to **generate a single file** from a package.

When comparing similarities with Dolos, it is important to **select the correct source files** which end up in the bundle.

Furthermore, it is best to **only consider specific compartments** which belong to the library that is analyzed.

The similarity metric can be improved as it is **inaccurate for small files**.

**Good values for  $k$  and  $w$**  may be chosen adaptively depending on properties of the library.

# Structural Analysis

---

Dynamic Analysis

Structural dynamic analysis compares the behavior of the system under test to those supplied by a reference system.

We consider structural dynamic analysis on the example of automata learning for TLS.

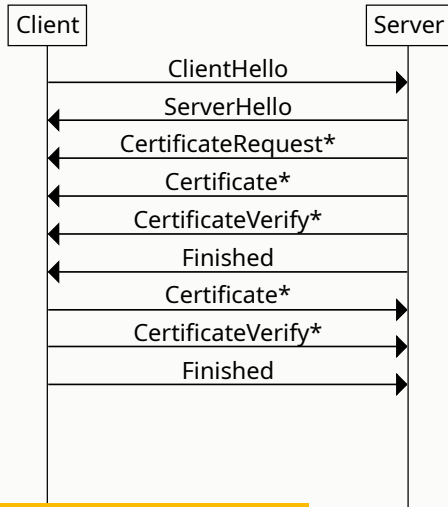
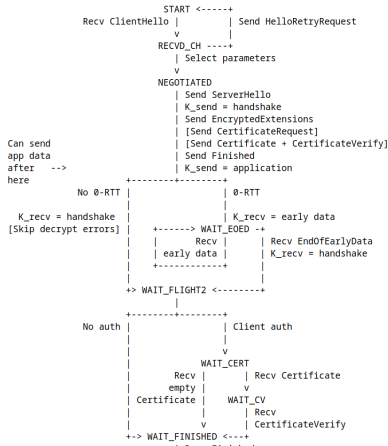
TLS provides confidentiality, authenticity and integrity for a connection.

RFC 8446

TLS

August 2018

## A.2. Server



## Version Detection for TLS

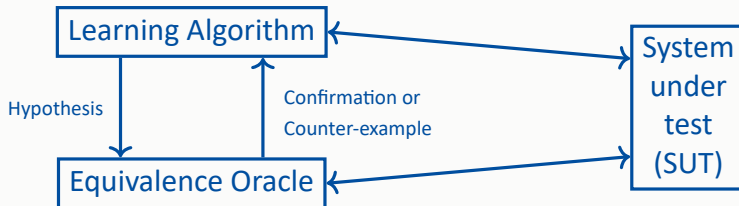
Existing TLS fingerprints focus on **application fingerprinting**. It is used in digital forensics and botnet detection.

Library identification and version detection strongly limits the suitable resources. All packets are processed by the application before being handed over to the library. Additionally, every application **must** configure the library when using it. Even default configurations may be depending on compilation flags.



# Automata Learning

Automata Learning is able to derive the implemented state machine in a black-box scenario. It works with an input alphabet  $\Sigma_I$  and an output alphabet  $\Sigma_O$ .



Examples for active learning algorithms:  $L^*$ ,  $NL^*$ , TTT, AAAR

Examples for equivalence methods: **random walk**, **W-method**, Wp-method, distinguishing bounds

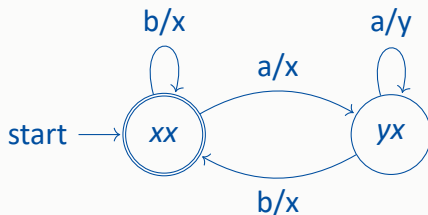
## The $L^*$ algorithm

The central datastructure of the  $L^*$  algorithm is the **observation table**. It classifies members as members or non-members of the SUT's language  $\mathcal{L}$ .

### Observation table

		$E$	
		$a$	$b$
$S$	$\epsilon$	$x$	$x$
	$a$	$y$	$x$
$S.\Sigma_I$	$b$	$x$	$x$
	$a.a$	$y$	$x$
	$a.b$	$x$	$x$

### State machine



## The $L^*$ algorithm

The central datastructure of the  $L^*$  algorithm is the **observation table**. It classifies members as members or non-members of the SUT's language  $\mathcal{L}$ .

The datastructure contains three elements  $(S, E, T)$ .

- $S \subset (\Sigma_I)^*$ : Non-empty prefix-closed set of strings
- $E \subset (\Sigma_I)^*$ : Non-empty suffix-closed set of strings
- $T : (S \cup S \times \Sigma_I) \times E \mapsto \Sigma_O$ : Observed outputs from the SUT

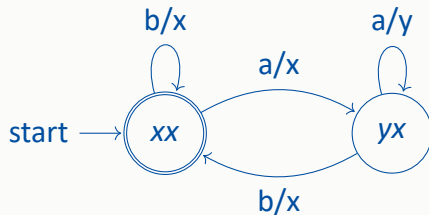
When viewed as a table, it has

- row captions from  $S \cup S \times \Sigma_I$
- column captions from  $E$
- entries given by  $T$

## Observation table

		$E$	
		$a$	$b$
$S$	$\epsilon$	$x$	$x$
	$a$	$y$	$x$
$S.\Sigma_I$	$b$	$x$	$x$
	$a.a$	$y$	$x$
	$a.b$	$x$	$x$

## State machine



The observation table corresponds to a state machine iff it is **closed** and **consistent**.

# The $L^*$ algorithm

## Observation table

		$E$	
		$a$	$b$
$S$	$\epsilon$	$x$	$x$
	$a$	$y$	$x$
$S.\Sigma_I$	$b$	$x$	$x$
	$a.a$	$y$	$x$
	$a.b$	$x$	$x$

An observation table is **closed** if for all  $t \in S \times \Sigma_I$  there exists  $s \in S$  such that  $\text{row}(t) = \text{row}(s)$ .

An observation table is **consistent** if for all  $(s_1, s_2) \in S \times S$  with  $\text{row}(s_1) = \text{row}(s_2)$  then for all  $a \in \Sigma_I$ ,  $\text{row}(s_1 \cdot a) = \text{row}(s_2 \cdot a)$ .

Intuitively, rows represent states. Closure means all states are defined. Consistency means that multiple representations of the same state have the same transitions.

## The $L^*$ algorithm

The learning algorithm consists of three steps.

If the observation table is **closed and consistent**, we can derive our hypothesis.

If the observation table is **not closed**, there exists  $t \in S \cdot \Sigma_i$  with  $\text{row}(t) \neq \text{row}(s)$  for all  $s \in S$ . Thus, add  $t$  to  $S$  and query the SUT for any empty cells.

If the observation table is **not consistent**, there exists  $(s_1, s_2) \in S \times S$  and  $a \in \Sigma_i$  such that  $\text{row}(s_1) = \text{row}(s_2)$  and  $\text{row}(s_1 \cdot a \cdot e) \neq \text{row}(s_2 \cdot a \cdot e)$  with  $e \in E$ . Thus, add  $a \cdot e$  to  $E$  and query the SUT for any empty cells.

Furthermore, if we receive a counter-example  $\mathcal{C}$ , add  $\mathcal{C}$  and all its prefixes to  $S$  and query the SUT for any empty cells.

# The equivalence oracle

There is no equivalence oracle.



Fortunately, we can approximate an equivalence oracle. If some assumptions are fulfilled, we might even get certain guarantees for the result.

**Equivalence method:** Random Walk

A random walk starts in the initial state, performs random transitions and restarts with a certain configurable probability.

# Automata Learning for TLS

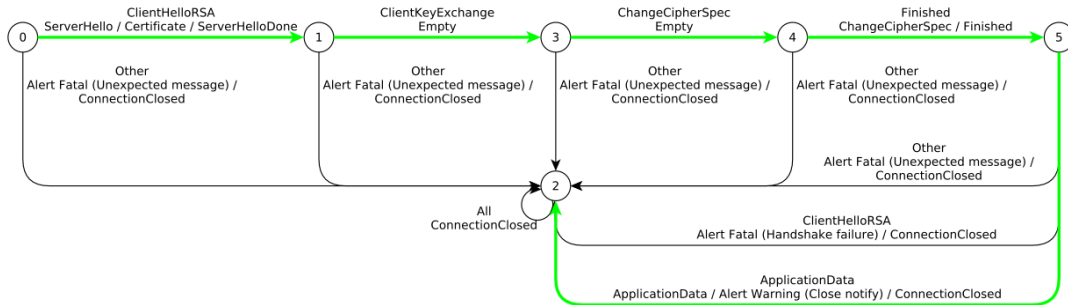
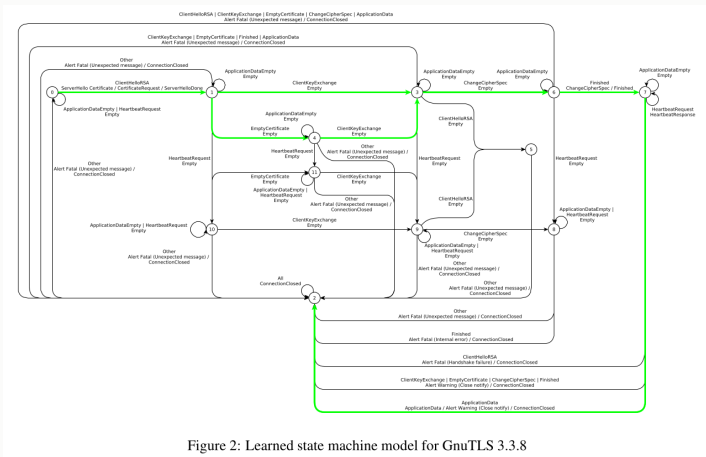


Figure 6: Learned state machine model for RSA BSAFE for Java 6.1.1



# Automata Learning for TLS



# Automata Learning for TLS

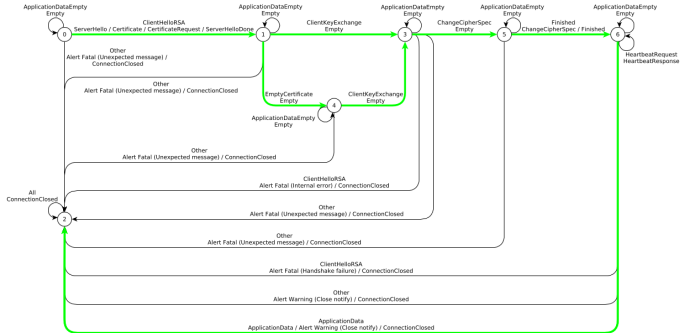
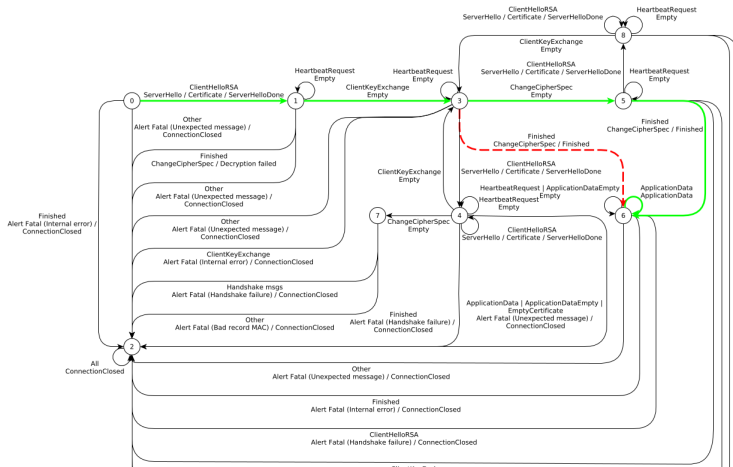


Figure 3: Learned state machine model for GnuTLS 3.3.12. A comparison with the model for GnuTLS 3.3.8 in Fig. 2 shows that the superfluous states (8, 9, 10, and 11) are now gone, confirming that the code has been improved.

# Automata Learning for TLS



# Automata Learning for TLS Library Version Detection

CVE #	Stack	Versions	Status	Comments
2014-0224	OpenSSL	$\leq 0.9.8za$ $\leq 1.0.0l$ $\leq 1.0.1h$	Detected	EarlyCCS (unexpected CCS transitions)
2015-0204	OpenSSL	$\leq 0.9.8zc$ $\leq 1.0.0o$ $\leq 1.0.1j$	Detected	FREAK (client- and server-side EXPORT RSA downgrade)
2015-0205	OpenSSL	$\leq 1.0.0p$ $\leq 1.0.1j$	Not Reproduced	Client auth. bypass. Requires DH certificate support
2020-24613	wolfSSL	$\leq 4.4.0$	Reproduced	TLS 1.3 server auth. bypass
2021-3336	wolfSSL	$\leq 4.6.0$	New	TLS 1.3 server auth. bypass
2022-25638	wolfSSL	$\leq 5.1.0$	New	TLS 1.3 server auth. bypass
2022-25640	wolfSSL	$\leq 5.1.0$	New	TLS 1.3 client auth. bypass

**Advantages:** Robust, configuration independent

**Limitations:** Runtime, narrow view

A. Rasoamanana, "Derivation and Analysis of Cryptographic Protocol Implementations," PhD Thesis, 2023

Questions?

## Possible Seminar/Lab/Master thesis topics:

- ST24 Lab: Automatic TLS client generation
- Active TLS Fingerprinting
- JavaScript Comment Analysis
- Improving JavaScript code comparisons
- WebAssembly Artifact Analysis
- ...

# Ben Swierzy

University of Bonn | Institute of Computer Science 4

[swierzy@cs.uni-bonn.de](mailto:swierzy@cs.uni-bonn.de)