GyoiThon

-- Fully automated penetration test tool --

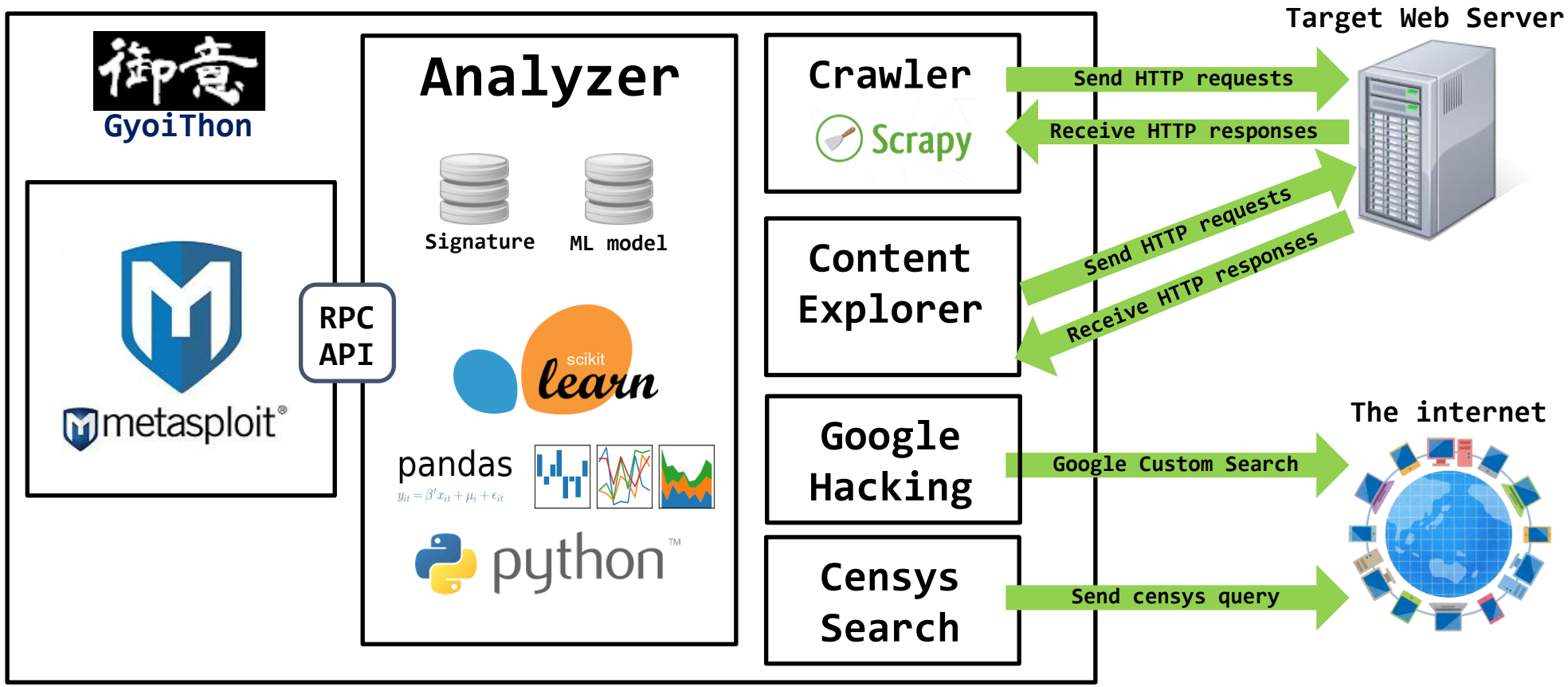
March 28th,2019

Presented by MBSD

#BHASIA @BLACK HAT EVENTS

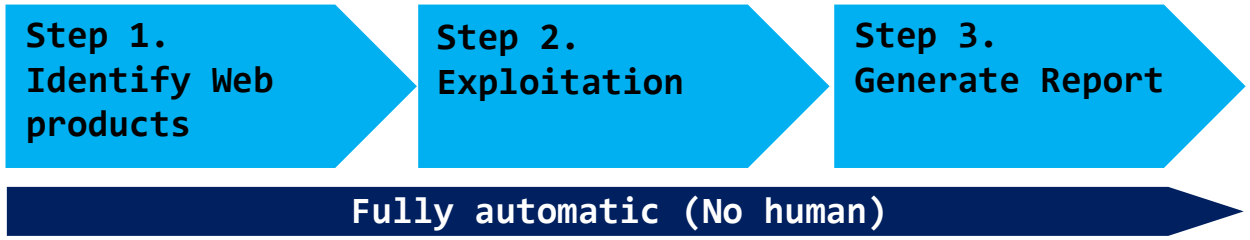
What is GyoiThon?

[*] The GyoiThon is specialized in intelligence gathering of Web Server.



It can gather target server information using several functions.

Processing flow

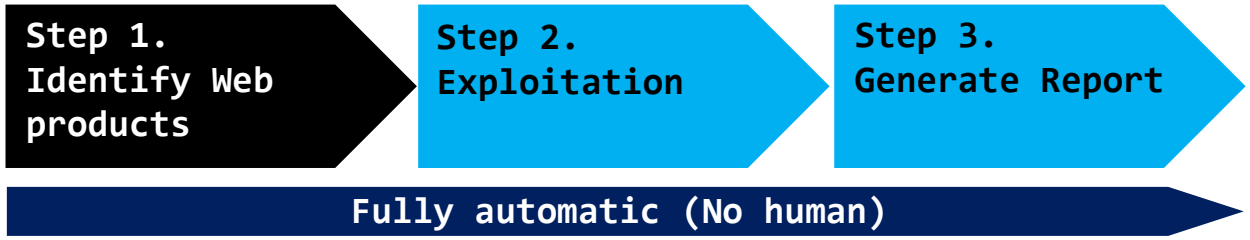


Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Identify Web products



Step 1. Identify Web products

- Gather the HTTP responses by Web crawling.
- Identify the WEB products by Signature (Pattern Matching).
- Identify the WEB products by Machine Learning.

Step 2. Exploitation

Step 3. Generate Report

Question #1

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Content-Length: 37496
Etag: "409ed-183-53c5f732641c0"

…snip…

<form action="/example/confirm.php">
```

What are included the Web products in this HTTP response?

Answer #1

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Content-Length: 37496
Etag: "409ed-183-53c5f732641c0"

…snip…

<form action="/example/confirm.php">
```

Identify **OpenSSL** and **PHP** using **Signature.**

Answer #1-2

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Content-Length: 37496
Etag: "409ed-183-53c5f732641c0"

…snip…

<form action="/example/confirm.php">
```

Identify **joomla!** and **Apache** using **Machine Learning.**

Question #2

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
<link rel="shortcut icon" href="/core/misc/favicon.ico" type="image/vnd.microsoft.icon" />
<link rel="canonical" href="/node/1" />
<link rel="shortlink" href="/node/1" />
<link rel="revision" href="/node/1" />

…snip…

<title>Gyoithon page | Gyoithon page</title>
<link rel="stylesheet" href="/sites/default/files/css/css_tkeOEWfM6OFUIhuI7-a5HCSwE3rIQepiuKNt5Zr7WWg.css
?pom0j6" media="all" />
<link rel="stylesheet" href="/sites/default/files/css/css_jOw317331T1b8jqEXN7KFhd_2yK9S_1cOWw6eHqkdv0.css
?pom0j6" media="all" />
```

What are included the Web product in this HTTP response?

Answer #2

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
<link rel="shortcut icon" href="/core/misc/favicon.ico" type="image/vnd.microsoft.icon" />
<link rel="canonical" href="/node/1" />
<link rel="shortlink" href="/node/1" />
<link rel="revision" href="/node/1" />

…snip…

<title>Gyoithon page | Gyoithon page</title>
<link rel="stylesheet" href="/sites/default/files/css/css_tkeOEWfM6OFUIhuI7-a5HCSwE3rIQepiuKNt5Zr7WWg.css
?pom0j6" media="all" />
<link rel="stylesheet" href="/sites/default/files/css/css_jOw317331T1b8jqEXN7KFhd_2yK9S_1cOWw6eHqkdv0.css
?pom0j6" media="all" />
```

Identify **Drupal** using **Signature** and **Machine Learning.**

Question #3

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

Fully automatic (No human)

```
<title>Gyoithon page</title>
<link href="/templates/protostar/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
<link href="/templates/protostar/css/style.css?f673d87b56120dcab0477660bf1c2384" rel="stylesheet" />

…snip…

<script src="/media/jui/js/jquery.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/jquery-noconflict.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/jquery-migrate.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/system/js/caption.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/bootstrap.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/templates/protostar/js/template.js?f673d87b56120dcab0477660bf1c2384"></script>
```

What are included the Web product in this HTTP response?

Answer #3

Step 1. Identify Web products

Step 2. Exploitation

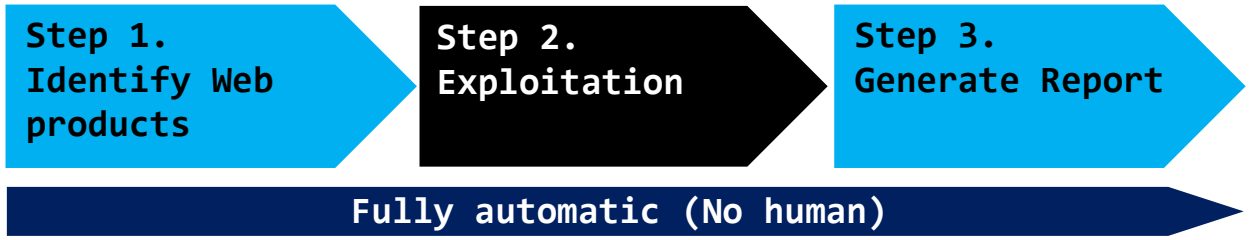
Step 3. Generate Report

Fully automatic (No human)

```
<title>Gyoithon page</title>
<link href="/templates/protostar/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
<link href="/templates/protostar/css/style.css?f673d87b56120dcab0477660bf1c2384" rel="stylesheet" />

…snip…

<script src="/media/jui/js/jquery.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/jquery-noconflict.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/jquery-migrate.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/system/js/caption.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/media/jui/js/bootstrap.min.js?f673d87b56120dcab0477660bf1c2384"></script>
<script src="/templates/protostar/js/template.js?f673d87b56120dcab0477660bf1c2384"></script>
```

Identify **Joomla!** using **Signature** and **Machine Learning.**

Exploitation



Step 1. Identify Web products

Step 2. Exploitation

- **Execute exploit** the identified Web products.
- **Open session** between **"GyoiThon" and target Web server**.

Step 3. Generate Report

Open session between "GyoiThon" and target Web server.

Generate report

Step 1. Identify Web products → Step 2. Exploitation → Step 3. Generate Report

Fully automatic (No human)

Step 1. Identify Web products

Step 2. Exploitation

Step 3. Generate Report

- Create report.

GyoiThon scan Report

Index	Item	Value
1	IP address	192.168.220.145
	Port number	21
	Product name	vsftpd
	Vuln name	VSFTPD v2.3.4 Backdoor Command Execution
	Type	shell
	Description	This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.
	Exploit module	exploit/unix/ftp/vsftpd_234_backdoor
	Target	0
	Payload	payload/cmd/unix/interact
	Reference	[OSVDB] 73573 [URL] http://pastebin.com/AetT9sS5 [URL] http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html

GyoiThon's overview demo.

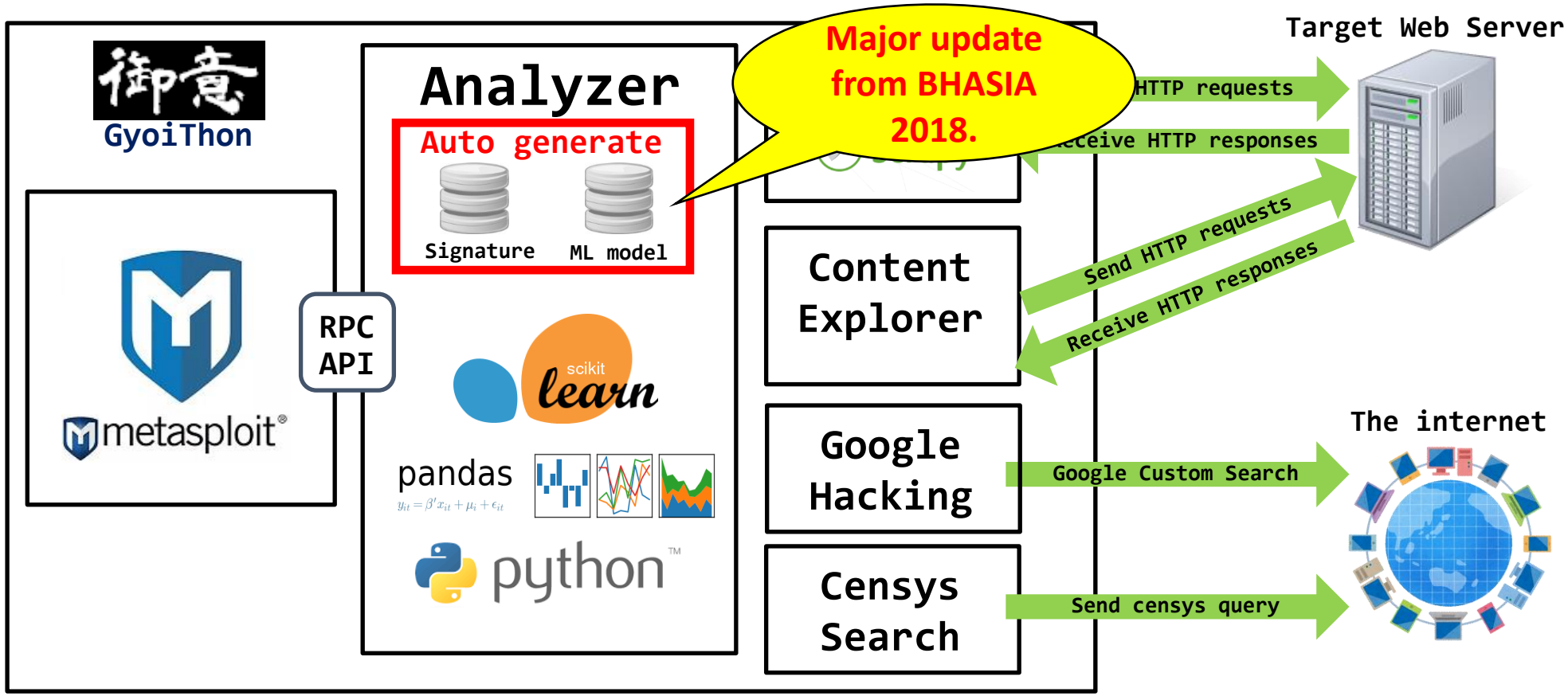
Scenario. Scan the Web server

Demo movie.

https://youtu.be/X8tW4S7c6s0

New function

[*] The GyoiThon can automatically generate the signatures/train data.



The GyoiThon can identify more products.

Why need to auto generate?

[*] Generating signatures/train data is very bother…

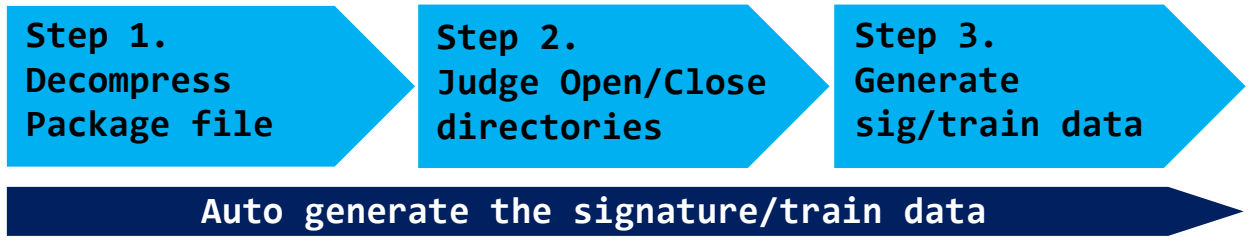
Signatures/train data examples.

```
Drupal    : (/core/misc/favicon.ico)
Drupal    : (/sites/default/files/css/css_tkeOEWfhuI7-a5HCSwE3rIQepiuKNt5Zr7WWg.css)
Drupal    : (/misc/icons/0074bd/chevron-right.svg)
Joomla    : (/templates/protostar/favicon.ico)
Joomla    : (/templates/protostar/css/style.css)
Joomla    : (/media/jui/js/jquery.min.js)
WordPress : (/wp-content/)
WordPress : (/wp-json/)
WordPress : (/wp-links-opml.php)

…snip…
```

Old GyoiThon : **Humans manually generate** signatures/train data.

New GyoiThon : **Machine automatically generates** signatures/train data.

How to automatically generate.



Step 1. Decompress package files.

Step 2. Judge Opened/Closed directories.

Step 3. Generate signature and train data.

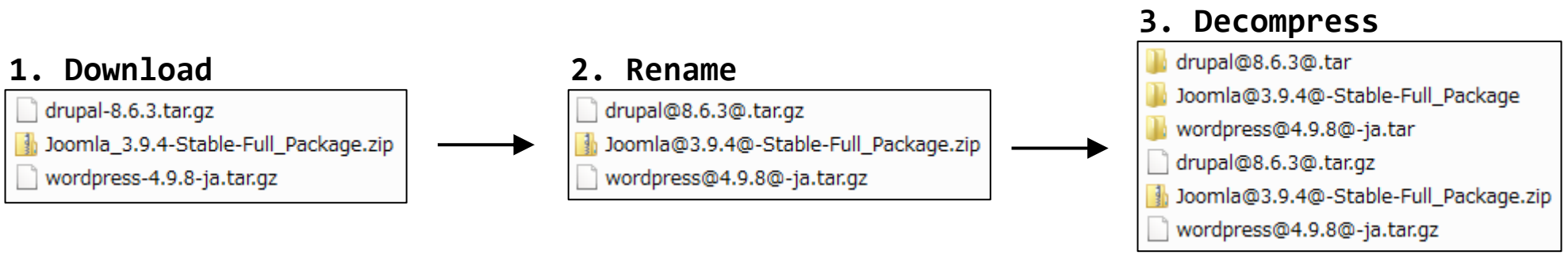
Step1. Decompress target package.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 1. Decompress package files.

1. **Download the package files** such as [drupal-8.6.3.tar.gz], [joomla!-3.9.4-*.zip].
2. **Rename** the package file name to [**product name@version**@extension].
3. **Decompress** the package files.



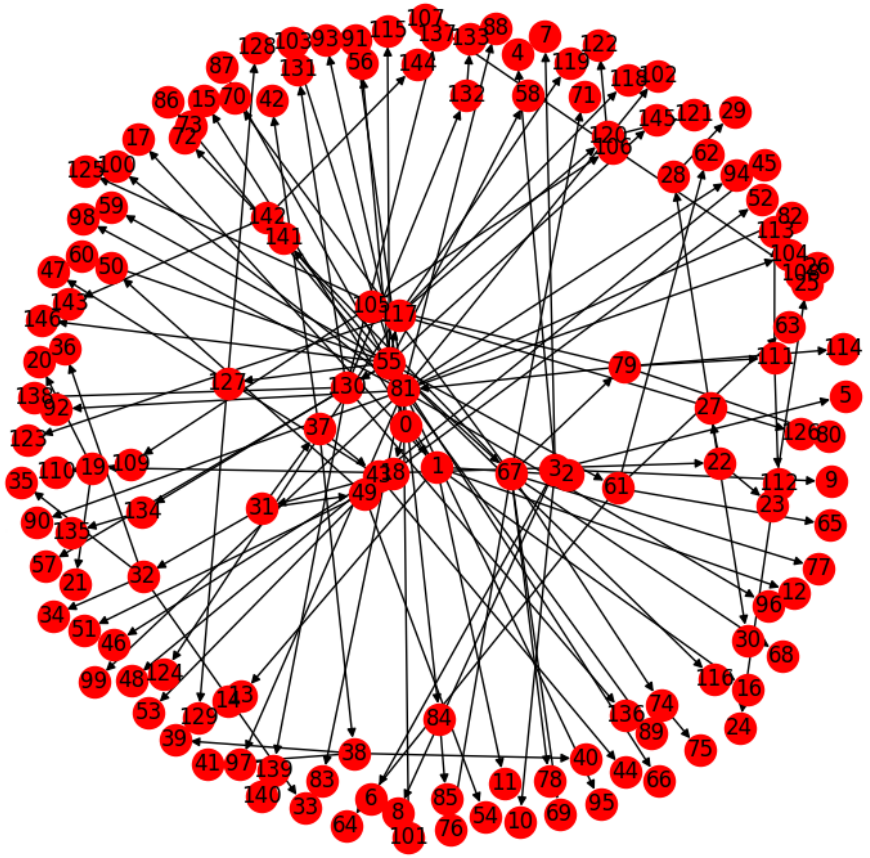
Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

- Create a **package graph** that directory relationships.
- Calculate directory score using score table.
- Judge Opened/Closed directory.



Ex) Joomla! package graph

Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

[*] The directory structure table.

```
Index : Directory path                          : File name on directory
-------------------------------------------------------------------------
 0    : /Joomla-3.9.4/                          : [‘web.config.txt’, ‘LISENCE.txt’]
 1    : /Joomla-3.9.4/administrator/            : [‘index.php’]
 2    : /Joomla-3.9.4/administrator/cache/      : [‘index.html’]

 …snip…

 1434 : /Joomla-3.9.4/media/                    : [‘index.html’]
 1435 : /Joomla-3.9.4/media/jui/                : []
 1436 : /Joomla-3.9.4/media/jui/js/             : [‘template.js’, ‘jquery.min.js’]
```

Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

· **Calculate directory score** using **score table**.

```
Index : Directory path                                  : Score
-------------------------------------------------------------
 0    : /Joomla-3.9.4/                                  : 0.2
 1    : /Joomla-3.9.4/administrator/                    : 0.2
 2    : /Joomla-3.9.4/administrator/cache/              : 0.8

 …snip…

 1434 : /Joomla-3.9.4/media/                            : 1.0
 1435 : /Joomla-3.9.4/media/jui/                        : 0.0
 1436 : /Joomla-3.9.4/media/jui/js/                     : 1.0
```

```
 Extension : Score
--------------------
 .txt      : 0.2
 .combine  : 1.0
 .config   : 0.2
 .css      : 1.0

 …snip…

 .html     : 0.8
 .ico      : 1.0
 .ini      : 0.2
 .jpg      : 1.0
 .js       : 1.0
 .php      : 0.2
```

Score table

Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

· Judge **Opened/Closed directory** (Opened: score≧threshold).

```
Index : Directory path                              : Score : Status
-------------------------------------------------------------------
 0    : /Joomla-3.9.4/                              : 0.2   : Closed
 1    : /Joomla-3.9.4/administrator/                : 0.2   : Closed
 2    : /Joomla-3.9.4/administrator/cache/          : 0.8   : Opened

 …snip…

 1434 : /Joomla-3.9.4/media/                        : 1.0   : Opened
 1435 : /Joomla-3.9.4/media/jui/                    : 0.0   : Closed
 1436 : /Joomla-3.9.4/media/jui/js/                 : 1.0   : Opened
```

Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

· **Turn inside out closed directory** between opened directories.

```
Index : Directory path                              : Score : Status
-------------------------------------------------------------------
 0    : /Joomla-3.9.4/                              : 0.2   : Closed
 1    : /Joomla-3.9.4/administrator/                : 0.2   : Closed
 2    : /Joomla-3.9.4/administrator/cache/          : 0.8   : Opened

 …snip…

 1434 : /Joomla-3.9.4/media/                        : 1.0   : Opened
 1435 : /Joomla-3.9.4/media/jui/                    : 1.0   : Opened
 1436 : /Joomla-3.9.4/media/jui/js/                 : 1.0   : Opened
```

Step2. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 2. Judge Open/Close directories.

· Judgement result.

```
Index : Directory path                                  : Score : Status
-------------------------------------------------------------------------
 0    : /Joomla-3.9.4/                                  : 0.2   : Closed
 1    : /Joomla-3.9.4/administrator/                    : 0.2   : Closed
 2    : /Joomla-3.9.4/administrator/cache/              : 0.8   : Opened

 …snip…

 1434 : /Joomla-3.9.4/media/                            : 1.0   : Opened
 1435 : /Joomla-3.9.4/media/jui/                        : 1.0   : Opened
 1436 : /Joomla-3.9.4/media/jui/js/                     : 1.0   : Opened
```

Step3. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → **Step 3. Generate sig/train data**

Auto generate the signature/train data

Step 3. Generate signature and train data.

· **Remove the closed path** from the **opened directory path**.

```
Index : Directory path                                  : Score : Status
-------------------------------------------------------------------------
 0    : /Joomla-3.9.4/                                  : 0.2   : Closed
 1    : /Joomla-3.9.4/administrator/                    : 0.2   : Closed
 2    : [  Removed  ]/[  Removed  ]/cache/              : 0.8   : Opened

 …snip…

 1434 : [  Removed  ]/media/                            : 1.0   : Opened
 1435 : [  Removed  ]/media/jui/                        : 1.0   : Opened
 1436 : [  Removed  ]/media/jui/js/                     : 1.0   : Opened
```

Step3. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → Step 3. Generate sig/train data

Auto generate the signature/train data

Step 3. Generate signature and train data.

· Combine opened path and files.

```
Opened path      : File name on the opened path
-------------------------------------------------
/cache/          : [‘index.html’]
/media/          : [‘index.html’]
/media/jui/      : []
/media/jui/js/   : [‘template.js’, ‘jquery.min.js’]
```

Step3. Analyze directory structure.

Step 1. Decompress Package file → Step 2. Judge Open/Close directories → **Step 3. Generate sig/train data**

Auto generate the signature/train data

Step 3. Generate signature and train data.

[*] Generated signature/train data.

```
/cache/
/cache/index.html
/media/
/media/index.html
/media/jui/
/media/jui/js/
/media/jui/js/template.js
/media/jui/js/jquery.min.js

…snip…
```

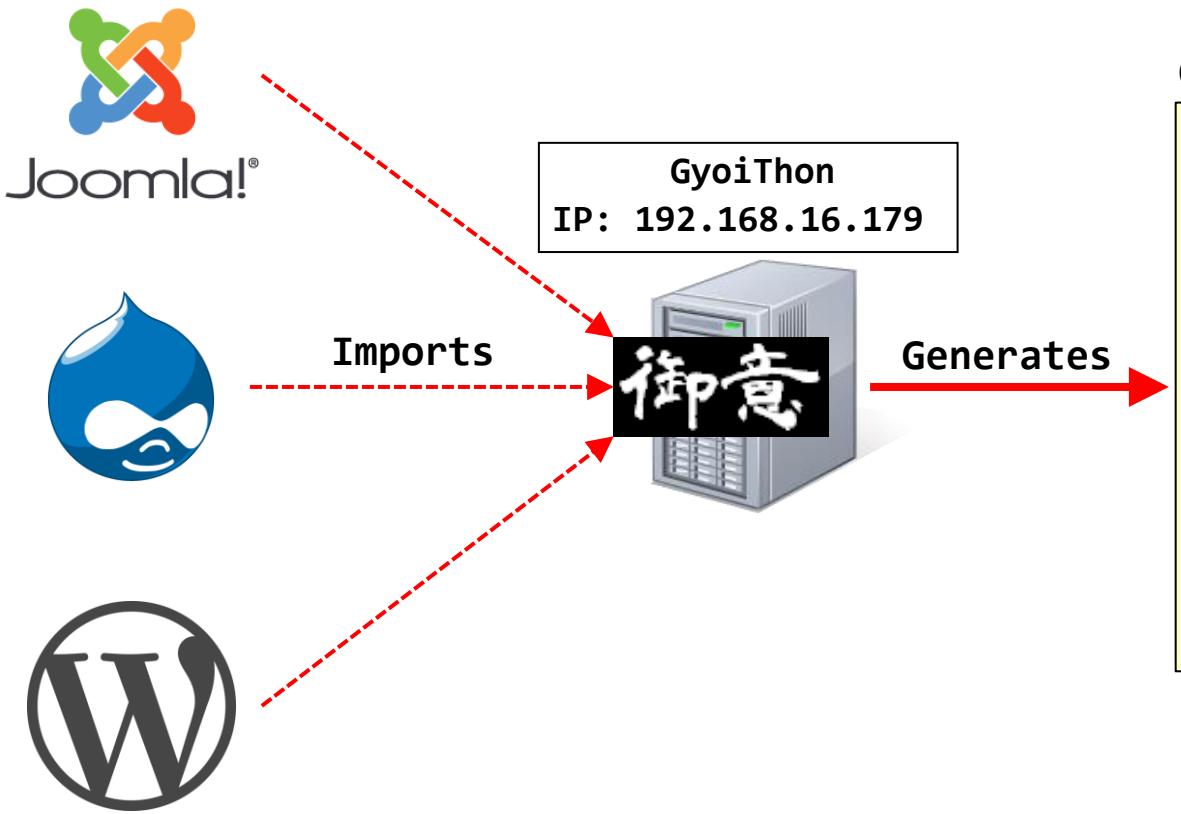
can be specified joomla!

```
<!DOCTYPE html>
<html lang="ja-jp" dir="ltr">
<head>
<title>This site is used Joomla!</title>

…snip…

<script src="/media/jui/js/jquery.min.js"></script>
<script src="/media/jui/js/jquery-migrate.min.js"></script>
<script src="/media/jui/js/template.js"></script>
<script src="/templates/protostar/js/template.js"></script>

…snip…
```

Ex) Sample site using Joomla!

Auto generate demo.

GyoiThon automatically generates three CMS signatures/train data.



Generated signatures/train data by GyoiThon.

```
Joomla    : (/templates/protostar/favicon.ico)
Joomla    : (/templates/protostar/css/style.css)
Joomla    : (/media/jui/js/jquery.min.js)
Drupal    : (/core/misc/favicon.ico)
Drupal    : (/sites/default/files/css/css_tkeOE.css)
Drupal    : (/misc/icons/0074bd/chevron-right.svg)
WordPress : (/wp-content/)
WordPress : (/wp-json/)
WordPress : (/wp-links-opml.php)

…snip…
```

Demo movie.

https://youtu.be/X8tW4S7c6s0

Resource

- Source codes & Usage

https://github.com/gyoisamurai/GyoiThon

Who we are

Company	:	**MBSD** - Mitsui Bussan Secure Directions, Inc.
Established	:	2001
Head office	:	Tokyo, Japan
Paid in capital	:	JPY 400 Mil (100% subsidiary of Mitsui & Co., Ltd)
Employees	:	256
Industry affiliations	:	Leading companies in Japan, such as telecoms, banks, retailers, internet business and the governments.
Businesses	:	Professional security services to protect business from cyber attacks. Vulnerability Assessment/Penetration test (Web/NW/Internet of Things…)
Services	:	Managed Security Services, Incident Response, GRC Consulting, R&D.

THANK YOU!

Reference all source codes and document:

https://github.com/gyoisamurai/GyoiThon