從冷知識到漏洞
你不懂的 Web，駭客懂

Huli @ WebConf Taiwan 2025

Huli

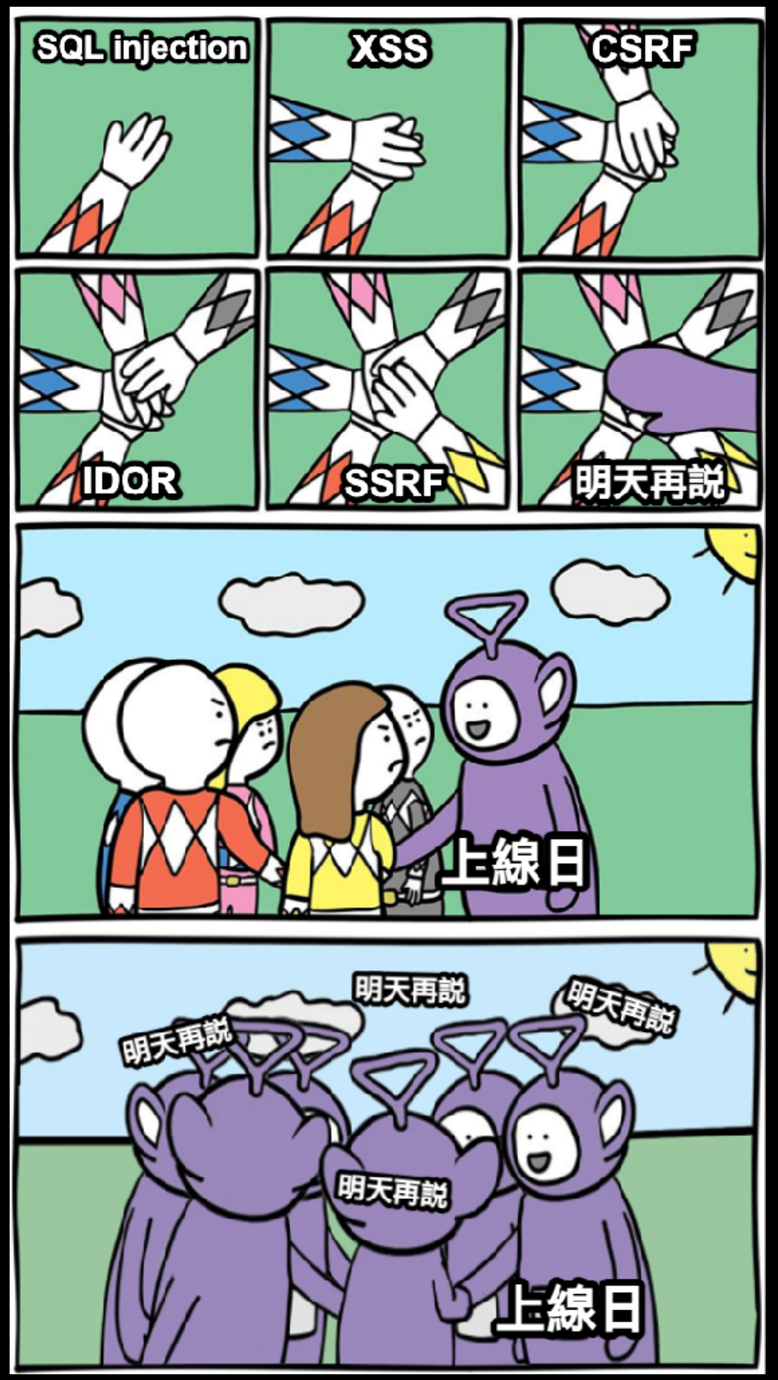
(Front-end | Security) Engineer

八成 coding 交給 AI，另外兩成自己來

漏洞分三種

「我知道有漏洞
但上線快來不及了 😠」

「你先建票，我們之後修」



「好像有碰過但我忘了 🤔」

「我不知道這樣寫有問題 😱」

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    userService.sendResetLink({
      link: userService.generateLink(user.id),
      to: email,
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    userService.sendResetLink({
      link: userService.generateLink(user.id),
      to: email,
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

用 email 找出 user

寄信

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    const resetLink = userService.generateLink(user.id);
    mailer.send({
      link: userService.generateLink(user.id),
      to: email,
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

用 email 找出 user

寄信

user.email 一定與 email 相等嗎？

```
mysql>
SELECT 'gmail.com' = 'GMAIL.com' COLLATE utf8mb4_unicode_ci;

+-------------------------------------------------+
| 'gmail.com' = ‘GMAIL.com' COLLATE utf8mb4_unicode_ci |
+-------------------------------------------------+
|                                               1 |
+-------------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql>
SELECT 'gmail.com' = 'gmaÎĺ.com' COLLATE utf8mb4_unicode_ci;

+-----------------------------------------------------+
| 'gmail.com' = 'gmaÎĺ.com' COLLATE utf8mb4_unicode_ci |
+-----------------------------------------------------+
|                                                   1 |
+-----------------------------------------------------+
1 row in set (0.00 sec)
```

12.14.1 Collation Implementation Types

```
mysql> SELECT c1, HEX(c1), HEX(WEIGHT_STRING(c1)) FROM t1;
+------+---------+--------------------------+
| c1   | HEX(c1) | HEX(WEIGHT_STRING(c1)) |
+------+---------+--------------------------+
| a    | 61      | 0041                     |
| A    | 41      | 0041                     |
| À    | C380    | 0041                     |
| á    | C3A1    | 0041                     |
+------+---------+--------------------------+
4 rows in set (0.00 sec)
```

utf8mb4_general_ci is an example: 'a', 'A', 'À', and 'á' each have different character codes but all have a weight of 0x0041 and compare as equal.

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    userService.sendResetLink({
      link: userService.generateLink(user.id),
      to: email,
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

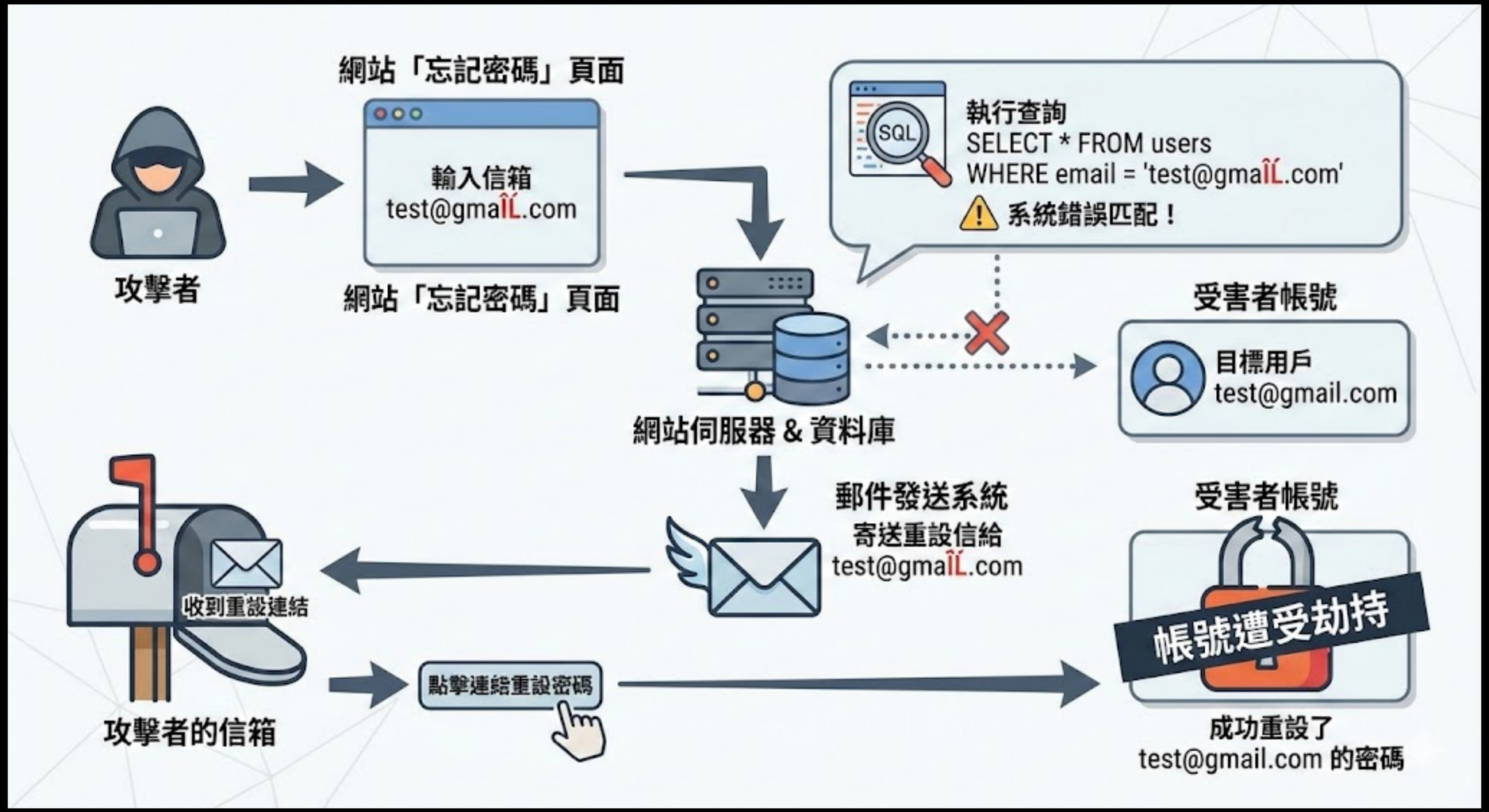
用 email 找出 user

寄信

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    userService.sendResetLink({
      link: userService.generateLink(user.id),
      to: email,
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

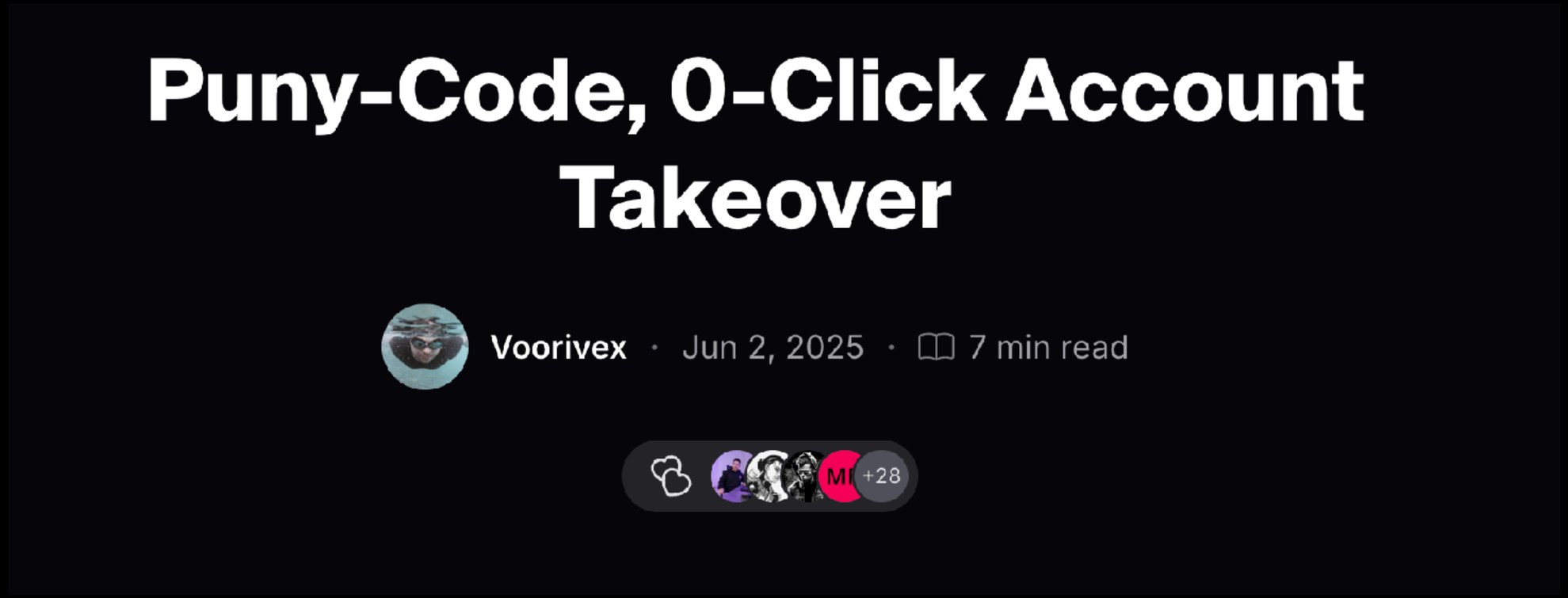
test@gmaÎĹ.com = test@gmail.com

寄信 test@gmaÎĹ.com


網站「忘記密碼」頁面
攻擊者
輸入信箱
test@gmaÎĹ.com
網站「忘記密碼」頁面
執行查詢
SELECT * FROM users
WHERE email = 'test@gmaÎĹ.com'
系統錯誤匹配！
受害者帳號
目標用戶
test@gmail.com
網站伺服器 & 資料庫
郵件發送系統
寄送重設信給
test@gmaÎĹ.com
受害者帳號
收到重設連結
帳號遭受劫持
攻擊者的信箱
點擊連結重設密碼
成功重設了
test@gmail.com 的密碼

```
async function forgotPassword(req, res) {
  const email = req.body.email.toLowerCase();
  const user = await safeSQL(
    'select * from users where email = ?',[email]
  )

  if (user) {
    userService.sendResetLink({
      link: userService.generateLink(user.id),
      to: email, user.email
    }).catch(console.error);
  }

  return res.status(204).end();
}
```

Credit to Voorivex



防禦心法第一條

不同 context 的相等就是不相等

```
const tmpl = '<button value="{{value}}">click</button>'
const value = new URL(location.href).searchParams.get('v')
const safeValue = value.replace(/[<>"]/g, '')
document.body.innerHTML = tmpl.replace('{{value}}', value)
```

```
const tmpl = '<button value="{{value}}">click</button>'
const value = new URL(location.href).searchParams.get('v')
const safeValue = value.replace(/[<>"]/g, '')
document.body.innerHTML = tmpl.replace('{{value}}', value)
```

移除特殊字元

取代

JavaScript 是世界上
最好的程式語言

~ 胡適

沒有說過

Specifying a string as the replacement

The replacement string can include the following special replacement patterns:

Pattern	Inserts
`$$`	Inserts a `"$"`.
`$&`	Inserts the matched substring.
`$``	Inserts the portion of the string that precedes the matched substring.
`$'`	Inserts the portion of the string that follows the matched substring.
`$n`	Inserts the `n`th (`1`-indexed) capturing group where `n` is a positive integer less than `100`.
`$<Name>`	Inserts the named capturing group where `Name` is the group name.

```
                    $`          $'
const tmpl = 'hello{name}world'
```

```
tmpl.replace("{name}","huli")
```

```
hellohuliworld
```

```
                    $`          $'
const tmpl = 'hello{name}world'
```

```
tmpl.replace("{name}","huli")
```

```
hellohuliworld
```

```
tmpl.replace("{name}","$`")
```

```
hellohelloworld
```

```
                    $`          $'
const tmpl = 'hello{name}world'
```

```
tmpl.replace("{name}","huli")   → hellohuliworld
tmpl.replace("{name}","$`")     → hellohelloworld
tmpl.replace("{name}","$'")     → helloworldworld
```

```
                    $`          $'
const tmpl = 'hello{name}world'
```

```
tmpl.replace("{name}","huli")    → hellohuliworld
tmpl.replace("{name}","$`")      → hellohelloworld
tmpl.replace("{name}","$'")      → helloworldworld
tmpl.replace("{name}","huli$'")  → hellohuliworldworld
```

```
$`                                  $'
<button value="{{value}}">click</button>
```

```
                $`                                  $'
  <button value="{{value}}">click</button>

  $` 123
→ <button value="<button value=" 123">click</button>
```

```
$`                                  $'
<button value="{{value}}">click</button>
```

```
$` 123
<button value="<button value=" 123">click</button>
```

→ `<button value="<button value=" 123">click</button>`

```
$`                                  $'
<button value="{{value}}">click</button>
```

```
$` 123
<button value="<button value=" 123">click</button>

<button value="<button value=" 123">click</button>
```

```
$` onclick='alert()'
<button value="<button value=" onclick='alert()'">click</button>
```

Credit to nitowa

Plaid CTF: Yet Another Calculator App

Participant: Peter Millauer / nitowa (01350868)

TL;DR / Short Summary

Classical XSS web exploit. The solution used special string replacement patterns to break out of string escapes.

Task Description

Type: Web

Task sources: https://gitea.nitowa.xyz/nitowa/PlaidCTF-YACA

Goal: Steal the cookie of the page-worker

防禦心法第二條

在 AI 取代人之前，先小心你的字串被取代

```go
func main() {
    filename := "test.js" // user-controlled
    pluginFilePath := filepath.Join("/etc/plugins", filename)

    fmt.Printf("Output: %q\n", pluginFilePath)
}
```

```go
func main() {
    filename := "../../etc/hosts" // user-controlled
    pluginFilePath := filepath.Join("/etc/plugins", filename)

    fmt.Printf("Output: %q\n", pluginFilePath)
}
```

```
func main() {
    filename := filepath.Clean("../../etc/hosts")
    pluginFilePath := filepath.Join("/etc/plugins", filename)

    fmt.Printf("Output: %q\n", pluginFilePath)
}
```

func Clean

```
func Clean(path string) string
```

Clean returns the shortest path name equivalent to path by purely lexical processing. It applies the following rules iteratively until no further processing can be done:

1. Replace multiple Separator elements with a single one.
2. Eliminate each . path name element (the current directory).
3. Eliminate each inner .. path name element (the parent directory) along with the non-.. element that precedes it.
4. Eliminate .. elements that begin a rooted path: that is, replace "/.." by "/" at the beginning of a path, assuming Separator is '/'.

func Clean

```
func Clean(path string) string
```

Clean returns the shortest path name equivalent to path by purely lexical processing. It applies the following rules iteratively until no further processing can be done:

1. Replace multiple Separator elements with a single one.
2. Eliminate each . path name element (the current directory).
3. Eliminate each inner .. path name element (the parent directory) along with the non-.. element that precedes it.
4. Eliminate .. elements that begin a rooted path: that is, replace "/.." by "/" at the beginning of a path, assuming Separator is '/'.

func Clean

```
func Clean(path string) string
```

Clean returns the shortest path name equivalent to path by purely lexical processing. It applies the following rules iteratively until no further processing can be done:

1. Replace multiple Separator elements with a single one.
2. Eliminate each . path name element (the current directory).
3. Eliminate each inner .. path name element (the parent directory) along with the non-.. element that precedes it.
4. Eliminate .. elements that begin a rooted path: that is, replace "/.." by "/" at the beginning of a path, assuming Separator is '/'.

func Clean

```
func Clean(p
```

Clean returns t . It applies the following rules

1. Replace m
2. Eliminate e
3. Eliminate e non-.. element that prece
4. Eliminate . beginning of a path, assu

```
  requestedFile := filepath.Clean(web.Params(c.Req)["*"])
  pluginFilePath := filepath.Join(plugin.PluginDir, requestedFile)

  // It's safe to ignore gosec warning G304 since we already clean
the requested file path and subsequently
  // use this with a prefix of the plugin's directory, which is set
during plugin loading
  // nolint:gosec
  f, err := os.Open(pluginFilePath)
```

```
requestedFile := filepath.Clean(web.Params(c.Req)["*"])
pluginFilePath := filepath.Join(plugin.PluginDir, requestedFile)

// It's safe to ignore gosec warning G304 since we already clean
the requested file path and subsequently
// use a prefix for the plugin folder, which is set
during plugin loading
// nolint:gosec
f, err := os.Open(pluginFilePath)
```

你的警告我知道了，但這很安全我已經 clean 了，你多慮了

GitHub Advisory Database / GitHub Reviewed / CVE-2021-43798

Grafana path traversal

High severity | GitHub Reviewed | Published on Dec 8, 2021 in **grafana/grafana** • Updated on Oct 23

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
github.com/grafana/grafana (Go)	>= 8.3.0, < 8.3.1	8.3.1
	>= 8.2.0, < 8.2.7	8.2.7
	>= 8.1.0, < 8.1.8	8.1.8
	>= 8.0.0-beta1, < 8.0.7	8.0.7

Severity

High 7.5 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None

Credit to Jordy Versmissen

SECURITY GUIDANCE

How I found the Grafana zero-day Path Traversal exploit that gave me access to your logs

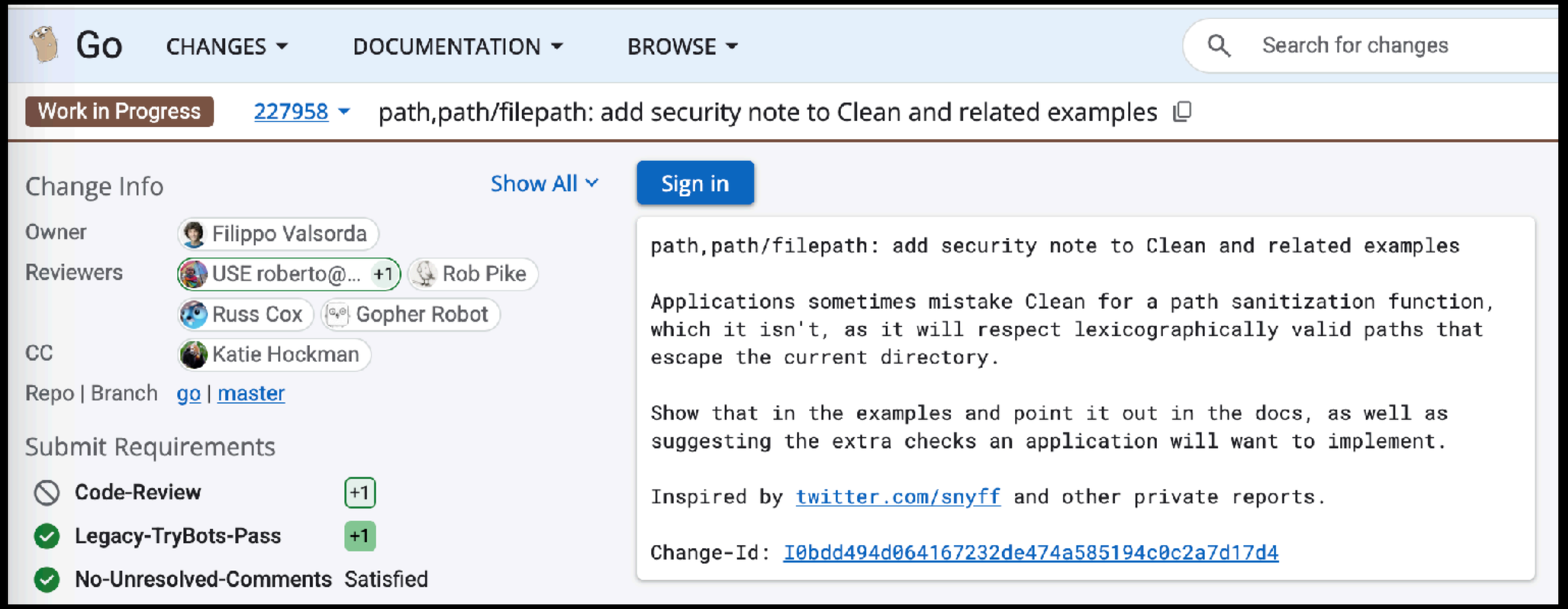
Jordy Versmissen Dec 15, 2021

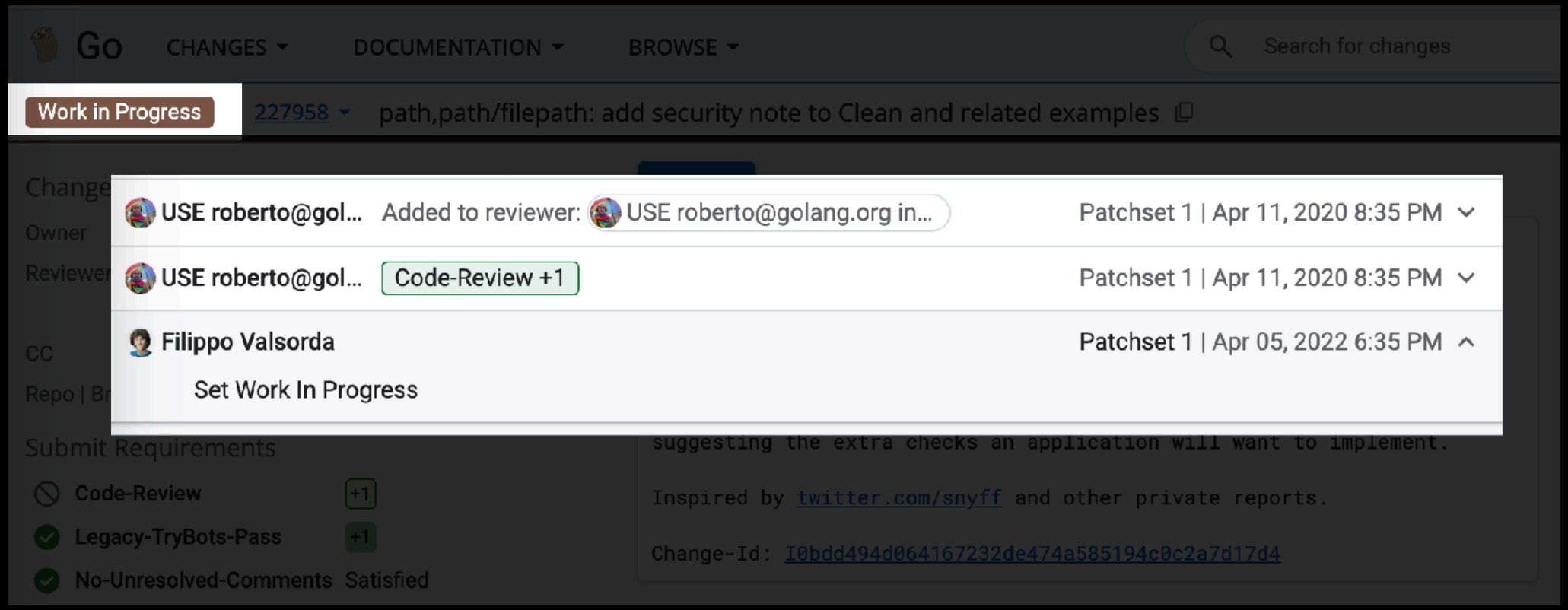
```python
import os

filename = '../../etc/hosts'
if '..' in filename:
  raise ValueError('invalid filename')
result = os.path.join('/tmp/test', filename)
print(result) # ValueError: invalid filename
```

```python
import os

filename = '/etc/hosts'
if '..' in filename:
  raise ValueError('invalid filename')
result = os.path.join('/tmp/test', filename)
print(result) # /etc/hosts
```

`os.path.join(path, /, *paths)`

Join one or more path segments intelligently. The return value is the concatenation of *path* and all members of **paths*, with exactly one directory separator following each non-empty part, except the last. That is, the result will only end in a separator if the last part is either empty or ends in a separator. If a segment is an absolute path (which on Windows requires both a drive and a root), then all previous segments are ignored and joining continues from the absolute path segment.


Go
CHANGES
DOCUMENTATION
BROWSE
Search for changes
Work in Progress
227958
path,path/filepath: add security note to Clean and related examples
Change Info
Show All
Sign in
Owner
Filippo Valsorda
Reviewers
USE roberto@... +1
Rob Pike
Russ Cox
Gopher Robot
CC
Katie Hockman
Repo | Branch
go | master
Submit Requirements
Code-Review
+1
Legacy-TryBots-Pass
+1
No-Unresolved-Comments
Satisfied
path,path/filepath: add security note to Clean and related examples
Applications sometimes mistake Clean for a path sanitization function,
which it isn't, as it will respect lexicographically valid paths that
escape the current directory.
Show that in the examples and point it out in the docs, as well as
suggesting the extra checks an application will want to implement.
Inspired by twitter.com/snyff and other private reports.
Change-Id: I0bdd494d064167232de474a585194c0c2a7d17d4


Go
CHANGES
DOCUMENTATION
BROWSE
Search for changes
Work in Progress
227958
path,path/filepath: add security note to Clean and related examples
USE roberto@gol...
Added to reviewer:
USE roberto@golang.org in...
Patchset 1 | Apr 11, 2020 8:35 PM
USE roberto@gol...
Code-Review +1
Patchset 1 | Apr 11, 2020 8:35 PM
Filippo Valsorda
Patchset 1 | Apr 05, 2022 6:35 PM
Set Work In Progress
Owner
Reviewer
CC
Repo | Br
Submit Requirements
Code-Review +1
Legacy-TryBots-Pass +1
No-Unresolved-Comments Satisfied
suggesting the extra checks an application will want to implement.
Inspired by twitter.com/snyff and other private reports.
Change-Id: I0bdd494d064167232de474a585194c0c2a7d17d4

防禦心法第三條

防禦心法第三條

寫好測試，確保你的 output 是對的

```
function App() {
  const params = new URLSearchParams(window.location.search)
  const obj = Object.fromEntries(params)

  return (
    <div {...obj}>Hello</div>
  )
}
```

```
function App() {
  const params = new URLSearchParams(window.location.search)
  const obj = Object.fromEntries(params)

  return (
    <div {...obj}>Hello</div>
  )
}
```

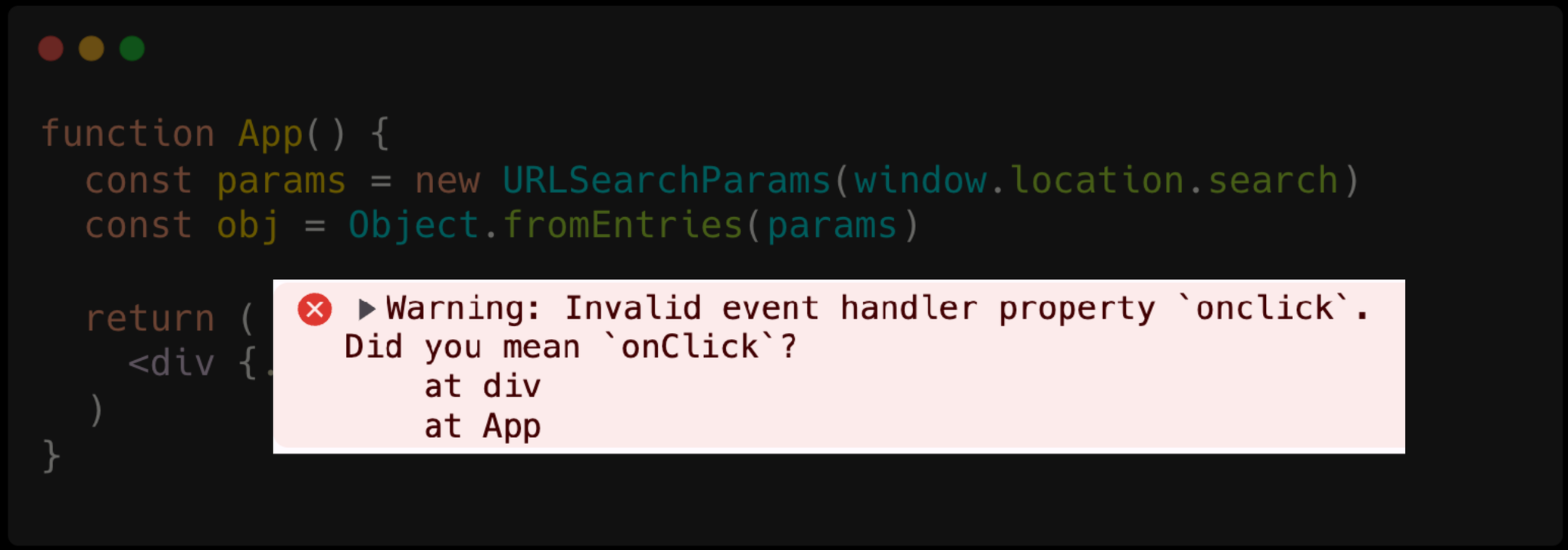
dangerouslySetInnerHTML?

```
function App() {
  const params = new URLSearchParams(window.location.search)
  const obj = Object.fromEntries(params)

  return (
    <div {...obj}>Hello</div>
  )
}
```

dangerouslySetInnerHTML: { __html: "..." }

```
function App() {
  const params = new URLSearchParams(window.location.search)
  const obj = Object.fromEntries(params)

  return (
    <div {...obj}>Hello</div>
  )
}
```

onclick / onClick?


function App() {
const params = new URLSearchParams(window.location.search)
const obj = Object.fromEntries(params)
return (
<div {
)
}
▶Warning: Invalid event handler property `onclick`.
Did you mean `onClick`?
at div
at App

```
function App() {
  const params = new URLSearchParams(window.location.search)
  const obj = Object.fromEntries(params)

  return (
    <div {
  )
}
```

```
▶ Warning: Expected `onClick` listener to be a
function, instead got a value of `string` type.
    at div
    at App
```

```jsx
function App() {
  const obj = {
    is: 'huli',
    onclick: 'alert(1)',
  }

  return (
    <div {...obj}>Hello</div>
  )
}
```

react / packages / react-dom / src / client / **DOMPropertyOperations.js**

Code | Blame | 263 lines (247 loc) · 8.05 KB

```
143     */
144    export function setValueForProperty(
145      node: Element,
146      name: string,
147      value: mixed,
148      isCustomComponentTag: boolean,
149    ) {
150      const propertyInfo = getPropertyInfo(name);
151      if (shouldIgnoreAttribute(name, propertyInfo, isCustomComponentTag)) {
152        return;
153      }
154
155      if (
156        enableCustomElementPropertySupport &&
157        isCustomComponentTag &&
158        name[0] === 'o' &&
159        name[1] === 'n'
160      ) {
161        let eventName = name.replace(/Capture$/, '');
162        const useCapture = name !== eventName;
163        eventName = eventName.slice(2);
```

on 開頭會被 ignore

react / packages / react-dom / src / shared / DOMProperty.js

Code | Blame | 634 lines (590 loc) · 16.8 KB

```
94  export function shouldIgnoreAttribute(
95    name: string,
96    propertyInfo: PropertyInfo | null,
97    isCustomComponentTag: boolean,
98  ): boolean {
99    if (propertyInfo !== null) {
100     return propertyInfo.type === RESERVED;
101   }
102   if (isCustomComponentTag) {
103     return false;
104   }
105   if (
106     name.length > 2 &&
107     (name[0] === 'o' || name[0] === 'O') &&
108     (name[1] === 'n' || name[1] === 'N')
109   ) {
110     return true;
111   }
112   return false;
113 }
```

react / packages / react-dom / src / shared / DOMProperty.js

Code | Blame | 634 lines (590 loc) · 16.8 KB

```
94  export function shouldIgnoreAttribute(
95    name: string,
96    propertyInfo: PropertyInfo | null,
97    isCustomComponentTag: boolean,
98  ): boolean {
99    if (propertyInfo !== null) {
100     return propertyInfo.type === RESERVED;
101   }
102   if (isCustomComponentTag) {
103     return false;
104   }
105   if (
106     name.length > 2 &&
107     (name[0] === 'o' || name[0] === 'O') &&
108     (name[1] === 'n' || name[1] === 'N')
109   ) {
110     return true;
111   }
112   return false;
113 }
```

機會在這

react / packages / react-dom / src / shared / **isCustomComponent.js**

Code | Blame | 33 lines (31 loc) · 967 Bytes

```
 5      * LICENSE file in the root directory of this source tree.
 6      *
 7      * @flow
 8      */
 9
10     function isCustomComponent(tagName: string, props: Object) {
11       if (tagName.indexOf('-') === -1) {
12         return typeof props.is === 'string';
13       }
```

react / packages / react-dom / src / shared / **isCustomComponent.js**

Code | Blame | 33 lines (31 loc) · 967 Bytes

```
 5      * LICENSE file in the root directory of this source tree.
 6      *
 7      * @flow
 8      */
 9
10     function isCustomComponent(tagName: string, props: Object) {
11       if (tagName.indexOf('-') === -1) {
12         return typeof props.is === 'string';
13       }
```

Bingo!

順帶一提，React 19 開始把這個好玩的 feature 改掉了...

Credit to zwade

防禦心法第四條

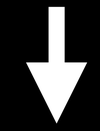
```
function isUnsafeString(str) {
    const regex = /.*[<].*[>=].*/s;
    return regex.test(str);
}
```

```
function isUnsafeString(str) {
    const regex = /.*[<].*[>=].*/s;
    return regex.test(str);
}

isUnsafeString('<svg>') // true
isUnsafeString('<svg onload=alert()') // true
isUnsafeString('<svg onload=alert()>') // true
isUnsafeString('hello <3') // false
```

```
function isUnsafeString(str) {
    const regex = /.*[<].*[>=].*/s;
    return regex.test(str);
}

console.time('test');
isUnsafeString('<svg>' + '<'.repeat(50000))
console.timeEnd('test');
// test: 1.218s
```

.* [<] .* [>=] .*

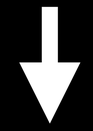
<svg><<<<<<<<<



.* [<] .* [>=] .*

<svg><<<<<<<<<

1. 配對所有 .*



```
.*  [<]  .*  [>=]  .*
```

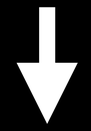
```
<svg><<<<<<<<<
```

2. 試圖配對 `[<]` 發現找不到

.* [<] .* [>=] .*

<svg><<<<<<<<<

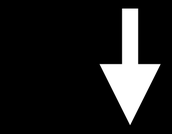
3. 回退一個字



.*　[<]　.*　[>=]　.*

<svg><<<<<<<<<

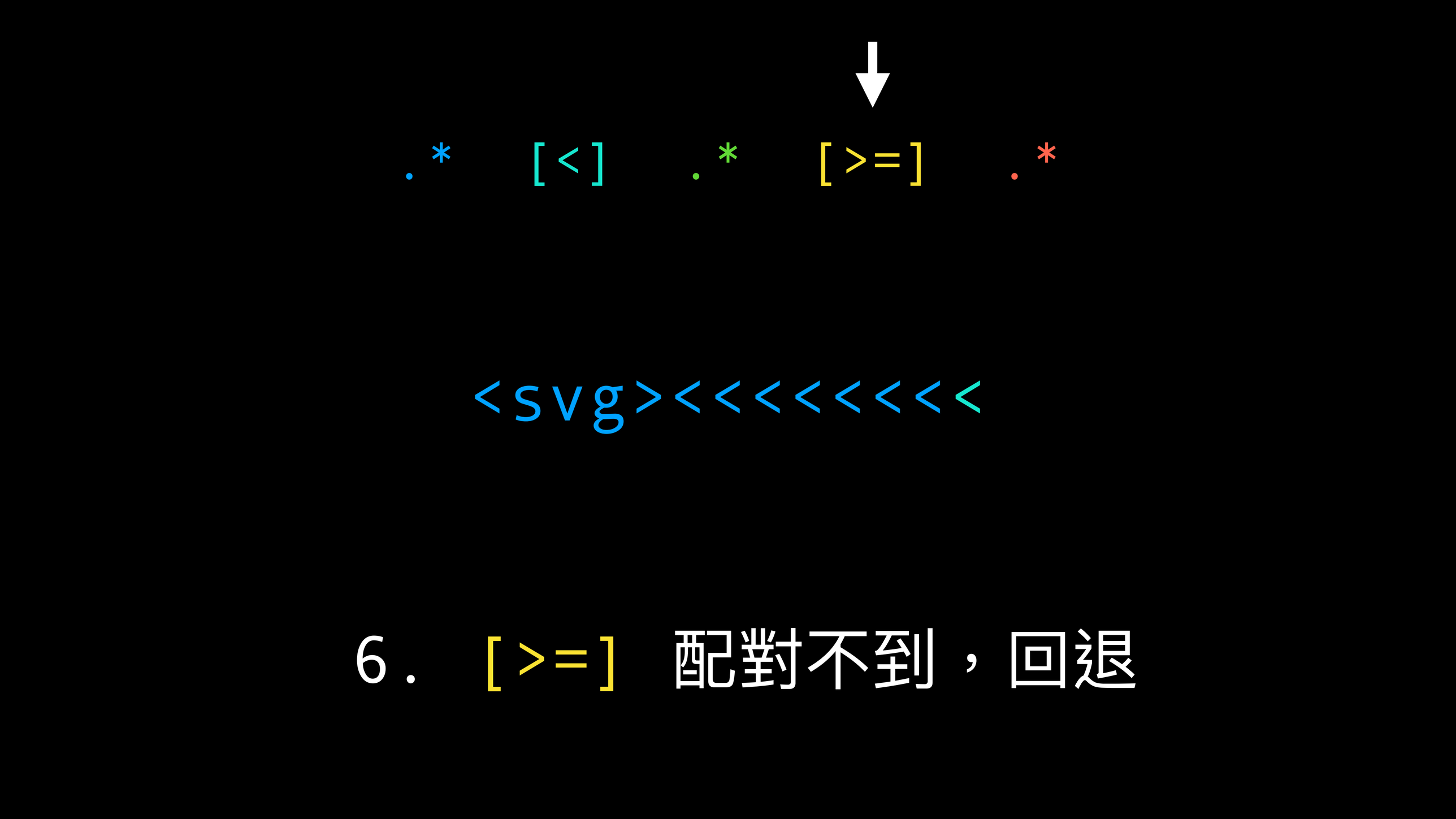
4. 現在配對到 [<] 了

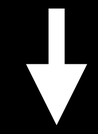


.*　[<]　.*　[>=]　.*

<svg><<<<<<<<<

5. .* 可以配對到空字元，下一步

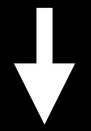

.*
[<]
.*
[>=]
.*
<svg><<<<<<<<<
6. [>=] 配對不到，回退



.*　[<]　.*　[>=]　.*

<svg><<<<<<<<<

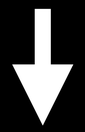
7. 回退完變這樣，留最後兩個



```
.*  [<]  .*  [>=]  .*
```

```
<svg><<<<<<<<<
```

8. [<] 配對一個字

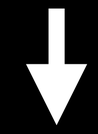


`.*` `[<]` `.*` `[>=]` `.*`

`<svg><<<<<<<<<`

9. `.*` 配對剩餘全部

10. 又配對不到，回退



.* [<] .* [>=] .*

<svg><<<<<<<<<

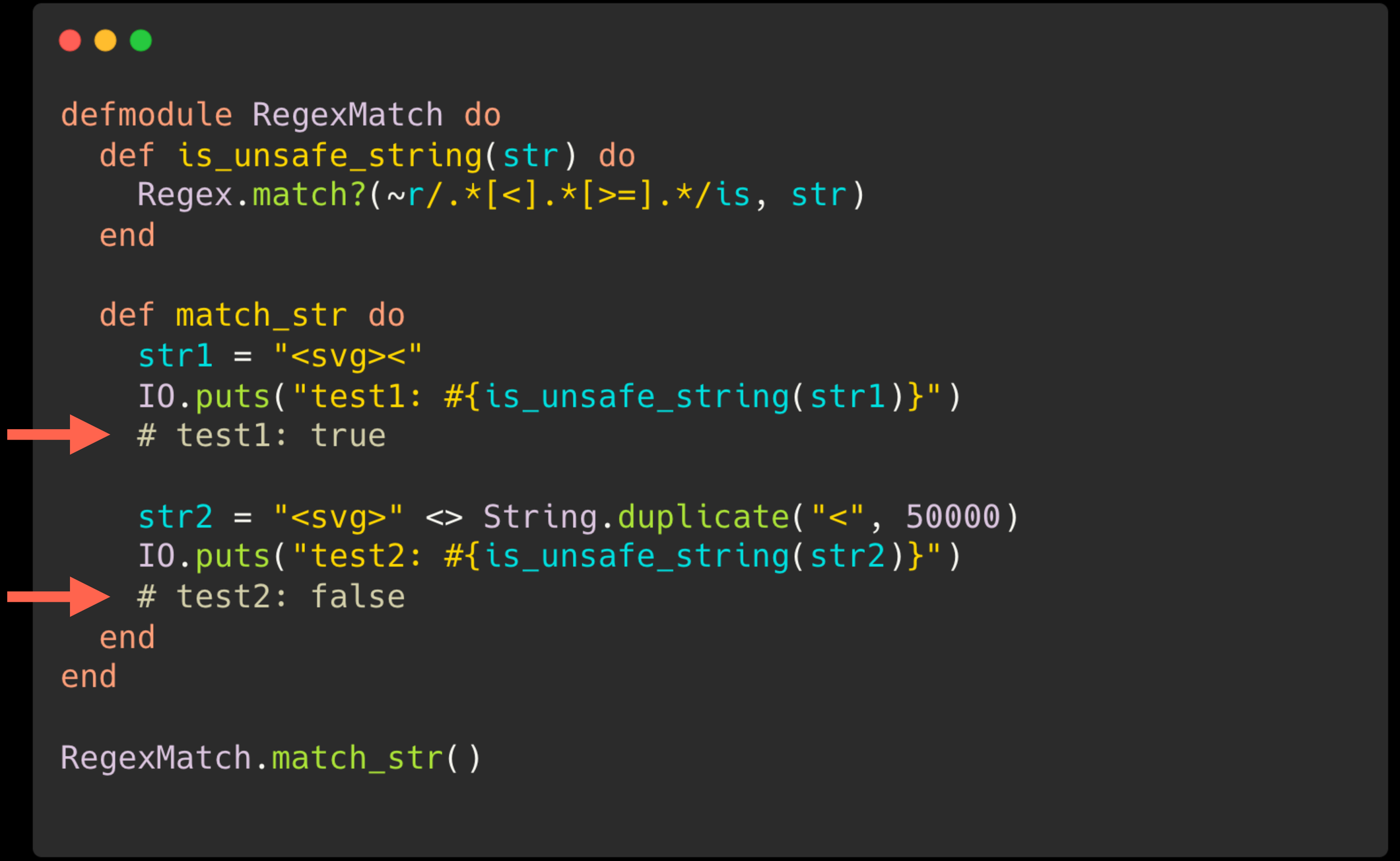
11. 留最後三個

不同程式語言的配對引擎不同

```php
<?php

  function isUnsafeString($str) {
    $regex = '/.*[<].*[>=].*/s';
    return preg_match($regex, $str);
  }

  $start = microtime(true);
  isUnsafeString('<svg>' . str_repeat('<', 50000));
  $end = microtime(true);

  $time = $end - $start;
  echo number_format($time, 3) . "s\n";
  // 0.001s
?>
```

```elixir
defmodule TimeTest do
  def is_unsafe_string(str) do
    Regex.match?(~r/.*[<].*[>=].*/is, str)
  end

  def run_test do
    str = "<svg>" <> String.duplicate("<", 50_000)

    {time_microseconds, _result} =
      :timer.tc(fn -> is_unsafe_string(str) end)

    IO.puts("test: #{time_microseconds / 1_000_000}s")
    # 0.098338s
  end
end

TimeTest.run_test()
```


我寫的 regex 出事了
誰叫你用JS，我都寫 Elixir
不會被 DoS 就沒事吧...對吧..


defmodule RegexMatch do
def is_unsafe_string(str) do
Regex.match?(~r/.*[<].*[>=].*/is, str)
end
def match_str do
str1 = "<svg><"
IO.puts("test1: #{is_unsafe_string(str1)}")
# test1: true
str2 = "<svg>" <> String.duplicate("<", 50000)
IO.puts("test2: #{is_unsafe_string(str2)}")
# test2: false
end
end
RegexMatch.match_str()

```php
<?php

  function isUnsafeString($str) {
    $regex = '/.*[<].*[>=].*/s';
    return preg_match($regex, $str);
  }

  $str1 = "<svg><";
  $str2 = "<svg>" . str_repeat('<', 50000);

  var_dump(isUnsafeString($str1));
  // int(1)

  var_dump(isUnsafeString($str2));
  // bool(false)
?>
```

Return Values

preg_match() returns 1 if the **`pattern`** matches given **`subject`**, 0 if it does not, or **`false`** on failure.

Warning This function may return Boolean **`false`**, but may also return a non-Boolean value which evaluates to **`false`**. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.

Credit to phith0n

PHP利用PCRE回溯次数限制绕过某些安全限制

PHITHON | Nov 26, 2018, 12:33 AM | 阅读：101595 | #网络安全 | #正则表达式

这次Code-Breaking Puzzles中我出了一道看似很简单的题目pcrewaf，将其代码简化如下：

```php
<?php
function is_php($data){
    return preg_match('/<\?.*[(`;?>].*/is', $data);
}

if(!is_php($input)) {
    // fwrite($f, $input); ...
}
```

MyBB Admin Panel RCE CVE-2023-41362

2023-09-11 #mybb #rce #regex #CVE-2023-41362 #redos

This blog post explores a critical vulnerability in MyBB's admin panel, leading to authenticated Remote Code Execution (RCE). MyBB is a popular forum software with a template system that utilizes eval() to render templates.

We will discuss how this vulnerability in the admin panel's template handling can be exploited for RCE.

We can change these templates in the admin panel. When we submit changes to templates they get put through the `check_template()` function in `admin/functions.php`. This function employs regex patterns to scan templates for specific patterns that may indicate malicious code.

```
function check_template($template){
    // Check to see if our database password is in the template
    if(preg_match('#\$config\[(([\'|"]database[\'|"])|([^\'"].*?))\]\[(([\'|"](database
        return true;
    }
```

防禦心法第五條

謹慎使用 /regex/

複習

1. MySQL unicode => context 的不相等

複習

1. MySQL unicode => context 的不相等
2. JS replace 時請小心謹慎

複習

1. MySQL unicode => context 的不相等
2. JS replace 時請小心謹慎
3. Clean 不是真的 clean，join 不是真的 join

複習

1. MySQL unicode => context 的不相等
2. JS replace 時請小心謹慎
3. Clean 不是真的 clean，join 不是真的 join
4. React is => 不要全盤相信使用者輸入

複習

1. MySQL unicode => context 的不相等
2. JS replace 時請小心謹慎
3. Clean 不是真的 clean，join 不是真的 join
4. React is => 不要全盤相信使用者輸入
5. 小心使用 regex

佛系分享各種東西，剛好偶爾講到資安

「Huli 隨意聊」