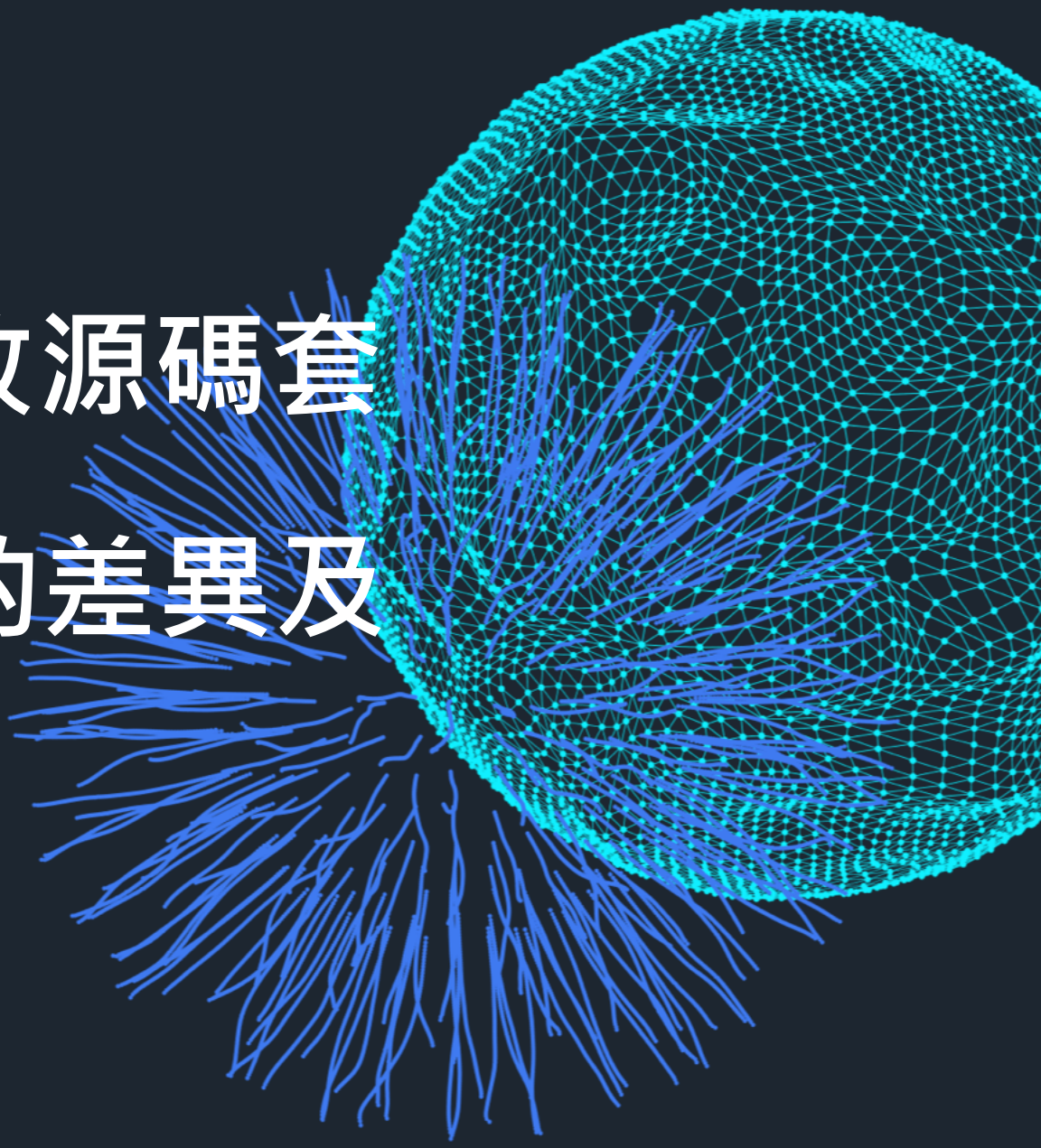
MICRO FOCUS Realize 2022

用白箱?用黑箱?還是開放源碼套件掃描?實例讓你了解安全測試的差異及應用



李柏厚 Bill
資深技術顧問
2022/06/24

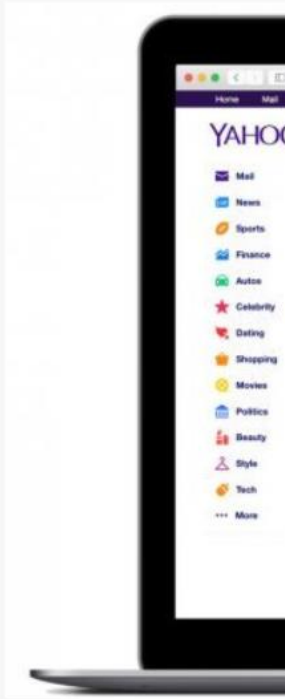
到底應用程式的問題在那裏?

新聞

Yahoo M

被發現的重大漏洞為Ya
件，被害者沒有點選連

文/ 陳曉莉 | 2016-12-12



示意圖，與新聞事件無關

當發現資
Equifax

2017 年 09 月 18
安, 資料外洩/個資外

美國三大信貸機構
消費者受到波及，
重大影響。

金融界對於網路安
的體認。

他們的資安團隊不
劃，所以才能在不

1.趨勢科技公布2022資安預測報告：供應鏈成駭客攻擊新場域

趨勢科技在12月16日發表「2022年資安年度預測報告」，提出三大重點觀察，包含：駭客藉四重勒索擴大獲利、供應鏈成為駭客攻擊新場域、個資外洩助長詐騙風潮，提醒有效的資訊安全防護需預作準備，方可控制風險。（趨勢科技: https://reurl.cc/NZVRkx）

2.Cloudflare正式推出Page Shield，可防禦惡意腳本攻擊提升網頁安全

Cloudflare推出Page Shield正式版本，供網路管理員能夠掌握網頁應用程式正在執行的腳本，並且在腳本被入侵或是執行洩露用戶資料等惡意行為時收到通知，用戶只需要在Cloudflare防火牆頁籤中，透過滑鼠點擊啟用Page Shield，不需要額外繁瑣的配置。平均而言，任何網頁應用程式都會從八個第三方主機載入腳本，這些腳本可能來自Google等大型企業，也有可能是來自小公司，所提供的隨插即用功能強化模組，像是聊天系統、日期選擇器或是結帳平台等，而這些第三方都可能成為供應鏈攻擊的目標。由於複雜的腳本來源，使得攻擊面變得很大難以監控，Cloudflare還指出，每月有50%的應用程式，會從新的第三方主機載入腳本，這使得攻擊面更加複雜。（iThome：https://pse.is/3x4h4p）

3.瞄準 5G、IT、OT 資安監控方案，資策會成立「資安鑄造」公司

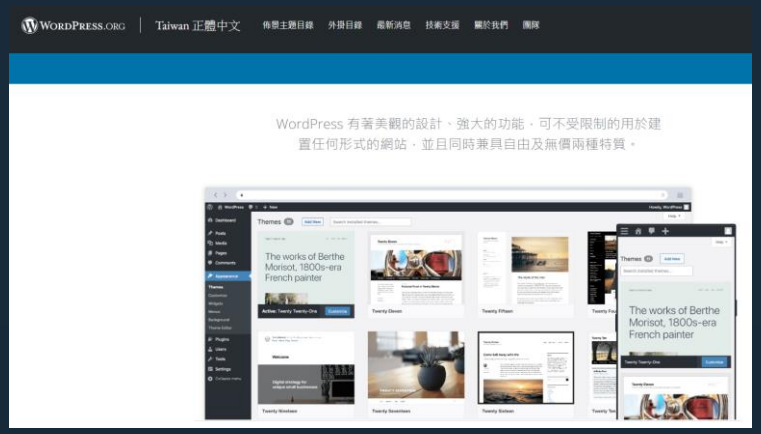
資策會指出，隨著第五代行動通訊技術（5G）的快速發展，將加速台灣產業數位轉型，卻產生工控設備（OT）數位化等資訊安全問題，例如駭客挖掘物聯網漏洞，並利用勒索軟體進行攻擊。了回應產業對資安的強烈需求，資策會於12月8日宣布成立「資安鑄造公司」，業務包含鑄造跨資訊（IT）、工控及 5G 的供應鏈

應用程式開發會有的元件

作業系統

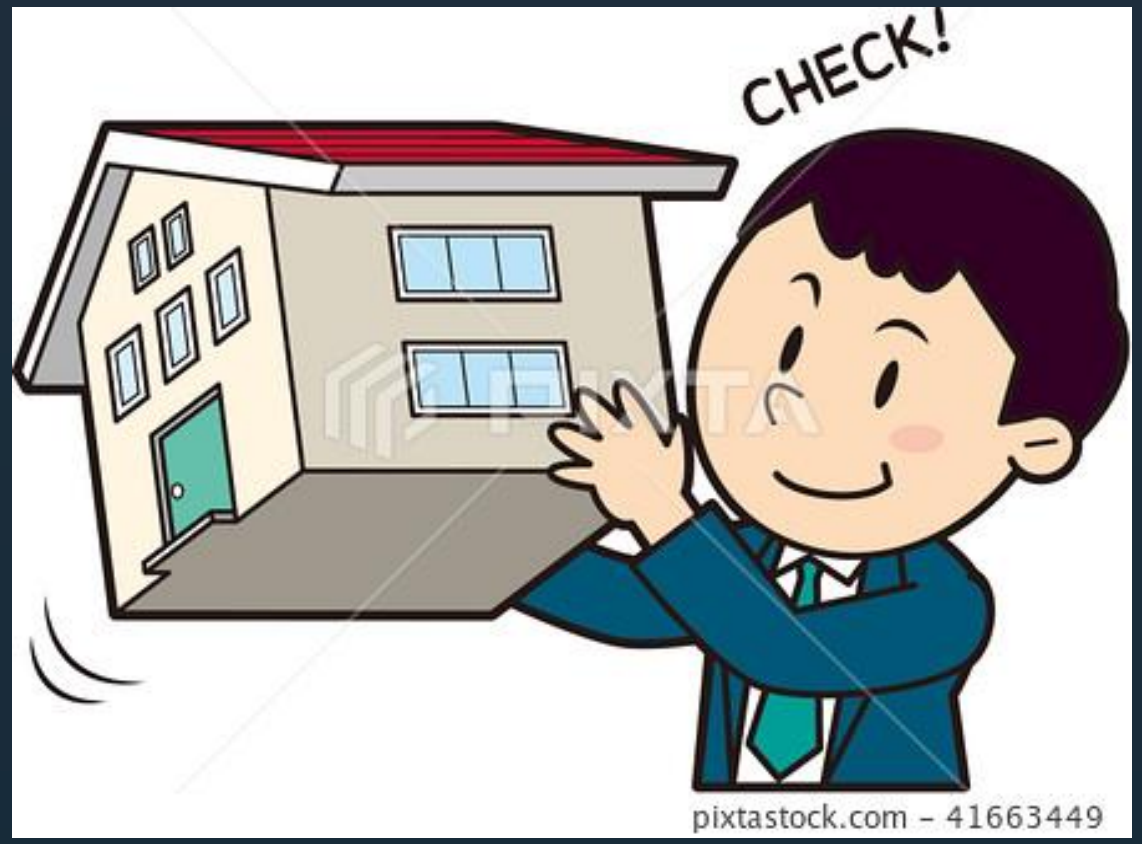
自有程式開發

第三方開放源碼套件



MICRO FOCUS

軟体安全測試就如房屋檢測

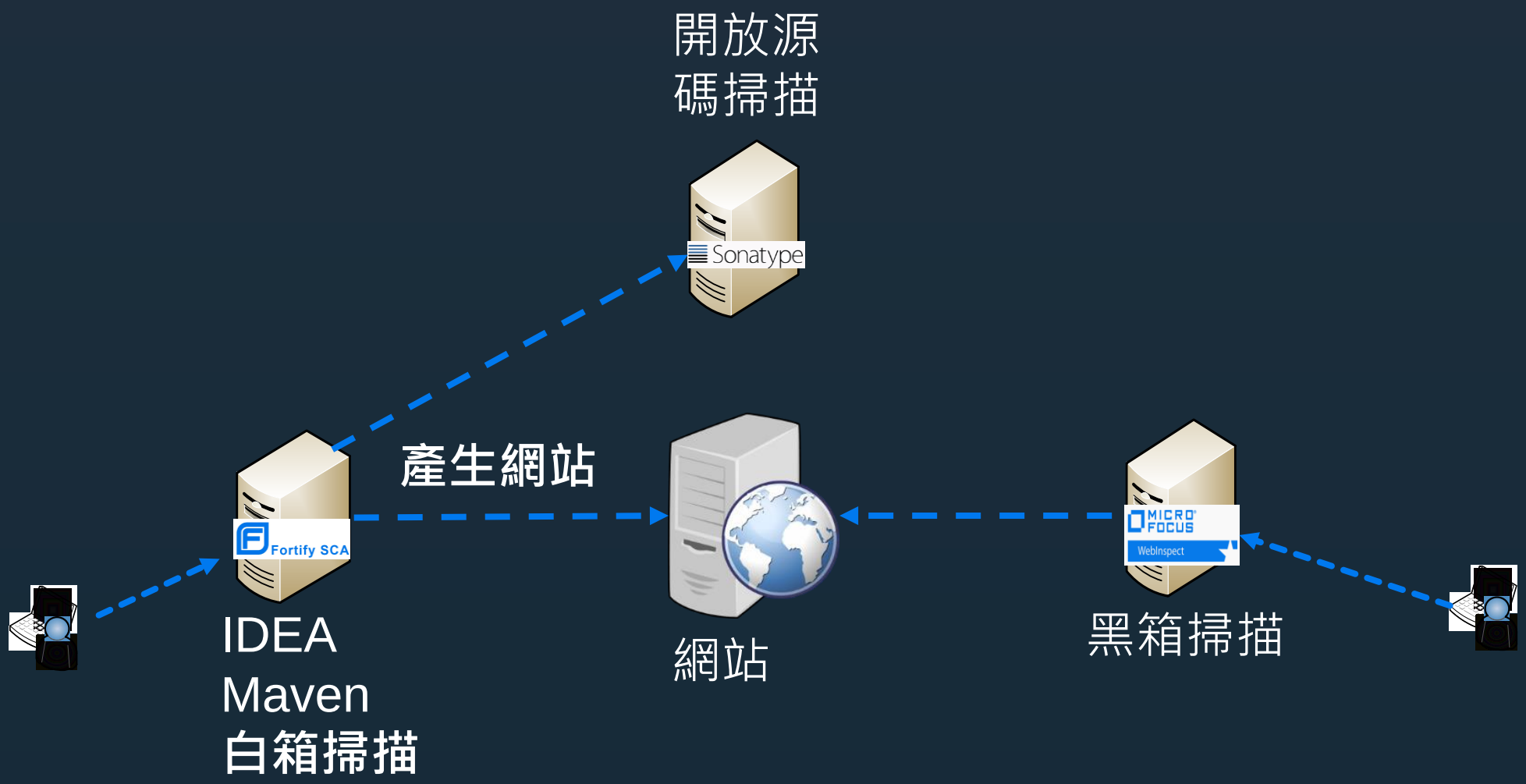


- 白箱-源碼掃描-由內部檢查
- 黑箱-網站掃描-由外部檢查
- 開放源碼掃描-構建材料檢查

10-20%
Custom
Code

80-90%
Open Source
Libraries

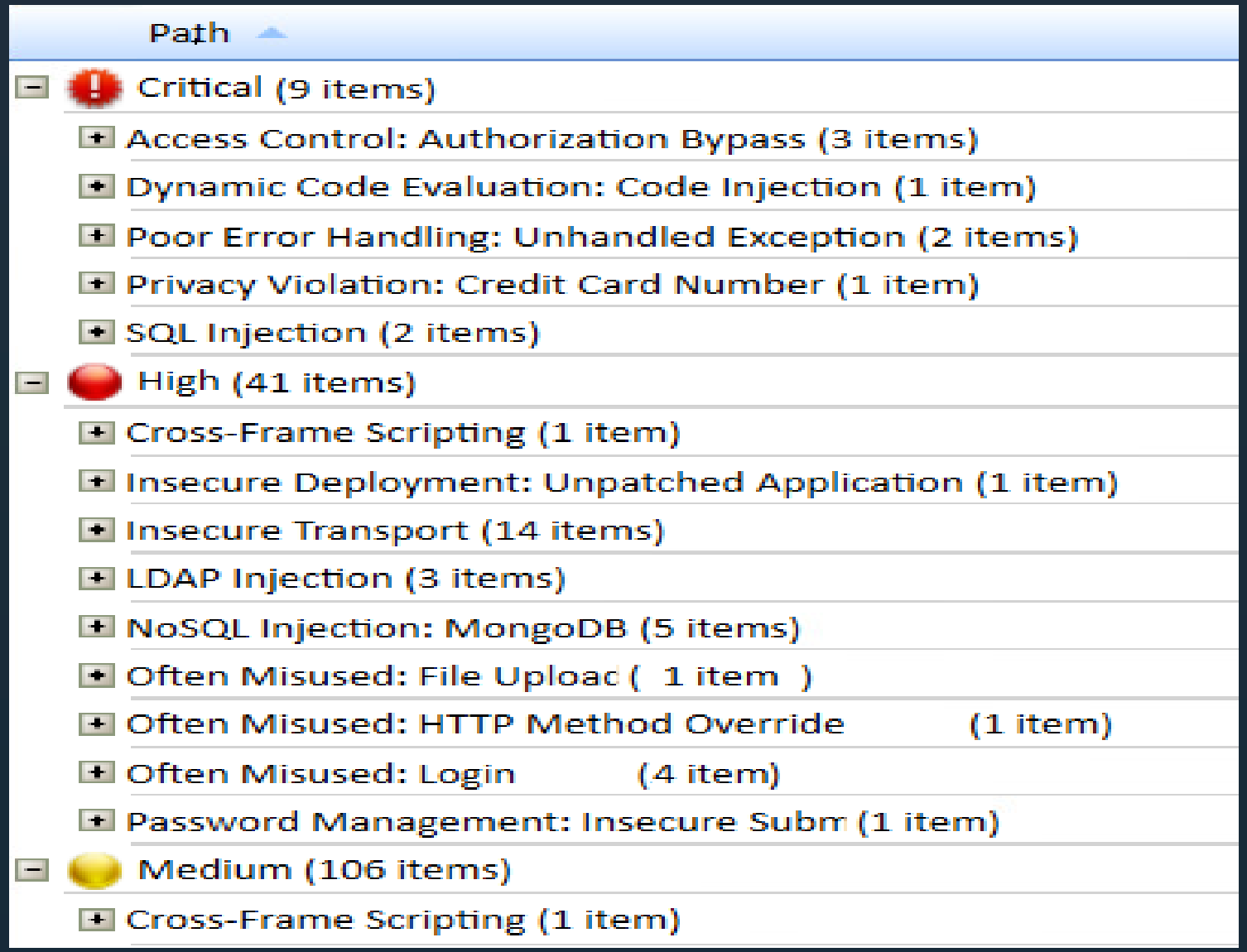
Webgoat-測試架構圖



網站掃描-黑箱測試

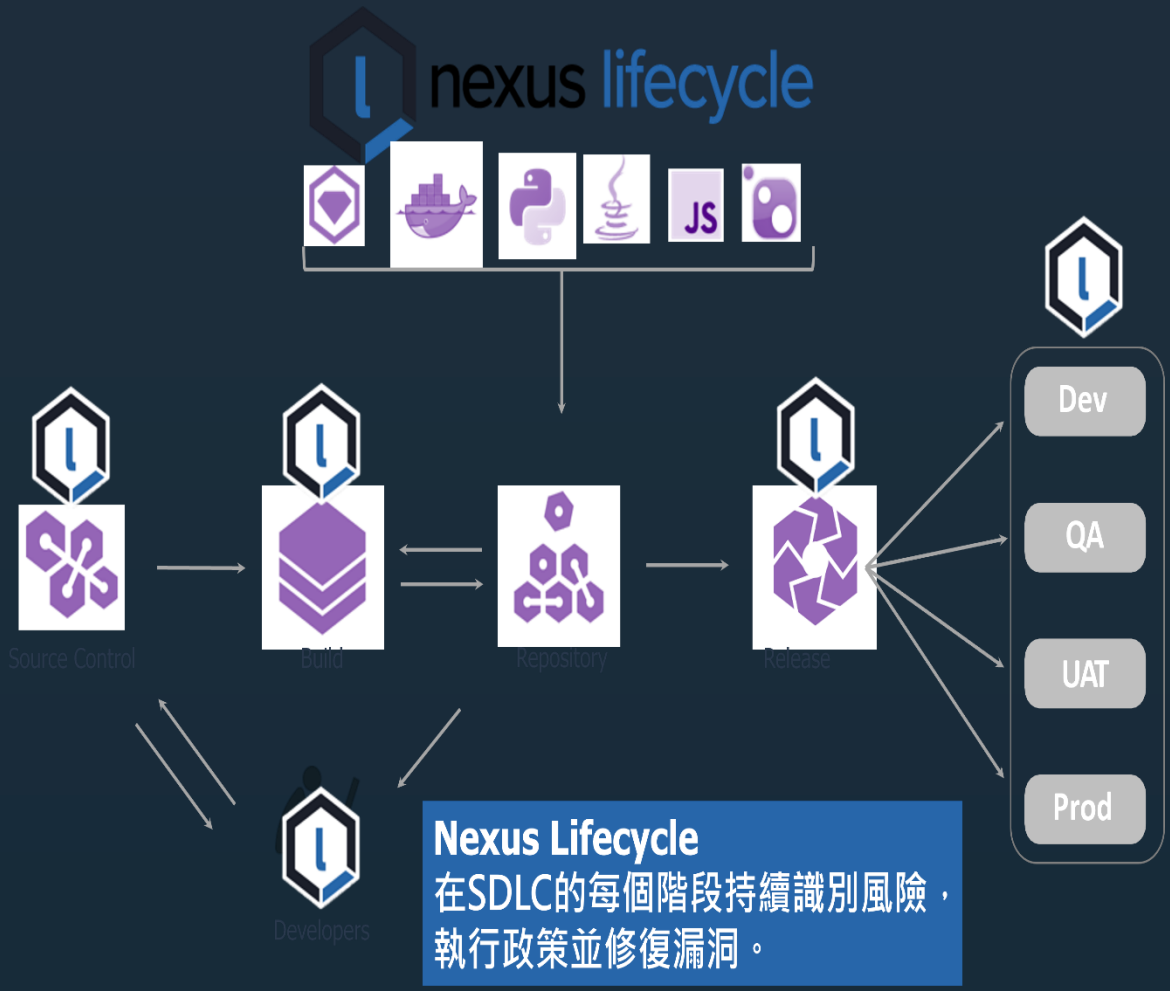
- 透過網站的互動來對網站的環境進行評估(SSL、Session)
- 對網站的漏洞(Tomcat)進行測試
- 對網站互動欄位進行入侵

黑箱掃描結果


Path
Critical (9 items)
Access Control: Authorization Bypass (3 items)
Dynamic Code Evaluation: Code Injection (1 item)
Poor Error Handling: Unhandled Exception (2 items)
Privacy Violation: Credit Card Number (1 item)
SQL Injection (2 items)
High (41 items)
Cross-Frame Scripting (1 item)
Insecure Deployment: Unpatched Application (1 item)
Insecure Transport (14 items)
LDAP Injection (3 items)
NoSQL Injection: MongoDB (5 items)
Often Misused: File Uploac (1 item)
Often Misused: HTTP Method Override (1 item)
Often Misused: Login (4 item)
Password Management: Insecure Subm (1 item)
Medium (106 items)
Cross-Frame Scripting (1 item)

開放源碼掃描-針對客戶所使用的開放源碼套件

- 主要是用hash的方式來識別惡意套件
- 偵測第三方套件的安全性、品質及授權等三方面。
- 可透過套件管理軟体如Nuget、Maven等來下載。



開放源碼掃描結果

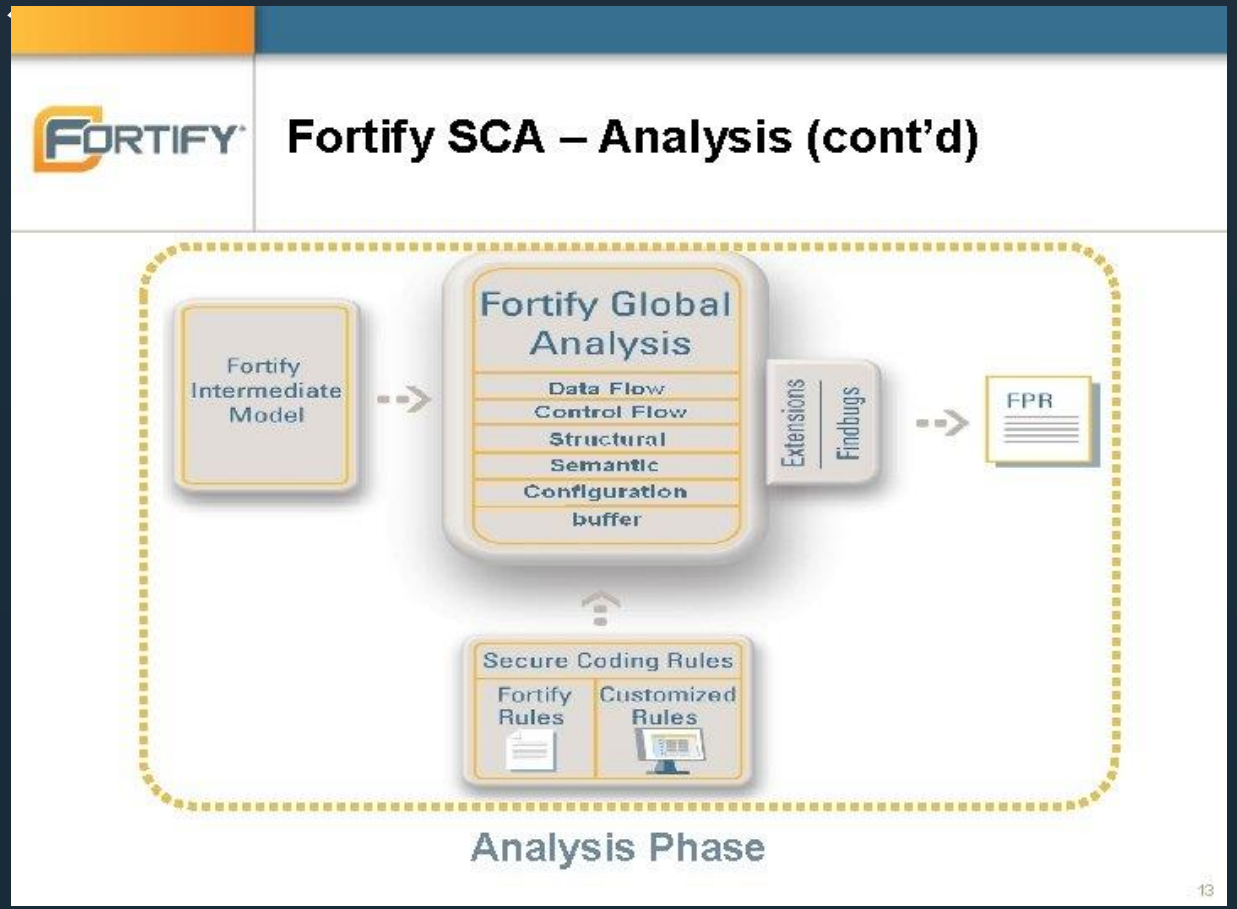
35 53 40 **128 VIOLATIONS**
Affecting 70 components

0 GRANDFATHERED
violations

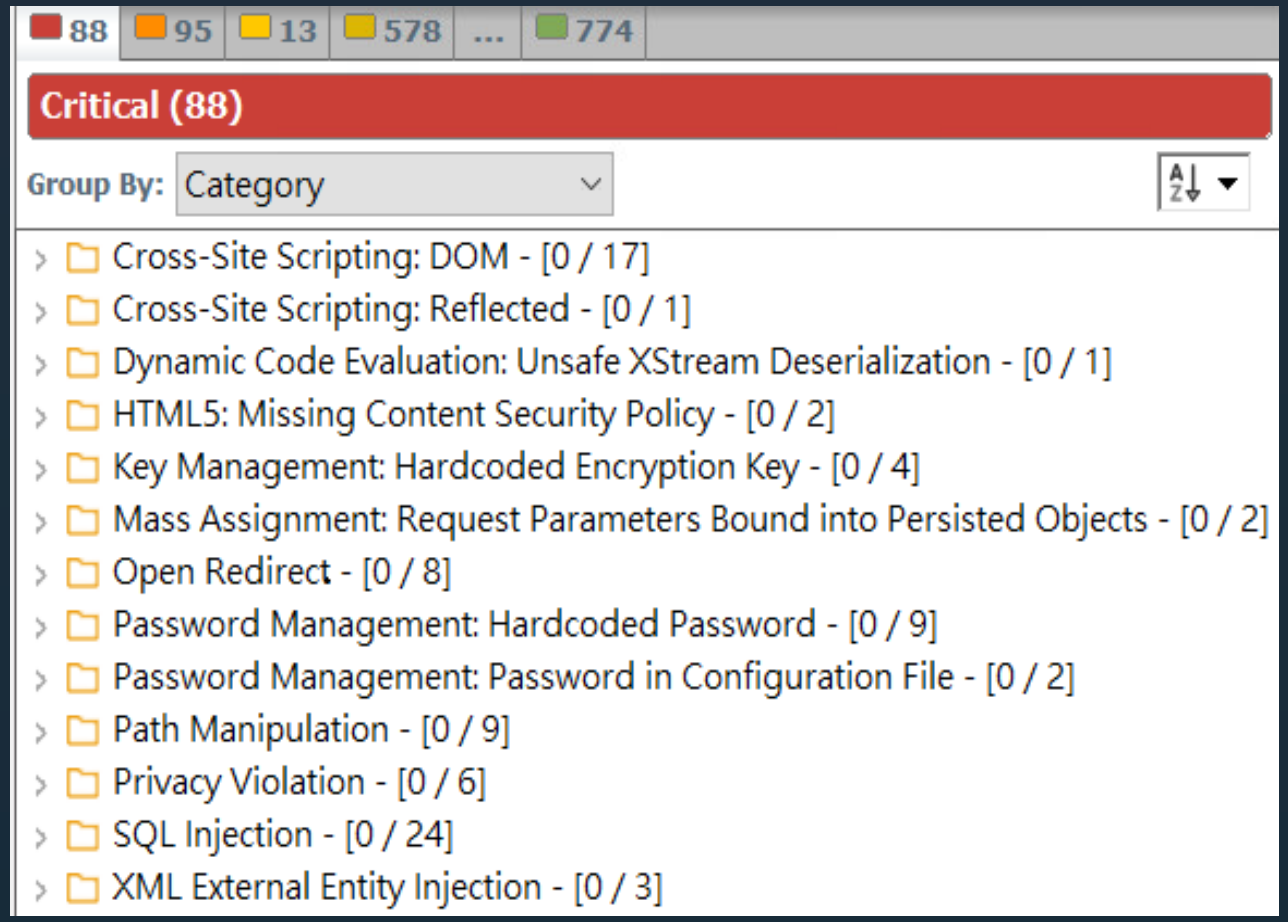
THREAT	POLICY NAME	POLICY TYPE	COMPONENT
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	org.postgresql : postgresql : 42.2.23
10	Security-Critical	Security	org.postgresql : postgresql : 42.2.23
10	Security-Critical	Security	org.springframework : spring-beans : 5.3.9
10	Security-Critical	Security	org.springframework : spring-beans : 5.3.9
10	Security-Critical	Security	org.springframework : spring-web : 5.3.9
10	Security-Critical	Security	org.thymeleaf : thymeleaf-spring5 : 3.0.12.RELEASE
9	Security-High	Security	ch.qos.logback : logback-core : 1.2.5
9	Security-High	Security	com.fasterxml.jackson.core : jackson-databind : 2.12.4
9	Security-High	Security	com.fasterxml.jackson.core : jackson-databind : 2.12.4
9	Security-High	Security	com.github.jnr : jnr-posix : 3.1.4
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	com.thoughtworks.xstream : xstream : 1.4.5
10	Security-Critical	Security	org.postgresql : postgresql : 42.2.23
10	Security-Critical	Security	org.postgresql : postgresql : 42.2.23
10	Security-Critical	Security	org.springframework : spring-beans : 5.3.9
10	Security-Critical	Security	org.springframework : spring-beans : 5.3.9
10	Security-Critical	Security	org.springframework : spring-web : 5.3.9
10	Security-Critical	Security	org.thymeleaf : thymeleaf-spring5 : 3.0.12.RELEASE
9	Security-High	Security	ch.qos.logback : logback-core : 1.2.5
9	Security-High	Security	com.fasterxml.jackson.core : jackson-databind : 2.12.4
9	Security-High	Security	com.fasterxml.jackson.core : jackson-databind : 2.12.4
9	Security-High	Security	com.github.jnr : jnr-posix : 3.1.4
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	com.thoughtworks.xstream : xstream : 1.4.5
9	Security-High	Security	io.undertow : undertow-core : 2.2.10.Final

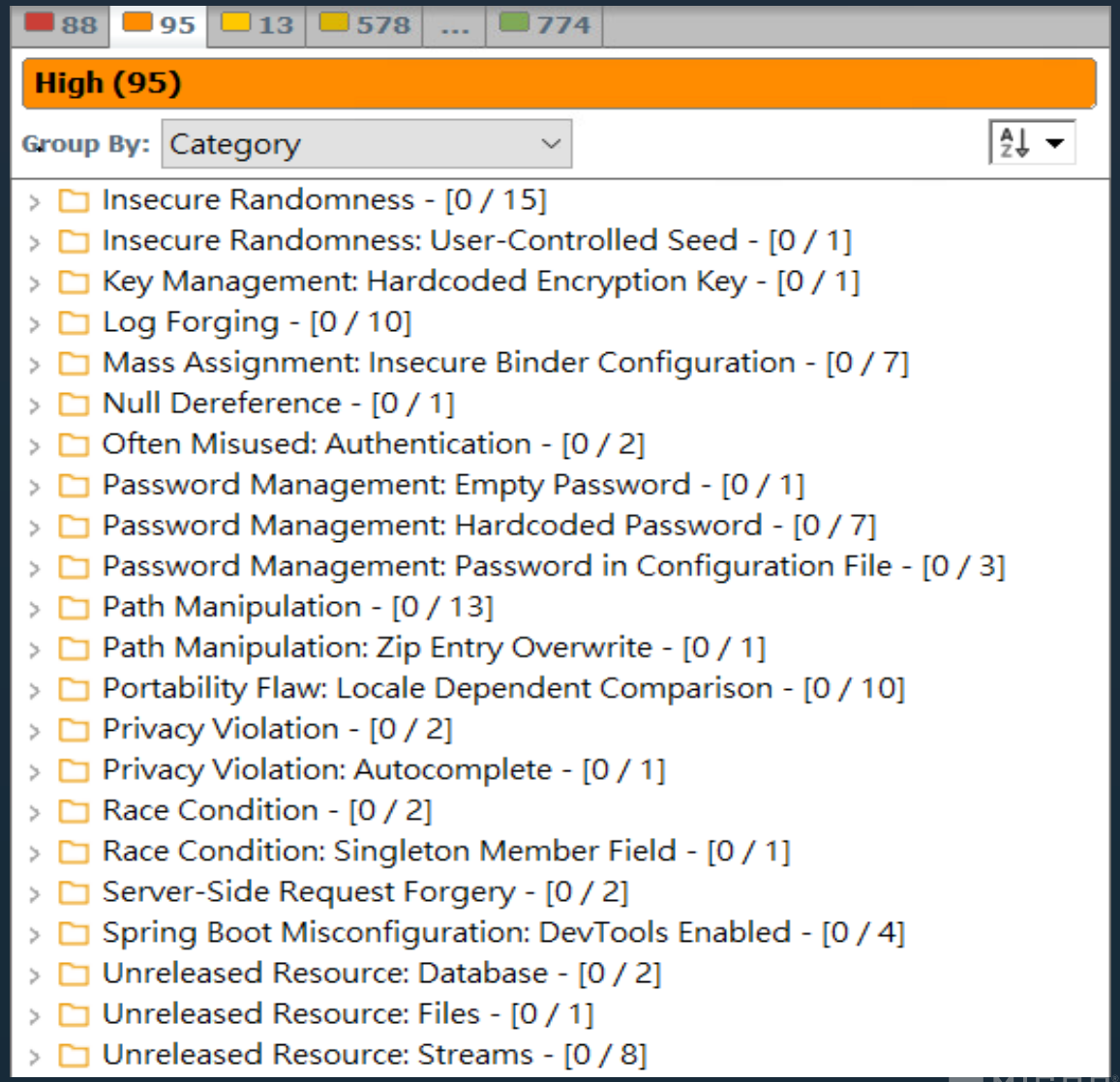
程式碼掃描-主要是針對客戶開發程式碼

- 主要是分析程式資料流、控制語法、結構及設定等面向的安全性問題。
- 以程式碼為主，若是編譯後的話，無法進行分析

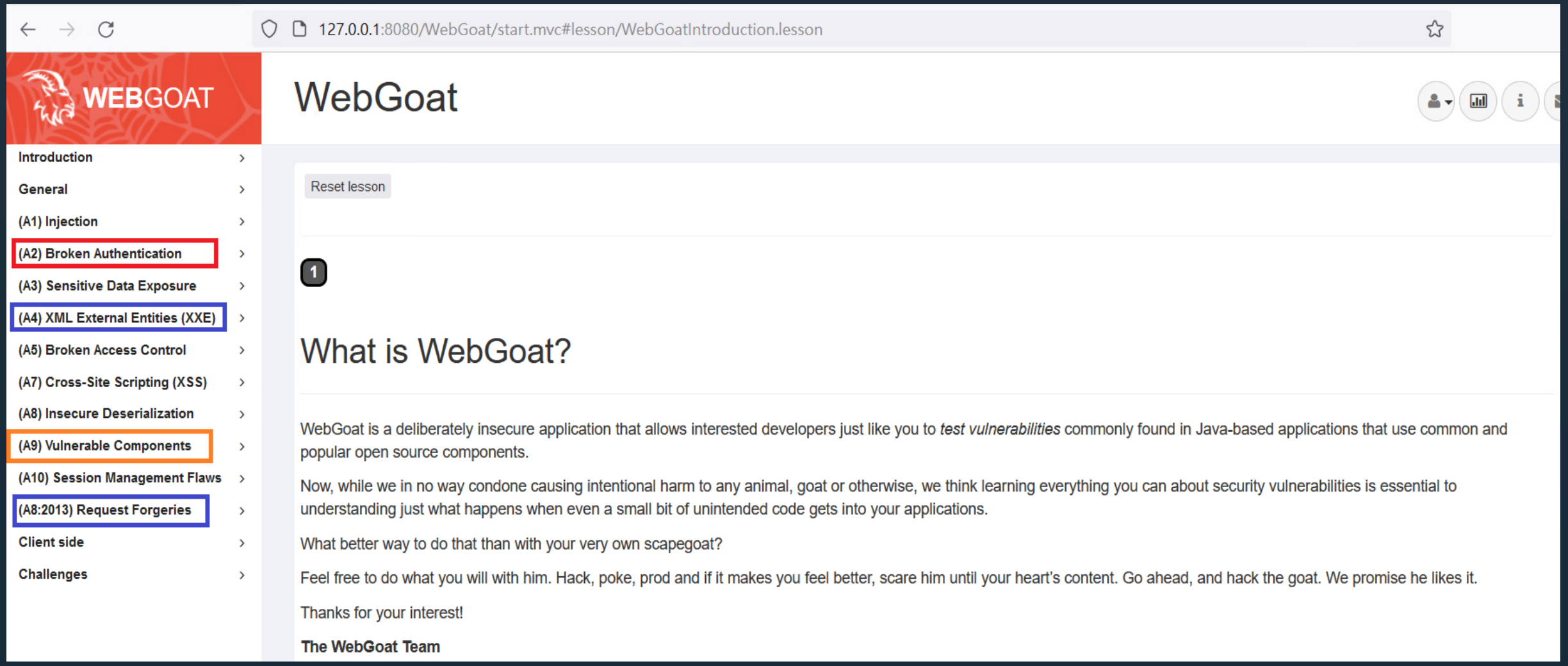


白箱掃描結果


88
95
13
578
...
774
Critical (88)
Group By:
Category
Cross-Site Scripting: DOM - [0 / 17]
Cross-Site Scripting: Reflected - [0 / 1]
Dynamic Code Evaluation: Unsafe XStream Deserialization - [0 / 1]
HTML5: Missing Content Security Policy - [0 / 2]
Key Management: Hardcoded Encryption Key - [0 / 4]
Mass Assignment: Request Parameters Bound into Persisted Objects - [0 / 2]
Open Redirect - [0 / 8]
Password Management: Hardcoded Password - [0 / 9]
Password Management: Password in Configuration File - [0 / 2]
Path Manipulation - [0 / 9]
Privacy Violation - [0 / 6]
SQL Injection - [0 / 24]
XML External Entity Injection - [0 / 3]


88
95
13
578
...
774
High (95)
Group By:
Category
Insecure Randomness - [0 / 15]
Insecure Randomness: User-Controlled Seed - [0 / 1]
Key Management: Hardcoded Encryption Key - [0 / 1]
Log Forging - [0 / 10]
Mass Assignment: Insecure Binder Configuration - [0 / 7]
Null Dereference - [0 / 1]
Often Misused: Authentication - [0 / 2]
Password Management: Empty Password - [0 / 1]
Password Management: Hardcoded Password - [0 / 7]
Password Management: Password in Configuration File - [0 / 3]
Path Manipulation - [0 / 13]
Path Manipulation: Zip Entry Overwrite - [0 / 1]
Portability Flaw: Locale Dependent Comparison - [0 / 10]
Privacy Violation - [0 / 2]
Privacy Violation: Autocomplete - [0 / 1]
Race Condition - [0 / 2]
Race Condition: Singleton Member Field - [0 / 1]
Server-Side Request Forgery - [0 / 2]
Spring Boot Misconfiguration: DevTools Enabled - [0 / 4]
Unreleased Resource: Database - [0 / 2]
Unreleased Resource: Files - [0 / 1]
Unreleased Resource: Streams - [0 / 8]

WebGoat問題點


127.0.0.1:8080/WebGoat/start.mvc#lesson/WebGoatIntroduction.lesson
WEBGOAT
WebGoat
Introduction
General
(A1) Injection
(A2) Broken Authentication
(A3) Sensitive Data Exposure
(A4) XML External Entities (XXE)
(A5) Broken Access Control
(A7) Cross-Site Scripting (XSS)
(A8) Insecure Deserialization
(A9) Vulnerable Components
(A10) Session Management Flaws
(A8:2013) Request Forgeries
Client side
Challenges
Reset lesson
1
What is WebGoat?
WebGoat is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-based applications that use common and popular open source components.
Now, while we in no way condone causing intentional harm to any animal, goat or otherwise, we think learning everything you can about security vulnerabilities is essential to understanding just what happens when even a small bit of unintended code gets into your applications.
What better way to do that than with your very own scapegoat?
Feel free to do what you will with him. Hack, poke, prod and if it makes you feel better, scare him until your heart's content. Go ahead, and hack the goat. We promise he likes it.
Thanks for your interest!
The WebGoat Team

Spring4Shell問題-code injection

Request/Response Packet

```
GET /WebGoat/registration?class.module.classLoader.class.name.bytes%5B99999%5D=0 HTTP/1.1
Referer: http://172.16.222.102:8080/WebGoat/login
Accept: */*
Pragma: no-cache
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0) Gecko/20100101 Firefox/83.0
Host: 172.16.222.102:8080
Connection: Keep-Alive
X-WIPP: AscVersion=22.1.0.99
X-Scan-Memo: Category="Audit.Attack";SID="15F397BE9BED78A9E31A240058D19AD2";PSID="C320CCF5D695997FBCDE79A8128A700E";SessionType="AuditAttack";CrawlType="None";AttackType="QueryParamManipulation";
OriginatingEngineID="1ff8b322-3862-4116-9788-e65f742cba24";AttackSequence="0";AttackParamDesc="class.module.classLoader.class.name.bytes%255B99999%255D";AttackParamIndex="0";AttackParamSubIndex="0
CheckId="11708";Engine="Spring4+Shell";SmartMode="4";tht="30";
X-RequestManager-Memo: stid="171";stmi="0";sc="1";rid="828b3507";
X-Request-Memo: rid="85c06891";sc="1";thid="255";
Cookie: CustomCookie=WebInspect182881ZX62EC846092974AEBA6B93A5638BE1249Y4DC4;JSESSIONID=Ter3s_7M6N7oRVo2LM_7Qa_q4IGzLkGw8uAV5yK2
```

'class.module.classLoader.class.name.bytes[99999]=0

```
HTTP/1.1 500 Internal Server Error
Connection: keep-alive
Content-Type: application/json
Date: Wed, 02 Nov 2022 15:46:34 GMT
Content-Length: 12811

{
  "timestamp" : "2022-11-02T15:46:34.973+00:00",
  "status" : 500,
  "error" : "Internal Server Error",
  "trace" : "org.springframework.beans.InvalidPropertyException: Invalid property 'class.module.classLoader.class.name.bytes[99999]' of bean class [java.lang.Class]: Invalid array index in propert
path 'bytes[99999]'; nested exception is java.lang.ArrayIndexOutOfBoundsException\n\tat org.springframework.beans.AbstractNestablePropertyAccessor.processKeyedProperty(
AbstractNestablePropertyAccessor.java:314)\n\tat org.springframework.beans.AbstractNestablePropertyAccessor.setPropertyValue(AbstractNestablePropertyAccessor.java:275)\n\tat org.springframework.
beans.AbstractNestablePropertyAccessor.setPropertyValue(AbstractNestablePropertyAccessor.java:266)\n\tat org.springframework.beans.AbstractPropertyAccessor.setPropertyValues(AbstractPropertyAccess
.java:104)\n\tat org.springframework.validation.DataBinder.applyPropertyValues(DataBinder.java:851)\n\tat org.springframework.validation.DataBinder.doBind(DataBinder.java:747)\n\tat org.
```

Summary: Dynamic Code Evaluation: Code Injection

Vulnerability ID: 11708
CWE ID: 94
Kingdom: Input Validation and Representation

WebInspect has detected a Spring4Shell remote code execution (RCE) vulnerability identified by CVE-2022-22965. This vulnerability affects a Spring Framework application that is running on JDK 9+ and uses data binding functionality to bind data stored within an HTTP request to certain objects used by the application. This vulnerability creates the risk of data leakage and remote code execution when special object classes are used. Pivotal Spring Framework versions affected by this vulnerability include 5.3.x prior to 5.3.18 and versions 5.2.x prior to 5.2.20 , and older unsupported versions.

Execution:

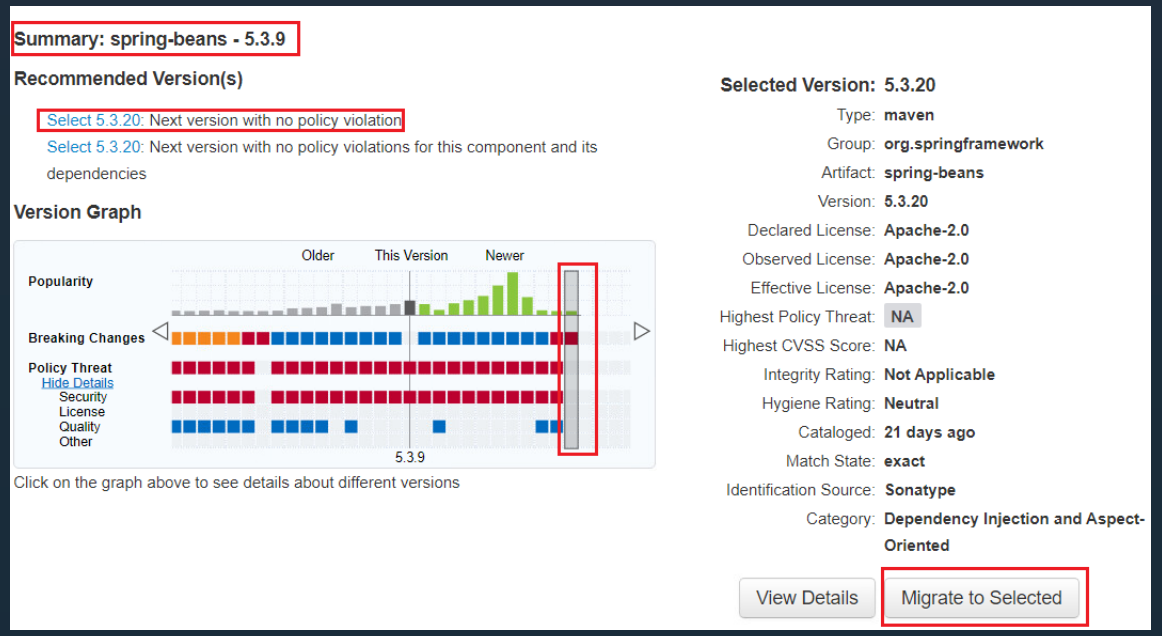
How to verify or exploit the issue.

To verify the vulnerability, repeat the vulnerable request, from the scan results, in the HTTP Editor provided in WebInspect Tools, or another proxy tool of your choice. The vulnerable target will return an error message.

MICRO FOCUS

Spring4Shell問題-code injection

開放源碼掃描



- 黑箱
  - 主要是透過發送一些特徵碼看主機的回應是不是期待的回應
  - 主要是偵測一種型態，而這些framework不屬於程式碼，所以白箱是無效
- 黑箱可以偵測屬於第三方套件的問題，但可以偵測的範圍小
- 第三方套件掃描為主要使用的工具較快、範圍較廣

系統錯誤訊息處理

URL	Post Parameters	Source	Crawl Link Type	Attack Type	Originating Engine
http://127.0.0.1:8080/WebGoat		Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/		Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/login		Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/login	username=billowen&password=%211qazxsw2	Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/welcome.mvc		Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/start.mvc		Start Macro	None	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/SqlInjectionMitigations/servers?column=id		AJAX Include	AJAX Include	Multipart Filename	null
http://127.0.0.1:8080/WebGoat/SqlInjectionMitigations/servers?column=%00		Probe	None	Probe	Parameter Injection Engine

Summary: Poor Error Handling: Unhandled Exception

Vulnerability ID: 742
CWE ID: 209
Kingdom: Errors

Critical database server error message vulnerabilities were identified in the web application, indicating that an unhandled exception was generated in your web application code. Unhandled exceptions are circumstances in which the application has received user input that it did not expect and does not know how to handle. When successfully exploited, an attacker can gain unauthorized access to the database by using the information recovered from seemingly innocuous error messages to pinpoint flaws in the web application and to discover additional avenues of attack. Recommendations include designing and adding consistent error-handling mechanisms that are capable of handling any user input to your web application, providing meaningful detail to end-users, and preventing error messages that might provide information useful to an attacker from being displayed.

Whitelabel Error Page

錯誤訊息

This application has no explicit mapping for /error, so you are seeing this as a fallback.

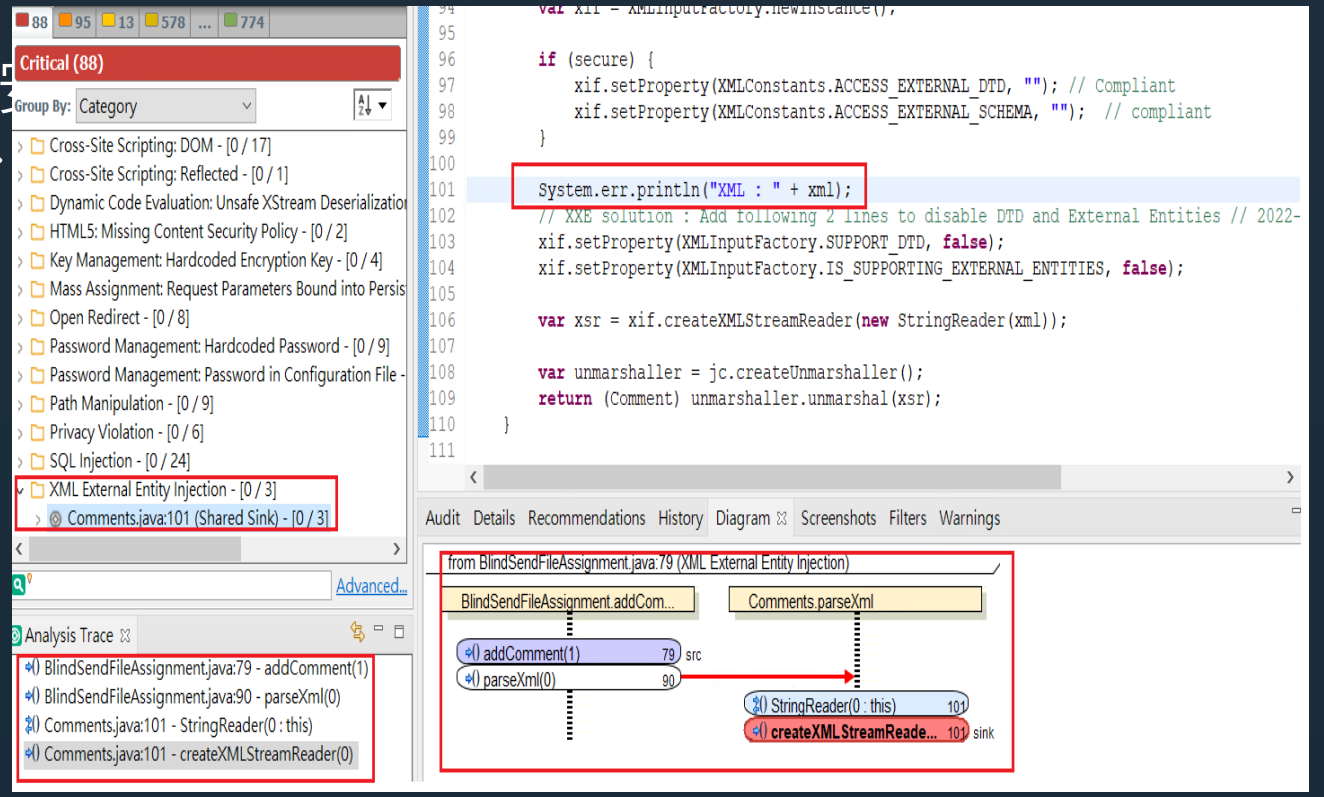
Thu Jun 02 09:09:23 CST 2022
There was an unexpected error (type=Internal Server Error, status=500).
Request processing failed; nested exception is java.sql.SQLSyntaxErrorException: unknown token: in statement [select id, hostname, ip, mac, status, description from se
by]
java.sql.SQLSyntaxErrorException: unknown token: in statement [select id, hostname, ip, mac, status, description from servers where status <> 'out of order' order by]
    at org.hsqldb.jdbc.JDBCUtil.sqlException(Unknown Source)
    at org.hsqldb.jdbc.JDBCUtil.sqlException(Unknown Source)
    at org.hsqldb.jdbc.JDBCPreparedStatement.<init>(Unknown Source)
    at org.hsqldb.jdbc.JDBCConnection.prepareStatement(Unknown Source)
    at jdk.internal.reflect.GeneratedMethodAccessor108.invoke(Unknown Source)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:564)
    at org.owasp.webgoat.lessons.LessonConnectionInvocationHandler.invoke(LessonConnectionInvocationHandler.java:31)
    at com.sun.proxy.$Proxy94.prepareStatement(Unknown Source)
    at org.owasp.webgoat.sql_injection.mitigation.Servers.sort(Servers.java:71)
    at jdk.internal.reflect.GeneratedMethodAccessor297.invoke(Unknown Source)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:564)
    at org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:197)
    at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:141)
    at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:106)
    at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:895)
    at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:808)
    at org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)
    at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1064)
    at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:963)

這類問題只有黑箱可以掃出
白箱則無法判定

XXE Injection問題說明

- XXE：XML External Entity 即外部實體，從安全形度理解成XML External Entity attack 外部實體注入攻擊
- 結構
  - XML聲明
  - DTD(Document type Definition)

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE test [
<!ENTITY file SYSTEM "file:///etc/passwd">
<!ENTITY copyright SYSTEM "http://www.w3school.com.cn/dtd/entities.dtd">
]>
<author>&file;©right;</author>
```



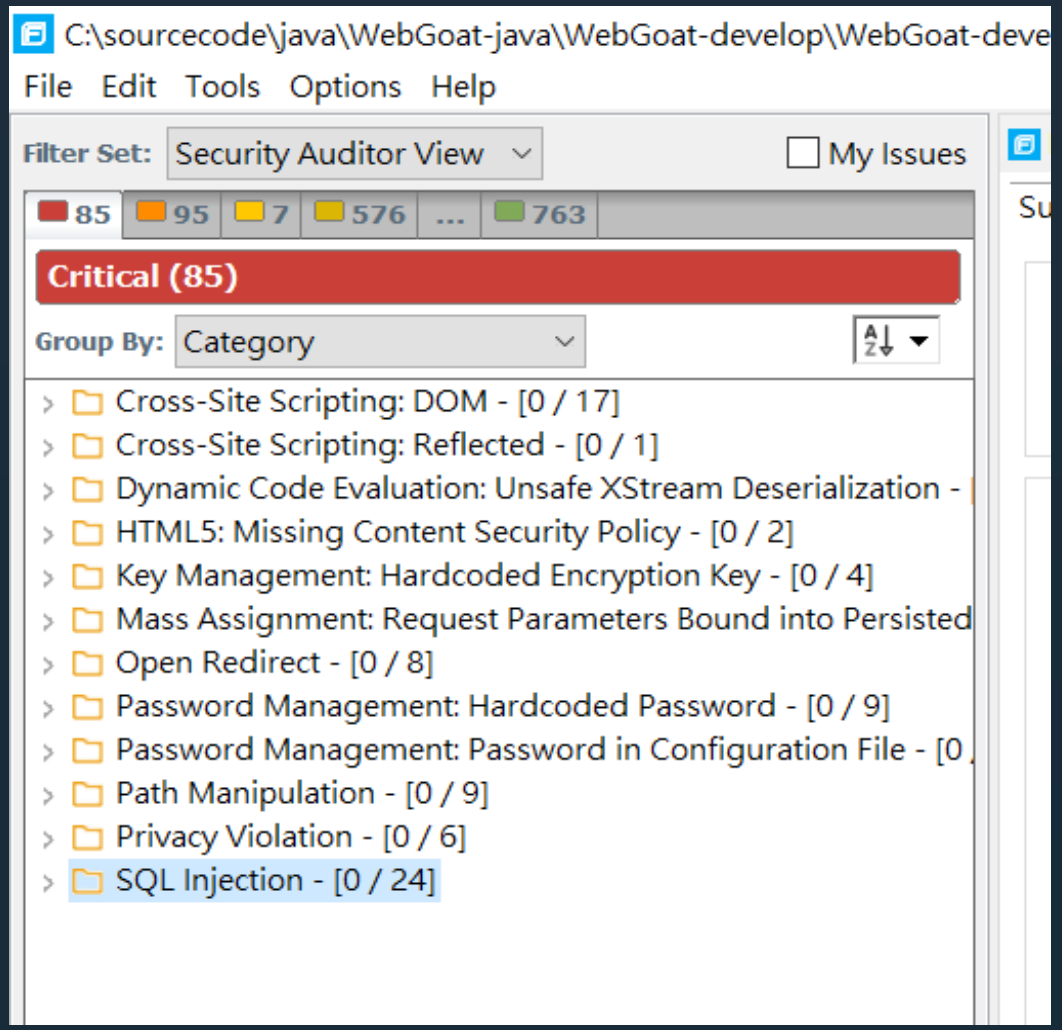
Demo

- 洩露本地文件
- CSRF/SSRF攻擊
- 命令執行
- 癱瘓攻擊(DoS)

XME問題修正

- Disable DTD and External Entities

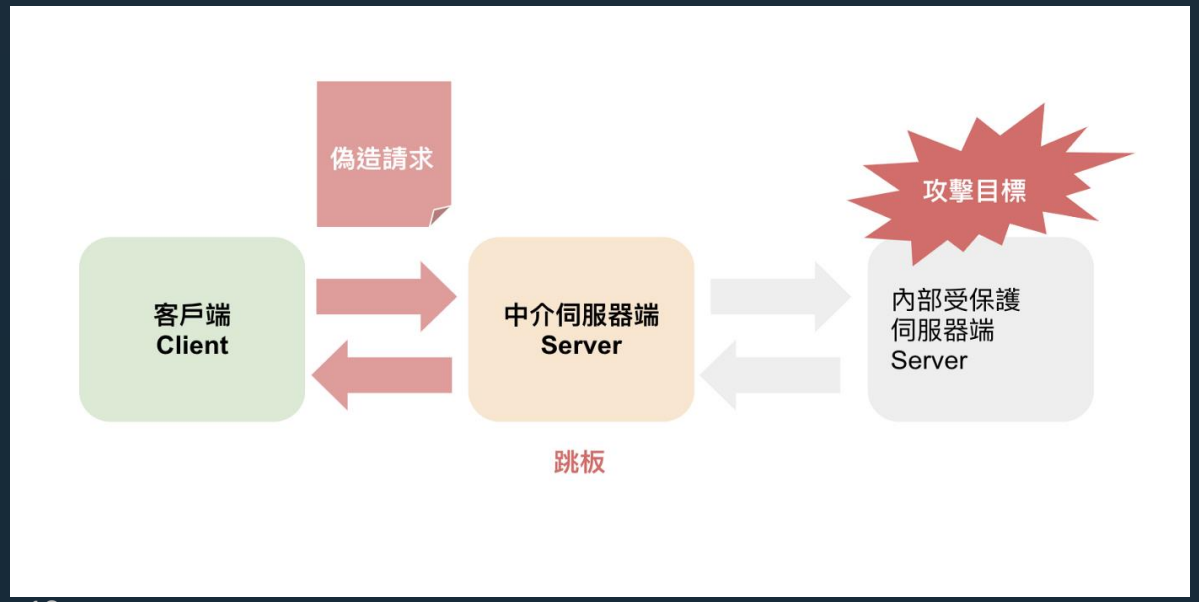
```
var jc :JAXBContext = JAXBContext.newInstance(Comment.class);
var xif :XMLInputFactory = XMLInputFactory.newInstance();

if (secure) {
    xif.setProperty(XMLConstants.ACCESS_EXTERNAL_DTD, ""); // Compliant
    xif.setProperty(XMLConstants.ACCESS_EXTERNAL_SCHEMA, ""); // compliant
}

System.err.println("XML : " + xml);
// XXE solution : Add following 2 lines to disable DTD and External Entities // 2022-05-30
xif.setProperty(XMLInputFactory.SUPPORT_DTD, false);
xif.setProperty(XMLInputFactory.IS_SUPPORTING_EXTERNAL_ENTITIES, false);

var xsr :XMLStreamReader = xif.createXMLStreamReader(new StringReader(xml));
```

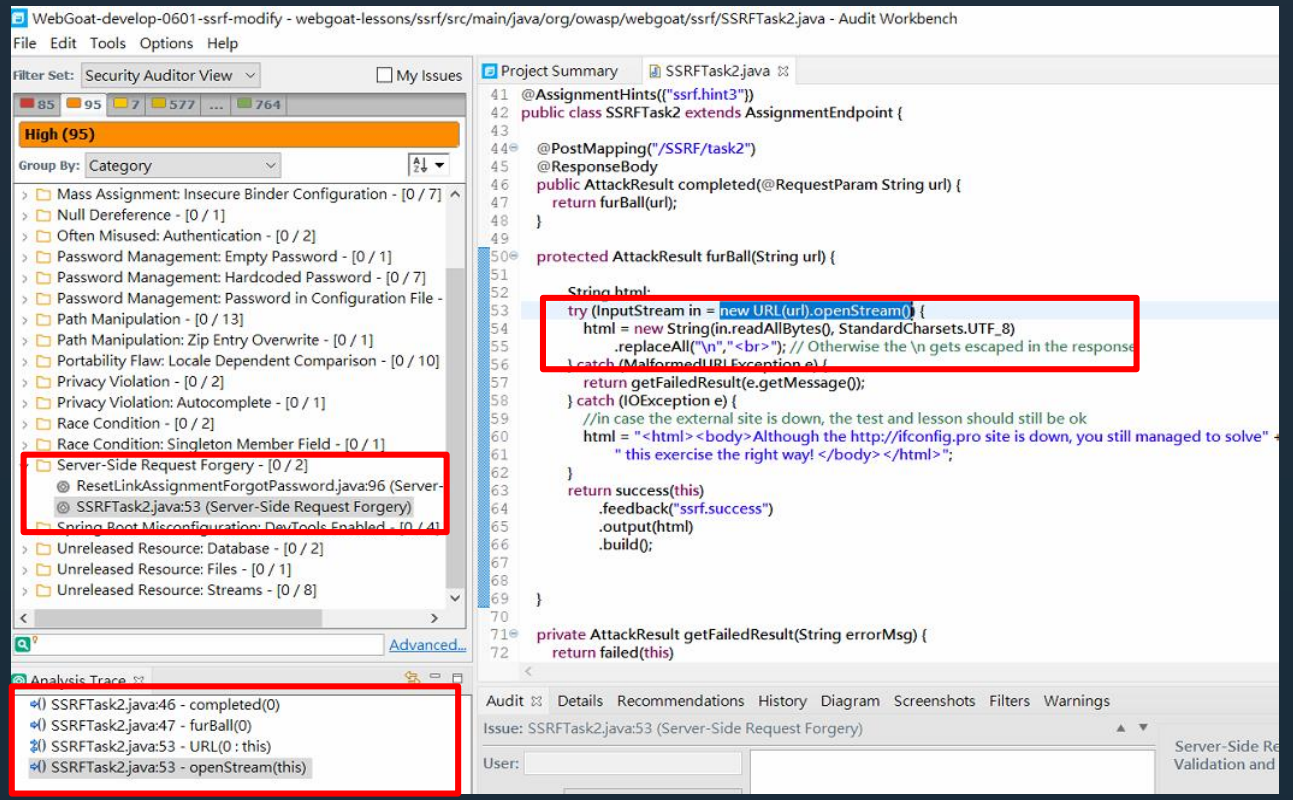


主要是在程式碼關閉注入的行為
所以黑箱很難知道要填什麼樣的值

SSRF問題說明

- 主要是伺服器端所提供的接口中，並未對客戶端所傳輸過來的URL參數進行過濾，導致攻擊者可以傳入惡意請求，來讓後端伺服器傳送攻擊者所需要的資訊。



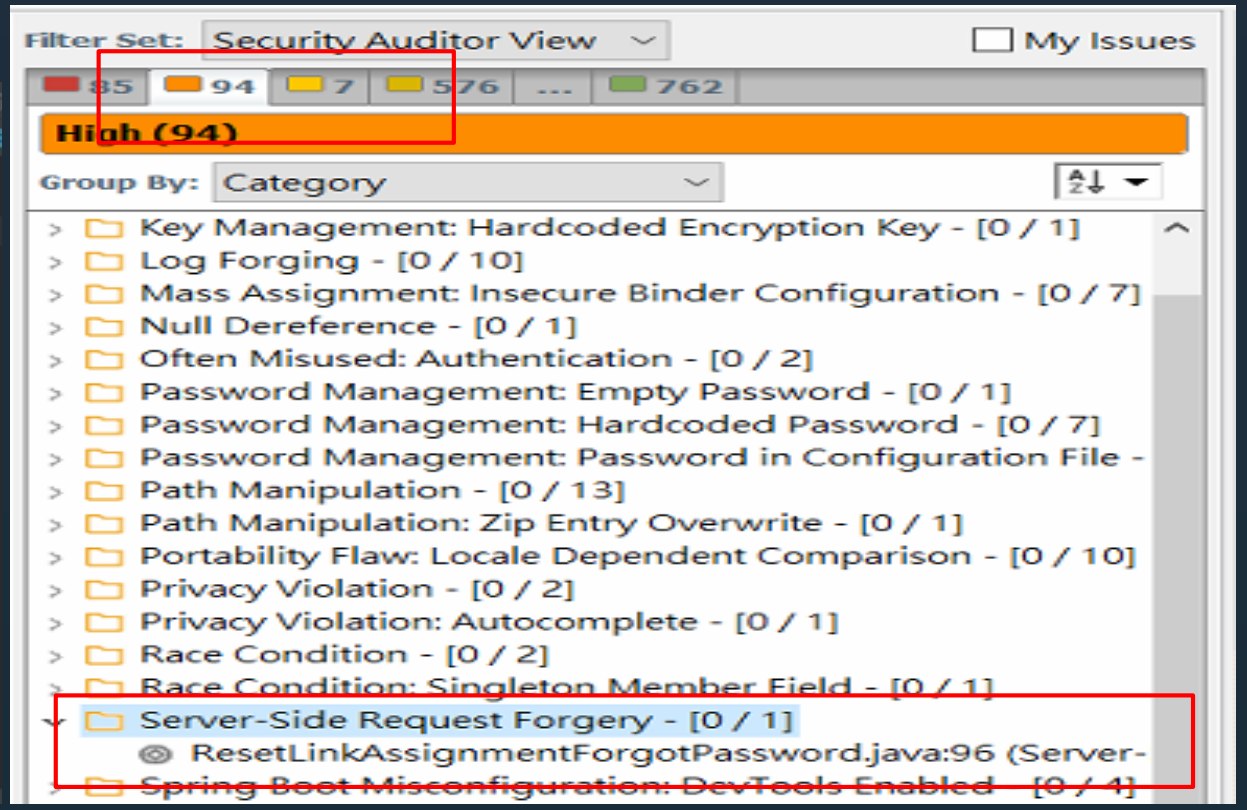


- 探測內部資源
- 獲取內部敏感文件訊息
- 配合其它漏洞攻擊內網其它機器

Demo

SSRF問題修正

```
@ResponseBody
public AttackResult completed(@RequestParam int index) { return furBall(index); }

protected AttackResult furBall(int index) {
        String[] allow_urls = {"http://ifconfig.pro", "https://www.google.com"};
        if(index < 0 && index > 1){
            return failed( assignment: this).output("Index only allow 0 and 1").build();
        }
        String html;
        try (InputStream in = new URL(allow_urls[index]).openStream()) {
            html = new String(in.readAllBytes(), StandardCharsets.UTF_8)
                    .replaceAll( regex: "\n", replacement: "<br>"); // Otherwise the \n gets escaped in the response
        } catch (MalformedURLException e) {
            return getFailedResult(e.getMessage());
        } catch (IOException e) {
            //in case the external site is down, the test and lesson should still be ok
            html = "<html><body>Although the http://ifconfig.pro site is down, you still managed to solve" +
                    " this exercise the right way!</body></html>";
        }
        return success( assignment: this)
                .feedback("ssrf.success")
```



- 過濾返回的信息，在返回結果展示給客戶前，先驗證返回的信息是否符合標準。
- 統一錯誤信息，避免用戶可以根據信息來判斷遠程服務器的端口狀態。
- 限制請求的端口，比如80,443等
- 禁止不常用的協議，儘儘允許HTTP和HTTPS請求，可以防止類似於File://. ftp等引起的問題。
- 使用白名單的方式

主要是因為刺探的格式不固定
所以白箱較黑箱適合

NULL Pointer Dereference

- 如果一個pointer的值是null的時會造成程式毀損或當機

```
[CanBeNull]
public static object Bar()
{
    return null;
}

public void TestBar()
{
    Console.WriteLine(Bar().ToString());
}
```

(method) object ValueAnalysisHelpers.Test.Bar()

Possible 'System.NullReferenceException'

```
int b = a ?? -1;
```

相等?

=

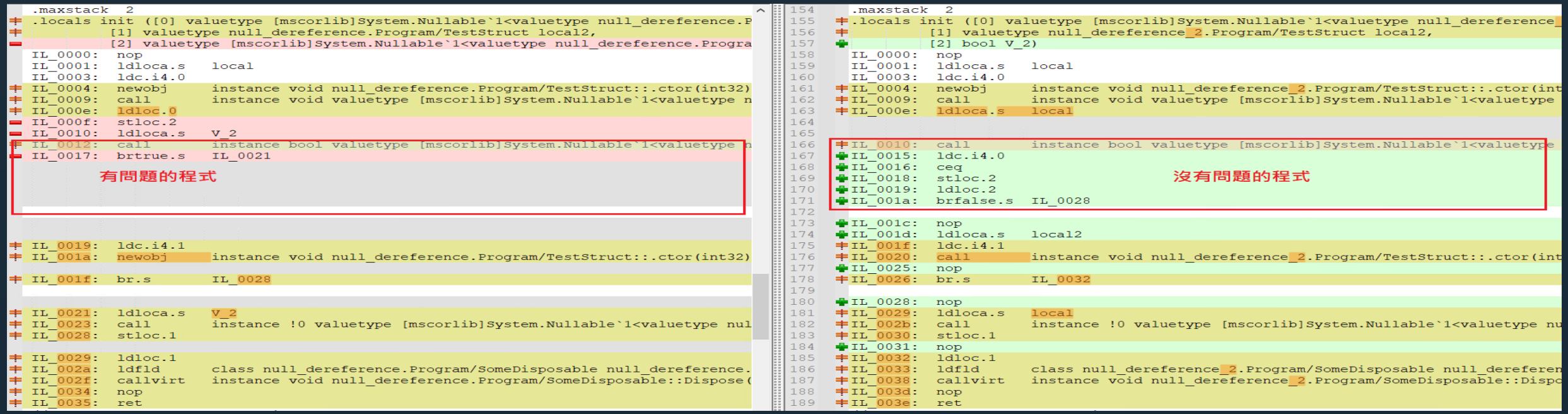
```
int b;
If( a == null ){
    b = -1;
}else{
    b = a;
}
```

上面的寫法是相等的嗎?

https://learn.microsoft.com/en-US/dotnet/csharp/language-reference/proposals/csharp-9.0/nullable-reference-types-specification

Null Pointer Dereference偵測

- 透過編譯掃描才能找到真的問題所在



- 此問題只有白箱能掃描出來
- 預防之道就是在使用pointer的時候，檢查其值

MICRO FOCUS

Postgres SQL Driver問題說明

CVE-2022-21724

Issue
CVE-2022-21724

Severity
CVE CVSS 3: 9.8
CVE CVSS 2.0: 7.5
Sonatype CVSS 3: 8.5

Weakness
CVE CWE: 665

Source
National Vulnerability Database

Categories
Data

Description from CVE

pgjdbc is the offical PostgreSQL JDBC Driver. A security hole was found in the jdbc driver for postgresql database while doing security research. The system using the postgresql library will be attacked when attacker control the jdbc url or properties. pgjdbc instantiates plugin instances based on class names provided via `authenticationPluginClassName`, `sslhostnameverifier`, `socketFactory`, `sslfactory`, `sslpasswordcallback` connection properties. However, the driver did not verify if the class implements the expected interface before instantiating the class. This can lead to code execution loaded via arbitrary classes. Users using plugins are advised to upgrade. There are no known workarounds for this issue.

Explanation

The `postgresql` package is vulnerable to Remote Code Execution (RCE). The `instantiate()` method in the `objectFactory` class allows any class type to be instantiated and executed without ensuring that the class implements the expected interface. This allows plugins and some factory implementations that can be used in a JDBC connection to instantiate Java classes that could allow users to execute arbitrary code. This gives attackers with the privileges to make a connection to a PostgreSQL database using a JDBC driver to instantiate a class that allows them to inject arbitrary code. This may also be used to download arbitrary code from a remote resource, that is then instantiated and executed, leading to RCE.

CVE-2022-26520

Issue
CVE-2022-26520

Severity
CVE CVSS 3: 9.8
CVE CVSS 2.0: 7.5
Sonatype CVSS 3: 7.5

Weakness
Sonatype CWE: 20

Source
National Vulnerability Database

Categories
Data
Operational

Description from CVE

** DISPUTED ** In pgjdbc before 42.3.3, an attacker (who controls the jdbc URL or properties) can call java.util.logging.FileHandler to write to arbitrary files through the loggerFile and loggerLevel connection properties. An example situation is that an attacker could create an executable JSP file under a Tomcat web root. NOTE: the vendor's position is that there is no pgjdbc vulnerability; instead, it is a vulnerability for any application to use the pgjdbc driver with untrusted connection properties.

Explanation

The `postgresql` package is vulnerable to Improper Input Validation. The `setupLoggerFromProperties()` method in the `Driver` class sets `loggerLevel` and `loggerFile` properties of the Java logger from parameters extracted from JDBC URLs. A remote attacker who can provide a JDBC URL or its properties can exploit this vulnerability to write to files in arbitrary locations on the affected server.

Note: The Sonatype security research team has determined that, while the project disputes this vulnerability, risk exists for applications that accept JDBC URLs or their properties from untrusted users. Ultimately, the project addressed this issue in version 42.3.3-rc1 by removing the functionality responsible for setting the aforementioned logger properties from JDBC URLs.

Risk Remediation

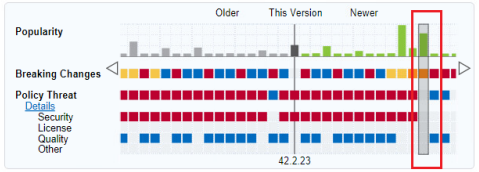
Recommended Versions

Upgrade to 42.3.3
Next version with no policy violation
Compare

Upgrade to 42.3.3
Next version with no policy violations for this component and its dependencies
Compare

The current version doesn't cause Build failure for this component and its dependencies

Version Explorer



Compare Versions

	CURRENT	SELECTED
Version	42.2.23	42.3.3
Highest Policy Threat	10 within 2 policies	None
Security Violation Threat	10	None
Highest CVSS Score	9.8	None
License Violation Threat	None	None
Effective License	BSD-2-Clause	BSD-2-Clause

Demo

此問題無法由外部偵測試，所以黑箱無法掃描，
這個問題也不是程式碼所以白箱也無法掃描

MICRO FOCUS

白箱無法協助項目

心跳（Heartbeat）停止：Heartbleed OpenSSL漏洞分析

2014 年 04 月 14 日 Trend Labs 趨勢科技全球技術支援與研發中心 Deep Security, Heartbleed, Heartbleed 漏洞, 漏洞攻擊, 趨勢科技產品, 重大資安事件

軟體會有漏洞是我們必須面對的現實，如果我們夠幸運且夠勤快，那就可以在網路犯罪份子攻擊它之前先加以修補。事實並不一定總是如此，但幸好那些算是例外，而不是常態。

先前Tomcat Server被揭露一個名為Ghostcat的漏洞，Apache Tomcat已在2月中旬於官方網站上，針對現行不同版本發布更新修補，近期，則有臺灣資安業者TeamT5杜浦數位安全發現，有中國駭客組織疑似利用該漏洞，在臺灣校園網站上傳BiFrost後門程式，因此呼籲企業與組織要注意此GhostCat漏洞的嚴重性。值得關注的是，該漏洞也波及多個Linux平臺，因而在最近一個多月的時間陸續修補，就連商用軟體也受影響。

關於這個Ghostcat漏洞（CVE-2020-1938），它在CVSS v3.1的分數是9.8，屬於重大風險漏洞，由資安業者長亭科技在2月20日揭露其風險，指出該漏洞存在於Apache JServ Protocol（AJP）中，而若是網站應用提供了文件上傳的功能，將導致遠端指令執行（RCE）漏洞。

值得注意的是，在Ghostcat漏洞被揭露後，隔沒幾日，開放原始碼資安公司Snyk

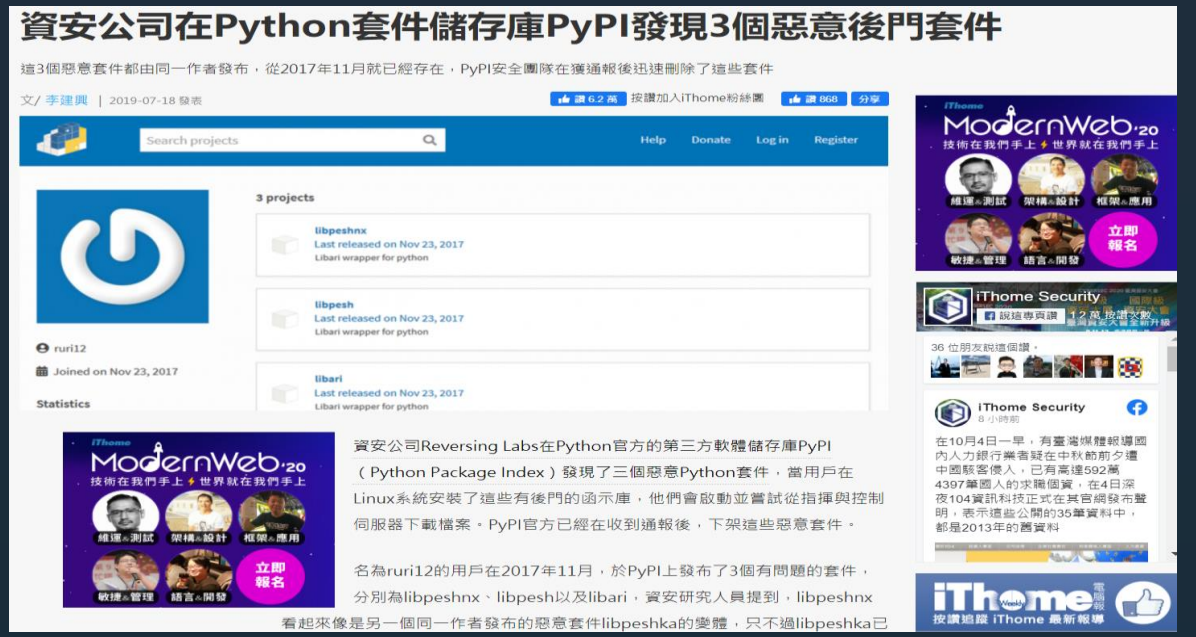
- 網站系統弱點問題-例如Tomcat網站漏洞檢查
- 認證授權等系統問題-密碼太簡單、SSL協定問題、未經授權的頁面
- 若是程式邏輯上合理的問題-上傳檔案、系統錯誤訊息

開放源碼掃描無法協助項目

美國第三大消費者信用報告業者Equifax周二（9/26）宣布，該公司執行長Richard Smith將卸下執行長與董事職位，即日退休，且將無償擔任顧問以協助交接。這是Equifax因駭客入侵造成1.43億名消費者的個人資料外洩之後，第三位下台的高階主管，首當其衝的Equifax資訊長Susan Mauldin與安全長David Webb皆已在9月15日下台。

駭客是透過Apache Struts2漏洞入侵了Equifax，Apache基金會已在今年3月修補該漏洞，而Equifax則是在5月被入侵，駭客存取了1.43億名的消費者姓名、社會安全碼、生日、地址，以及21萬筆的消費者信用卡資料，有接近一半的美國人口受到影響。

目前Equifax仍在尋找執行長的繼任人選。該公司的非執行董事Mark Feidler表示，董事會相當關心且完全專注在這次的網路安全意外上，他們正全力提供消費者



- 主要是檢查Library的hash值，所以並無法掃描程式的問題
- 有些需要靠套件管理軟体(package management)來輸入檔案掃描，所以整合上有侷限
- 遇到library是自己開發的話，那就無法掃描，此時還需要回到程式碼掃描

黑箱無法協助的部份


```
1 env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
2 #env [OPTION]... [NAME=VALUE]... [COMMAND [ARGS]...]
```

- 沒有注入點的問題無法掃描
- 沒有規則可執行的無法掃描

OWASP Top 10 2017

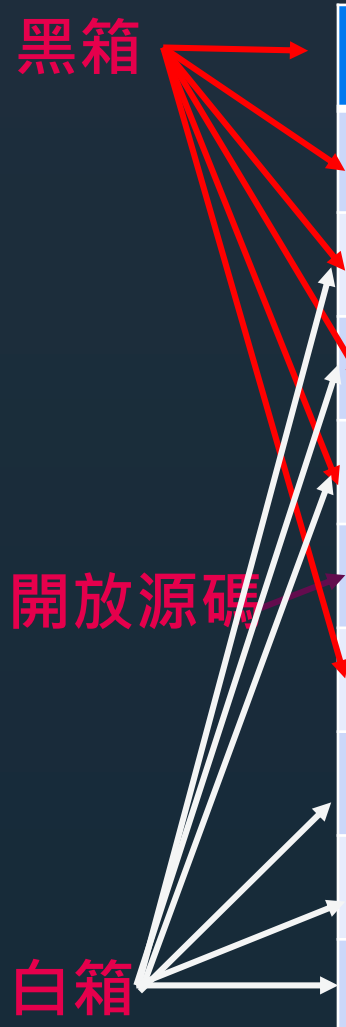
白箱

開放源碼

黑箱

https://www.ithome.com.tw/news/118411

OWASP TOP 10 2021



1)權限控制失效(Broken Access Control)
2)加密機制失效(Cryptographic Failures)
3)注入式攻擊(Injection)
4)不安全設計(Insecure Design)
5)安全設定的錯誤(Security Misconfiguration)
6)危險或過時的元件(Vulnerable and Outdated Components)
7)認證及驗證機制失效(Identification and Authentication Failure)
8)軟体及資料完整性失效(Software and Data Integrity Failure)
9)資安記錄及監控失效(Security Logging and Monitoring Failure)
10)伺服器端請求偽造(Server-Side Request Forgery(SSRF))

總結

- 如果網站有互動欄位的話，黑白箱可以檢測問題
  - Sql injection、XSS等問題
- 如果網站沒有注入點的話，白箱是主要偵測工具
- 網站漏洞、認證及加密的話，黑箱會是主要的工具
- 黑箱可以偵測部份的開放源碼套件的問題，但主体還是開放源碼套件掃描工具
- 白箱、黑箱及開放源碼套件掃描工具是主要網站掃描工具

Fortify能帶來的好處

- 本身擁有白黑箱的技術，可檢測這兩項的問題，同時也提供黑白箱交叉技術，可以讓管理者快速的了解問題發生的原因及如何檢測
- OEM開放源碼檢測工具Sonatype，所以可同時掃描程式碼及開放源碼，不用一個掃描後再掃描另外一個
- demo

MICRO FOCUS®