

**Find, Fix and Fortify**

**If you want better security,  
think like a bad guy.**

Get to threats before they get to you. Today, a global threat marketplace collaborates and innovates to attack our organizations 24/7. It's time to think like a bad guy. HP draws on decades of security experience to take the fight to adversaries before they attack. We can help you predict and disrupt threats so they don't become headlines - by using insights from big data. And with HP Threat Exchange, we're continuously trading intelligence to keep your business safe to innovate. Better security. See how it leads to a better enterprise. Visit [hp.com/go/security](http://hp.com/go/security)

# Find, Fix and Fortify in a HYBRID Way

부제: 포티파이 Hybrid 2.0 보안 취약점 융복합 분석 기법 소개

April 2014

엔터프라이즈 시큐리티 프로젝트 | 한국휴렛팩커드



# Find, Fix and Fortify



최근 보안 이슈 분석

---



보안 취약점 분석 기법 리뷰

---



Hybrid 2.0 보안취약점 분석

## OpenSSL Security Advisory [07 Apr 2014]

### ===== TLS heartbeat read overrun (CVE-2014-0160) =====

A missing bounds check in the handling of the TLS heartbeat extension can be used to reveal up to 64k of memory to a connected client or server.

**Only 1.0.1 and 1.0.2-beta releases of OpenSSL are affected including 1.0.1f and 1.0.2-beta1.**

.....

Affected users should **upgrade to OpenSSL 1.0.1g**. Users unable to immediately upgrade can alternatively **recompile OpenSSL with -DOPENSSL\_NO\_HEARTBEATS**.

**1.0.2 will be fixed in 1.0.2-beta2.**



## Blog posts for Heartbleed bugs on HP Security Research Blog :

- Heartbleed protection with **HP TippingPoint** : <http://h30499.www3.hp.com/t5/HP-Security-Products-Blog/Heartbleed-protection-with-HP-TippingPoint/ba-p/6444226#.U0ilCvmSw-0>
- HPSR Software security content update - Heartbleed bug detection (**HP WebInspect**) : <http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/HPSR-Software-security-content-update-Heartbleed-bug-detection/ba-p/6445654#.U0ii3vmSw-0>
- Heartbleed causes heartache : <http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/Heartbleed-causes-heartache/ba-p/6444868#.U0ikJPmSw-0>
- Thoughts on the Heartbleed bug : [http://h30499.www3.hp.com/t5/Fortify-Application-Security/Thoughts-on-the-Heartbleed-Bug/ba-p/6442708#.U0Xp-\\_mSw-0](http://h30499.www3.hp.com/t5/Fortify-Application-Security/Thoughts-on-the-Heartbleed-Bug/ba-p/6442708#.U0Xp-_mSw-0)

properties of their respective owners.



```
diff --git a/ssl/d1_both.c b/ssl/d1_both.c
index 7a5596a..2e8cf68 100644 (file)
--- a/ssl/d1_both.c
+++ b/ssl/d1_both.c
@@ -1459,26 +1459,36 @@ dtls1_process_hear
     unsigned int payload;
     unsigned int padding = 16; /* RFC 6347 sec. 4.2.1.2.1 */

     /* Read type and payload length first */
     hbtype = *p++;
     n2s(p, payload);
     pl = p;

     if (s->msg_callback)
         s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
             s->s3->rrec.data[0], s->s3->rrec.length,
             s, s->msg_callback_arg);

     /* Read type and payload length first */
     if (1 + 2 + 16 > s->s3->rrec.length)
         return 0; /* silently discard per RFC 6520 sec. 4 */
     hbtype = *p++;
     n2s(p, payload);
     if (1 + 2 + payload + 16 > s->s3->rrec.length)
         return 0; /* silently discard per RFC 6520 sec. 4 */
     pl = p;

     if (hbtype == TLS1_HB_REQUEST)
     {
         unsigned char *buffer, *bp;
         unsigned int write_length = 1 + 2 + payload + padding;
         if (write_length > SSL3_RT_MAX_PLAIN_LENGTH)
             return 0;
     }

```

```
diff --git a/ssl/t1_lib.c b/ssl/t1_lib.c
index b82fada..bddffd9 100644 (file)
--- a/ssl/t1_lib.c
+++ b/ssl/t1_lib.c
@@ -1459,26 +1459,36 @@ dtls1_process_hear
     unsigned int payload;
     unsigned int padding = 16; /* RFC 6347 sec. 4.2.1.2.1 */

     /* Read type and payload length first */
     hbtype = *p++;
     n2s(p, payload);
     pl = p;

     if (s->msg_callback)
         s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
             s->s3->rrec.data[0], s->s3->rrec.length,
             s, s->msg_callback_arg);

     /* Read type and payload length first */
     if (1 + 2 + 16 > s->s3->rrec.length)
         return 0; /* silently discard per RFC 6520 sec. 4 */
     hbtype = *p++;
     n2s(p, payload);
     if (1 + 2 + payload + 16 > s->s3->rrec.length)
         return 0; /* silently discard per RFC 6520 sec. 4 */
     pl = p;

     if (hbtype == TLS1_HB_REQUEST)
     {
         unsigned char *buffer, *bp;

```

# 보안 사고 시작은 소스 코드상의 보안 취약점!

오픈SSL 하트블리드 취약점이 세상에 알려지며 보안 제품은 물론이고  
**모든 소프트웨어 개발에 ‘시큐어  
코딩’을 의무화** 해야 한다는 목소리에 힘이 실린다

Source : <http://www.etnews.com/20140415000041>



# Find, Fix and Fortify



최근 보안 이슈 분석

---



보안 취약점 분석 기법 리뷰

---



Hybrid 2.0 보안 취약점 분석

# 소프트웨어 보안 취약점 분석 기법

## 보안 취약점 자동화 분석 기법

Static 정적분석 <span style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 5px;">1</span>	<span style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 5px; font-size: 2em;">?</span>	Dynamic 동적분석 <span style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 5px;">2</span>
소스코드	새로운 보안 취약점 분석 기법은?	웹 애플리케이션(침투테스트)
<ul style="list-style-type: none"> <li>• <b>White Box Testing</b></li> <li>• 보안 취약점 전수검사(多)</li> <li>• 소스코드 레벨의 보안 취약점 상세결과 제공</li> <li>• 근원적인 보안취약점 제거방법 제공</li> </ul>	<div style="font-size: 4em; color: white; background-color: #dc3545; border-radius: 50%; width: 100px; height: 100px; display: flex; align-items: center; justify-content: center; margin: 0 auto;"> <span>?</span> </div>	<ul style="list-style-type: none"> <li>• <b>Black Box Testing</b></li> <li>• 실제 공격가능한 보안 취약점 검출(小)</li> <li>• 보안취약점 우선순위제공 (제한적인 분석범위)</li> <li>• 근원적인 보안취약점 해결 한계점 보유</li> <li>• 웹 <b>Request /Response</b>를 통한 보안취약점 검출</li> </ul>
사용자: 개발자		사용자 : 보안팀/관제팀





# Find, Fix and Fortify



최근 보안 이슈 분석

---



보안 취약점 분석 기법 리뷰

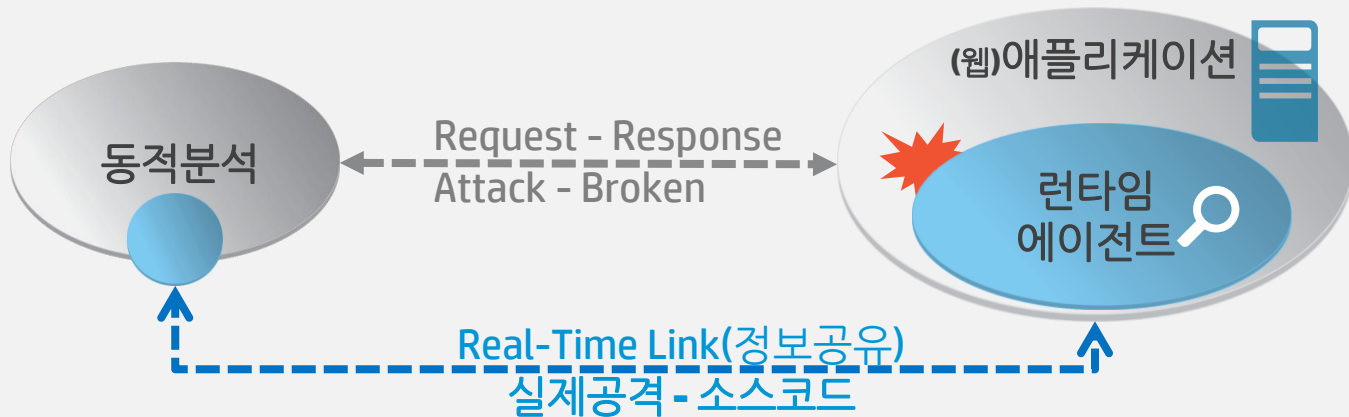
---



**Hybrid 2.0 보안 취약점 분석**

# Hybrid 2.0 보안 취약점 분석

## 분석기법 소개: 런타임 분석 기법 (Gray Box Testing)



**R**

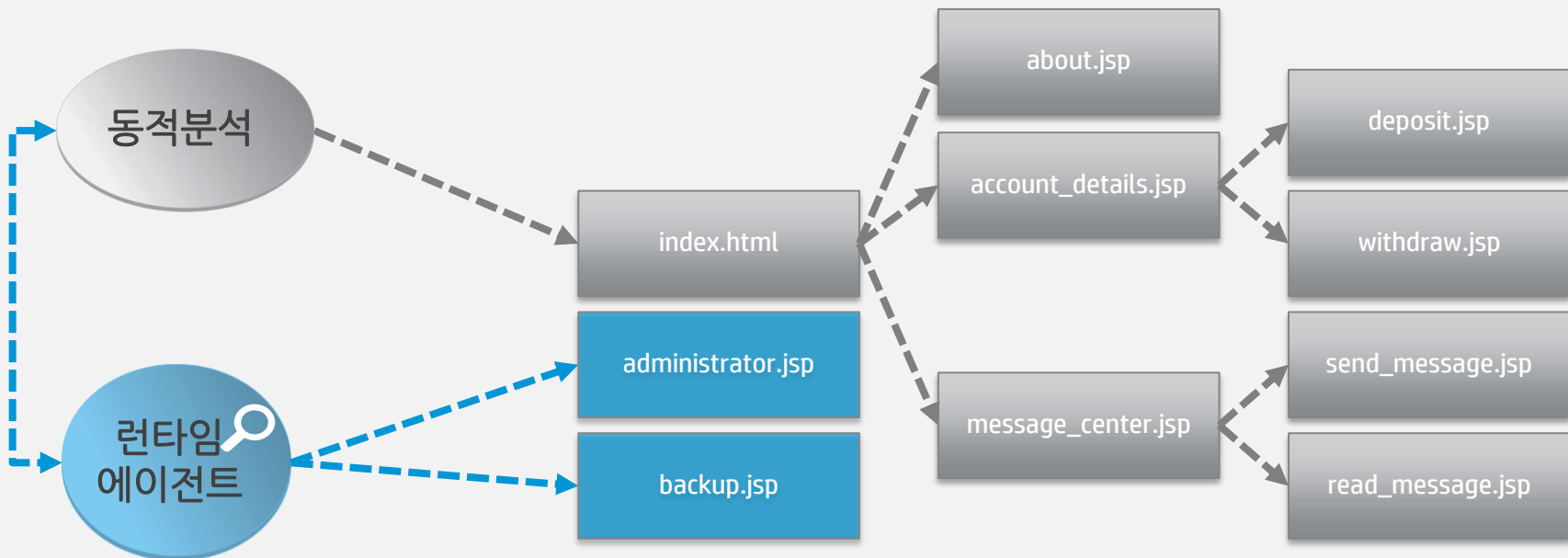
### 동적분석과의 협업을 통한 런타임 분석기법 기대효과

- **Find More** (동적분석에서 놓친) 보안 점검범위 확대(PT)
- **Fix Faster** (실제공격과 소스코드 연계 통한) 소스코드 라인상 보안 취약점 정보 제공

# Hybrid 2.0 보안 취약점 분석

분석기법 소개: 런타임 분석 기법 (Gray Box Testing) – Find More (Missing Paths)

**R Find More** : 런타임 분석기법은 동적분석의 Crawling에 의해서 **Missing**되었던 자원을 발견하고 이를 동적분석기에 전달, 보안점검범위를 확대합니다.



# Hybrid 2.0 보안 취약점 분석

## 분석기법 소개: 동적분석 기법 (White Box Testing)

### D 동적분석 예제 : WebGoat 5.4 / WebInspect 10.1 (OWASP Top 10 App Security Policy)

Crawl 983 of 983

Audit 2,567 of 2,567

#### Scan Status

Completed

#### Activity

Req/Sec

Crawling

-

Auditing

-

#### Other

Evt/Sec

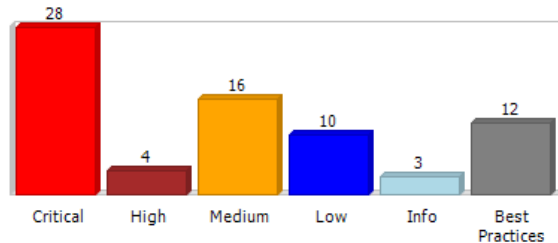
Script Execution

-

SecurityScope

Not Detected

#### Vulnerabilities



Attack Type	Attacks						
Manipulation	31,571	28	2	9	0	0	0
Exploratory	18,081	0	0	0	1	0	0
Other	30,437	0	2	7	9	3	12

#### Scan

Type: Site  
Duration: 08:05:46  
Policy: OWASP...  
Deleted Items:

#### Crawl

Hosts: 1  
Sessions: 748

#### Audit

Attacks Sent: 80,089  
Issues: 73

#### Network

Total Requests: 81,292  
Failed Requests: 4  
Script Includes: 15  
Macro Requests: 1  
404 Probes: 407  
404 Check Redirects: 19  
Verify Requests: 0  
Logouts: 0  
Macro Playbacks: 1  
AJAX Requests: 2  
Script Events: 45,803  
Kilobytes Sent: 98,908K  
Kilobytes Received: 1,393,9...


- Crawl(Session) : 748
- Issues(보안취약점) : 58

# Hybrid 2.0 보안 취약점 분석

## 분석기법 소개: 동적분석 기법 (White Box Testing)

### D 동적분석 예제 : WebGoat 5.4 / WebInspect 10.1 (OWASP Top 10 App Security Policy)

http://172.16.100.20:8080/WebGoat-5.4/attack?Screen=601&menu=800

Vulnerability to Review:  SQL Injection (c



Browser Request Response Stack Tra

```
POST /WebGoat-5.4/attack?Screen=601&menu=800
Referer: http://172.16.100.20:8080/
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Accept: */*
Pragma: no-cache
User-Agent: Mozilla/5.0 (Windows; U; MSIE 9.0; en-US; Windows NT 6.0)
Host: 172.16.100.20:8080
Connection: Keep-Alive
X-WIPP: AscVersion=10.1.177.0; CapVersion=10.1.177.0; WIPP-RequestID=620417d2-d95a-456a-9b5e-06b6bfc08797
X-Scan-Memo: Category="Audit.Attack"; WIPP-RequestID=620417d2-d95a-456a-9b5e-06b6bfc08797
X-RequestManager-Memo: StateID="36"; WIPP-RequestID=620417d2-d95a-456a-9b5e-06b6bfc08797
X-Request-Memo: ID="620417d2-d95a-456a-9b5e-06b6bfc08797"; WIPP-RequestID=620417d2-d95a-456a-9b5e-06b6bfc08797
Cookie: JSESSIONID=9C9FEEE6264F9A0E8E8E8E8E8E8E8E8E
```

username=%27%09OR&SUBMIT=Submit

http://172.16.100.20:8080/WebGoat-5.4/attack?Screen=601&menu=800

Vulnerability to Review:  SQL Injection (confirmed)

 Retest  M

HTTP/1.1 200 OK

```
Server: Apache-Coyote/1.1
X-WIPP-Version: java / 1.3 / WIN-1BICR1EIJ7G_2776
X-WIPP-RequestID: 620417d2-d95a-456a-9b5e-06b6bfc08797
X-WIPP-Update: Application
X-WIPP-User-Logged-On: true
Content-Type: text/html; charset=ISO-8859-1
Date: Thu, 10 Apr 2014 13:15:49 GMT
Content-Length: 30473
```

- HTTP Request/Response를 통한 공격성공유무 제공 Only

# Hybrid 2.0 보안 취약점 분석

분석기법 소개: 런타임 분석 기법 (Gray Box Testing) – Find More & Fix Faster

**R** 동적+런타임분석 예제 : WebGoat 5.4 / WebInspect 10.1 (OWASP Top 10 App Security Policy)

**Crawl** 1,751 of 1,751

**Audit** 5,756 of 5,756

**Scan Status**  
Completed ✔

**Activity**      **Req/Sec**

Crawling      -

Auditing      -

**Other**      **Evt/Sec**

Script Execution      -

SecurityScope      Detected

**Vulnerabilities**

Attack Type	Attacks	!	●	●	●	●	✔
Manipulation	32,286	34	2	9	0	0	0
Exploratory	385	0	0	0	0	0	0
Other	2,784	0	9	12	22	5	20

**Scan**

Type: Site

Duration: 17:38:21

Policy: OWASP...

Deleted Items:

**Crawl**

Hosts: 1

Sessions: 1,288

**Audit**

Attacks Sent: 35,455

Issues: 113

**NETWORK**

Total Requests: 46,685

Failed Requests: 3

Script Includes: 13

Macro Requests: 1

404 Probes: 476

404 Check Redirects: 23

Verify Requests: 0

Logouts: 0

Macro Playbacks: 1

AJAX Requests: 2

Script Events: 56,525

Kilobytes Sent: 56,963K

Kilobytes Received: 756,998K

- **Crawl(Session) : 1,248**
- **Issues(보안취약점) : 88**



# Hybrid 2.0 보안 취약점 분석

분석기법 소개: 런타임 분석 기법 (Gray Box Testing) – Find More & Fix Faster

**R** 동적+런타임분석 예제 : WebGoat 5.4 / WebInspect 10.1 (OWASP Top 10 App Security Policy)

The screenshot shows the WebInspect interface for a confirmed SQL Injection vulnerability. The URL is `http://172.16.100.20:8080/WebGoat-5.4/attack?Screen=601&menu=800`. The vulnerability is confirmed, and the SecurityScope Trigger is `SELECT * FROM usersystemdata WHERE username = ''\tOR'`. The SecurityScope Stack Trace shows the following internal activity:

```

at org.hsqldb.jdbc.jdbcStatement.executeQuery(Unknown Source)
at org.owasp.webgoat.lessons.ThreadSafetyProblem.createContent(ThreadSafetyProblem.java:98)
at org.owasp.webgoat.lessons.AbstractLesson.handleRequest(AbstractLesson.java:771)
at org.owasp.webgoat.lessons.ThreadSafetyProblem.handleRequest(ThreadSafetyProblem.java:189)
at org.owasp.webgoat.HammerHead.makeScreen(HammerHead.java:367)
at org.owasp.webgoat.HammerHead.doPost(HammerHead.java:146)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:643)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:723)
at org.apache.jsp._jspService(_jspService.java:2003)
at org.apache.jsp._jspService(_jspService.java:2003)
at org.apache.jsp._jspService(_jspService.java:2003)
at org.apache.jsp._jspService(_jspService.java:2003)
at org.apache.jsp._jspService(_jspService.java:2003)
at org.apache.jsp._jspService(_jspService.java:2003)

```

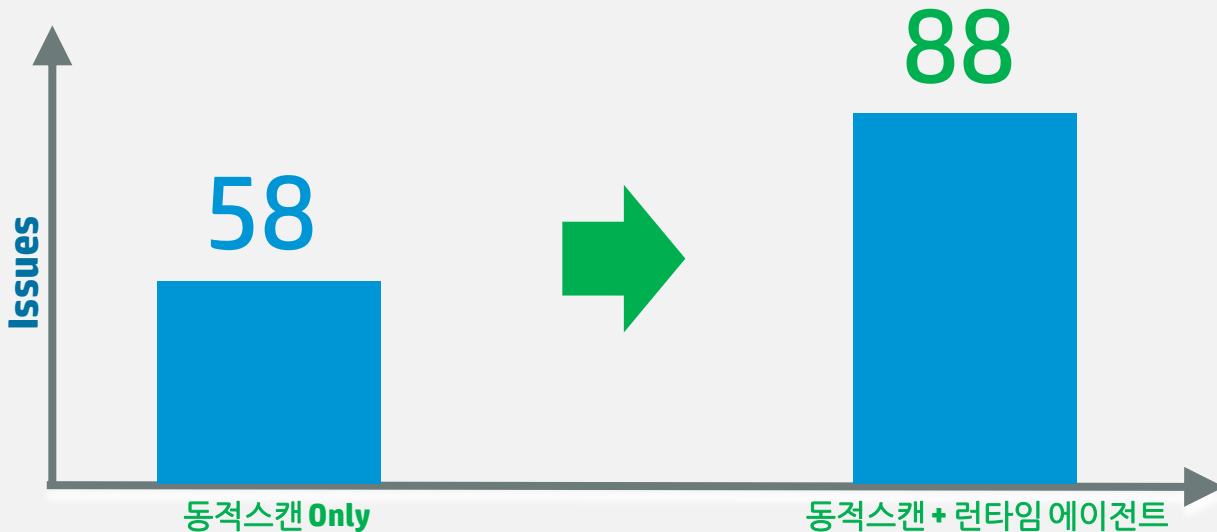
The request details show a POST request to `/WebGoat-5.4/attack?Screen=601&menu=800` with a `username=%27%09OR&SUBMIT=Submit` parameter.

• 공격시 발생하는 Stack Trace를 통한 내부 Activity 및 보안 취약점 보유 (소스코드: 라인) 제공

# Hybrid 2.0 보안 취약점 분석

분석기법 소개: 런타임 분석 기법 (Gray Box Testing) – Find More & Fix Faster

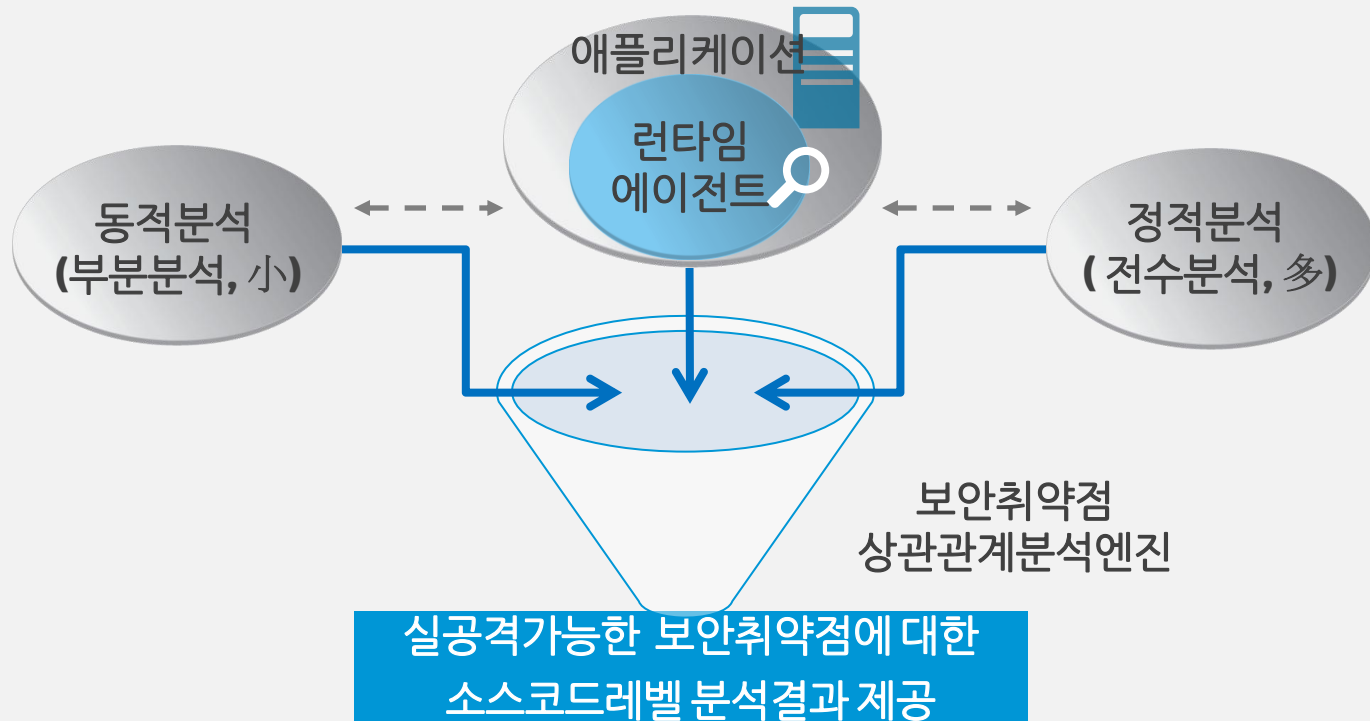
**R Find More & Fix Faster:** 런타임 분석기법은 동적분석의 **Crawling**에 의해서 **Missing**되었던 자원을 발견하고 이를 동적분석기에 전달, **보안점검범위를 확대**하며 **Stack Trace**를 통해 **보안취약점과 소스코드 라인을 연결**합니다.





# Hybrid 2.0 보안 취약점 분석

## 분석기법 소개



# Hybrid 2.0 보안 취약점 분석

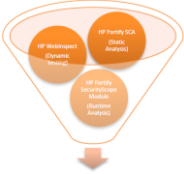
## 분석기법 소개

Static 정적분석 <span style="float: right;">1</span>	Runtime 런타임분석 <span style="float: right;">+</span>	Dynamic 동적분석 <span style="float: right;">2</span>
소스코드	웹 애플리케이션	웹 애플리케이션(침투테스트)
<ul style="list-style-type: none"> <li>• <b>White Box Testing</b></li> <li>• 보안 취약점 전수검사(多)</li> <li>• 소스코드 레벨의 보안 취약점 상세결과 제공</li> <li>• 근원적인 보안취약점 제거방법 제공</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Gray Box Testing</b></li> <li>• 애플리케이션 내부 실행 프로세스 모니터링</li> <li>• 동적분석 솔루션과 연계하여 오탐축소, 보안 점검대상 확대(<b>Hidden applications</b>)</li> <li>• <b>Stack Trace</b>를 통한 새로운 보안 취약점 대상 공격 식별</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Black Box Testing</b></li> <li>• 실제 공격가능한 보안 취약점 검출(小)</li> <li>• 보안취약점 우선순위제공(제한적인 분석범위)</li> <li>• 근원적인 보안취약점 해결 한계점 보유</li> <li>• 웹 <b>Request /Response</b>를 통한 보안취약점 검출</li> </ul>
사용자: 개발자	사용자 : 테스트/보안팀	사용자 : 보안팀/관제팀



# Hybrid 2.0 보안 취약점 분석

분석기법 소개: *Lining up with Attack and Source Code*

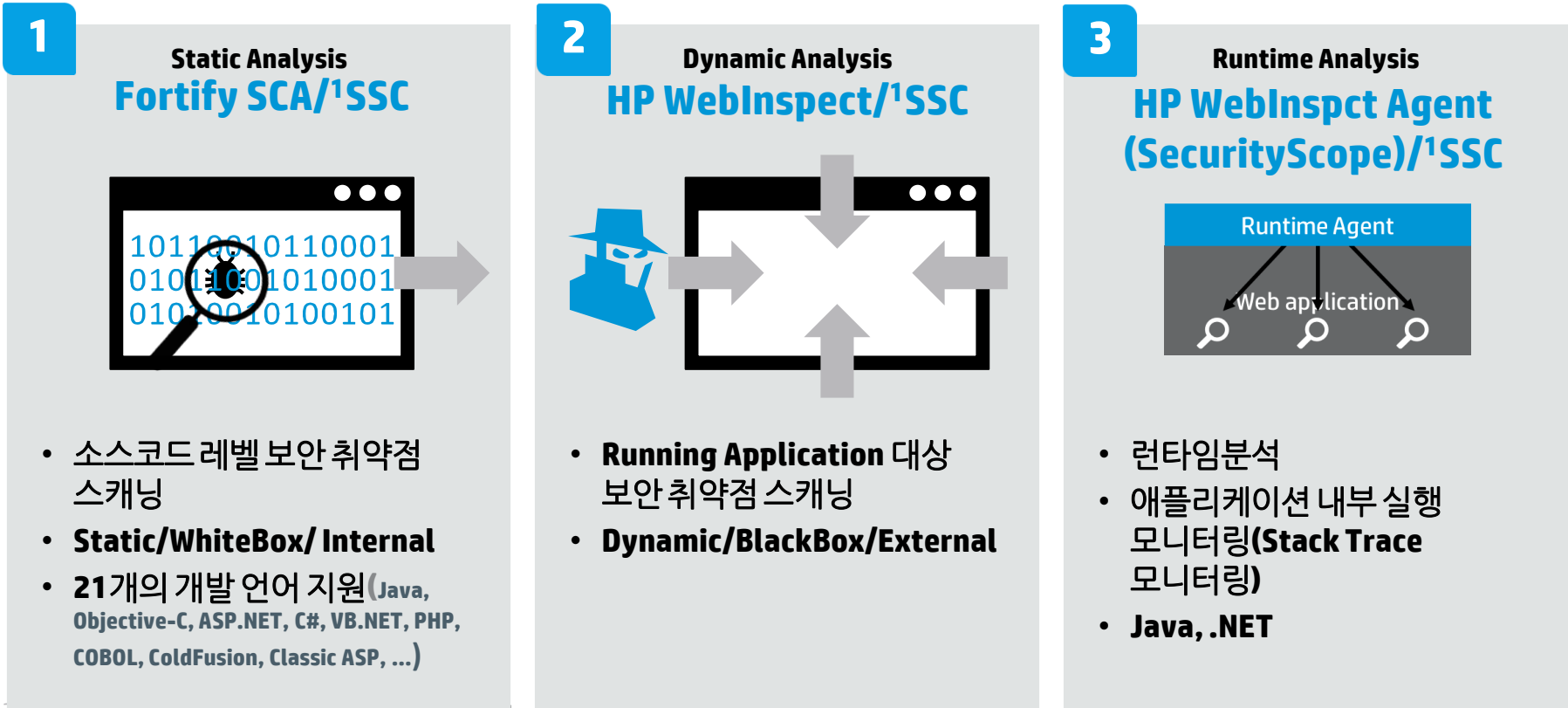


보안취약점 I	보안취약점 II	보안취약점 III
정적분석	1 런타임분석	+ 동적분석
<ul style="list-style-type: none"> <li>• Category : SQL Injection</li> <li>• Package : path8.subdir</li> <li>• Class : Cart</li> <li>• Filename : Cart.java</li> <li>• Line : 359</li> <li>• ID : 482</li> </ul>	<ul style="list-style-type: none"> <li>• Category : SQL Injection</li> <li>• Package : path8.subdir</li> <li>• Class : Cart</li> <li>• Filename : Cart.java</li> <li>• Line : 359</li> <li>• ID : 297</li> </ul>	<ul style="list-style-type: none"> <li>• Category : SQL Injection</li> <li>• ID : 297</li> </ul>
(1) 보안 취약점 그룹핑	(2) 보안 취약점 상관관계 분석	(3) 상관분석된 보안취약점 위험도별 분류 및 소스코드 레벨 취약점 제거방법 제공



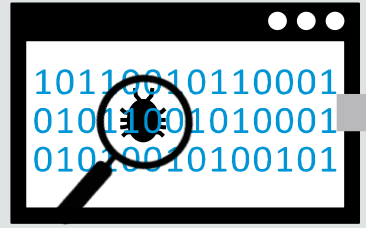
# Hybrid 2.0 보안 취약점 분석

## 구성요소



1

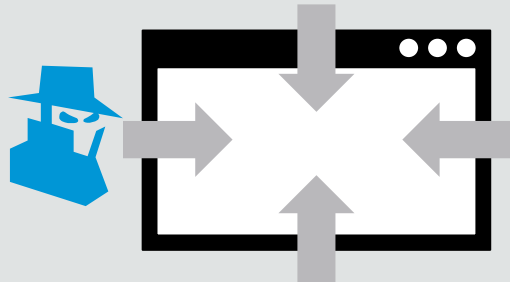
### Static Analysis Fortify SCA/SSC



- 소스코드 레벨 보안 취약점 스캐닝
- **Static/WhiteBox/Internal**
- 21개의 개발 언어 지원 (Java, Objective-C, ASP.NET, C#, VB.NET, PHP, COBOL, ColdFusion, Classic ASP, ...)

2

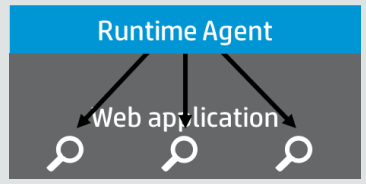
### Dynamic Analysis HP WebInspect/SSC



- **Running Application** 대상 보안 취약점 스캐닝
- **Dynamic/BlackBox/External**

3

### Runtime Analysis HP WebInspect Agent (SecurityScope)/SSC



- 런타임분석
- 애플리케이션 내부 실행 모니터링 (Stack Trace 모니터링)
- **Java, .NET**

Copyright © 2014 armr Company. All rights reserved. Other trademarks are registered trademarks and the properties of their respective owners.

Note : SSC (Software Security Center) : 정적/동적/런타임 분석기에 의해 수집된 보안취약점 관리/상관분석 제공 시스템

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### 동작원리 준비

#### 대상 애플리케이션 :

- **OWASP WebGoat 5.4 ( Security Practice를 위해 의도적으로 보안취약점을 탑재한 Open Source )**

#### 단위 보안 취약점 분석 결과물 :

- 정적분석결과(Static) : **HP Fortify SCA**
- 동적분석결과(Dynamic) : **HP WebInspect**
- 런타임분석결과(Runtime) : **HP WebInspect Agent (SecurityScope)**

#### Fortify Software Security Center(SSC) :

- 정적/동적/런타임 분석기에 의해 수집된 보안취약점 관리 분석 (상관분석) 시스템

# Hybrid 2.0 보안 취약점 분석

## 동작원리

프로젝트 생성 : *Runtime Hybrid 2.0*

1

The screenshot displays the HP Fortify Software Security Center interface. On the left, a sidebar shows the 'Projects' section with a search filter and a table containing one record for 'WebGoat' with version '5.4'. An 'Add' button is highlighted in a blue box. The main area shows the 'Create Project Version' dialog box. The 'Project' section has 'Name' set to 'Runtime Hybrid 2.0'. The 'Version' section has 'Name' set to '2.0'. Below the dialog, the 'Projects' page is visible, showing a table with 2 records found. The table has columns for 'Version' and 'State'. The first record is 'Runtime Hybrid 2.0' with version '2.0' and state 'No analysis results exist'.

Version	State
Runtime Hybrid 2.0	No analysis results exist

# Hybrid 2.0 보안 취약점 분석

동작원리

SCA(정적), SecurityScope(런타임), 동적(WebInspect) 단위 보안점검결과 업로드

2

**Static/Dynamic/Runtime 개별 보안점검결과 업로드**

**Analysis Results**  
3 records found

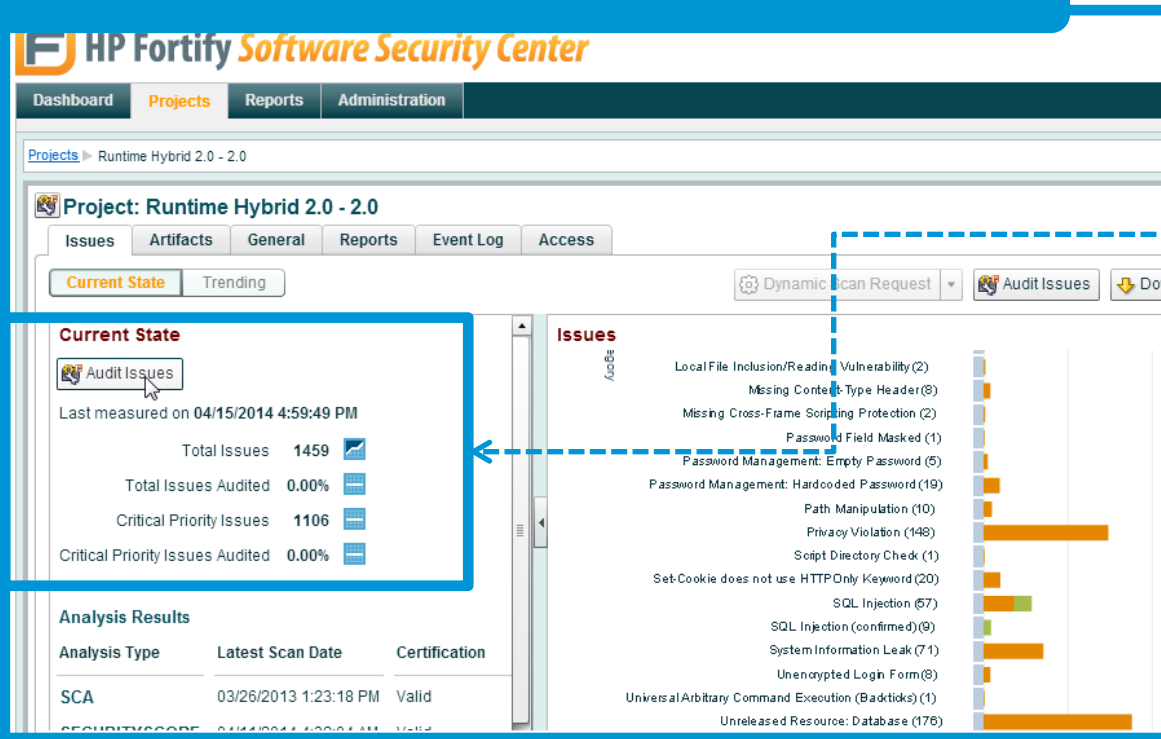
Upload Date	Uploaded By	SCA	SecurityScope	WebInspe	Othe	Audit	Status
04/15/2014 4:58:24 PM	admin (default user,...	●					Processing Complete
04/15/2014 3:53:24 PM	admin (default user,...		●				Processing Complete
04/15/2014 3:52:02 PM	admin (default user,...			●			Processing Complete

# Hybrid 2.0 보안 취약점 분석

동작원리

## Audit Issues ( WebGoat 프로젝트 보안취약점 현황 개요 )

3



전체 보안 취약점 1459개



# Hybrid 2.0 보안 취약점 분석

동작원리

## Audit Issues ( 보안취약점 Correlation )

3

Welcome admin  
[Logout](#) | [Account](#) | [Preferences](#) | [eLearning](#) | [About](#)

Dashboard Projects Reports Administration

Projects > Runtime Hybrid 2.0 - 2.0 > Issue List

Filter Set: Developer View  Issues for: admin

Critical (1106) High (284) Medium (21) Low (48) All (1459)

Critical (1106)

Group By: Correlated

Select: [All](#) [None](#)

false - [0 / 1055]

true - [0 / 51]

Issues: 1106 issues

Select item and... View Details

Primary Location	Category	Analysis Type
AbstractLesson.java:534	Cross-Site Scripting: Reflected	SCA
AbstractLesson.java:769	Cross-Site Scripting: Reflected	SCA
AbstractLesson.java:769	Cross-Site Scripting: Reflected	SCA
AbstractLesson.java:771	Cross-Site Scripting: Reflected	SCA
AbstractLesson.java:771	Cross-Site Scripting: Reflected	SCA
attack	Cross-Site Scripting	WEBINSPECT
attack	SQL Injection (confirmed)	WEBINSPECT
attack	Cross-Site Scripting	WEBINSPECT
attack	Cross-Site Scripting	WEBINSPECT
attack	Cross-Site Scripting	WEBINSPECT
attack	Cross-Site Scripting	WEBINSPECT

Search

[View Options](#)

**보안 취약점 Correlation ( Group By : Correlated )**

- true – Correlated Issues
- false – uncorrelated issues

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

[Logout](#) | [Account](#) | [Preferences](#)
[Dashboard](#) | [Projects](#) | [Reports](#) | [Administration](#)
[Projects](#) > [Runtime Hybrid 2.0 - 2.0](#) > Issue List

 Filter Set Developer View  Issues for: admin

Critical (1106) | High (284) | Medium (21) | Low (48) | All (1459)

 Critical (3 of 1106)

 Group By Correlation Group

 Select: [All](#) [None](#)

- Correlation Group 1 - [0 / 3]
- Correlation Group 10 - [0 / 3]
- Correlation Group 11 - [0 / 3]
- Correlation Group 12 - [0 / 1]
- Correlation Group 2 - [0 / 2]
- Correlation Group 3 - [0 / 18]
- Correlation Group 4 - [0 / 4]
- Correlation Group 5 - [0 / 3]

 Search 
[View Options](#)

### Correlation Group 1

#### • WebInspect + SCA + SECURITYSCOPE

Issues

3 issues in current search filter

 Select item and... [View Details](#)

	Primary Location	Category	Analysis Type
<input type="checkbox"/>	attack	SQL Injection (confirmed)	<span style="color: yellow;">●</span> WEBINSPECT
<input type="checkbox"/>	ThreadSafetyProblem.java:98	SQL Injection	<span style="color: blue;">●</span> SCA
<input type="checkbox"/>	ThreadSafetyProblem.java:98	SQL Injection	<span style="color: cyan;">●</span> SECURITYSCOPE

### 보안취약점 Correlation ( Group By : Correlation Group )

- 관련있는 보안 취약점 그룹핑 : 정적, 런타임, 동적 분석분석 결과

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

Logout | Account | Preferences

Dashboard Projects Reports Administration

Projects > Runtime Hybrid 2.0 - 2.0 > Issue List

Filter Set Developer View  Issues for: admin

Critical (1106) High (284) Medium (21) Low (48) All (1459)

Critical (3 of 1106)

Group By Correlation Group

Select: [All](#) [None](#)

- Correlation Group 1 - [0 / 3]
- Correlation Group 10 - [0 / 3]
- Correlation Group 11 - [0 / 3]
- Correlation Group 12 - [0 / 1]
- Correlation Group 2 - [0 / 2]
- Correlation Group 3 - [0 / 18]
- Correlation Group 4 - [0 / 4]
- Correlation Group 5 - [0 / 3]
- Correlation Group 6 - [0 / 3]

Search

[View Options](#)

Issues

3 issues in current search filter

Select item and...

	Primary Location	Category	Analysis Type
<input type="checkbox"/>	attack	SQL Injection (confirmed)	WEBINSPECT
<input type="checkbox"/>	ThreadSafetyProblem.java:98	SQL Injection	SCA
<input type="checkbox"/>	Includes correlated issues ThreadSafetyProblem.java:98	SQL Injection	SECURITYSCOPE

개별 취약점 상세정보를 확인 위해서는 **Double Click**

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

### 정적분석 SCA

- ThreadSafetyProblem.java : 98
- SQL Injection (Input Validation and Representation, Data flow)

The screenshot displays a security analysis tool interface. At the top, there are navigation tabs: Dashboard, Projects, Reports, and Administration. The main content area shows an audit issue for 'ThreadSafetyProblem.java:98'. The issue title is 'SQL Injection (Input Validation and Representation, Data flow)'. The description states: 'On line 98 of ThreadSafetyProblem.java, the method createContent() invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.' The analysis trace shows the flow from the method call to the SQL query construction and execution. The code snippet shows a conditional statement where a user input is concatenated into a SQL query string. The issue is highlighted with a red box, and the corresponding code lines are also highlighted. The bottom section of the interface includes a 'Details' tab, an 'Abstract' section with the same description, and an 'Explanation' section. The bottom left corner has a 'Comments' section with a 'Click to add comment' button.

Dashboard Projects Reports Administration

Projects Runtime Hybrid 2.0 - 2.0 Issue List ThreadSafetyProblem.java:98

ThreadSafetyProblem.java:98

Issue 1 of 3 Filter Set: Security Auditor View, Folder:

**SQL Injection (Input Validation and Representation, Data flow)**

On line 98 of ThreadSafetyProblem.java, the method createContent() invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Analysis Trace

File: tomcat/webapps/WebGoat/WEB-INF/classes/org/owasp/webgoat/lessons/ThreadSafetyProblem.java

```

90     if (!"".equals(currentUser))
91     {
92         Thread.sleep(1500);
93
94         // Get the users info from the DB
95         := (8) Assignment to query
96         String query = "SELECT * FROM user_system_data WHERE user_name = '" + currentUser + "'";
97         Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
98                                     ResultSet.CONCUR_READ_ONLY);
99         * (9) executeQuery(0)
100        ResultSet results = statement.executeQuery(query);

```

Details Recommendations History Correlated Issues Screenshots

**Abstract:**

On line 98 of ThreadSafetyProblem.java, the method createContent() invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

**Explanation:**

User: Not Set

Analysis: Not Set

Comments

Click to add comment

# Hybrid 2.0 보안 취약점 분석

동작원리

## Audit Issues ( 보안취약점 Correlation ) 3

### 런타임분석 SecurityScope

- ThreadSafetyProblem.java : 98
- Rule ID: 9B5F0161-88EC-4104-B70B-0182FEB53BF2
- Taint Flags: WEB, XSS
- Direct Function Call: **java.sql.Statement.executeQuery()**

The screenshot displays the Fortify Hybrid 2.0 interface. The top section shows the source code for `ThreadSafetyProblem.java` at line 98, where `executeQuery` is called. A red box highlights this call, and a blue arrow points from the `SecurityScope` callout to it. The bottom section shows the 'Correlated Issues' tab, which displays a request log with the following details:

Request	Function Call
Path: /attack Referer: http://172.16.100.20:8080/WebGoat-5.4/attack?Screen=601&menu=800 Method: POST Parameters: Screen:601 Cookies: JSESSIONID=9C9FEEE6264F9ACC9513C2A412569B57	org.hsqldb.jdbc.jdbcStatement.executeQuery() Triggers: SELECT * FROM user_system_data WHERE user_name = " OR'

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

### 동적분석 WebInspect

- SQL Injection (confirmed) (Input Validation and Representation, pentest)

Projects > Runtime Hybrid 2.0 - 2.0 > Issue List > attack

File: /WebGoat-5.4/attack

URL `http://172.16.100.20:8080/WebGoat-5.4/attack`

Method `POST`

Vulnerable Parameter `username`

Attack Payload `username: %27%09OR`

Request Response Stack Trace

Auto-scroll  Wrap Text

```
X-RequestManager-Memo: StateID="36"; ID="a5040421-f5d5-418a-97ff-13cc2e7116a5";
X-Request-Memo: ID="620417d2-d95a-456a-9b5e-06b6bfc08797"; ThreadId="65";
Cookie: JSESSIONID=9C9FEEEE6264F9ACC9513C2A412569B57

username=%27%09OR&SUBMIT=Submit
```

User `Not Set`

Analysis `Not Set`

Comments

Click to add comment

Details Recommendations History Correlated Issues Steps Screenshots

Location	Type
attack	WEBINSPECT
ThreadSafetyProblem.java:98	SECURITYSCOPE
ThreadSafetyProblem.java:98	SCA

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

File: /WebGoat-5.4/attack

URL **http://172.16.100.20:8080/WebGoat-5.4/attack**

Method **POST**

Vulnerable Parameter **username**

Attack Payload **username: %27%09OR**

**동적분석 WebInspect**

- **HTTP/1.1 200 OK**

Request **Response** Stack Trace

HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
X-WIPP-Version: java / 1.3 / WIN-1BICR1EIJ7G\_2776  
X-WIPP-RequestID: 620417d2-d95a-456a-9b5e-06b6bfc08797  
X-WIPP-Update: Application  
X-WIPP-User-Logged-On: true  
Content-Type: text/html;charset=ISO-8859-1  
Date: Thu, 10 Apr 2014 13:15:49 GMT  
Content-Length: 30473

Auto-scroll  Wrap Text

# Hybrid 2.0 보안 취약점 분석

## 동작원리

### Audit Issues ( 보안취약점 Correlation )

3

File: /WebGoat-5.4/attack

URL **http://172.16.100.20:8080/WebGoat-5.4/attack**Method **POST**Vulnerable Parameter **username**Attack Payload **username: %27%09OR**

### 동적분석 WebInspect

- StackTrace
- SQL Injection (Input Validation and Representation, runtime)

Request Response **Stack Trace**

#### SecurityScope Trigger:

```
SELECT * FROM user_system_data WHERE user_name = 'tOR'
```

#### SecurityScope Stack Trace:

```
at org.hsqldb.jdbc.jdbcStatement.executeQuery(Unknown Source)
at org.owasp.webgoat.lessons.ThreadSafetyProblem.createContent(ThreadSafetyProblem.java:98)
at org.owasp.webgoat.lessons.AbstractLesson.handleRequest(AbstractLesson.java:771)
at org.owasp.webgoat.lessons.ThreadSafetyProblem.handleRequest(ThreadSafetyProblem.java:189)
at org.owasp.webgoat.HammerHead.makeScreen(HammerHead.java:367)
at org.owasp.webgoat.HammerHead.doPost(HammerHead.java:146)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:643)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:723)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
```



# Hybrid 2.0 보안 취약점 분석

동작원리

## Audit Issues ( 보안취약점 Correlation - All ) 3

Filter Set Developer View  Issues for: admin

Critical (1106) High (284) Medium (21) Low (48) All (1459)

All (1459)

Group By Correlation Group

Select: [All](#) [None](#)

- Correlation Group 1 - [0 / 3]
- Correlation Group 10 - [0 / 3]
- Correlation Group 11 - [0 / 3]
- Correlation Group 12 - [0 / 2]
- Correlation Group 2 - [0 / 2]
- Correlation Group 3 - [0 / 18]
- Correlation Group 4 - [0 / 4]
- Correlation Group 5 - [0 / 3]
- Correlation Group 6 - [0 / 3]

Search

[View Options](#)

Filter Set Developer View  Issues for: admin

Critical (1106) High (284) Medium (21) Low (48) All (1459)

All (1459)

Group By Correlated

Select: [All](#) [None](#)

- false - [0 / 1407]
- true - [0 / 52]

Search

[View Options](#)

전체 1459개의 보안취약점 중 52개는 상관관계분석된 보안취약점 그룹으로서 점검 우선순위 1순위로 분류, 소스코드상의 보안 취약점 수정 필요

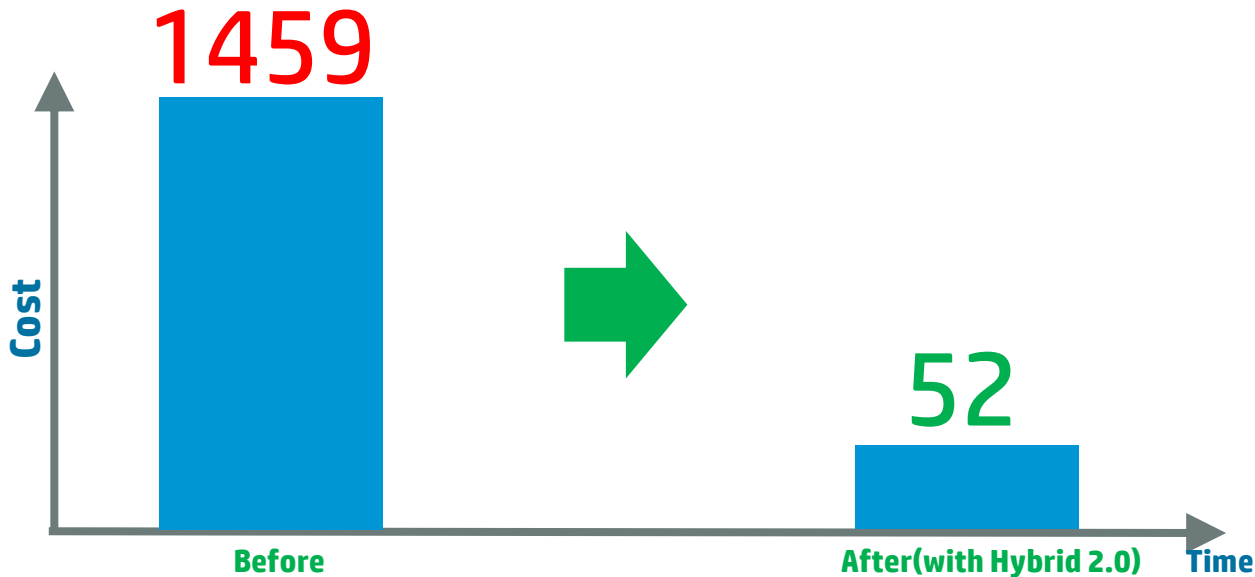


# Hybrid 2.0 보안 취약점 분석

기대효과

## Hybrid 2.0 분석기법 H

- 소스코드 분석결과와 동적/런타임 분석결과를 연계, 보안취약점에 대한 우선순위화/근원적인 보안취약점 제거 방안 제공합니다.



***Find, Fix and Fortify***

**Thank You**

보안솔루션 관련 문의처: [espkorea@hp.com](mailto:espkorea@hp.com)

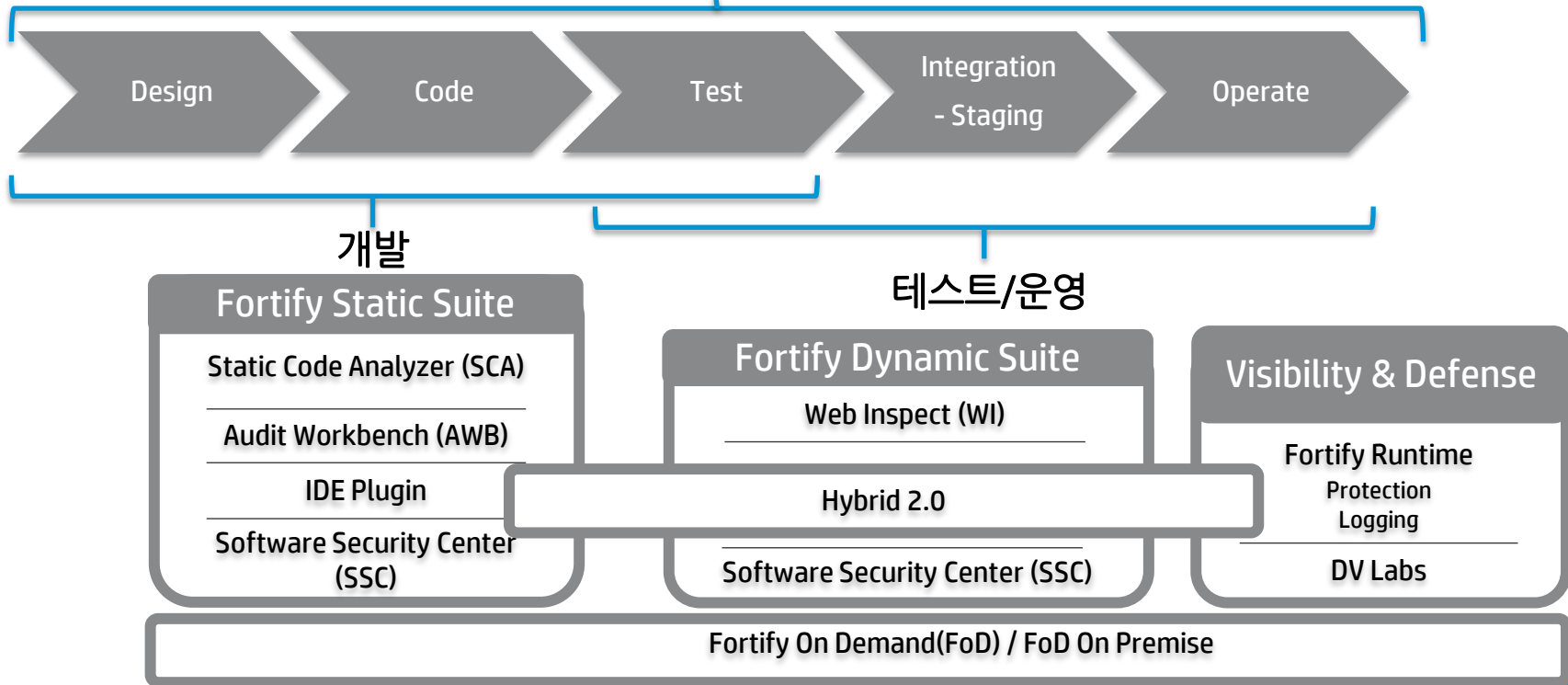
한국HP | 보안사업부(Enterprise Security Products)



# 소프트웨어 보안 품질 보증

포티파이 애플리케이션 보안 취약점 관리 솔루션

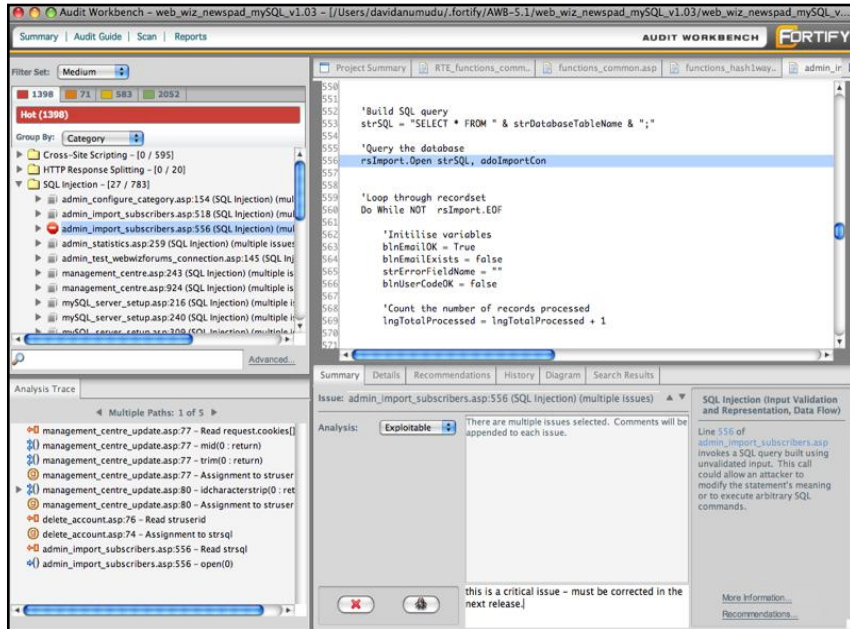
소프트웨어 보안



# Fortify Static Code Analyzer (SCA)

정적분석기/Static analysis - 소스코드 보안 취약점 발견 및 조치 가이드

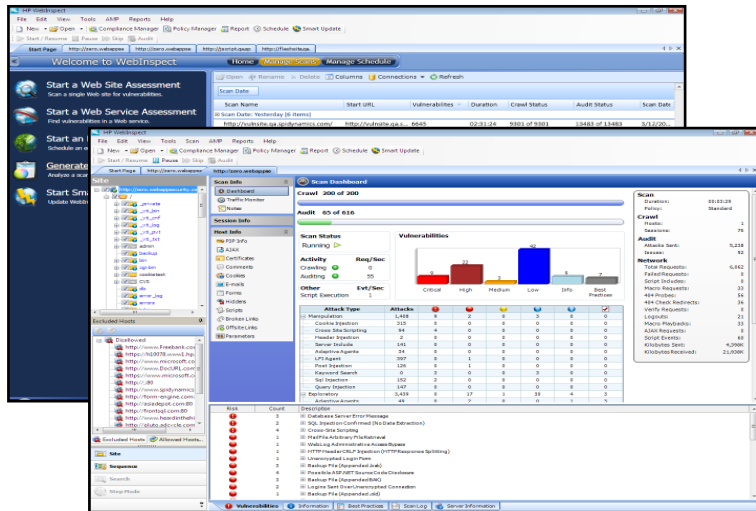
- 소스코드 정적보안취약점 분석 **De facto standard** 제품
- **500개 이상** 보안카테고리 및 업계최다인 **21개의** 개발 언어 대상 코드레벨 정적 보안 취약점 분석지원 (**Java7, HTML5 포함**)
- 업계 유일 **Mobile app** 개발 언어 분석지원 (**Apple Objective-C/Xcode, Android Java** 모두 지원)
- 업계유일 **Hybrid** 분석 기술(**SecurityScope**) 채용을 통해 **Static/Dynamic** 에만 국한된 분석기법의 한계 극복, 보안취약점에 대한 우선순위기반 보안취약점 관리 제공
- **On-Premise/On-Demand** 솔루션지원 통한 유연성 제공



# WebInspect

## 동적 분석기/Dynamics analysis - 웹애플리케이션 보안취약점 발견 및 조치 가이드

- 네트워크를 통한 웹서버 대상 보안 스캐닝을 수행해 짧은 시간안에 고위험도 웹애플리케이션 보안취약점 탐지 및 조치 가이드
- 개발 프로젝트 진행, 보안검수시점과 운영 중 웹보안품질 측정을 실시해 보안취약점에 대한 상시 점검
- **1Hybrid 분석 기술을 채용해 소스코드 분석결과와 동적 분석결과를 연계, 보안취약점에 대한 우선순위기반 보안취약점관리방안 제공(SecurityScope)**



1Hybrid 분석 기술: 정적분석기법인 소스코드 분석(Static code analysis)과 런타임 분석 기법인 프로그램 실행분석(SecurityScope), 동적분석기법인 웹 보안취약점 스캐너인 WebInspect를 상호 연계, 애플리케이션 보안취약점에 대해 통합분석함으로써, 동적분석의 한계인 분석 범위를 확장하고 정적분석결과를 실제 공격이 가능한 순으로 분류하여 분석결과를 제공하므로 위험도와 실행 가능성이 높은 보안취약점을 빠르게 식별 및 우선적으로 수정할 수 있도록하여 전체 개발비용과 소요시간을 급격히 절감할 수 있는 신기술

Dynamic Application Security Testing (DAST), Static Application Security Testing (SAST), Interactive Application Security Testing (IAST)



# WebInspect with 런타임 에이전트

Interactive Application Security Testing 기법 채용한 차세대 동적 분석기

**Parameters**

**Line of Code**

**Stack Trace**

**Confirmed SQL Injection**

HP WebInspect

File Edit View Tools Scan AMP Reports Help

New Open Compliance Manager Policy Manager Report Schedule

vulnerability Stack Traces

Start / Resume Pause Skip Audit Rescan

Start Page Riches with Security

Site

Scan Info

Dashboard

Transfers

SendView

UploadPr

PerformChar

PerformChec

PerformTran

Security.jsp

ShowLocatio

titles

Transfer

ViewMes

web

profilepictu

rwi-1.swf

Security.ac

ShowLocatio

(Post) ad

(Post) ad

(Query) err

(Query) err

(Query) err

(Query) err

(Query) err

(Query) que

Excluded Hosts Allowed Hosts

Site

Sequ

Search

Step

SQL Injection (confirmed)

This stack trace is from the running application and was returned by SecurityScope. It can be used to identify the root cause of the vulnerability.

**SecurityScope Trigger:**

```
SELECT * FROM location WHERE zip = '30346'\tOR'
```

**SecurityScope Stack Trace:**

```
at org.apache.tomcat.dbcp.dbcp.PoolingDataSource$PoolGuardConnectionWrapper.prepareStatement(
----- start of user application code -----
at com.fortify.samples.riches.model.LocationService.findByZip(LocationService.java:110)
at com.fortify.samples.riches.FindLocations.execute(FindLocations.java:45)
sor64.invoke(Unknown Source)
sorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
ethod.java:597)
tActionInvocation.invokeAction(DefaultActionInvocation.java:404)
at com.opensymphony.xwork2.DefaultActionInvocation.invokeActionOnly(DefaultActionInvocation.java:
at com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:229)
ot(DefaultWorkflo
odFilterInterce
onInvocation
onInvocation
```

Path

Check:SQL Injection (confi

Duplicates:SQL Injection (confi

http://riches.spidynamics.com/riches/pages/common/hidden\_AdminControl.jsp(1 iter

for=%2bnull&zip=30346%09OR

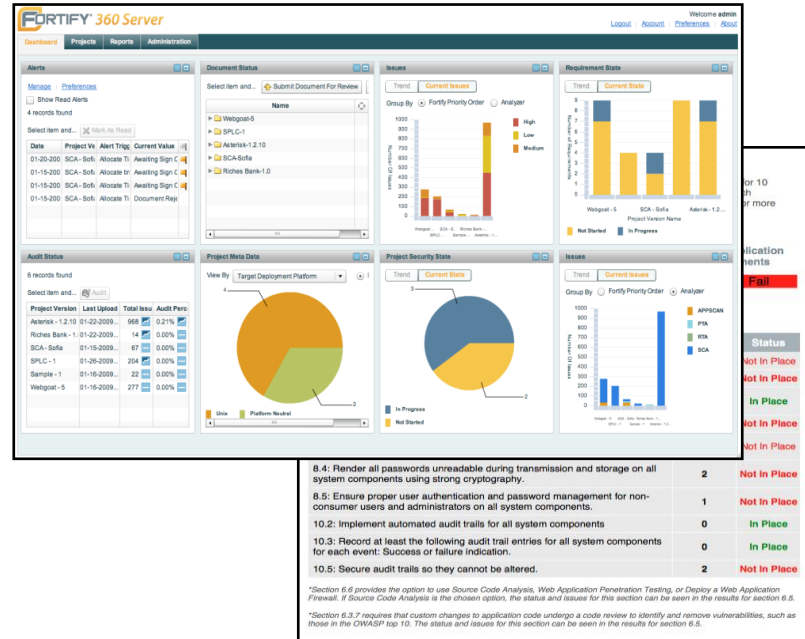
locationGo.y=5&

Duplicates:Universal Arbitrary Command Execution (Pipe/Ampersand/Double Quote) - http://riches.spidynamics.com/riches/pages/common/hidden\_AdminControl.jsp(1 iter

# Fortify Software Security Center server

## 총체적인 보안 취약점 관리 센터

- 정적/동적/런타임 분석기에 의해 수집된 보안취약점 관리/연동분석/추적/제거 방안 제공 시스템
- 역할 기반 접근, 프로세스 기반의 보안취약점 관리
- 핵심개발환경과의 연동
  - Build integration, Defect tracking, Source control, 3rd party analysis engines
- 유연한 보안취약점 저장소 관리 및 리포팅 제공
  - Normalized, correlated vulnerability repository
  - Aggregated risk metrics





# Mobile Application Security Testing

모바일 인프라 전반에 걸친 애플리케이션 스택 보안취약점 발견 및 조치

## 모바일 지원 플랫폼

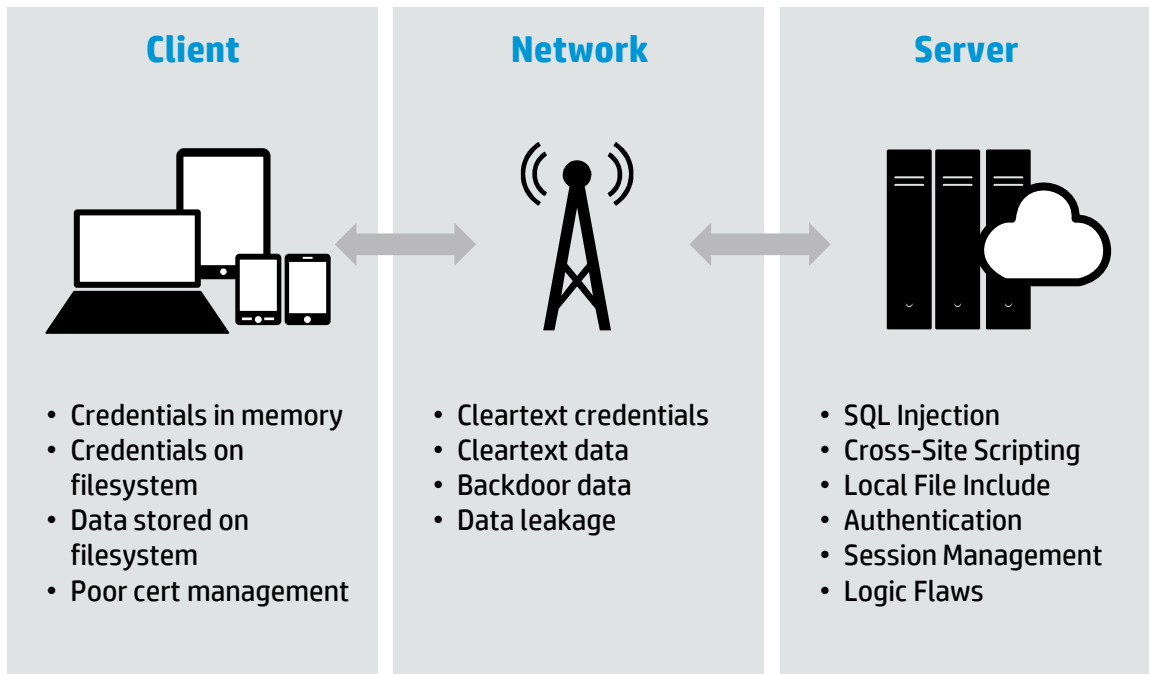
- Apple iOS (**Objective C**)
- Android
- Windows Phone / Blackberry

## Hybrid 분석 기술 채용

- 소스코드 분석결과와 동적분석결과와의 **Correlation**

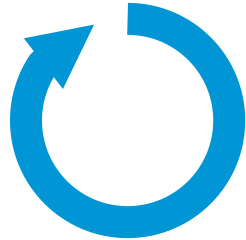
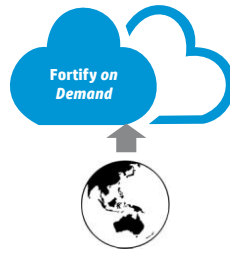
## 모바일 인프라 3-Tier 지원

- 모바일 단말상의 애플리케이션
- 모바일 백엔드 서버상의 애플리케이션



# HP Fortify On Demand

애플리케이션 보안, 적은 비용으로 간단하게 빠르게 유연하게!



## Simple



## Fast



## Flexible

**1일 이내 Application** 보안 시작

- 하드웨어, 소프트웨어 **先투자 필요 - No!**
- 보안 전문인력 필요 - **No!**

신속, 뛰어난 확장성 제공

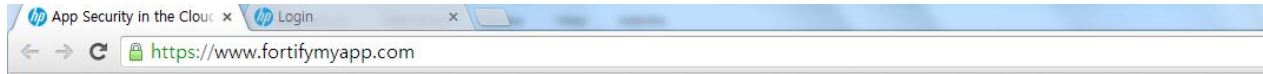
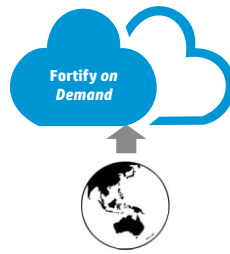
- **Application** 보안 테스트 결과 - **최단기간 소요**
- 데스크탑, **모바일** 혹은 **클라우드** 상의 **1,000**여개 이상 **Application** 지원

유연한 테스트

- 상용, 오픈소스, **3rd Party** 애플리케이션 보안테스트 지원
- **On-Premise, On-Demand**, 혹은 **both** 지원



# HP Fortify On Demand



Fast, affordable, on-demand  
Quickly test the security of any soft



Presented by HP Fortify on Demand

hp App Security in the Cloud x hp Login x

← → ↻ https://www.hpod.com/login?ReturnUrl=%2fTenantData%2fApplications

SERVICES & SOLUTIONS FORTIFY BLOG SALES

hp

## Login to Fortify on Demand

Username

Password

Tenant ID

Login [Forgot Password](#)

[Don't have an account? Learn More](#)

Dashboard Applications Reports Administration

### Corporate Website | Release 1.0

Rating: ★★★★★

Category	Severity Count	Assignment	Analysis Type	Status Trend	Scan Type
Low	34	43	Medium	Low	77
High	0	0	High	High	77

Status: Complete Last Scan: 2/18/2013

Dynamic: All Inflight Last Scan: none

Start Scan = [+](#) [-](#) [Refresh](#)

0

# HP Fortify Coverage

## *Static Analysis*



## Static analysis supports 21 languages and growing

- ABAP
- Android
- C#
- Classic ASP
- Cold Fusion
- HTML
- JavaScript/AJAX
- Objective C
- PL/SQL
- T-SQL
- VB6
- XML
- ASP.NET
- C/C++
- COBOL
- Flex
- Java
- JSP
- PHP
- Python
- Ruby
- VB.NET
- VBScript

## Dynamic analysis covers all web environments

- QA or production environments
- Web services

## Mobile application security solution covers

- Objective C
- Android
- Blackberry
- Microsoft



# Gartner Magic Quadrant for Application Security Testing

- “HP offers comprehensive SAST capabilities with Fortify's strong brand name and breadth of languages tested.
- The company has innovative IAST capability with Fortify SecurityScope, which integrates with its WebInspect DAST.
- There is strong integration within HP's security portfolio, such as integration of AST knowledge into ArcSight and DAST knowledge into TippingPoint's IPS for WAF-like protection.
- HP uniquely offers runtime application self-protection (RASP) technology



As of July 2013