How safe am I? Choosing the right security tool for the job!

There is an abundance of security tools out there: network scanners, dynamic and static analysis tools, fuzzing tools, etc. However, it is not always easy to determine which tool is best at finding what, and choosing the wrong security tools can not only result in providing useless results, but also result in creating an excessive burden for the organization trying to clear “false positives.” In this session, you will learn about the many types of security tools available to your organization, how to select the proper tools, and how to use these tools in the most effective fashion to assess your on-premises and cloud security posture.

Session ID
CON6303

ORACLE

How safe am I? Choosing the right security tool for the job!

CON6303

ORACLE
OPEN
WORLD

October 1–5, 2017
SAN FRANCISCO, CA

John Heimann
Vice President – Security Program Management
Global Product Security
Oracle Corp.

ORACLE

Safe Harbor Statement

Program Agenda

1. Challenge: too many things, too little time!
2. Testing new deployments: finding misconfigurations and proper use of pen-testing
3. Testing new code: using dynamic analysis
4. Testing new code: using static analysis
5. Conclusion and call to action!

How Safe Am I?

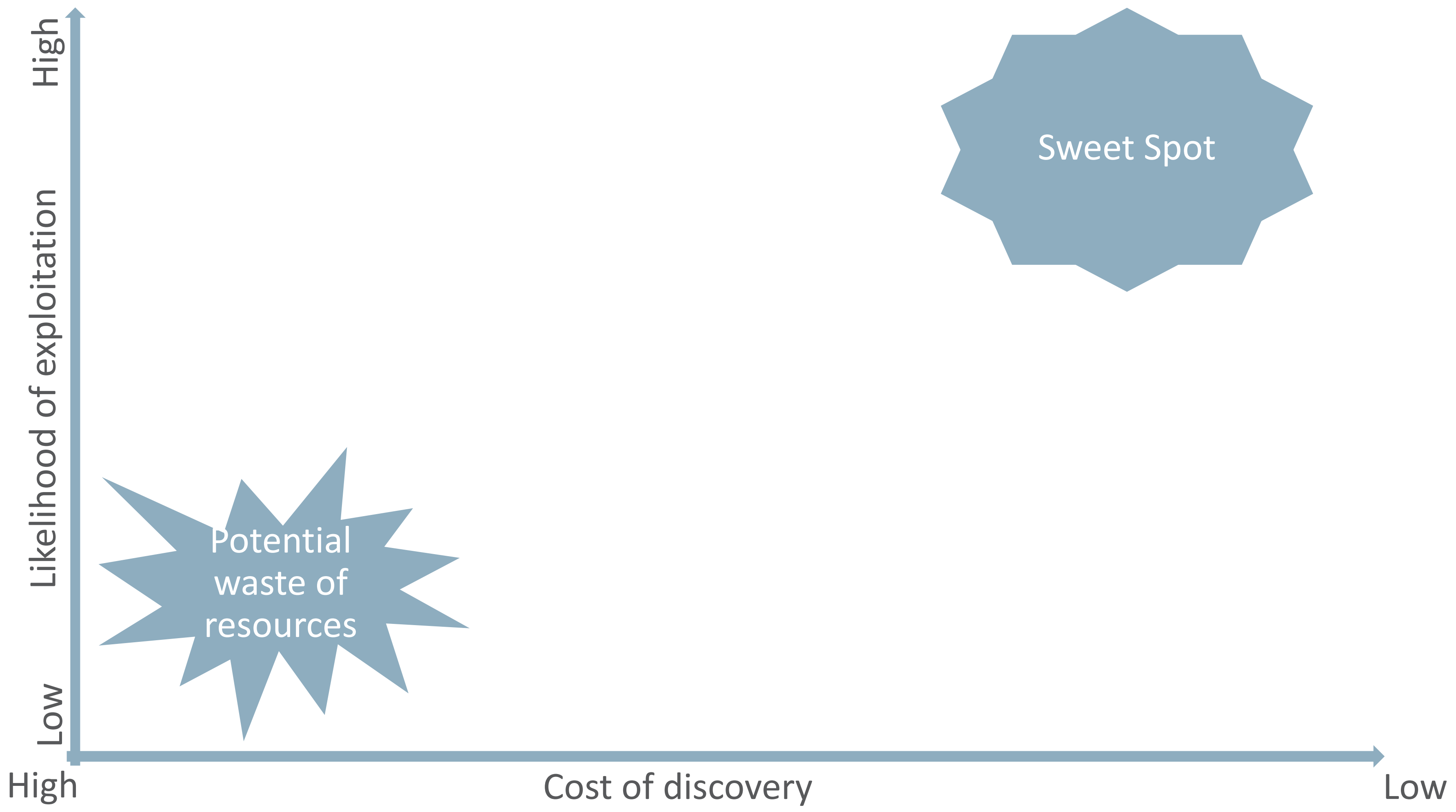
- It is very important to periodically assess your security posture…
  - Security testing should not be performed solely against new deployments
  - Periodic testing against the whole production environment will help detect “creeping problems” such as unwanted configuration changes and unpatched vulnerabilities
  - And then… Do you really know what vulnerable components may be used throughout your environment?
    - Newly-deployed and existing systems may make use of obsolete and vulnerable components (e.g., OpenSSL, GNU Bash, etc.)

How Safe Am I?....Really, How Safe Am I?

- …But production requirements are resource and time-consuming
  - Security testing takes the backseat to production
  - Time-to-market considerations often result in rushing new applications without proper testing
  - Operation staff may feel that security testing is the responsibility of the corporate security and audit staff
    - While hoping that nothing is found on the next security audit!!!

Challenge: “Getting the Best Bang for Your Bucks!”

Obviously, you want to spend your security testing resources finding likely exploitable vulnerabilities at the lowest possible cost.

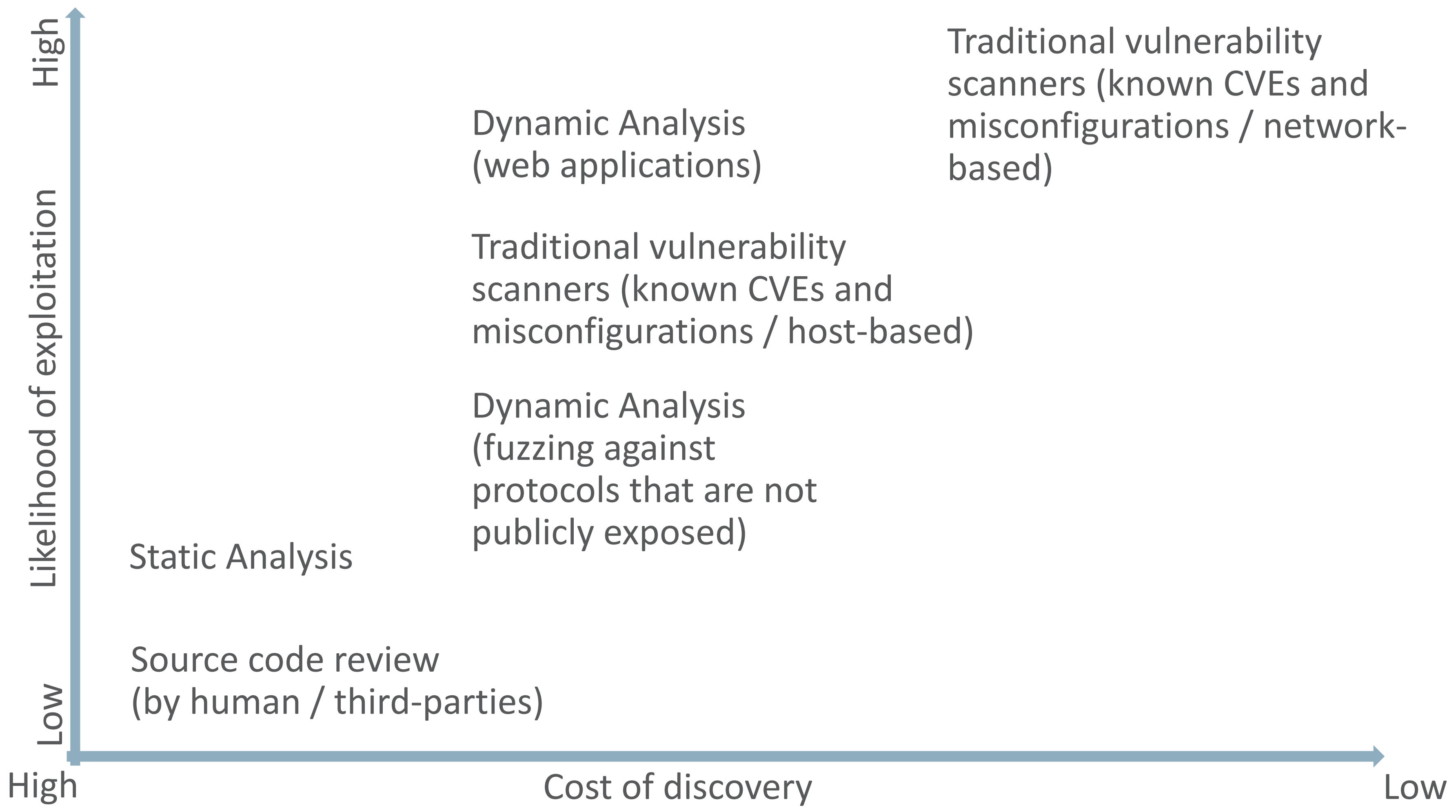


How Easily Can You Find Security Issues?

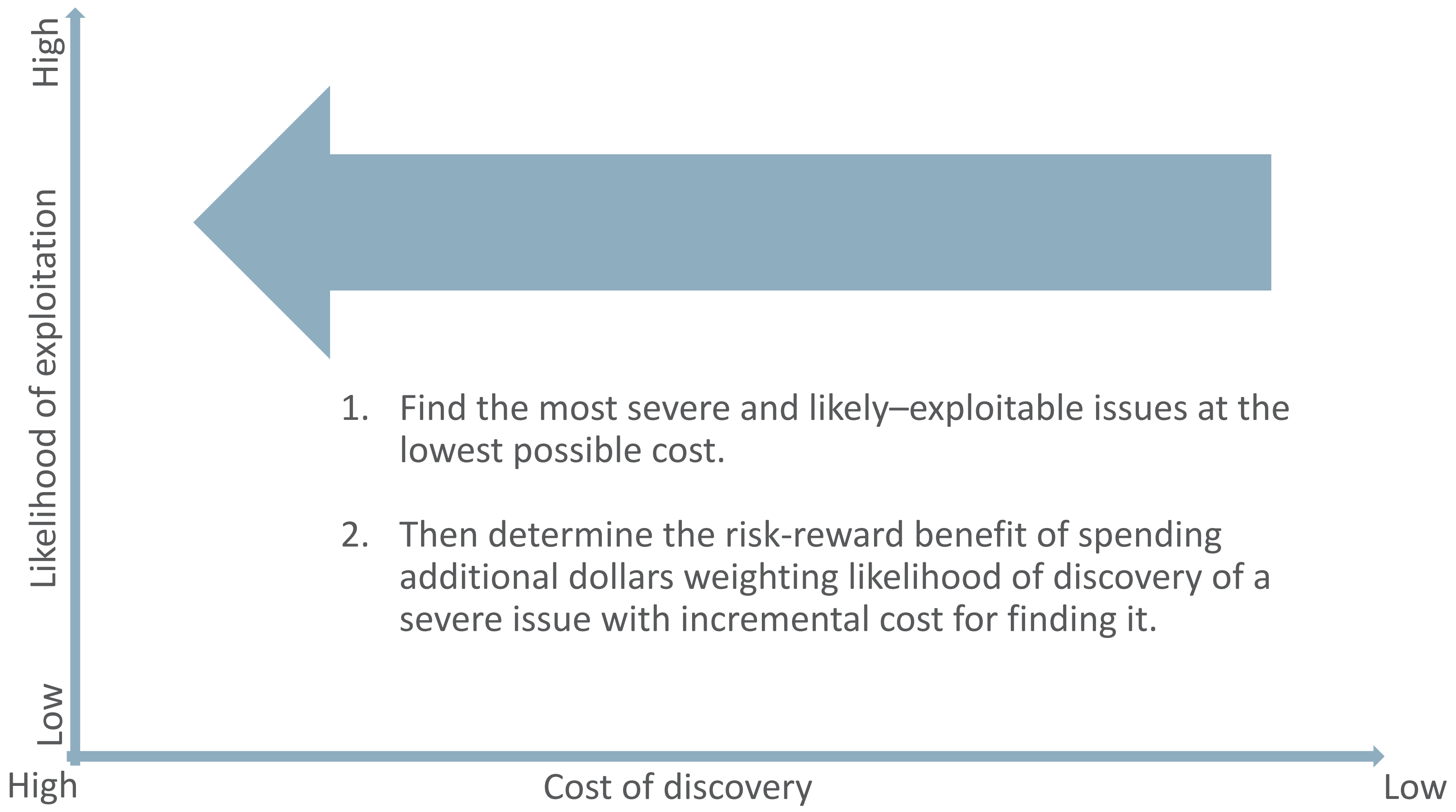
	Known vulnerabilities	**Errors and misconfigurations**	**Hidden vulnerabilities and new vectors (i.e., 0-days)**
Examples	• Unpatched vulnerabilities • Known vulnerable components • Publicly-disclosed vulnerabilities	• Default account passwords • Use of vulnerable protocols (e.g., obsolete crypto) • Improper permissions (files, user privileges)	• New exploit vectors (e.g., exploit of undocumented interfaces) • New and previously undisclosed vulnerabilities
How can they typically be found?	• Inventory scans / patch discovery • Leaked information • Network scanners	• Network scanners • Review product deployment specs. • Pen testing	• Static analysis / code review • Dynamic testing/Fuzzing • Expert "hacking" skills
Who can typically find them?	• Administrators • Auditors • Pen testers	• Administrators • Auditors • Pen testers	• Users of static/dynamic analysis tools • Expert hackers

Challenge: the Picture for Traditional IT Shop

Pen-testing and ethical hacking engagements will yield various benefits based on a number of factors.



Challenge: Prioritizing Security Testing Activities?

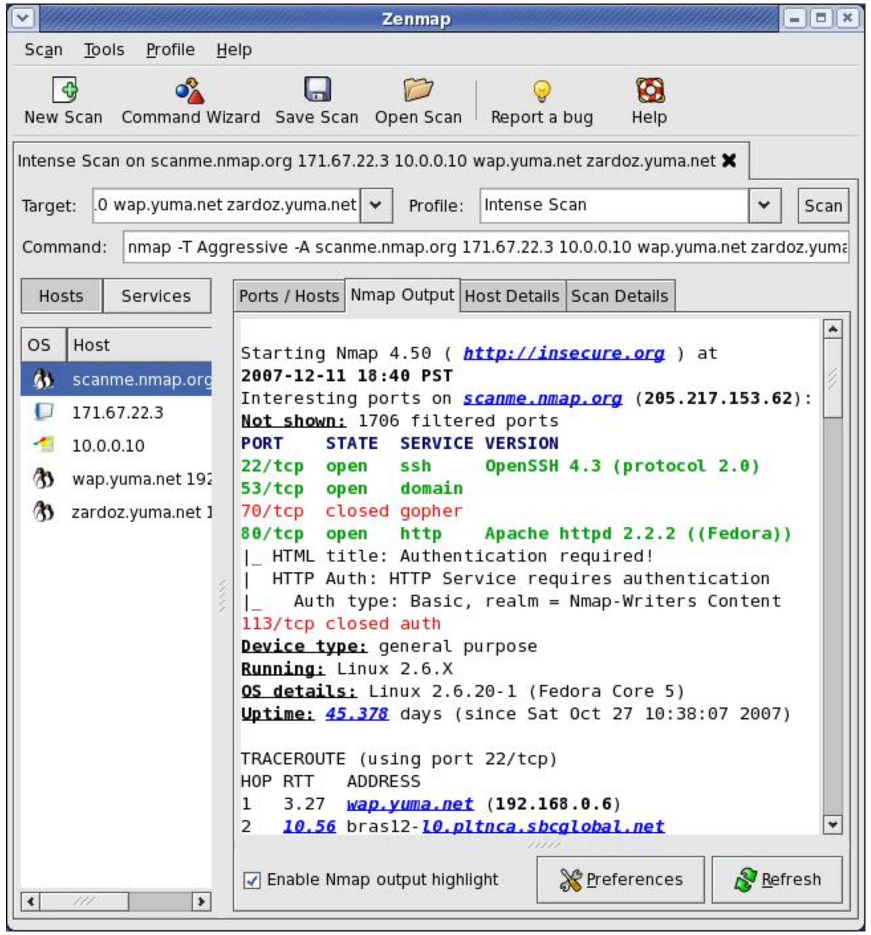


1. Find the most severe and likely–exploitable issues at the lowest possible cost.

2. Then determine the risk-reward benefit of spending additional dollars weighting likelihood of discovery of a severe issue with incremental cost for finding it.

Start With the Basics! Network Scanning Tools

- Widely-available tools (e.g., NMAP) allow script-kiddies to quickly detect vulnerable systems
- Helps define “best” payload
- … And don’t forget Google!
  - inurl:”viewerframe?mode=motion”
  - inurl:”admin.php”
  - Etc.



Intense Scan on scanme.nmap.org 171.67.22.3 10.0.0.10 wap.yuma.net zardoz.yuma.net

Target: .0 wap.yuma.net zardoz.yuma.net   Profile: Intense Scan   Scan

Command: nmap -T Aggressive -A scanme.nmap.org 171.67.22.3 10.0.0.10 wap.yuma.net zardoz.yuma

```
Starting Nmap 4.50 ( http://insecure.org ) at
2007-12-11 18:40 PST
Interesting ports on scanme.nmap.org (205.217.153.62):
Not shown: 1706 filtered ports
PORT     STATE  SERVICE VERSION
22/tcp   open   ssh     OpenSSH 4.3 (protocol 2.0)
53/tcp   open   domain
70/tcp   closed gopher
80/tcp   open   http    Apache httpd 2.2.2 ((Fedora))
|_ HTML title: Authentication required!
|  HTTP Auth: HTTP Service requires authentication
|_   Auth type: Basic, realm = Nmap-Writers Content
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.20-1 (Fedora Core 5)
Uptime: 45.378 days (since Sat Oct 27 10:38:07 2007)

TRACEROUTE (using port 22/tcp)
HOP RTT   ADDRESS
1   3.27  wap.yuma.net (192.168.0.6)
2   10.56 bras12-10.pltnca.sbcglobal.net
```

Go After Known Bugs!

- Generally, publishing security patches means:
  - Vulnerabilities become quickly known to potential attackers
    - Weaponization begins
    - “Nearby” vulnerabilities are identified
    - Weaponized attack scripts will be published with configurable payloads
- Continued use of unpatched vulnerable components creates critical risks
  - Script kiddies can attack and can select payloads
- Many “Patch Inventory Tools” can report lack of applied, published fixes in specific products distributions or their components
- Increasing numbers of industry groups demand quick deployment of published security fixes

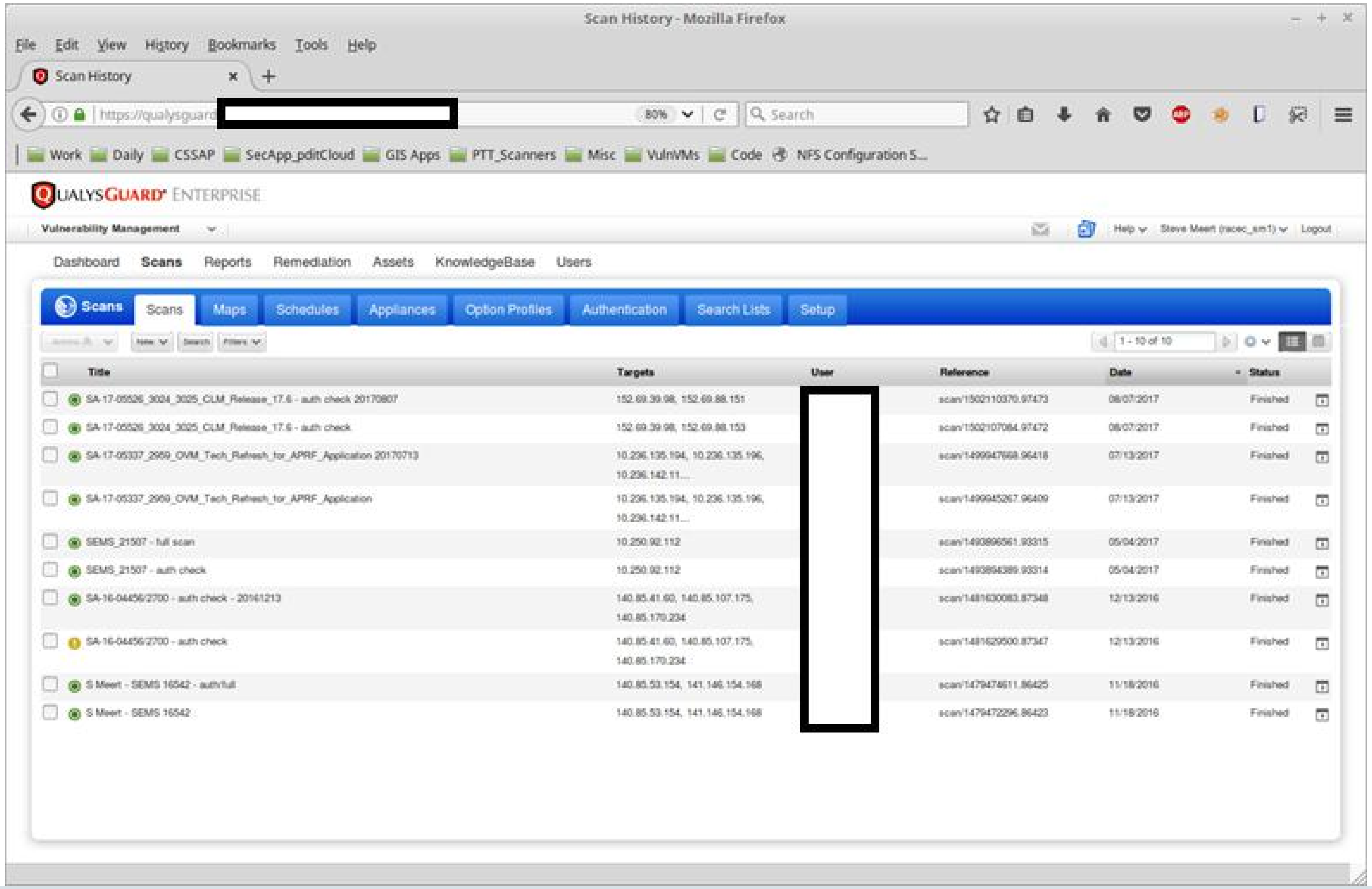
Use of Patch Inventory Tools and Facilities

- Reports of published fixes that have not been applied
  - Both full products and internal components
  - Biggest bang for the buck: Easy remediation and high risk due to public knowledge
  - Published fixes are attacker focus, especially in widely used components
- But, many patch inventory tools are new and have significant issues
  - False negatives (i.e. not comprehensive)
  - False positives due to poor identification of component release versions
  - “False positives” because component defects are not exploitable via the application

Product Misconfigurations Are Significant Risks

- After applying published security fixes, next address misconfiguration
- Bad actors read security guides to identify best attack strategies
  - Attackers know that users often fail to follow the recommended configuration advice
  - Vendors respond by providing “secure by default” deployments
  - For compatibility reasons, sites sometimes change defaults to less secure and obsolete configurations – these should be remediated as soon as possible
  - Quick and dirty configuration changes made for interoperability, quick enhancement or testing reasons often have very high risk factors
- Configuration issues can be as easily exploitable as known vulnerabilities
  - Less than secure deployments made for compatibility reasons are chronic targets
  - Sometimes the customer base must be inconvenienced (e.g. white lists)

Patch and Configuration Tools


Scan History - Mozilla Firefox

https://qualysguard[redacted]

Vulnerability Management

Dashboard Scans Reports Remediation Assets KnowledgeBase Users

Scans | Scans | Maps | Schedules | Appliances | Option Profiles | Authentication | Search Lists | Setup

1 - 10 of 10

Title	Targets	User	Reference	Date	Status
SA-17-05526_3024_3025_CLM_Release_17.6 - auth check 20170807	152.69.39.98, 152.69.88.151	[redacted]	scan/1502110370.97473	08/07/2017	Finished
SA-17-05526_3024_3025_CLM_Release_17.6 - auth check	152.69.39.98, 152.69.88.153	[redacted]	scan/1502107084.97472	08/07/2017	Finished
SA-17-05337_2959_OVM_Tech_Refresh_for_APRF_Application 20170713	10.236.135.194, 10.236.135.196, 10.236.142.11....	[redacted]	scan/1499947668.96418	07/13/2017	Finished
SA-17-05337_2959_OVM_Tech_Refresh_for_APRF_Application	10.236.135.194, 10.236.135.196, 10.236.142.11....	[redacted]	scan/1499945267.96409	07/13/2017	Finished
SEMS_21507 - full scan	10.250.92.112	[redacted]	scan/1493896561.93315	05/04/2017	Finished
SEMS_21507 - auth check	10.250.92.112	[redacted]	scan/1493894389.93314	05/04/2017	Finished
SA-16-04456/2700 - auth check - 20161213	140.85.41.60, 140.85.107.175, 140.85.170.234	[redacted]	scan/1481630083.87348	12/13/2016	Finished
SA-16-04456/2700 - auth check	140.85.41.60, 140.85.107.175, 140.85.170.234	[redacted]	scan/1481629500.87347	12/13/2016	Finished
S Meert - SEMS 16542 - auth/full	140.85.53.154, 141.146.154.168	[redacted]	scan/1479474611.86425	11/18/2016	Finished
S Meert - SEMS 16542	140.85.53.154, 141.146.154.168	[redacted]	scan/1479472296.86423	11/18/2016	Finished

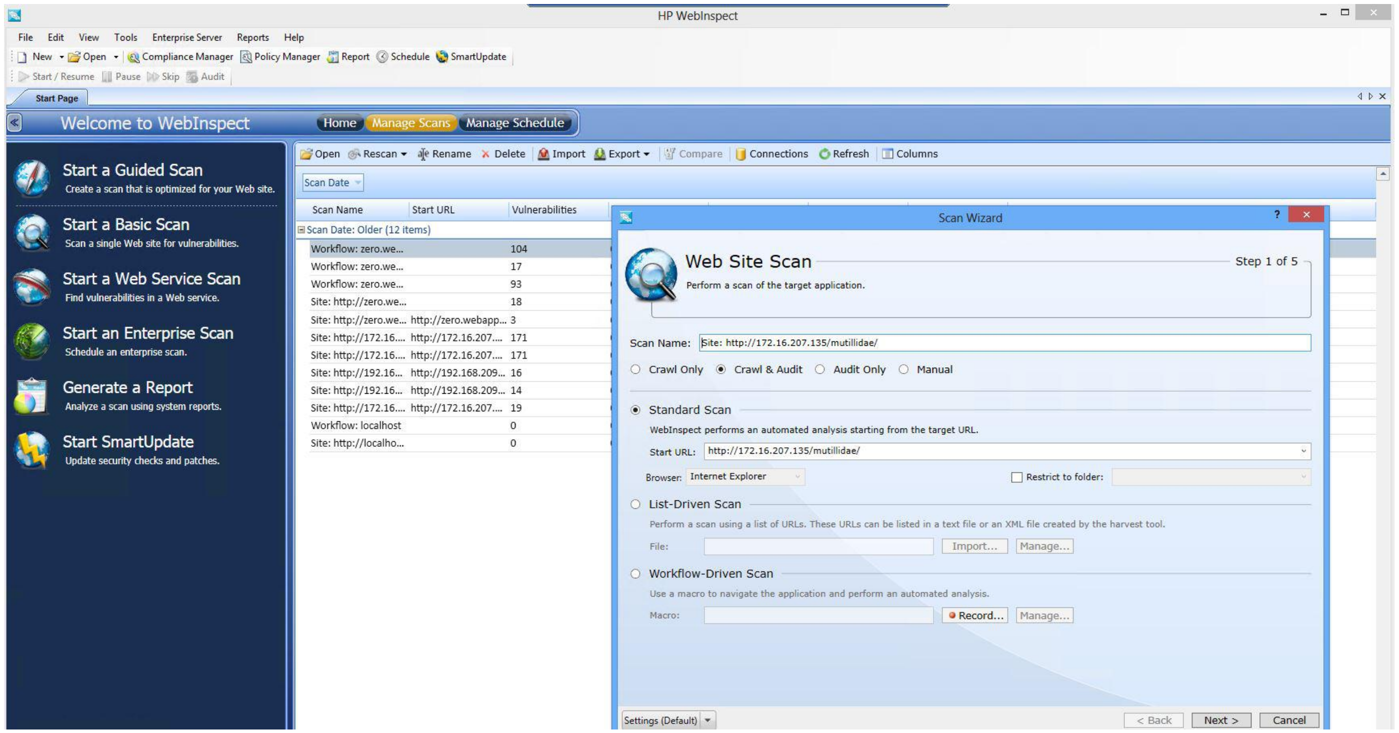
File Edit View History Bookmarks Tools Help
Scan History
Work Daily CSSAP SecApp_pditCloud GIS Apps PTT_Scanners Misc VulnVMs Code NFS Configuration S...
QUALYSGUARD ENTERPRISE
Help Steve Meert (racec_sm1) Logout
New Search Filters

A Few Words About Pen-Testing

- Prior sections discussed patch application and correcting configurations
  - All customers can apply published fixes and correct their product configurations
  - No experts required!!!
- Some customers have expertise and resources to employ comprehensive static and dynamic tests while others will rely on pen-testers (consultants)
- For customers with expertise and resources, best testing sequence is:
  - Dynamic tests first (discussed next)
  - Static tests next
  - Finally Penetration Testers should be employed (pen-testers should always be used, but more on that later)

Dynamic Testing

- Test should be against actual deployment configurations
- Tests can be generic – not specific to applications
  - Fuzzers (Both network and UI) are a subset of this category
  - Configuration testers (e.g., SSLv2 not allowed)
  - Urgency is high since exploiter can run such tools also


HP WebInspect
File Edit View Tools Enterprise Server Reports Help
New Open Compliance Manager Policy Manager Report Schedule SmartUpdate
Start / Resume Pause Skip Audit
Start Page
Welcome to WebInspect
Home Manage Scans Manage Schedule
Start a Guided Scan
Create a scan that is optimized for your Web site.
Start a Basic Scan
Scan a single Web site for vulnerabilities.
Start a Web Service Scan
Find vulnerabilities in a Web service.
Start an Enterprise Scan
Schedule an enterprise scan.
Generate a Report
Analyze a scan using system reports.
Start SmartUpdate
Update security checks and patches.
Open Rescan Rename Delete Import Export Compare Connections Refresh Columns
Scan Date

Scan Date: Older (12 items)

Scan Name	Start URL	Vulnerabilities
Workflow: zero.we...		104
Workflow: zero.we...		17
Workflow: zero.we...		93
Site: http://zero.we...		18
Site: http://zero.we...	http://zero.webapp...	3
Site: http://172.16....	http://172.16.207....	171
Site: http://172.16....	http://172.16.207....	171
Site: http://192.16...	http://192.168.209...	16
Site: http://192.16...	http://192.168.209...	14
Site: http://172.16....	http://172.16.207....	19
Workflow: localhost		0
Site: http://localho...		0

Scan Wizard

Web Site Scan — Step 1 of 5

Perform a scan of the target application.

Scan Name: Site: http://172.16.207.135/mutillidae/

Crawl Only [✓ Crawl & Audit] Audit Only Manual

[✓ Standard Scan]

WebInspect performs an automated analysis starting from the target URL.

Start URL: http://172.16.207.135/mutillidae/

Browser: Internet Explorer

Restrict to folder:

List-Driven Scan

Perform a scan using a list of URLs. These URLs can be listed in a text file or an XML file created by the harvest tool.

File:

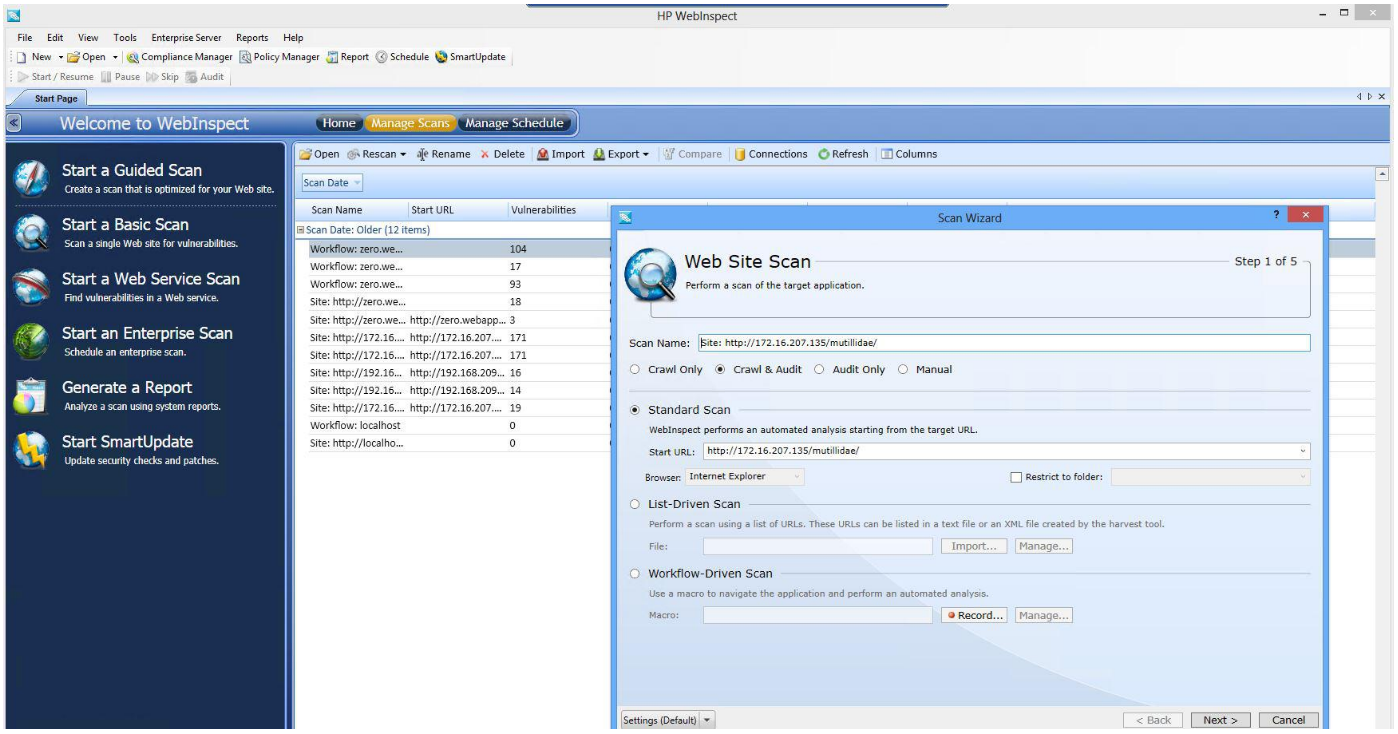
Workflow-Driven Scan

Use a macro to navigate the application and perform an automated analysis.

Macro:

Dynamic Testing, cont.

- Tests can be specific to an application and to detect regressions
  - Include regression tests (does application still fix security errors)
  - Can have specialized instrumentation to detect load, lock conflict or other issues
  - Can be run as part of change check-in (recommended)


HP WebInspect
File Edit View Tools Enterprise Server Reports Help
New Open Compliance Manager Policy Manager Report Schedule SmartUpdate
Start / Resume Pause Skip Audit
Start Page
Welcome to WebInspect
Home Manage Scans Manage Schedule
Open Rescan Rename Delete Import Export Compare Connections Refresh Columns
Scan Date

Start a Guided Scan
Create a scan that is optimized for your Web site.

Start a Basic Scan
Scan a single Web site for vulnerabilities.

Start a Web Service Scan
Find vulnerabilities in a Web service.

Start an Enterprise Scan
Schedule an enterprise scan.

Generate a Report
Analyze a scan using system reports.

Start SmartUpdate
Update security checks and patches.

Scan Name	Start URL	Vulnerabilities
Scan Date: Older (12 items)		
Workflow: zero.we...		104
Workflow: zero.we...		17
Workflow: zero.we...		93
Site: http://zero.we...		18
Site: http://zero.we...	http://zero.webapp...	3
Site: http://172.16....	http://172.16.207....	171
Site: http://172.16....	http://172.16.207....	171
Site: http://192.16...	http://192.168.209...	16
Site: http://192.16...	http://192.168.209...	14
Site: http://172.16....	http://172.16.207....	19
Workflow: localhost		0
Site: http://localho...		0

Scan Wizard

Web Site Scan
Perform a scan of the target application.
Step 1 of 5

Scan Name: Site: http://172.16.207.135/mutillidae/

Crawl Only [✓ Crawl & Audit] Audit Only Manual

[✓ Standard Scan]
WebInspect performs an automated analysis starting from the target URL.
Start URL: http://172.16.207.135/mutillidae/
Browser: Internet Explorer Restrict to folder:

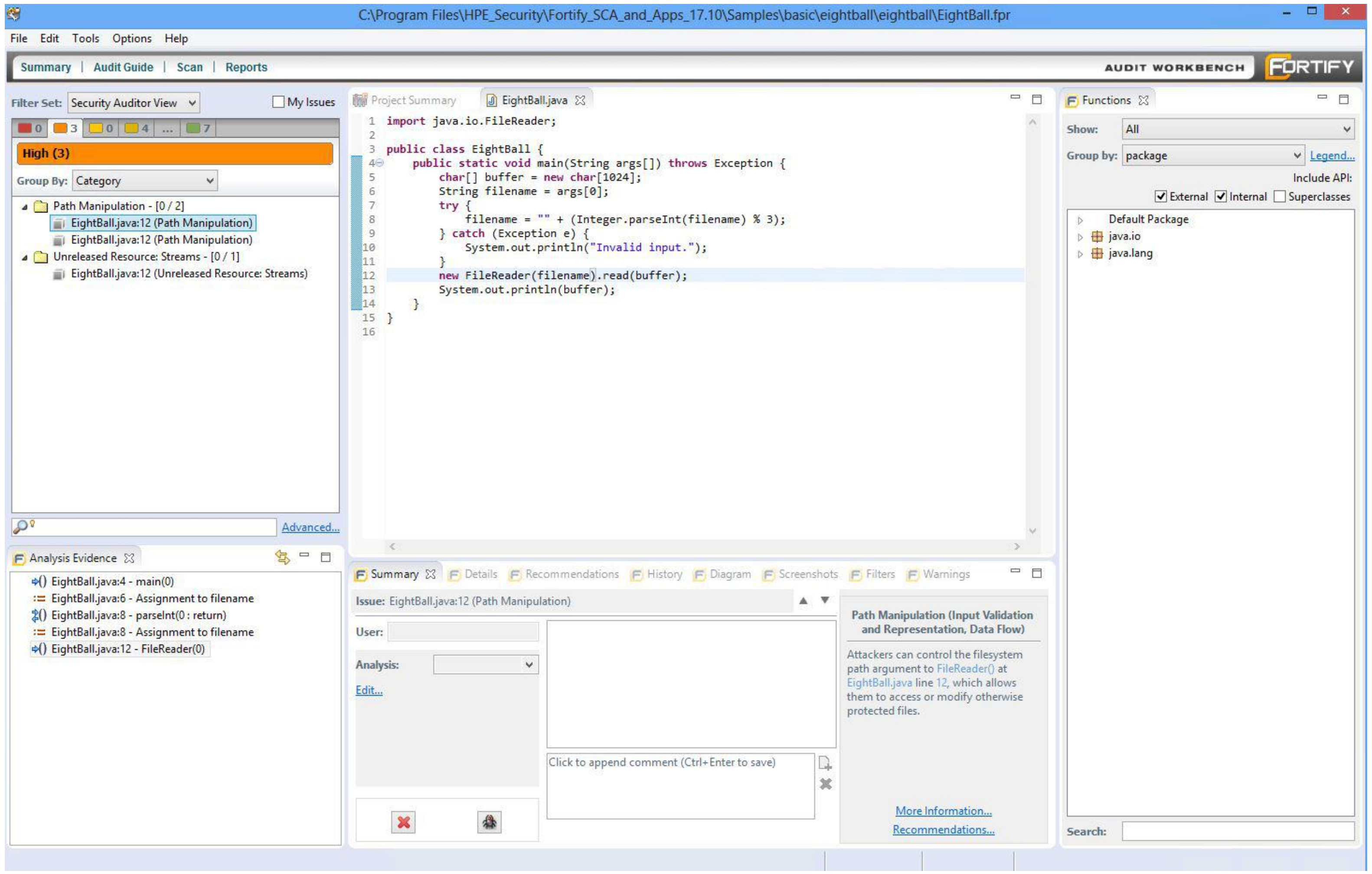
List-Driven Scan
Perform a scan using a list of URLs. These URLs can be listed in a text file or an XML file created by the harvest tool.
File: Import... Manage...

Workflow-Driven Scan
Use a macro to navigate the application and perform an automated analysis.
Macro: Record... Manage...

Settings (Default) < Back Next > Cancel

Static Testing

- Analysis of source code – requires source code be available
- Tools can be expensive
  - Some tools require lots of expert IT customization to become effective
  - False positive can be major issue until IT gains expertise for use


C:\Program Files\HPE_Security\Fortify_SCA_and_Apps_17.10\Samples\basic\eightball\eightball\EightBall.fpr

Filter Set: Security Auditor View

High (3)

Group By: Category

- Path Manipulation - [0 / 2]
  - EightBall.java:12 (Path Manipulation)
  - EightBall.java:12 (Path Manipulation)
- Unreleased Resource: Streams - [0 / 1]
  - EightBall.java:12 (Unreleased Resource: Streams)

EightBall.java

```java
import java.io.FileReader;

public class EightBall {
    public static void main(String args[]) throws Exception {
        char[] buffer = new char[1024];
        String filename = args[0];
        try {
            filename = "" + (Integer.parseInt(filename) % 3);
        } catch (Exception e) {
            System.out.println("Invalid input.");
        }
        new FileReader(filename).read(buffer);
        System.out.println(buffer);
    }
}
```

Analysis Evidence

- EightBall.java:4 - main(0)
- EightBall.java:6 - Assignment to filename
- EightBall.java:8 - parseInt(0 : return)
- EightBall.java:8 - Assignment to filename
- EightBall.java:12 - FileReader(0)

Issue: EightBall.java:12 (Path Manipulation)

Click to append comment (Ctrl+Enter to save)

Path Manipulation (Input Validation and Representation, Data Flow)

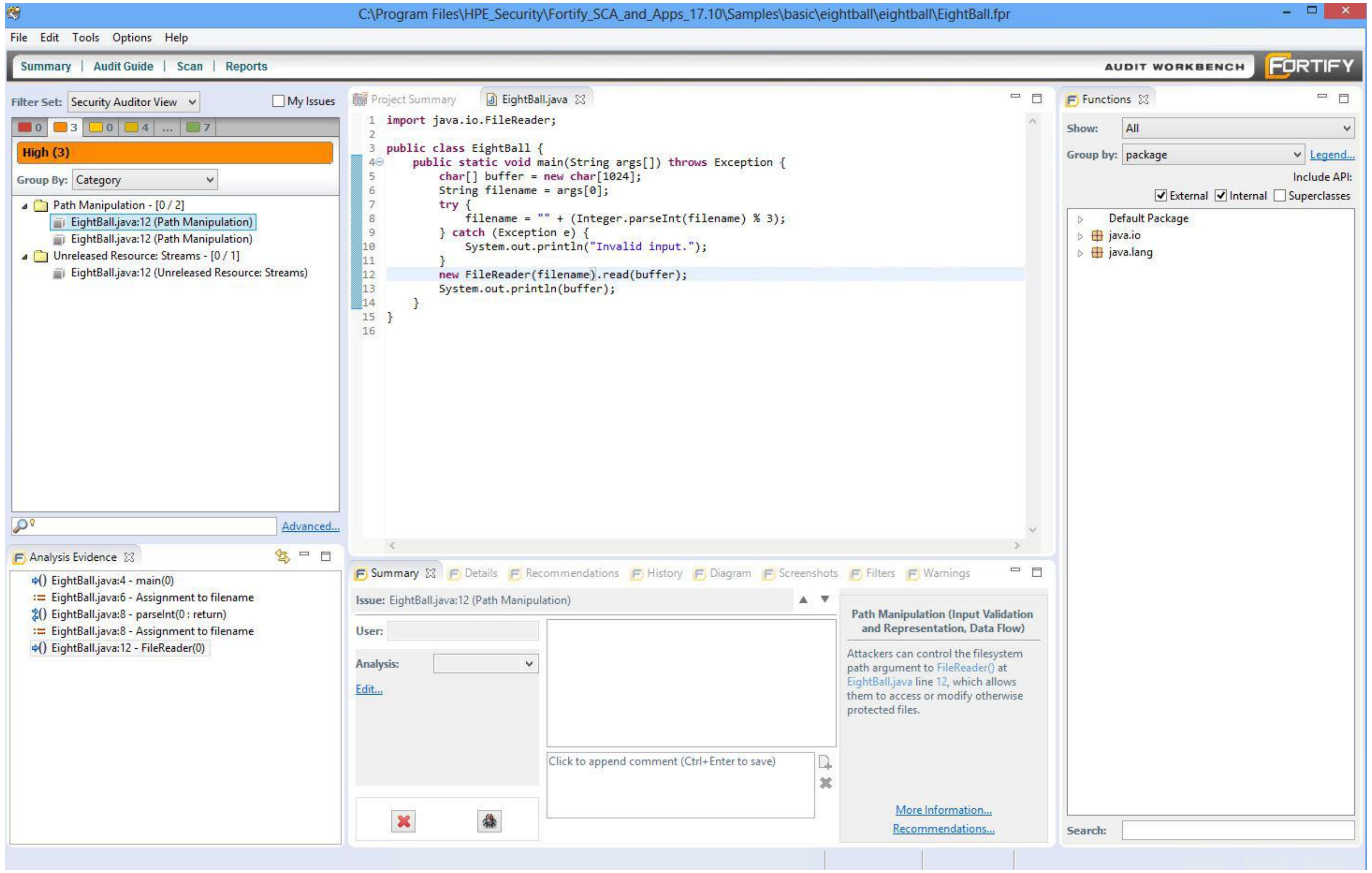
Attackers can control the filesystem path argument to FileReader() at EightBall.java line 12, which allows them to access or modify otherwise protected files.

Functions — Show: All; Group by: package; Include API: External ✓, Internal ✓, Superclasses

- Default Package
- java.io
- java.lang

Static Testing, cont.

- But, static testing is much less expensive per vulnerability discovered than pen-testers or code review
- Exploiters can't perform static testing unless they have source available



C:\Program Files\HPE_Security\Fortify_SCA_and_Apps_17.10\Samples\basic\eightball\eightball\EightBall.fpr

File Edit Tools Options Help

Summary | Audit Guide | Scan | Reports — AUDIT WORKBENCH FORTIFY

Filter Set: Security Auditor View — My Issues

0 | 3 | 0 | 4 | ... | 7

High (3)

Group By: Category

- Path Manipulation - [0 / 2]
  - EightBall.java:12 (Path Manipulation)
  - EightBall.java:12 (Path Manipulation)
- Unreleased Resource: Streams - [0 / 1]
  - EightBall.java:12 (Unreleased Resource: Streams)

Advanced...

Analysis Evidence

- EightBall.java:4 - main(0)
- EightBall.java:6 - Assignment to filename
- EightBall.java:8 - parseInt(0 : return)
- EightBall.java:8 - Assignment to filename
- EightBall.java:12 - FileReader(0)

Project Summary | EightBall.java

```
 1 import java.io.FileReader;
 2
 3 public class EightBall {
 4     public static void main(String args[]) throws Exception {
 5         char[] buffer = new char[1024];
 6         String filename = args[0];
 7         try {
 8             filename = "" + (Integer.parseInt(filename) % 3);
 9         } catch (Exception e) {
10             System.out.println("Invalid input.");
11         }
12         new FileReader(filename).read(buffer);
13         System.out.println(buffer);
14     }
15 }
16
```

Summary | Details | Recommendations | History | Diagram | Screenshots | Filters | Warnings

Issue: EightBall.java:12 (Path Manipulation)

User:

Analysis:

Edit...

Click to append comment (Ctrl+Enter to save)

Path Manipulation (Input Validation and Representation, Data Flow)

Attackers can control the filesystem path argument to FileReader() at EightBall.java line 12, which allows them to access or modify otherwise protected files.

More Information...

Recommendations...

Functions

Show: All

Group by: package — Legend...

Include API: ✓ External ✓ Internal ☐ Superclasses

- Default Package
- java.io
- java.lang

Search:

Penetration Testing

- Pen testing can be a very valuable exercise
  - If performed against a deployment that closely mirrors your production environment
  - If performed after patches applied and configuration is secure
  - If performed after vulnerabilities discovered by automated means are remediated (which pen-testing organizations can run on behalf of customers)
- Pen testing can yield high-value findings
  - Complex attacks involving multiple components may be difficult to uncover with automated tools but may be discovered by penetration testers
  - New vulnerabilities not yet discoverable via automated tools may be discovered
  - Again, all organizations should employ pen-testers

ORACLE

What should you do if you think you have discovered a security bug in an Oracle product?

- For Customers
  - One SR per vulnerability with proof of concept (POC)
  - Security tools reports should be verified (e.g., through POC) before submitting SR
  - POC not required for published fixes in 3rd party components known to be part of product (POC is self-evident)
- For Researchers
  - Contact secalert_us@oracle.com
    - We encourage use of encryption via our public key http://www.oracle.com/technology/deploy/security/alerts.htm

ORACLE