BİMAR BİLGİ İŞLEM HİZMETLERİ A.Ş.

Secure Web Application Development

METHODOLOGIES AND AUTOMATED TOOLS

MURAT KAYA – SOFTWARE SECURITY SPECIALIST

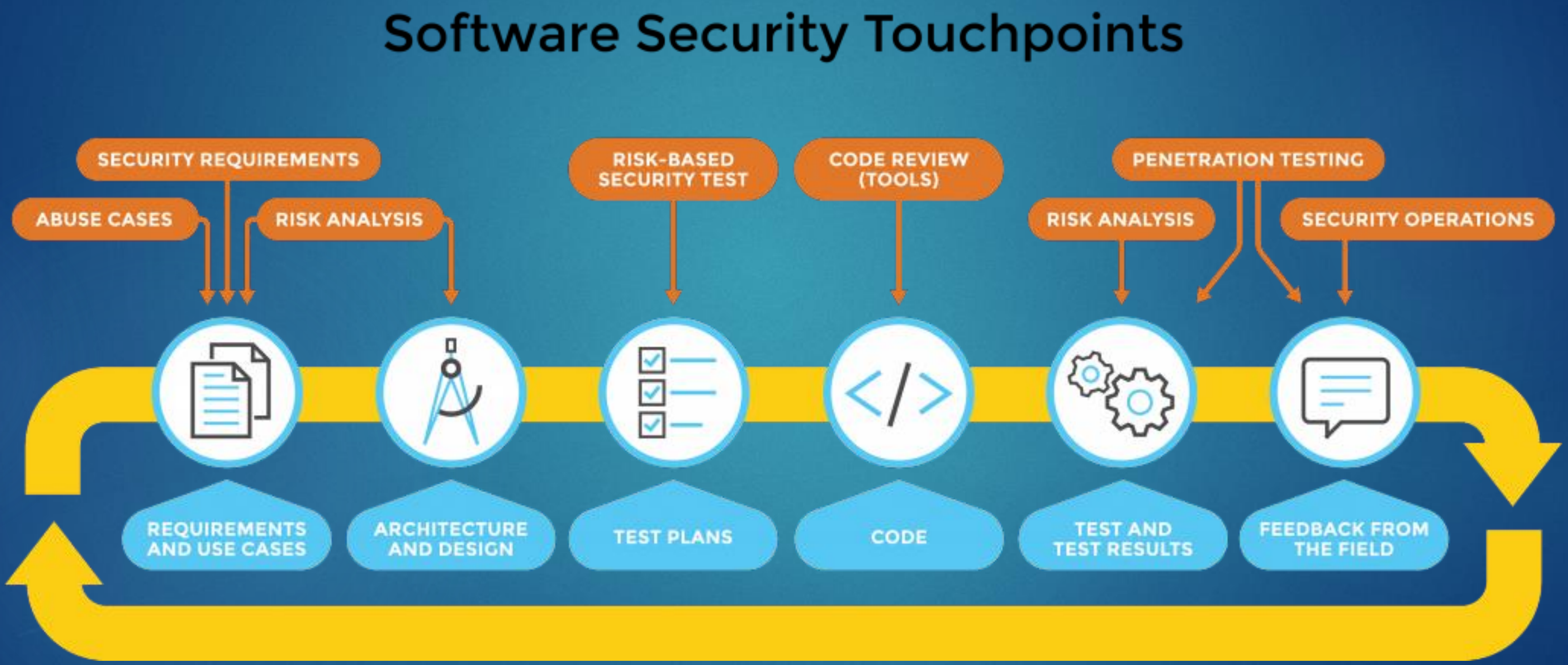
Agenda

- Software Security Overview
- Software Security Methodologies
- Security Test Methods
- Analysis Tools
- Tools Auditing Samples
- Preparing Analysis Results
- Sample Analysis Results
- Managing of Findings and Vulnerabilities
- Q&A

Software Security Overview

- Security is a "**risk management**" with the most basic expression.
- Secure software refers to that has been designed from the ground up with all security processes, enhanced with security checks, and completed with security tests.
- Software security is based on internationally accepted methods.
- It's not a **one-time** process
- In reality there is no **%100** secured software

Software Security Overview



Methodologies

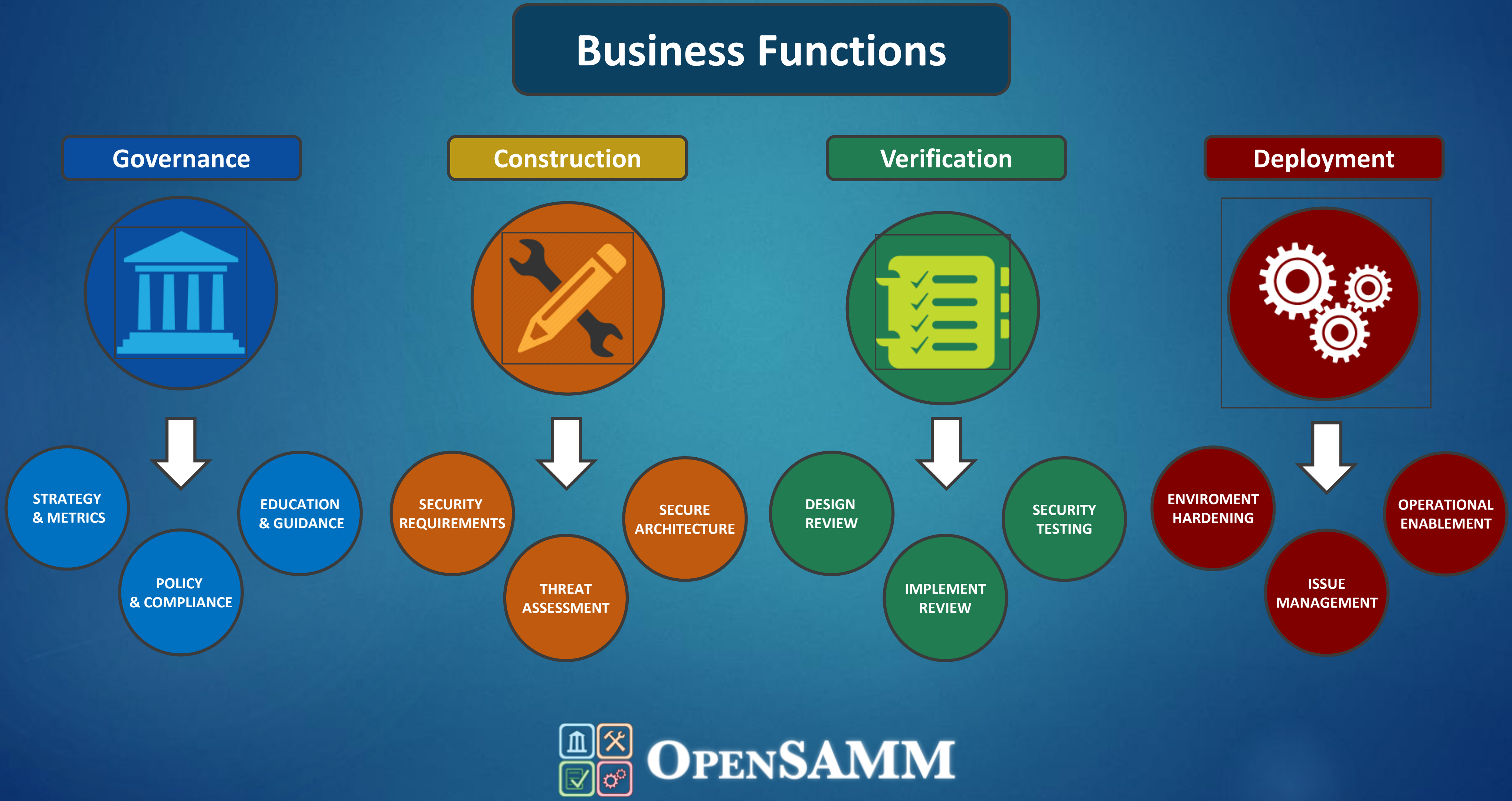
- **OpenSAMM (Software Assurance Maturity Model)**
  - SAMM is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization
  - As an open project, SAMM content shall always remain vendor-neutral and freely available for all to use
  - SAMM was defined with flexibility in mind such that it can be utilized by small, medium, and large organizations using any style of development.
- **SDL (Secure Development Lifecycle)**
  - SDL is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost
  - Developed by Microsoft Security Technology Unit.
  - The ability of an application to be included in the SDL process depends on the use of the application. If an application needs Internet access, it will be included in a process that can contain important information.

Methodologies - OpenSAMM

- **The resources provided by SAMM will aid in:**
  - Evaluating an organization’s existing software security practices
  - Building a balanced software security program in well-defined iterations
  - Demonstrating concrete improvements to a security assurance program
  - Defining and measuring security-related activities within an organization

OpenSAMM

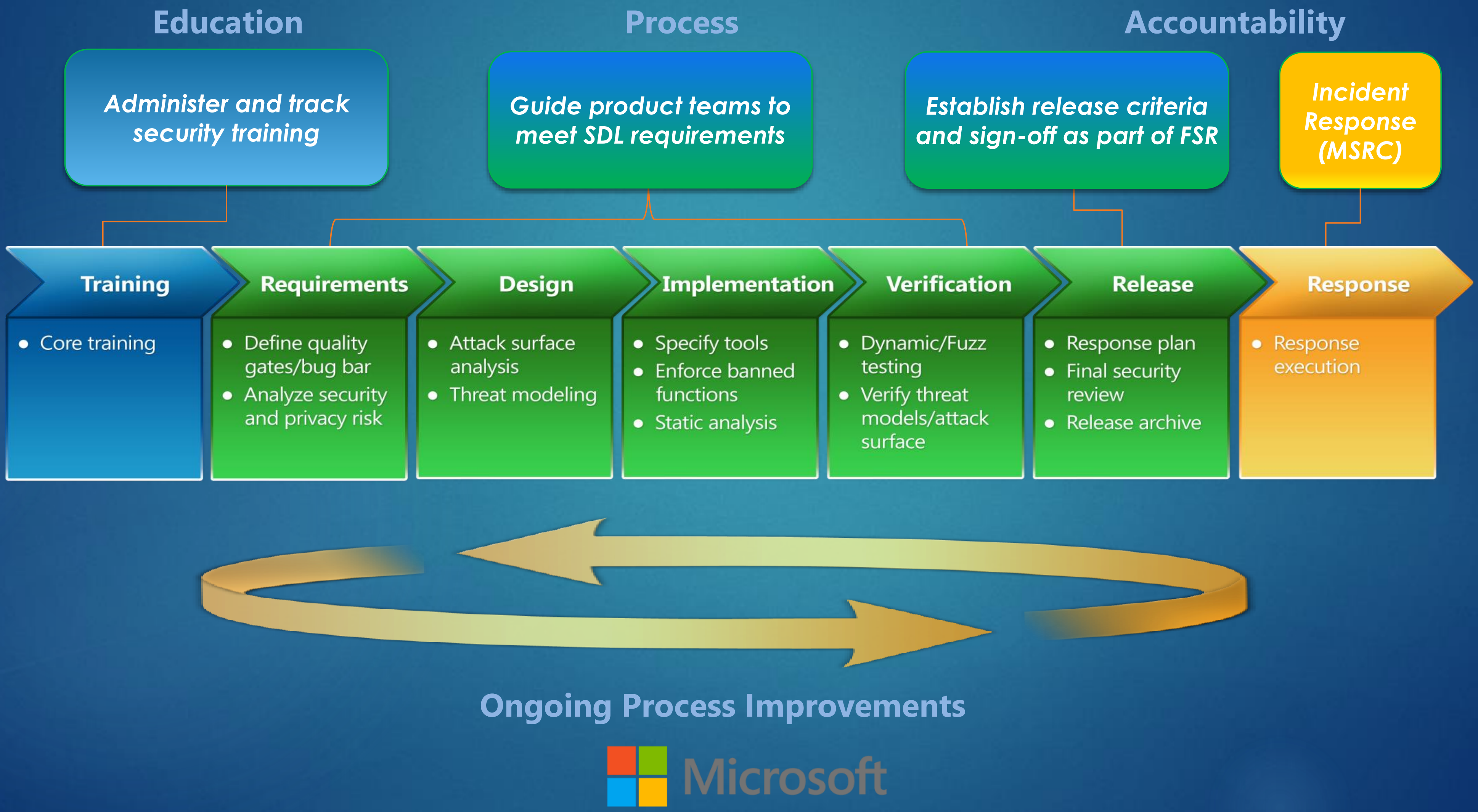
Methodologies - OpenSAMM



Methodologies – SDL

- **The resources provided by SDL will aid in:**
  - The SDL Helps you build software that's more secure by reducing the number and severity of vulnerabilities in your code
  - Incorporating the SDL into the application development process helps meet compliance requirements and produce a return on investment (ROI) by guiding organizations to make smart choices early in the design process, thereby minimizing expensive inefficiencies.
  - The SDL systematically addresses software security during the development phase, ensuring that vulnerabilities are more likely to be found and fixed prior to application deployment and thereby reducing your total cost of software development.

Methodologies – SDL



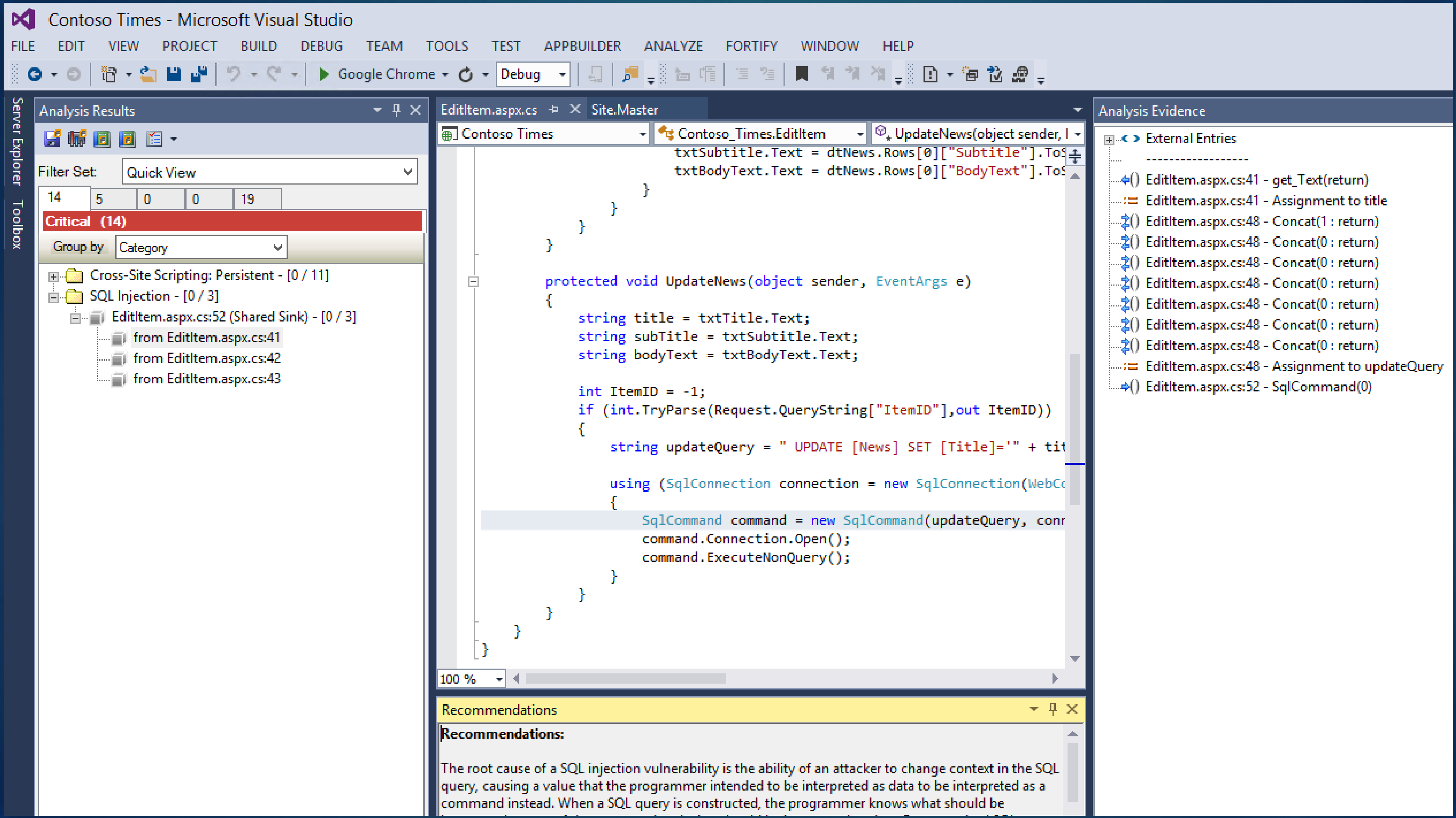
Security Test Methods

- Static Code Analysis (Source Code Analysis)
  - Static Code Analysis (also known as Source Code Analysis) is usually performed as part of a Code Review (also known as white-box testing) and is carried out at the Implementation phase of a Security Development Lifecycle (SDL)
- Penetration testing options include black box, white box and gray box tests.
  - **White box**, or authenticated tests, target the security of your underlying technology with full knowledge of your IT department.
  - **Black box**, or unauthenticated, tests closely represent a hacker attempting to gain unauthorized access to a system or IT infrastructure to obtain and exfiltrate data.
  - **Gray box** testing lies between black and white. Testers will have knowledge of some areas but not others.

Analysis Tools

- Static Code Analysis Tools
  - SonarQube
  - HPE Fortify SCA
  - Telerik Platforms
- Pentration Testing Tools
  - Accunetix Web Vulnerability Scanner
  - HPE WebInspect
  - OWASP Zap Proxy
  - SoapUI, Burp e.t.c
- Fuzzing Tools

Auditing – Fortify SCA Visual Studio


Contoso Times - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST APPBUILDER ANALYZE FORTIFY WINDOW HELP
Google Chrome
Debug
Server Explorer
Toolbox
Analysis Results
EditItem.aspx.cs
Site.Master
Contoso Times
Contoso_Times.EditItem
UpdateNews(object sender, I
Analysis Evidence
100 %
Recommendations

Filter Set: Quick View

14	5	0	0	19

Critical (14)

Group by: Category

- Cross-Site Scripting: Persistent - [0 / 11]
- SQL Injection - [0 / 3]
  - EditItem.aspx.cs:52 (Shared Sink) - [0 / 3]
    - from EditItem.aspx.cs:41
    - from EditItem.aspx.cs:42
    - from EditItem.aspx.cs:43

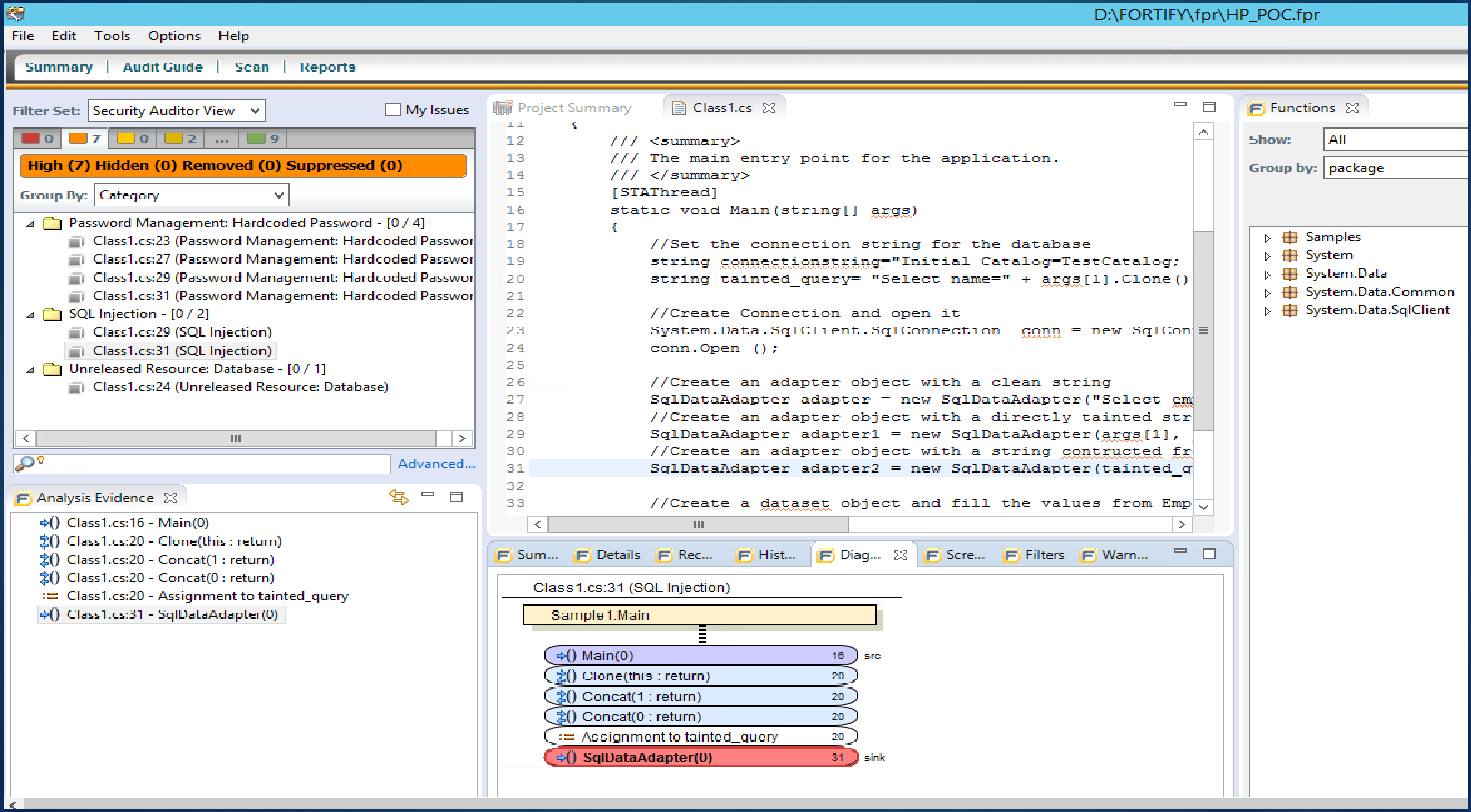
```
                    txtSubtitle.Text = dtNews.Rows[0]["Subtitle"].ToS
                    txtBodyText.Text = dtNews.Rows[0]["BodyText"].ToS
                }
            }
        }
    }

    protected void UpdateNews(object sender, EventArgs e)
    {
        string title = txtTitle.Text;
        string subTitle = txtSubtitle.Text;
        string bodyText = txtBodyText.Text;

        int ItemID = -1;
        if (int.TryParse(Request.QueryString["ItemID"],out ItemID))
        {
            string updateQuery = " UPDATE [News] SET [Title]='" + tit

            using (SqlConnection connection = new SqlConnection(WebCc
            {
                SqlCommand command = new SqlCommand(updateQuery, conr
                command.Connection.Open();
                command.ExecuteNonQuery();
            }
        }
    }
  }
}
```

- External Entries
- ------------------
- EditItem.aspx.cs:41 - get_Text(return)
- EditItem.aspx.cs:41 - Assignment to title
- EditItem.aspx.cs:48 - Concat(1 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Concat(0 : return)
- EditItem.aspx.cs:48 - Assignment to updateQuery
- EditItem.aspx.cs:52 - SqlCommand(0)

Recommendations:

The root cause of a SQL injection vulnerability is the ability of an attacker to change context in the SQL query, causing a value that the programmer intended to be interpreted as data to be interpreted as a command instead. When a SQL query is constructed, the programmer knows what should be

Auditing – Fortify SCA Workbench



D:\FORTIFY\fpr\HP_POC.fpr

File Edit Tools Options Help

Summary | Audit Guide | Scan | Reports

Filter Set: Security Auditor View   My Issues

0 | 7 | 0 | 2 | ... | 9

High (7) Hidden (0) Removed (0) Suppressed (0)

Group By: Category

- Password Management: Hardcoded Password - [0 / 4]
  - Class1.cs:23 (Password Management: Hardcoded Passwor
  - Class1.cs:27 (Password Management: Hardcoded Passwor
  - Class1.cs:29 (Password Management: Hardcoded Passwor
  - Class1.cs:31 (Password Management: Hardcoded Passwor
- SQL Injection - [0 / 2]
  - Class1.cs:29 (SQL Injection)
  - Class1.cs:31 (SQL Injection)
- Unreleased Resource: Database - [0 / 1]
  - Class1.cs:24 (Unreleased Resource: Database)

Advanced...

Analysis Evidence

- Class1.cs:16 - Main(0)
- Class1.cs:20 - Clone(this : return)
- Class1.cs:20 - Concat(1 : return)
- Class1.cs:20 - Concat(0 : return)
- Class1.cs:20 - Assignment to tainted_query
- Class1.cs:31 - SqlDataAdapter(0)

Project Summary | Class1.cs

```
12          /// <summary>
13          /// The main entry point for the application.
14          /// </summary>
15          [STAThread]
16          static void Main(string[] args)
17          {
18              //Set the connection string for the database
19              string connectionstring="Initial Catalog=TestCatalog;
20              string tainted_query= "Select name=" + args[1].Clone()
21
22              //Create Connection and open it
23              System.Data.SqlClient.SqlConnection  conn = new SqlCon
24              conn.Open ();
25
26              //Create an adapter object with a clean string
27              SqlDataAdapter adapter = new SqlDataAdapter("Select em
28              //Create an adapter object with a directly tainted str
29              SqlDataAdapter adapter1 = new SqlDataAdapter(args[1],
30              //Create an adapter object with a string contructed fr
31              SqlDataAdapter adapter2 = new SqlDataAdapter(tainted_q
32
33              //Create a dataset object and fill the values from Emp
```

Sum... | Details | Rec... | Hist... | Diag... | Scre... | Filters | Warn...

Class1.cs:31 (SQL Injection)

Sample1.Main

Main(0)	16	src
Clone(this : return)	20	
Concat(1 : return)	20	
Concat(0 : return)	20	
Assignment to tainted_query	20	
SqlDataAdapter(0)	31	sink

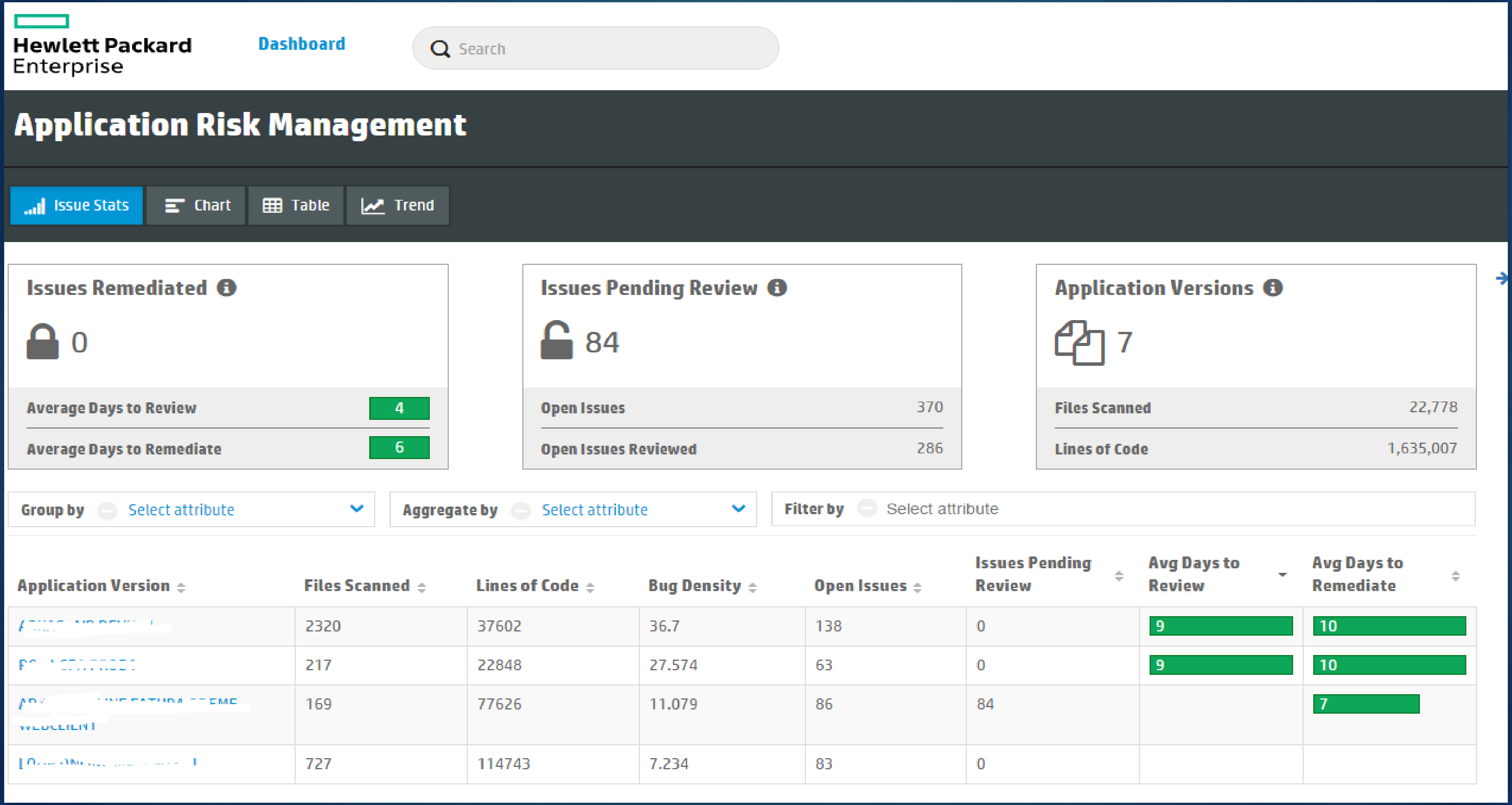
Functions

Show: All

Group by: package

- Samples
- System
- System.Data
- System.Data.Common
- System.Data.SqlClient

Auditing – Fortify SSC Portal



Hewlett Packard Enterprise

Dashboard

Search

Application Risk Management

Issue Stats | Chart | Table | Trend

Issues Remediated	0
Average Days to Review	4
Average Days to Remediate	6

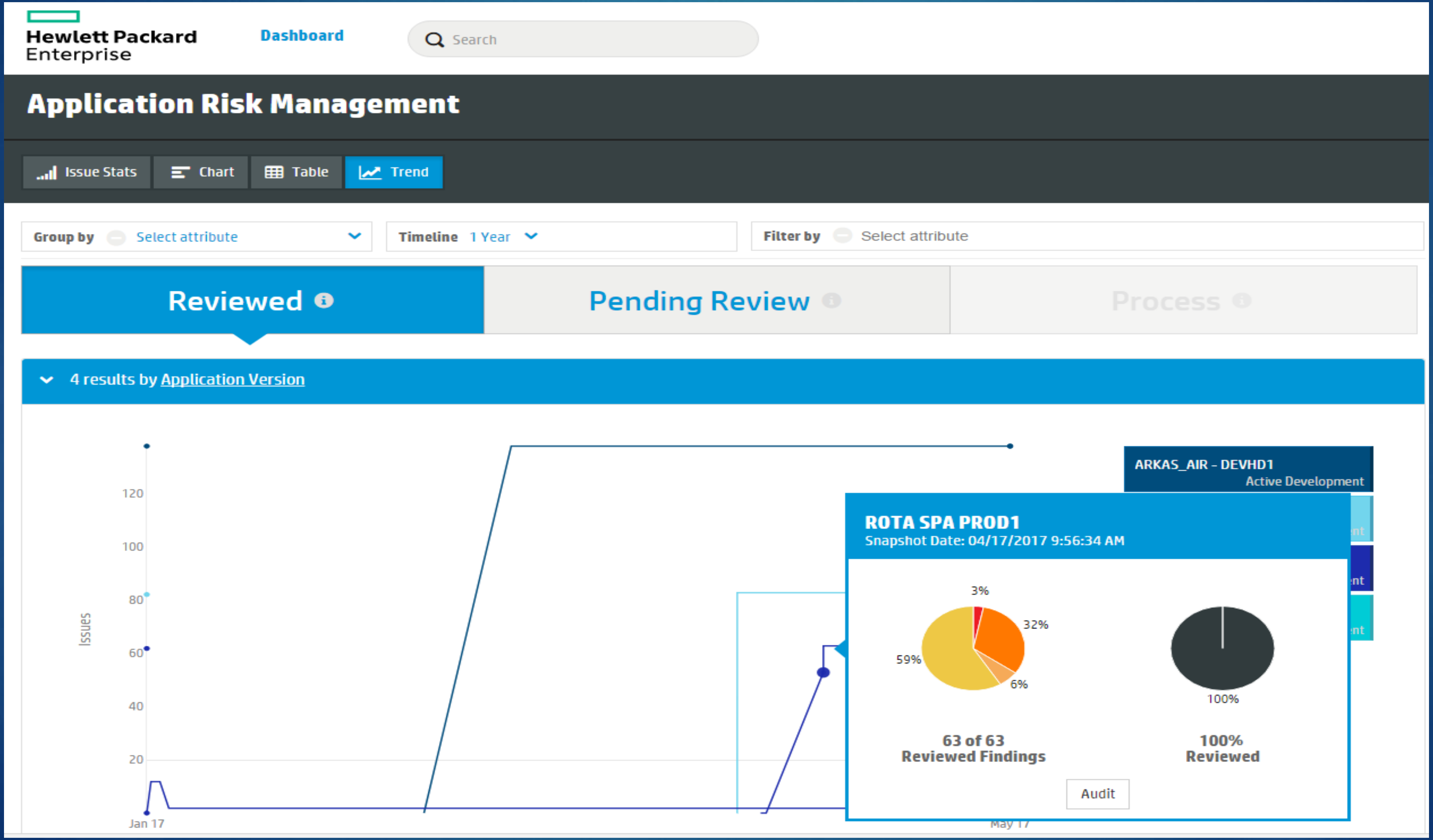
Issues Pending Review	84
Open Issues	370
Open Issues Reviewed	286

Application Versions	7
Files Scanned	22,778
Lines of Code	1,635,007

Group by: Select attribute | Aggregate by: Select attribute | Filter by: Select attribute

Application Version	Files Scanned	Lines of Code	Bug Density	Open Issues	Issues Pending Review	Avg Days to Review	Avg Days to Remediate
[redacted]	2320	37602	36.7	138	0	9	10
[redacted]	217	22848	27.574	63	0	9	10
[redacted]	169	77626	11.079	86	84		7
[redacted]	727	114743	7.234	83	0		

Auditing – Fortify SSC Portal



Hewlett Packard Enterprise Dashboard Search

Application Risk Management

Issue Stats | Chart | Table | Trend

Group by: Select attribute | Timeline: 1 Year | Filter by: Select attribute

Reviewed | Pending Review | Process

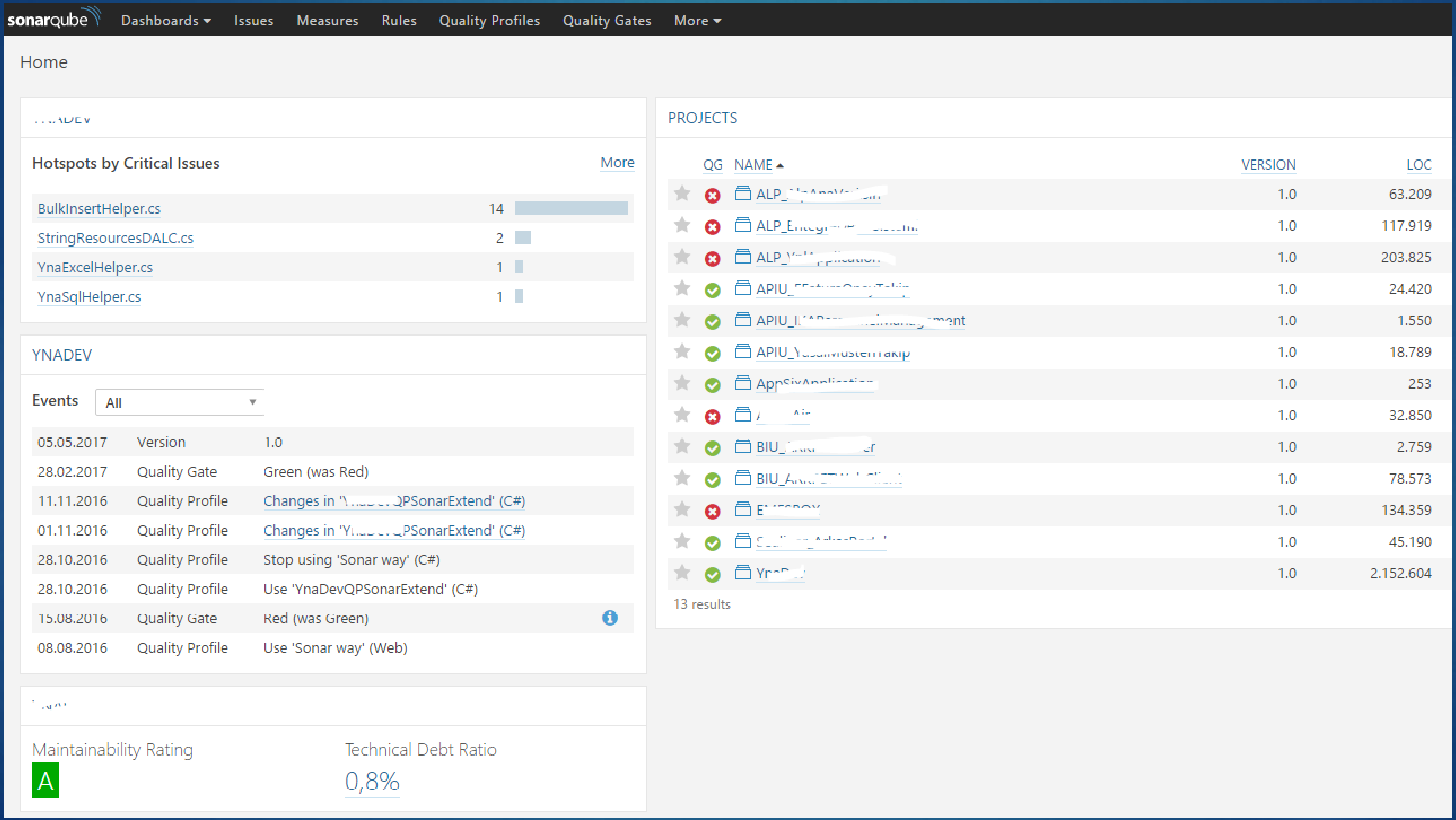
4 results by Application Version

ROTA SPA PROD1
Snapshot Date: 04/17/2017 9:56:34 AM

63 of 63 Reviewed Findings

100% Reviewed

Auditing – SonarQube



Home

[redacted]

Hotspots by Critical Issues More

BulkInsertHelper.cs	14
StringResourcesDALC.cs	2
YnaExcelHelper.cs	1
YnaSqlHelper.cs	1

YNADEV

Events All

05.05.2017	Version	1.0
28.02.2017	Quality Gate	Green (was Red)
11.11.2016	Quality Profile	Changes in '[redacted]QPSonarExtend' (C#)
01.11.2016	Quality Profile	Changes in 'Y[redacted]PSonarExtend' (C#)
28.10.2016	Quality Profile	Stop using 'Sonar way' (C#)
28.10.2016	Quality Profile	Use 'YnaDevQPSonarExtend' (C#)
15.08.2016	Quality Gate	Red (was Green)
08.08.2016	Quality Profile	Use 'Sonar way' (Web)

[redacted]

Maintainability Rating A

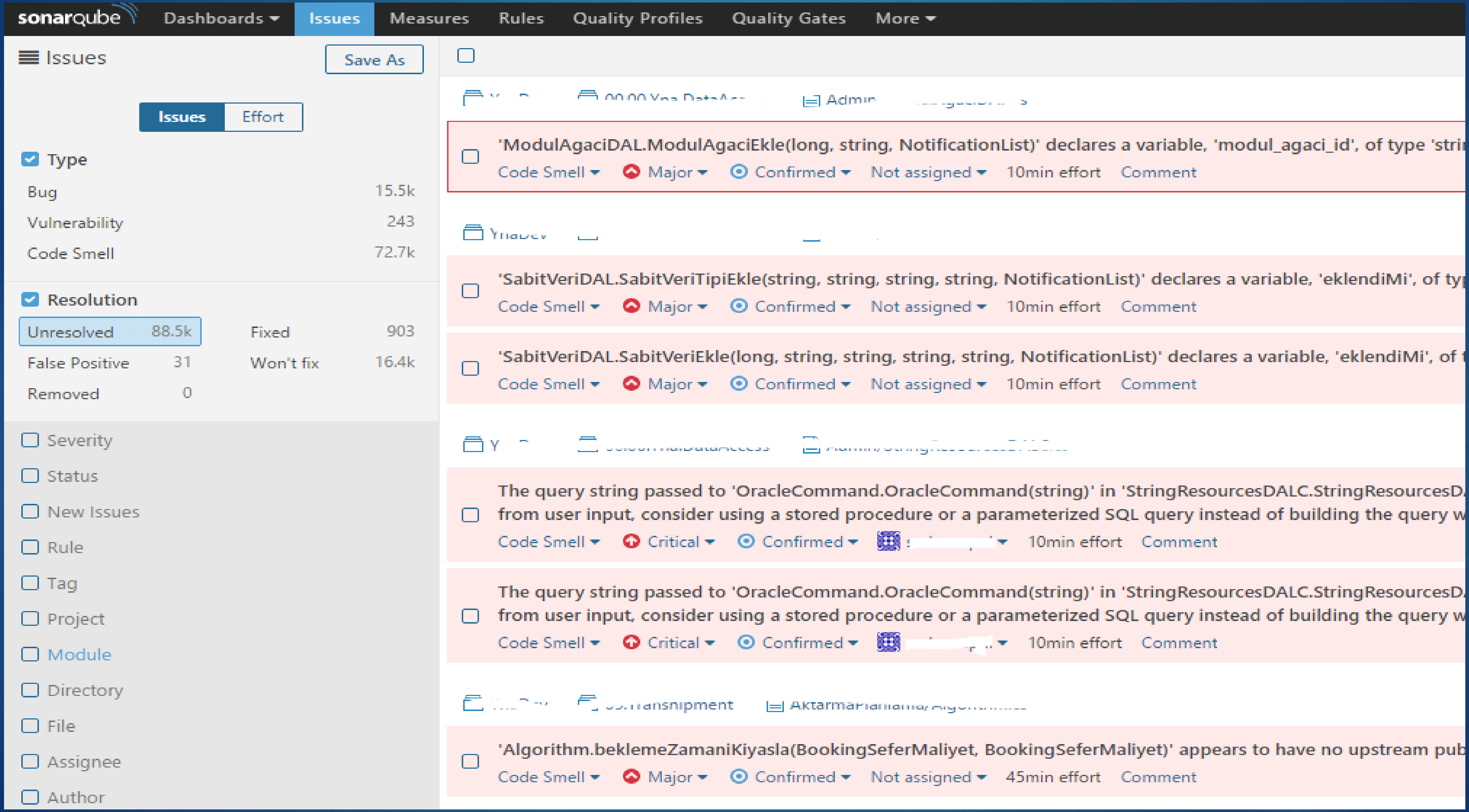
Technical Debt Ratio 0,8%

PROJECTS

QG	NAME	VERSION	LOC
✗	ALP_[redacted]	1.0	63.209
✗	ALP_E[redacted]	1.0	117.919
✗	ALP_Y[redacted]	1.0	203.825
✓	APIU_[redacted]	1.0	24.420
✓	APIU_I[redacted]ment	1.0	1.550
✓	APIU_Y[redacted]akip	1.0	18.789
✓	App[redacted]	1.0	253
✗	[redacted]	1.0	32.850
✓	BIU_[redacted]er	1.0	2.759
✓	BIU_A[redacted]	1.0	78.573
✗	E[redacted]	1.0	134.359
✓	S[redacted]	1.0	45.190
✓	Yn[redacted]	1.0	2.152.604

13 results

Auditing – SonarQube



Type	
Bug	15.5k
Vulnerability	243
Code Smell	72.7k

Resolution			
Unresolved	88.5k	Fixed	903
False Positive	31	Won't fix	16.4k
Removed	0		

Severity
Status
New Issues
Rule
Tag
Project
Module
Directory
File
Assignee
Author

'ModulAgaciDAL.ModulAgaciEkle(long, string, NotificationList)' declares a variable, 'modul_agaci_id', of type 'stri
Code Smell · Major · Confirmed · Not assigned · 10min effort · Comment

'SabitVeriDAL.SabitVeriTipiEkle(string, string, string, string, NotificationList)' declares a variable, 'eklendiMi', of ty
Code Smell · Major · Confirmed · Not assigned · 10min effort · Comment

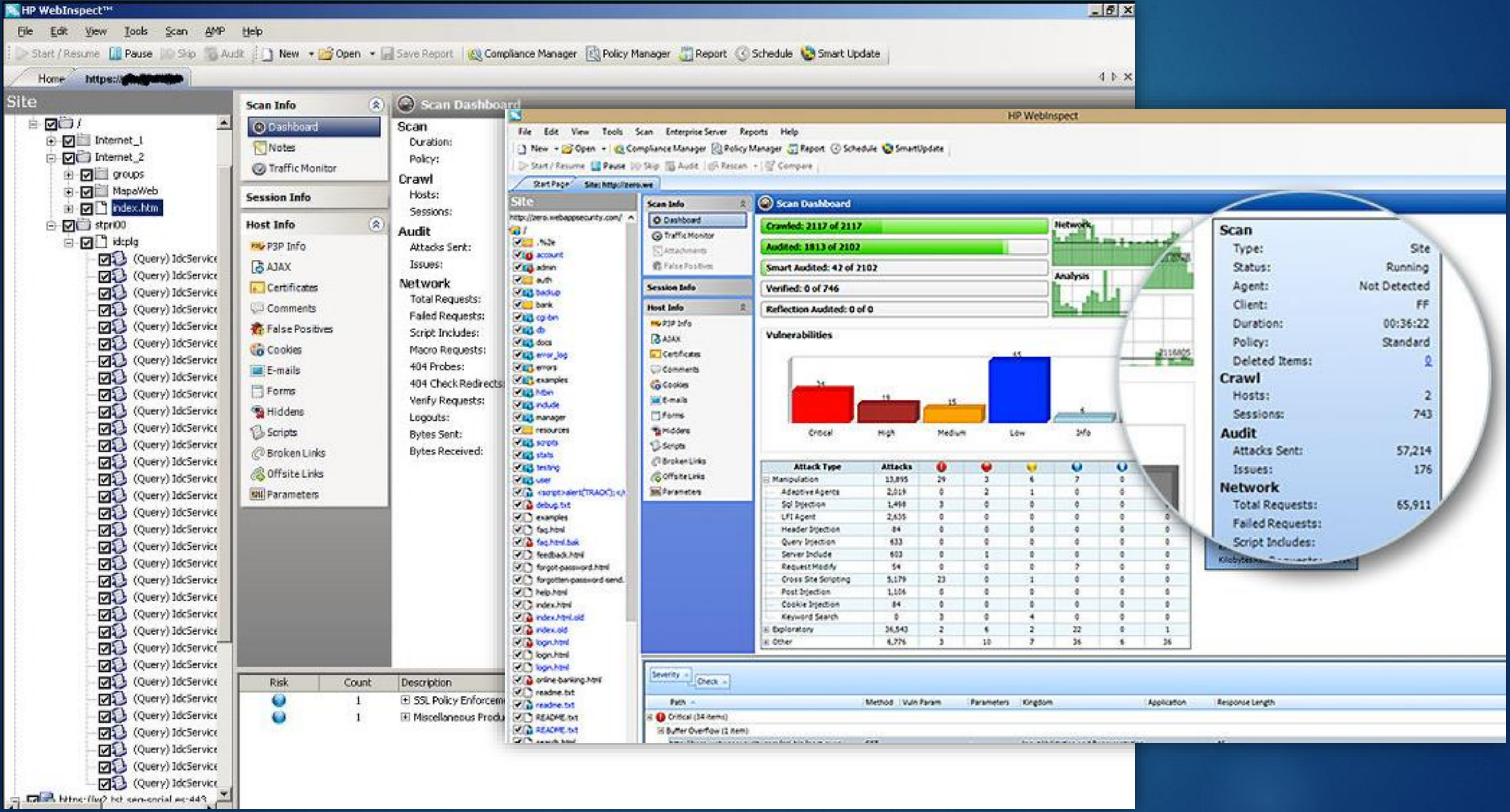
'SabitVeriDAL.SabitVeriEkle(long, string, string, string, string, NotificationList)' declares a variable, 'eklendiMi', of t
Code Smell · Major · Confirmed · Not assigned · 10min effort · Comment

The query string passed to 'OracleCommand.OracleCommand(string)' in 'StringResourcesDALC.StringResourcesD
from user input, consider using a stored procedure or a parameterized SQL query instead of building the query w
Code Smell · Critical · Confirmed · 10min effort · Comment

The query string passed to 'OracleCommand.OracleCommand(string)' in 'StringResourcesDALC.StringResourcesD
from user input, consider using a stored procedure or a parameterized SQL query instead of building the query w
Code Smell · Critical · Confirmed · 10min effort · Comment

'Algorithm.beklemeZamaniKiyasla(BookingSeferMaliyet, BookingSeferMaliyet)' appears to have no upstream pub
Code Smell · Major · Confirmed · Not assigned · 45min effort · Comment

Auditing – HPE WebInspect



Crawled: 2117 of 2117

Audited: 1813 of 2102

Smart Audited: 42 of 2102

Verified: 0 of 746

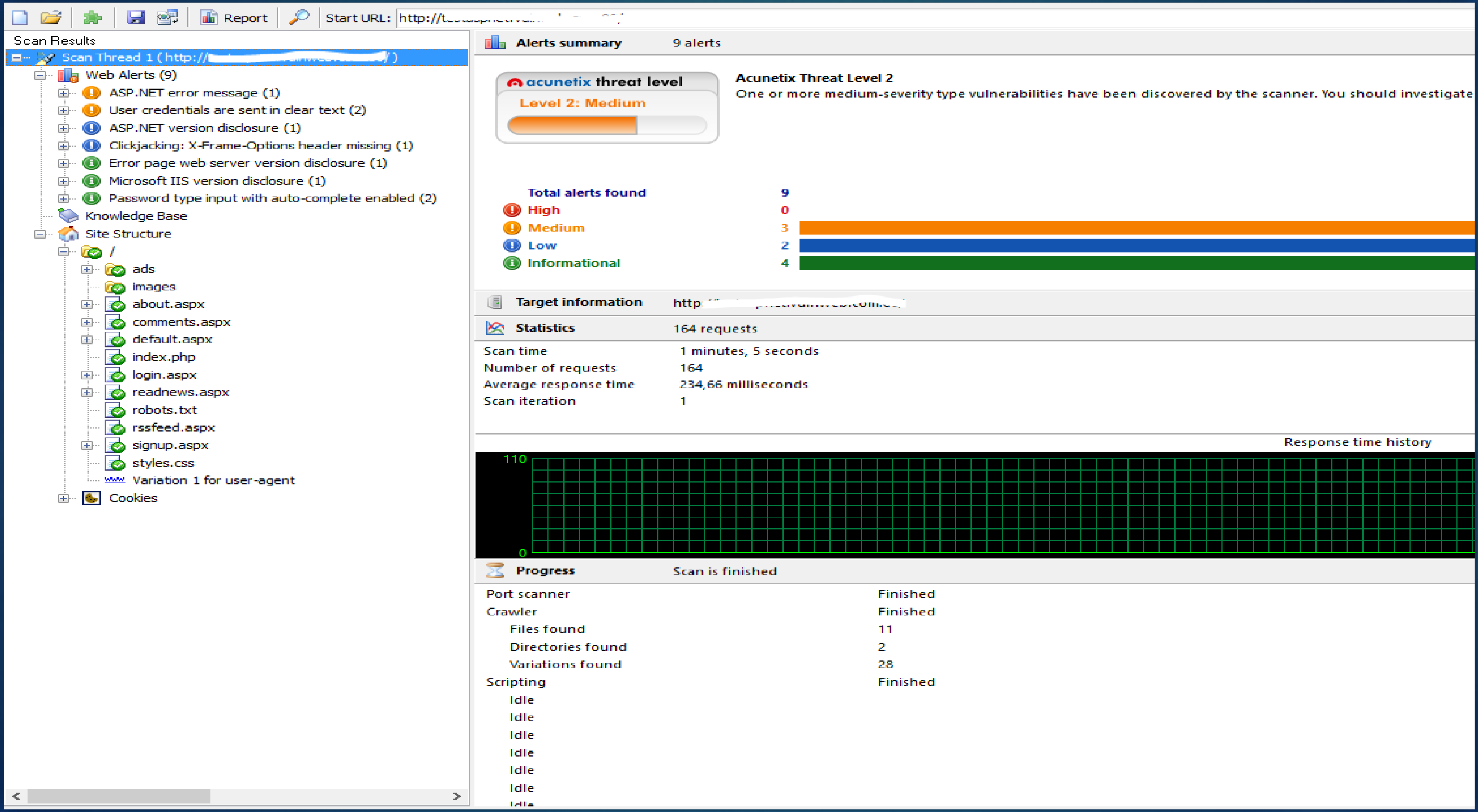
Reflection Audited: 0 of 0

Vulnerabilities

Attack Type	Attacks	🔴	🔴	🟡	🔵	🔵	
Manipulation	13,895	29	3	6	7	0	
Adaptive Agents	2,019	0	2	1	0	0	
Sql Injection	1,498	3	0	0	0	0	
LFI Agent	2,635	0	0	0	0	0	0
Header Injection	84	0	0	0	0	0	0
Query Injection	633	0	0	0	0	0	0
Server Include	603	0	1	0	0	0	0
Request Modify	54	0	0	0	7	0	0
Cross Site Scripting	5,179	23	0	1	0	0	0
Post Injection	1,106	0	0	0	0	0	0
Cookie Injection	84	0	0	0	0	0	0
Keyword Search	0	3	0	4	0	0	0
Exploratory	36,543	2	6	2	22	0	1
Other	6,776	3	10	7	36	6	36

Scan	
Type:	Site
Status:	Running
Agent:	Not Detected
Client:	FF
Duration:	00:36:22
Policy:	Standard
Deleted Items:	0
Crawl	
Hosts:	2
Sessions:	743
Audit	
Attacks Sent:	57,214
Issues:	176
Network	
Total Requests:	65,911
Failed Requests:	
Script Includes:	

Auditing – Acunetix WVS


Report | Start URL: http://[redacted]

Scan Results
- Scan Thread 1 (http://[redacted])
  - Web Alerts (9)
    - ASP.NET error message (1)
    - User credentials are sent in clear text (2)
    - ASP.NET version disclosure (1)
    - Clickjacking: X-Frame-Options header missing (1)
    - Error page web server version disclosure (1)
    - Microsoft IIS version disclosure (1)
    - Password type input with auto-complete enabled (2)
  - Knowledge Base
  - Site Structure
    - /
      - ads
      - images
      - about.aspx
      - comments.aspx
      - default.aspx
      - index.php
      - login.aspx
      - readnews.aspx
      - robots.txt
      - rssfeed.aspx
      - signup.aspx
      - styles.css
      - Variation 1 for user-agent
    - Cookies

Alerts summary 9 alerts

acunetix threat level — Level 2: Medium

Acunetix Threat Level 2
One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate

Total alerts found	9
High	0
Medium	3
Low	2
Informational	4

Target information http[redacted]

Statistics 164 requests

Scan time	1 minutes, 5 seconds
Number of requests	164
Average response time	234,66 milliseconds
Scan iteration	1

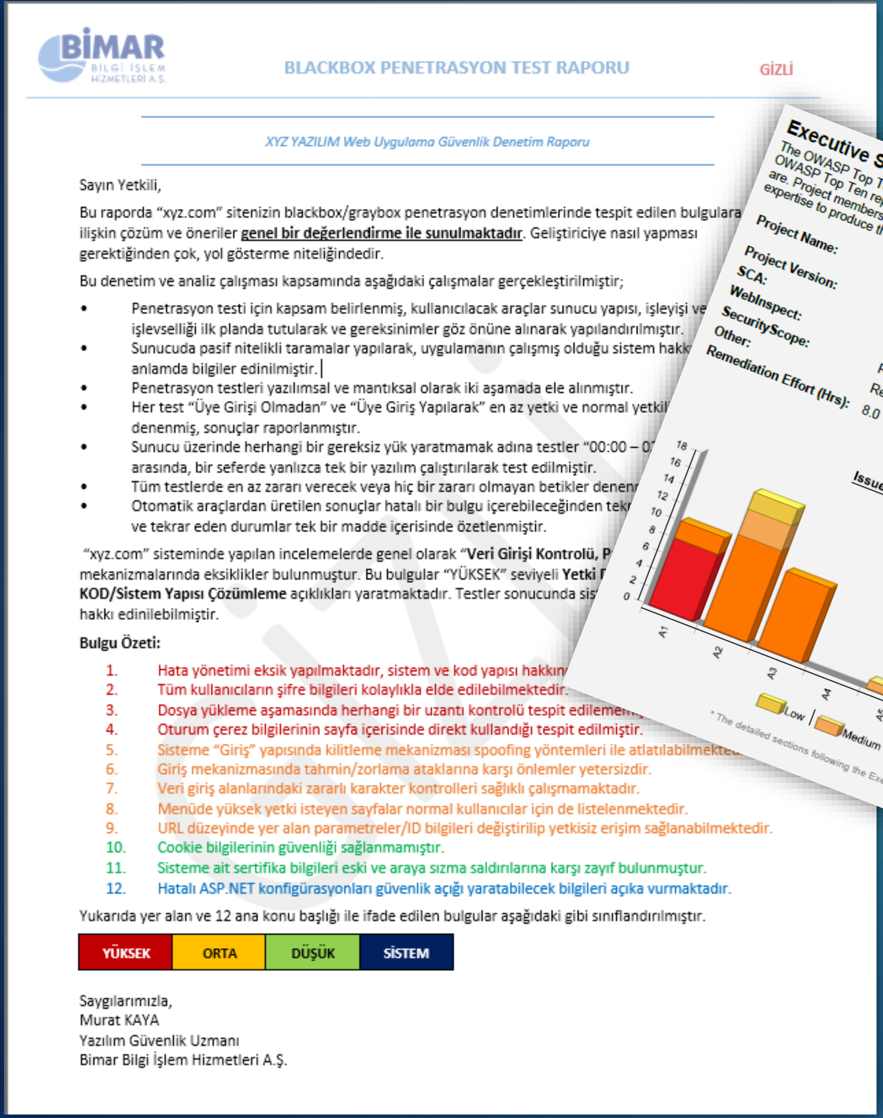
Progress Scan is finished

Port scanner	Finished
Crawler	Finished
Files found	11
Directories found	2
Variations found	28
Scripting	Finished
Idle	
Idle	
Idle	
Idle	
Idle	
Idle	
Idle	

Preparing Analysis Results

- All findings that obtained from automated tools or manual analyzes are reviewed by security auditor before assigning to developer.
  - False positives.
  - Unrelated codes etc.
- Reports are generated with below main headings;
  - Security Level
  - Owasp Category
  - Type of Evidence
  - Effect of Evidence
  - Complete Analysis of Steps
  - Base Solutions
- The report that prepared with above details are sent to team and technical leaders after encrypted with PGP

Sample Analysis Results - Executive



BLACKBOX PENETRASYON TEST RAPORU GİZLİ

XYZ YAZILIM Web Uygulama Güvenlik Denetim Raporu

Sayın Yetkili,

Bu raporda "xyz.com" sitenizin blackbox/graybox penetrasyon denetimlerinde tespit edilen bulgulara ilişkin çözüm ve öneriler **genel bir değerlendirme ile sunulmaktadır**. Geliştiriciye nasıl yapması gerektiğinden çok, yol gösterme niteliğindedir.

Bu denetim ve analiz çalışması kapsamında aşağıdaki çalışmalar gerçekleştirilmiştir;

- Penetrasyon testi için kapsam belirlenmiş, kullanılacak araçlar sunucu yapısı, işleyişi ve işlevselliği ilk planda tutularak ve gereksinimler göz önüne alınarak yapılandırılmıştır.
- Sunucuda pasif nitelikli taramalar yapılarak, uygulamanın çalışmış olduğu sistem hakk anlamda bilgiler edinilmiştir.
- Penetrasyon testleri yazılımsal ve mantıksal olarak iki aşamada ele alınmıştır.
- Her test "Üye Girişi Olmadan" ve "Üye Giriş Yapılarak" en az yetki ve normal yetkil denenmiş, sonuçlar raporlanmıştır.
- Sunucu üzerinde herhangi bir gereksiz yük yaratmamak adına testler "00:00 – 0 arasında, bir seferde yanlızca tek bir yazılım çalıştırılarak test edilmiştir.
- Tüm testlerde en az zararı verecek veya hiç bir zararı olmayan betikler denen
- Otomatik araçlardan üretilen sonuçlar hatalı bir bulgu içerebileceğinden tek ve tekrar eden durumlar tek bir madde içerisinde özetlenmiştir.

"xyz.com" sisteminde yapılan incelemelerde genel olarak "**Veri Girişi Kontrolü, P** mekanizmalarında eksiklikler bulunmuştur. Bu bulgular "YÜKSEK" seviyeli **Yetki KOD/Sistem Yapısı Çözümleme** açıklıkları yaratmaktadır. Testler sonucunda sis hakkı edinilebilmiştir.

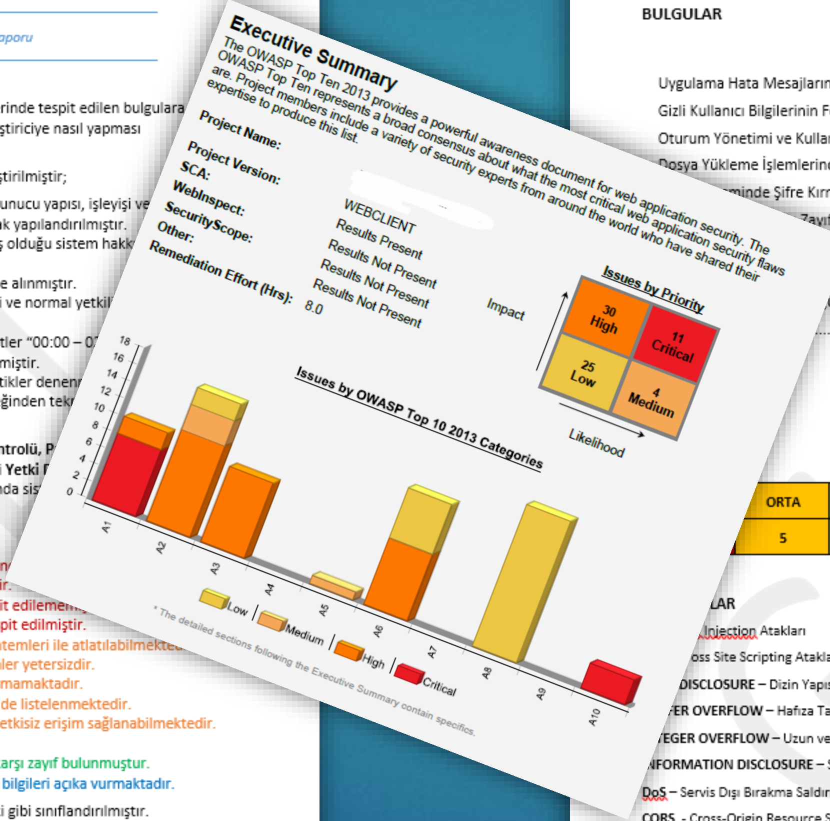
Bulgu Özeti:

1. Hata yönetimi eksik yapılmaktadır, sistem ve kod yapısı hakkın
2. Tüm kullanıcıların şifre bilgileri kolaylıkla elde edilebilmektedir.
3. Dosya yükleme aşamasında herhangi bir uzantı kontrolü tespit edileme
4. Oturum çerez bilgilerinin sayfa içerisinde direkt kullandığı tespit edilmiştir.
5. Sisteme "Giriş" yapısında kilitleme mekanizması spoofing yöntemleri ile atlatılabilmekte
6. Giriş mekanizmasında tahmin/zorlama ataklarına karşı önlemler yetersizdir.
7. Veri giriş alanlarındaki zararlı karakter kontrolleri sağlıklı çalışmamaktadır.
8. Menüde yüksek yetki isteyen sayfalar normal kullanıcılar için de listelenmektedir.
9. URL düzeyinde yer alan parametreler/ID bilgileri değiştirilip yetkisiz erişim sağlanabilmektedir.
10. Cookie bilgilerinin güvenliği sağlanmamıştır.
11. Sisteme ait sertifika bilgileri eski ve araya sızma saldırılarına karşı zayıf bulunmuştur.
12. Hatalı ASP.NET konfigürasyonları güvenlik açığı yaratabilecek bilgileri açıka vurmaktadır.

Yukarıda yer alan ve 12 ana konu başlığı ile ifade edilen bulgular aşağıdaki gibi sınıflandırılmıştır.

YÜKSEK	ORTA	DÜŞÜK	SİSTEM

Saygılarımızla,
Murat KAYA
Yazılım Güvenlik Uzmanı
Bimar Bilgi İşlem Hizmetleri A.Ş.

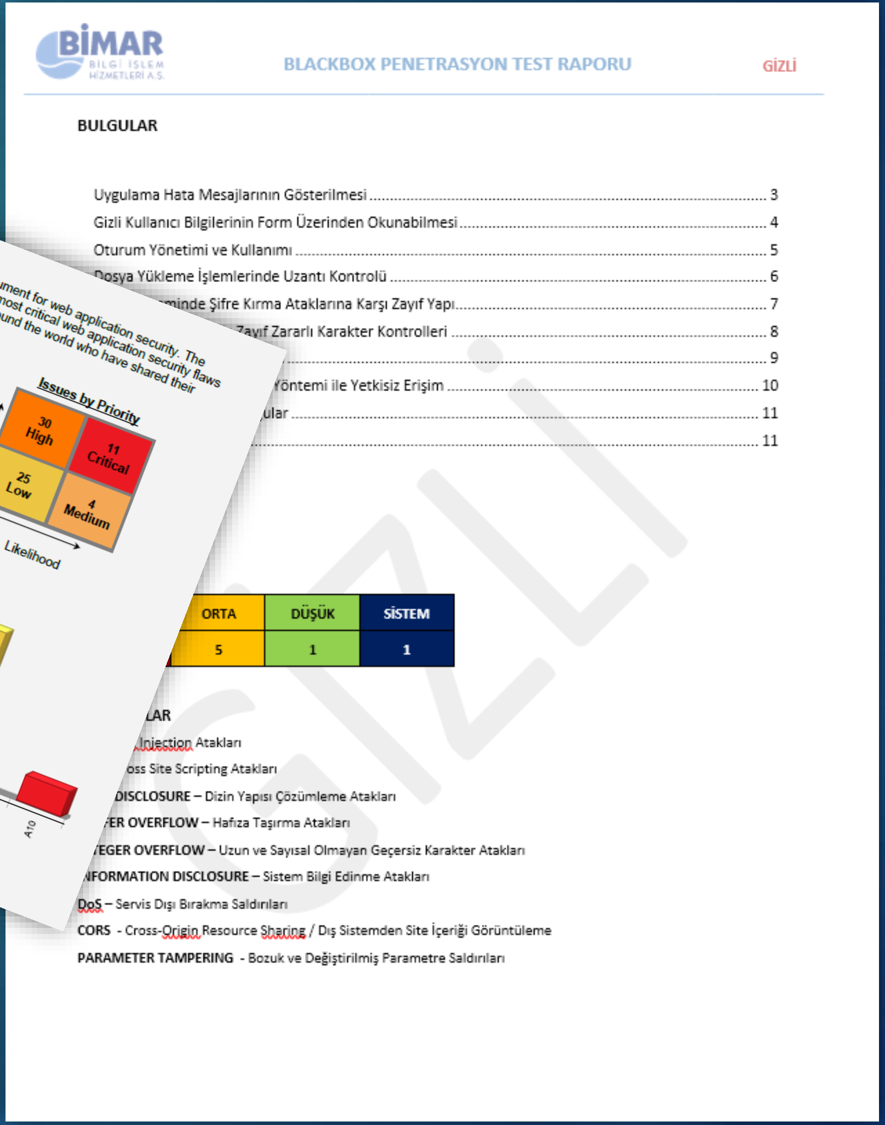


Executive Summary

The OWASP Top Ten 2013 provides a powerful awareness document for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

Project Name: [redacted]
Project Version: WEBCLIENT
SCA: Results Present
WebInspect: Results Not Present
SecurityScope: Results Not Present
Other: Results Not Present
Remediation Effort (Hrs): 8.0

* The detailed sections following the Executive Summary contain specifics.



BLACKBOX PENETRASYON TEST RAPORU GİZLİ

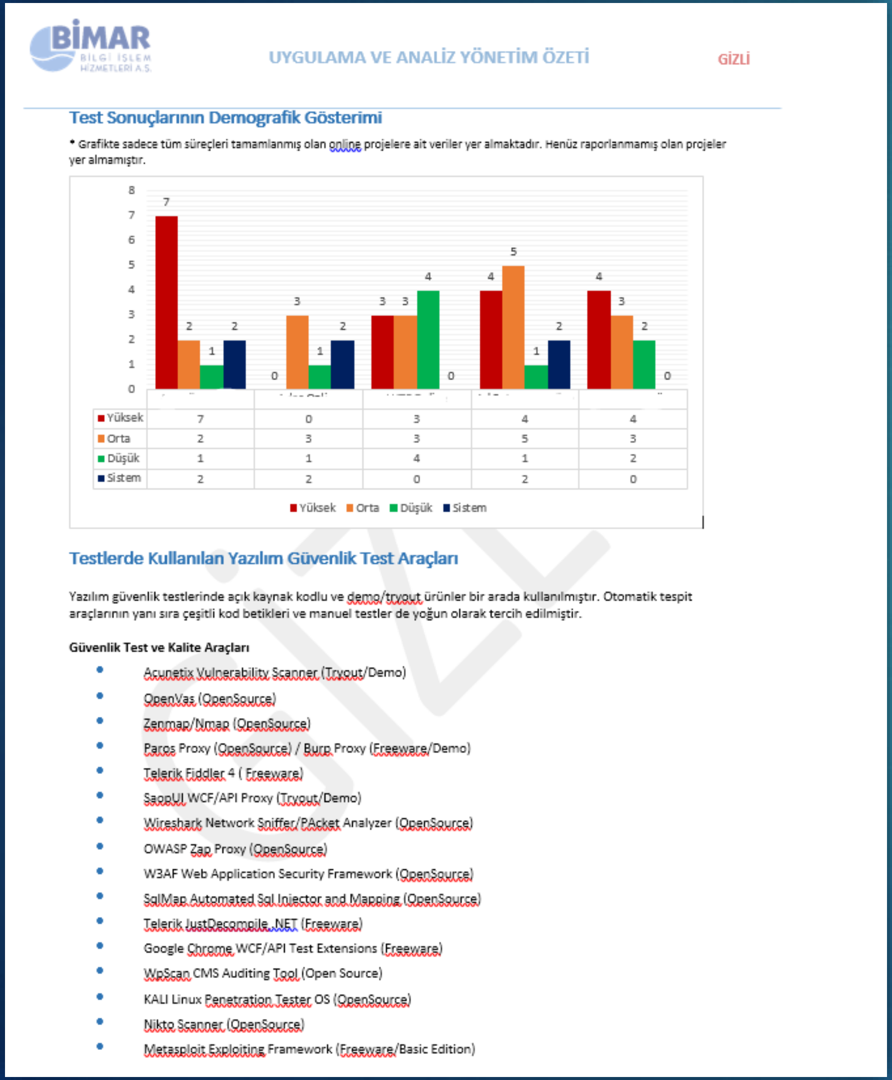
BULGULAR

ORTA	DÜŞÜK	SİSTEM
5	1	1

LAR

Injection Atakları
oss Site Scripting Atakları
DISCLOSURE – Dizin Yapısı Çözümleme Atakları
FER OVERFLOW – Hafıza Taşırma Atakları
EGER OVERFLOW – Uzun ve Sayısal Olmayan Geçersiz Karakter Atakları
NFORMATION DISCLOSURE – Sistem Bilgi Edinme Atakları
DoS – Servis Dışı Bırakma Saldırıları
CORS - Cross-Origin Resource Sharing / Dış Sistemden Site İçeriği Görüntüleme
PARAMETER TAMPERING - Bozuk ve Değiştirilmiş Parametre Saldırıları

Sample Analysis Results - Annual



BİMAR BİLGİ İŞLEM HİZMETLERİ A.Ş.

UYGULAMA VE ANALİZ YÖNETİM ÖZETİ

GİZLİ

Test Sonuçlarının Demografik Gösterimi

* Grafikte sadece tüm süreçleri tamamlanmış olan online projelere ait veriler yer almaktadır. Henüz raporlanmamış olan projeler yer almamıştır.

🟥 Yüksek	7	0	3	4	4
🟧 Orta	2	3	3	5	3
🟩 Düşük	1	1	4	1	2
🟦 Sistem	2	2	0	2	0

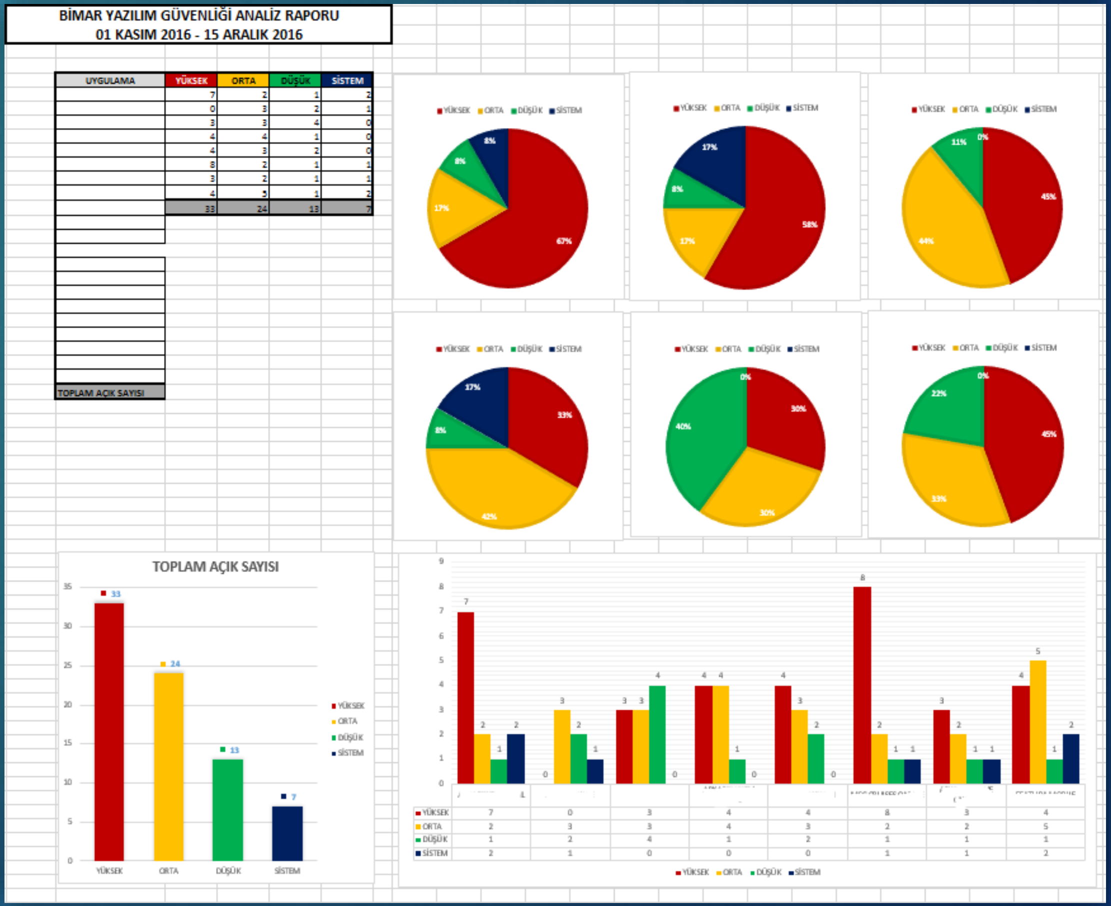
🟥 Yüksek 🟧 Orta 🟩 Düşük 🟦 Sistem

Testlerde Kullanılan Yazılım Güvenlik Test Araçları

Yazılım güvenlik testlerinde açık kaynak kodlu ve demo/tryout ürünler bir arada kullanılmıştır. Otomatik tespit araçlarının yanı sıra çeşitli kod betikleri ve manuel testler de yoğun olarak tercih edilmiştir.

Güvenlik Test ve Kalite Araçları

- Acunetix Vulnerability Scanner (Tryout/Demo)
- OpenVas (OpenSource)
- Zenmap/Nmap (OpenSource)
- Paros Proxy (OpenSource) / Burp Proxy (Freeware/Demo)
- Telerik Fiddler 4 (Freeware)
- SaopUI WCF/API Proxy (Tryout/Demo)
- Wireshark Network Sniffer/PAcket Analyzer (OpenSource)
- OWASP Zap Proxy (OpenSource)
- W3AF Web Application Security Framework (OpenSource)
- SqlMap Automated Sql Injector and Mapping (OpenSource)
- Telerik JustDecompile .NET (Freeware)
- Google Chrome WCF/API Test Extensions (Freeware)
- WpScan CMS Auditing Tool (Open Source)
- KALI Linux Penetration Tester OS (OpenSource)
- Nikto Scanner (OpenSource)
- Metasploit Exploiting Framework (Freeware/Basic Edition)



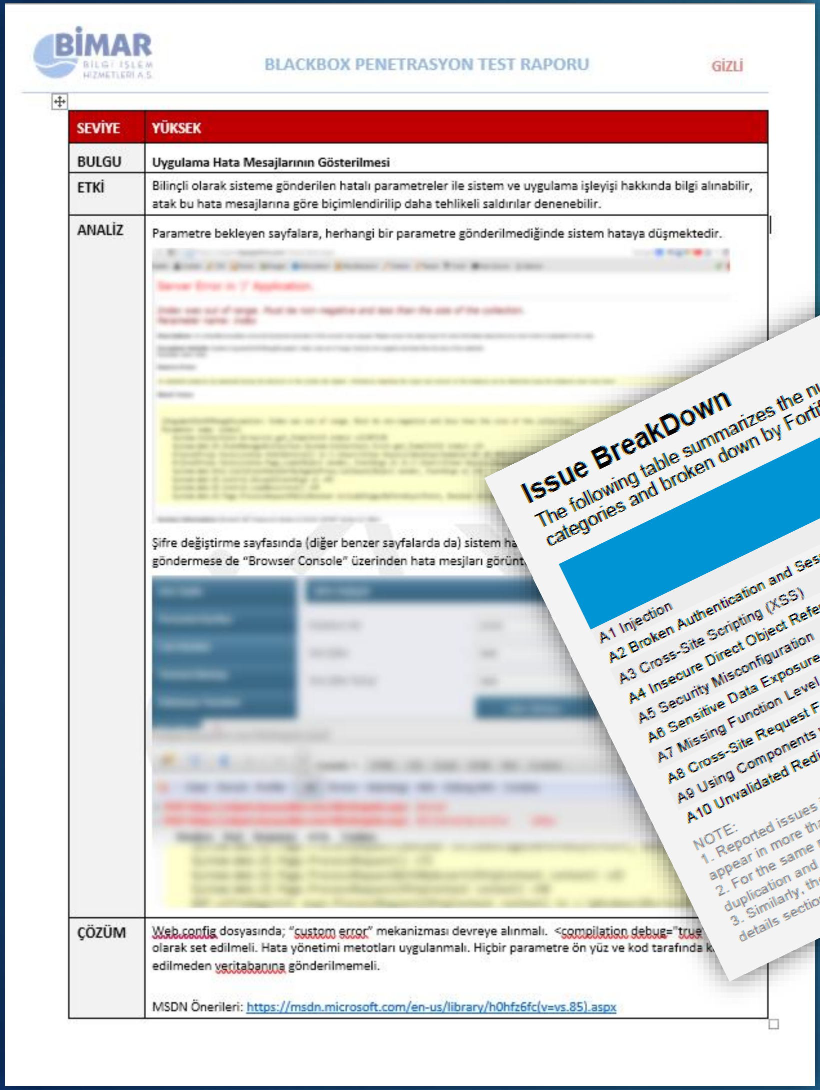
BİMAR YAZILIM GÜVENLİĞİ ANALİZ RAPORU
01 KASIM 2016 - 15 ARALIK 2016

UYGULAMA	YÜKSEK	ORTA	DÜŞÜK	SİSTEM
	7	2	1	2
	0	3	2	1
	3	3	4	0
	4	4	1	0
	4	3	2	0
	8	2	1	1
	3	2	1	1
	4	5	1	2
	33	24	13	7
TOPLAM AÇIK SAYISI				

🟥 YÜKSEK	7	0	3	4	4	8	3	4
🟨 ORTA	2	3	3	4	3	2	2	5
🟩 DÜŞÜK	1	2	4	1	2	1	1	1
🟦 SİSTEM	2	1	0	0	0	1	1	2

🟥 YÜKSEK 🟨 ORTA 🟩 DÜŞÜK 🟦 SİSTEM

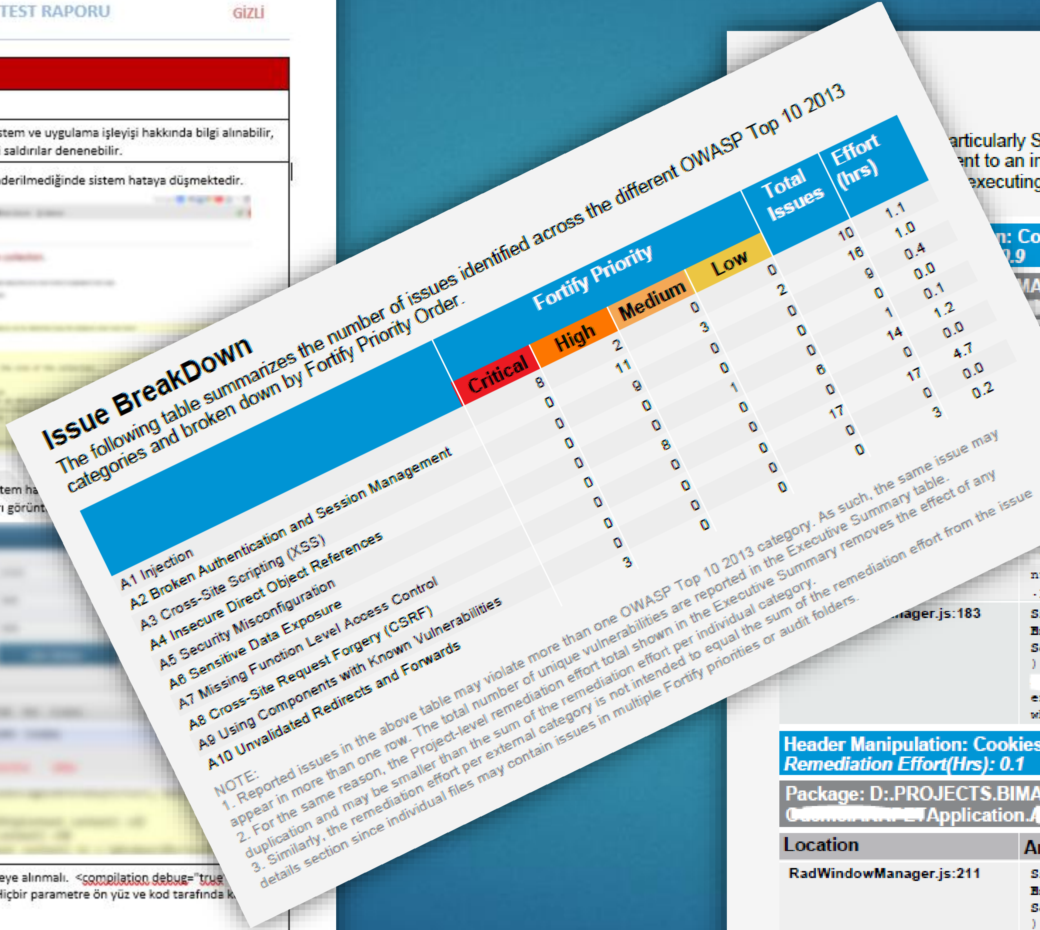
Sample Analysis Results - Developer



BLACKBOX PENETRASYON TEST RAPORU

GİZLİ

SEVİYE	YÜKSEK
BULGU	**Uygulama Hata Mesajlarının Gösterilmesi**
ETKİ	Bilinçli olarak sisteme gönderilen hatalı parametreler ile sistem ve uygulama işleyişi hakkında bilgi alınabilir, atak bu hata mesajlarına göre biçimlendirilip daha tehlikeli saldırılar denenebilir.
ANALİZ	Parametre bekleyen sayfalara, herhangi bir parametre gönderilmediğinde sistem hataya düşmektedir. Şifre değiştirme sayfasında (diğer benzer sayfalarda da) sistem ha göndermese de "Browser Console" üzerinden hata mesjları görünt
ÇÖZÜM	Web.config dosyasında; "custom error" mekanizması devreye alınmalı. <compilation debug="true olarak set edilmeli. Hata yönetimi metotları uygulanmalı. Hiçbir parametre ön yüz ve kod tarafında k edilmeden veritabanına gönderilmemeli. MSDN Önerileri: https://msdn.microsoft.com/en-us/library/h0hfz6fc(v=vs.85).aspx

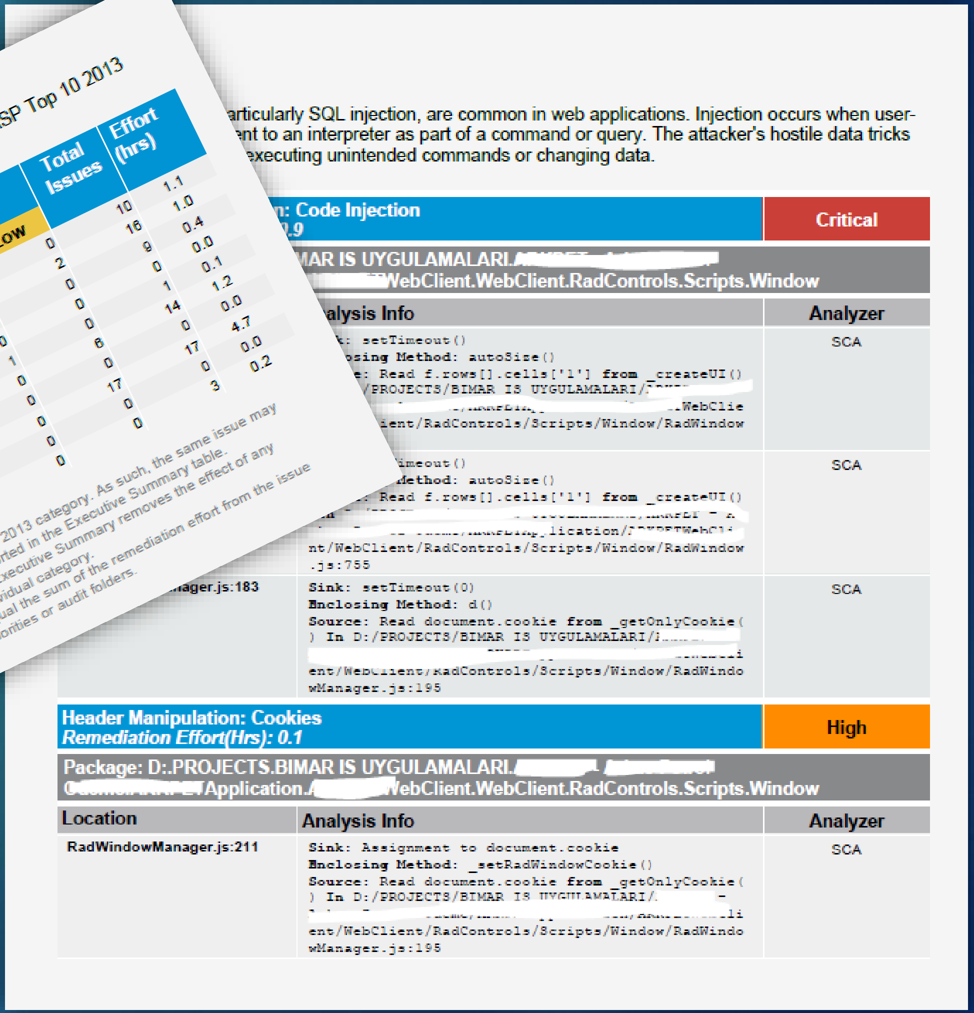


Issue BreakDown

The following table summarizes the number of issues identified across the different OWASP Top 10 2013 categories and broken down by Fortify Priority Order.

	Fortify Priority				Total Issues	Effort (hrs)
	Critical	High	Medium	Low		
A1 Injection	8	2	0	0	10	1.1
A2 Broken Authentication and Session Management	0	11	3	2	16	1.0
A3 Cross-Site Scripting (XSS)	0	9	0	0	9	0.4
A4 Insecure Direct Object References	0	0	0	0	0	0.0
A5 Security Misconfiguration	0	0	1	0	1	0.1
A6 Sensitive Data Exposure	0	8	0	6	14	1.2
A7 Missing Function Level Access Control	0	0	0	0	0	0.0
A8 Cross-Site Request Forgery (CSRF)	0	0	0	17	17	4.7
A9 Using Components with Known Vulnerabilities	0	0	0	0	0	0.0
A10 Unvalidated Redirects and Forwards	3	0	0	0	3	0.2

NOTE:
1. Reported issues in the above table may violate more than one OWASP Top 10 2013 category. As such, the same issue may appear in more than one row. The total number of unique vulnerabilities are reported in the Executive Summary table.
2. For the same reason, the Project-level remediation effort total shown in the Executive Summary removes the effect of any duplication and may be smaller than the sum of the remediation effort per individual category.
3. Similarly, the remediation effort per external category is not intended to equal the sum of the remediation effort from the issue details section since individual files may contain issues in multiple Fortify priorities or audit folders.



articularly SQL injection, are common in web applications. Injection occurs when user-
ent to an interpreter as part of a command or query. The attacker's hostile data tricks
executing unintended commands or changing data.

n: Code Injection — **Critical**
.9

MAR IS UYGULAMALARI.[redacted]
[redacted]WebClient.WebClient.RadControls.Scripts.Window

Location	Analysis Info	Analyzer
	k: setTimeout() osing Method: autoSize() e: Read f.rows[].cells['1'] from _createUI() /PROJECTS/BIMAR IS UYGULAMALARI/[redacted]WebClie ient/RadControls/Scripts/Window/RadWindow	SCA
	imeout() Method: autoSize() Read f.rows[].cells['1'] from _createUI() [redacted]lication/[redacted] nt/WebClient/RadControls/Scripts/Window/RadWindow .js:755	SCA
ager.js:183	Sink: setTimeout(0) Enclosing Method: d() Source: Read document.cookie from _getOnlyCookie( ) In D:/PROJECTS/BIMAR IS UYGULAMALARI/[redacted] ent/WebClient/RadControls/Scripts/Window/RadWindo wManager.js:195	SCA

Header Manipulation: Cookies — **High**
Remediation Effort(Hrs): 0.1

Package: D:.PROJECTS.BIMAR IS UYGULAMALARI.[redacted]
[redacted]Application.[redacted]WebClient.WebClient.RadControls.Scripts.Window

Location	Analysis Info	Analyzer
RadWindowManager.js:211	Sink: Assignment to document.cookie Enclosing Method: _setRadWindowCookie() Source: Read document.cookie from _getOnlyCookie( ) In D:/PROJECTS/BIMAR IS UYGULAMALARI/[redacted] ent/WebClient/RadControls/Scripts/Window/RadWindo wManager.js:195	SCA

Managing Vulnerabilities

- TFS (Team Foundation Server)
  - Just «critical» vulnerabilities
  - Automatically opened «Bug» type workitems
  - «High» priority
- HPE Fortify SSC (Software Security Center)
  - All vulnerabilities with categorized dashboards
  - All kind of reports can be generated
  - Historically scan results and metrics
  - Automatic/Manual assigment to team members

Q & A ?