



Make your developers the first line of defence for application security

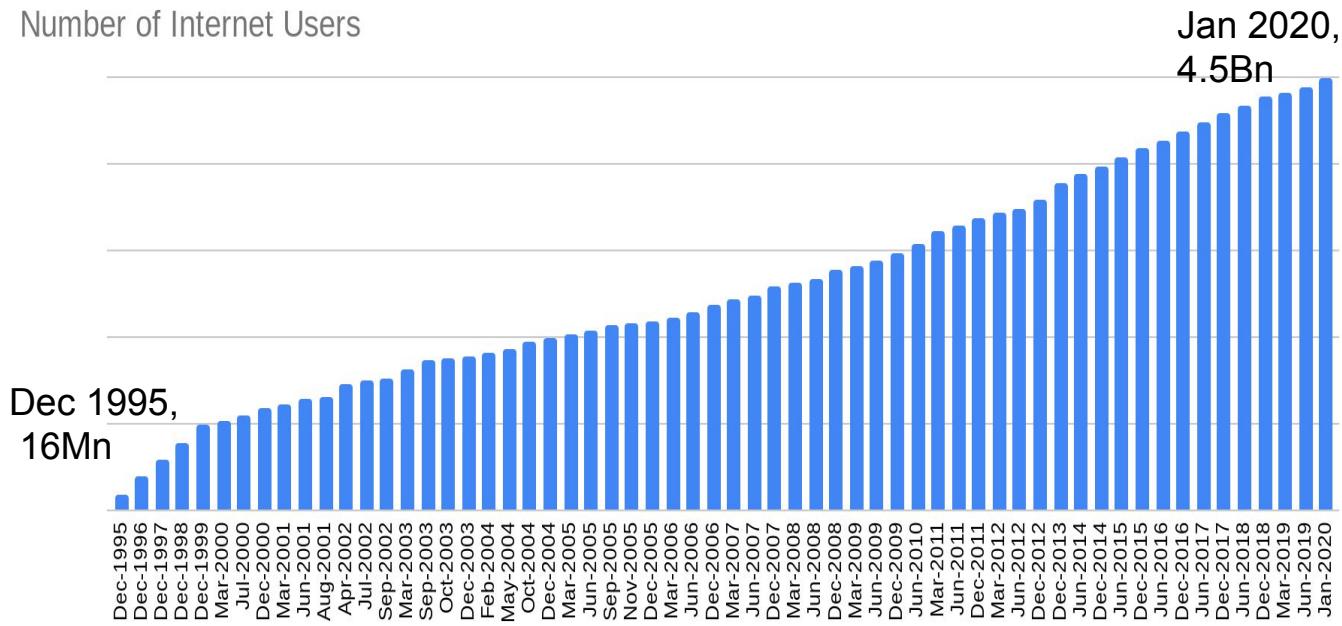
Klaudia Zabek

kzabek@securecodewarrior.com

Partnerships Manager

Digital Landscape is growing

Number of Internet Users



Internet World Stats

Top Mobile Publishers Worldwide for July 2020 by Downloads

Overall Downloads	App Store Downloads	Google Play Downloads
1 Google	1 Google	1 Google
2 Facebook	2 Tencent	2 Facebook
3 Voodoo	3 Voodoo	3 Voodoo
4 Outfit7	4 Voodoo	4 Outfit7
5 Crazy Labs	5 Bytedance	5 Crazy Labs
6 TikTok	6 Bytedance	6 TikTok
7 AppLovin	7 Alibaba	7 AppLovin
8 InShot	8 Microsoft	8 SayGames
9 SayGames	9 Amazon	9 Bytedance
10 Tencent	10 Douyin	10 BabyBus

Note: Does not include downloads from third-party Android stores in China or other regions. TikTok includes downloads of Douyin.

Sensor Tower Data That Drives App Growth

sensortower.com

Why Application Security Training?



The Application Layer is the fastest growing attack vector !

In a 2019 Forrester Research survey, 42% of organizations asked who had experienced an external attack blamed the incident on a software security flaw.

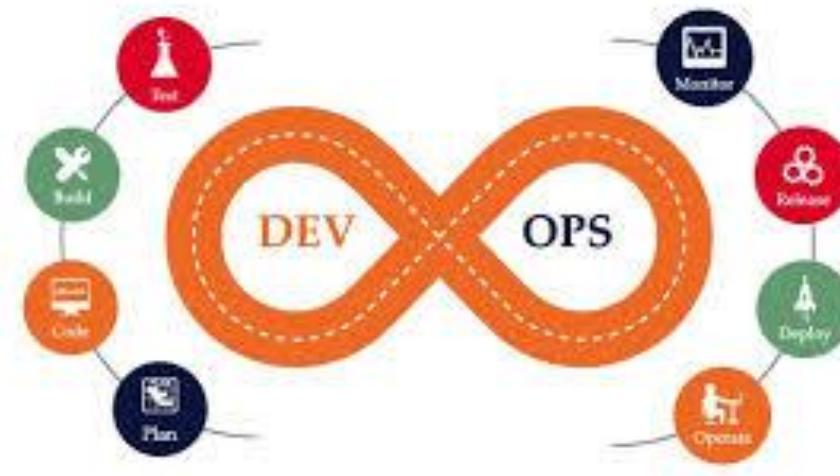
20,000: Number of times the average web app was attacked, January and February 2020



We are investigating the theft of customer data from our website and our mobile app, as a matter of urgency. For more information, please click the following link:

Data Breach at Sears and Delta May Have 'Several Hundred Thousand' Customers



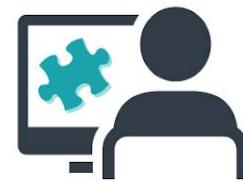


Demanding release cycles leave little room for error, and a critical security breach can delay or stop a release schedule

Maintain high production volumes and their fast coding, and now develop even more securely and incorporate them into every application you create



SAST DAST IAST



SECURITY EXPERTS
TEST AND FIND
VULNERABILITIES

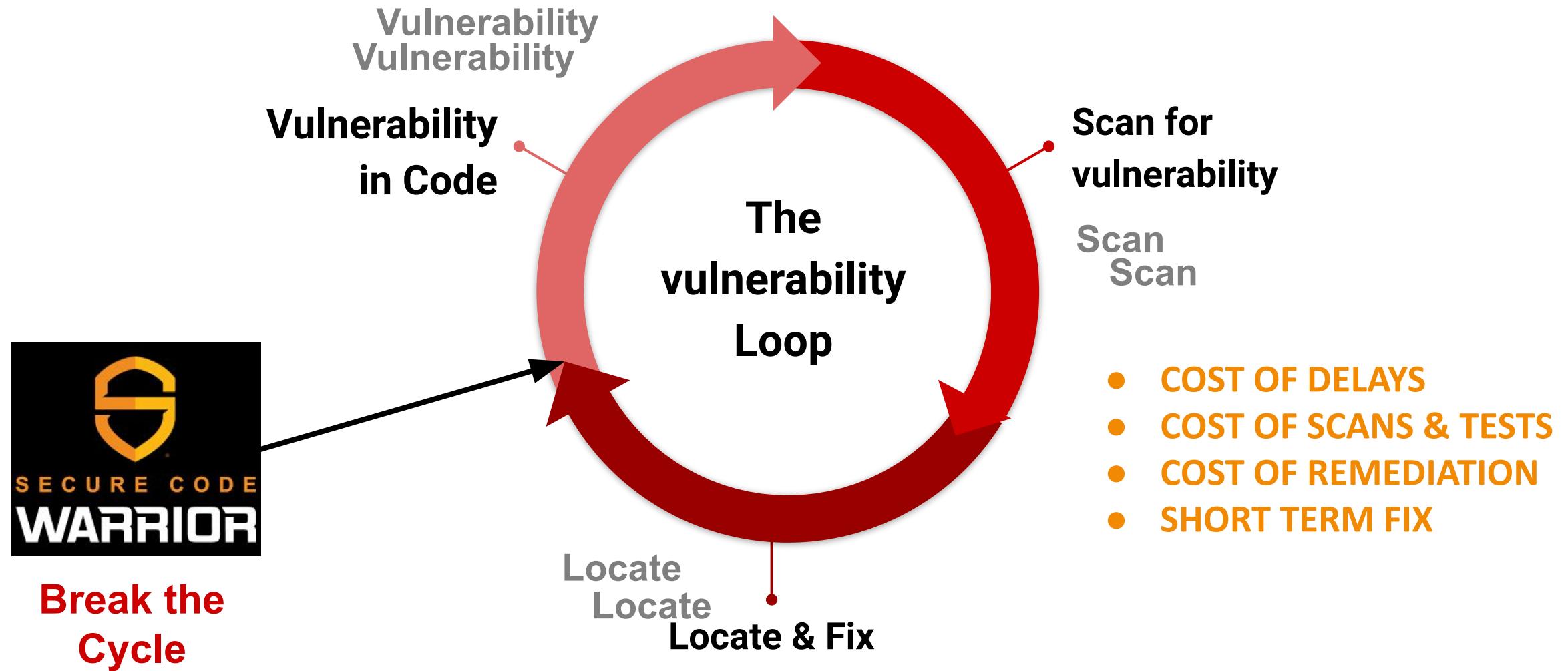
RESULTS ARE LOADED
INTO THE BUG
TRACKING SYSTEM

DEVELOPER
FINDS WAY TO FIX THE
PROBLEM

KNOWLEDGE
DISAPPEARS INTO A
BLACK HOLE

BUG
REAPPEARS

The Vulnerability Loop



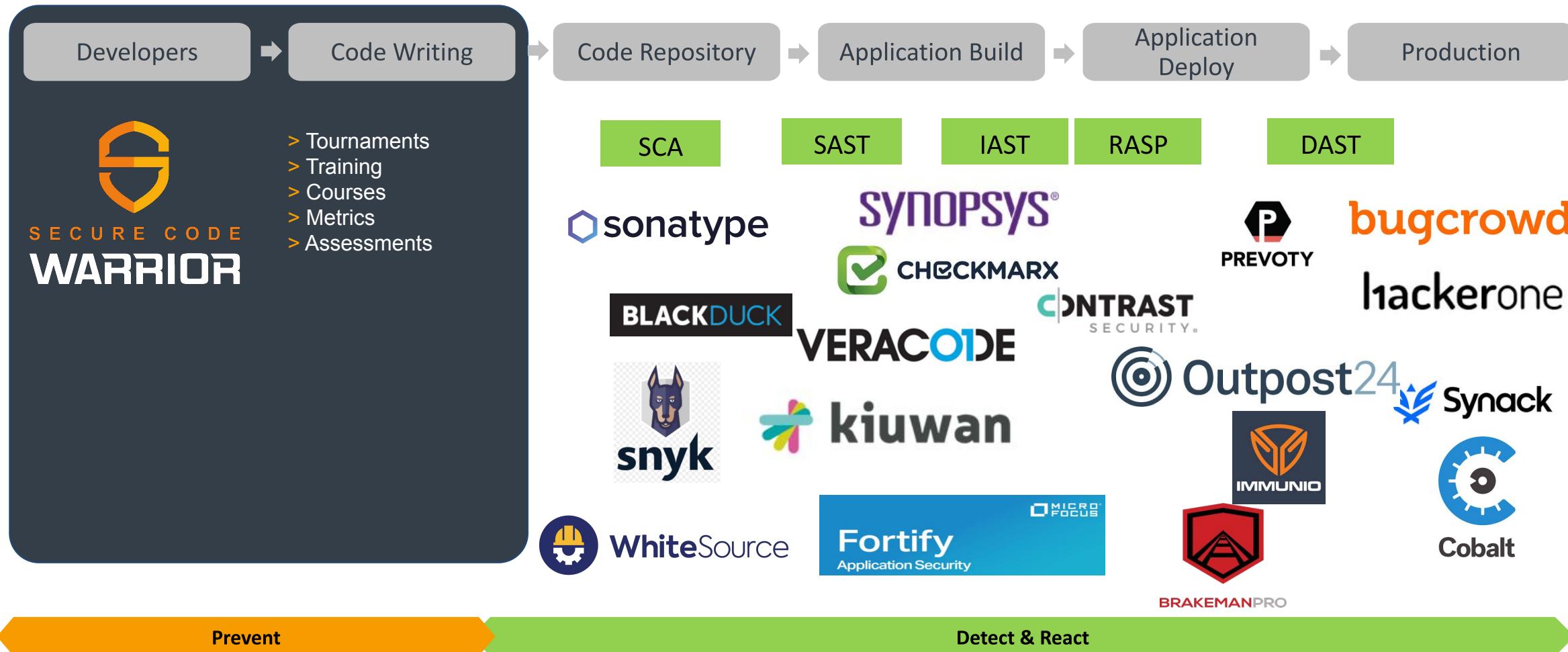
Why Application Security Training?



**TESTING ALONE IS
NOT THE
ANSWER...**

**SOFTWARE NEEDS
TO BE CODED
SECURELY FROM
THE START**

Starting Left In the SDLC



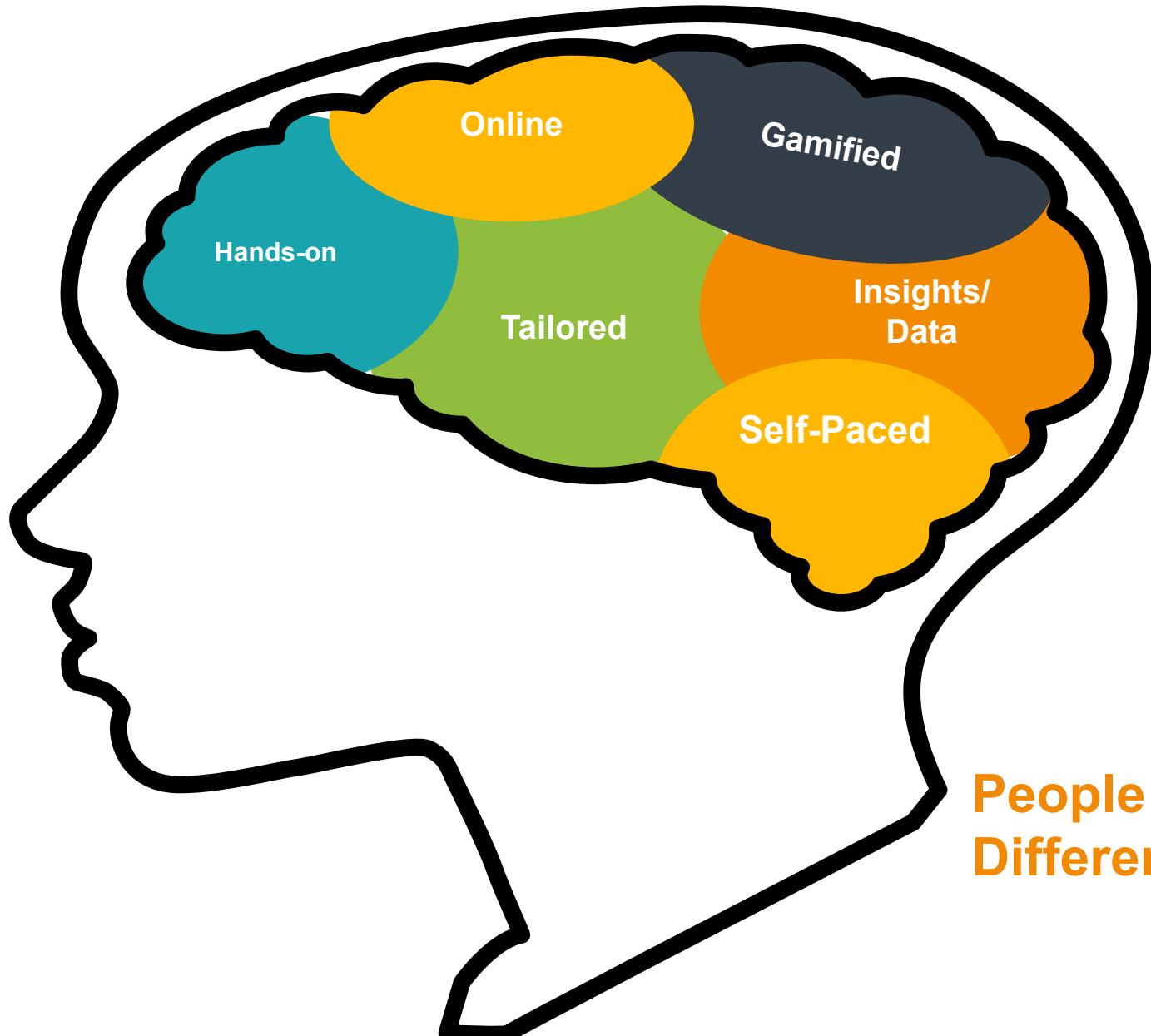
Cost to fix a security bug

SHIFTING FROM REACTION TO PREVENTION WITH ONE POWERFUL SECURE CODING PLATFORM



A DIFFERENT APPROACH

**Built by developers
for developers**



**People Learn In
Different Ways**

Secure Code Warrior is the smartest and easiest way to improve your application security program.

Learning Resources

Get started with security fundamentals and application security concepts.

60+ e-learning videos and presentation resources, covering security fundamentals, mobile and web application security weaknesses.

Training

Build secure coding skills with interactive language:framework specific coding challenges.

Grow awareness in identifying vulnerabilities and how they work level up skills in locating vulnerabilities during code review, and finally, understand how to mitigate and fix the vulnerability.

Courses

Curated learning-pathways to build competency within your overall cyber security program.

Configure and assign training activities to assist in achieving compliance requirements, like NIST and PCI-DSS or target specific skill gaps.



Tournaments

Create awareness and drive continuous engagement for secure coding

Run competitive and engaging events that get the whole coding community involved.

Assessments

Verify secure coding skills in a fully customisable and controllable environment

Be confident that your developers have a base level of competency when it comes to securing your code. Qualify and benchmark the secure coding skills of your existing developers, off-shore developers, new hires and graduates.

Data and Insights

Reporting to track and monitor training progress across your organisation

Role specific dashboards, pre-built reports and reporting API make it easy to measure and analyse, team and company performance, plus personal skills development.

Integrations (APIs)

Connect with your core business systems to streamline your workflow

Streamline user management and save time by programmatically managing users and building management reports within your existing toolset with RESTful APIs.

Make training hands-on

The screenshot shows a user interface for a training platform. At the top, there's a navigation bar with links for Home, Tournaments, Training (which is highlighted in orange), Courses, Assessments, Sensei, and Resources. On the far right, there are links for Metrics, Administration, Help, and a user profile icon. Below the navigation, a progress bar indicates three steps: 'Locate Vulnerability' (step 1), 'Identify Solution' (step 2), and 'Challenge Complete' (step 3). A large 'Stage Completed' badge is prominently displayed. To the left of the badge, a message discusses a stored XSS vulnerability, mentioning the sanitizeHTML method and its configuration. Below this message are two thumbs-up icons. At the bottom of this section are buttons for 'Review Solutions' and 'Continue'. To the right of the progress bar is a file tree and a code editor. The file tree shows a directory structure with folders like api, e2e, src (containing app, components, guards, interfaces, routing, services, validators, assets, environments, favicon.ico, index.html, main.ts, polyfills.ts, and styles.css), and files like app.component.css, app.component.html, app.component.spec.ts, and app.module.ts. The code editor displays a portion of a TypeScript file named single-reply.component.ts. The code is annotated with several yellow warning signs, indicating potential security issues or code smells. The code itself handles user authentication, reply management, and sanitization logic.

```
1 import { Observable } from 'rxjs/Rx';
2 import sanitizeHtml from 'sanitize-html';
3 import { QuestionService } from '../../../../../services/question/question.service';
4 import { UserService } from '../../../../../services/user/user.service';
5 import { Component, Input, Output, EventEmitter, OnInit } from '@angular/core';
6 import { User, Reply } from '../../../../../interfaces';
7
8 @Component({
9   selector: 'app-single-reply',
10  templateUrl: './single-reply.component.html',
11  styleUrls: ['./single-reply.component.css']
12 })
13 export class SingleReplyComponent implements OnInit {
14
15   private authUser: User;
16
17   @Input('reply')
18   reply: Reply;
19
20   @Input('index')
21   index: any;
22
23   @Input('questionId')
24   questionId: string;
25
26   @Output('deleteReplyEvent')
27   deleteReplyEvent = new EventEmitter<{questionId: string, replyId: string}>();
28
29   constructor(private userService: UserService, private questionService: QuestionService) {
30     this.userService.userChange.subscribe((value) => {
31       this.authUser = value;
32     });
33   }
34
35   ngOnInit() {
36     const username = this.reply.user ? this.reply.user.username : '';
37     const allElements = document.getElementsByClassName('reply-body');
38     const lastElement = allElements[this.index];
39
40     const replyBody = sanitizeHtml(`<a href="#" class="anchor-username">${this.authUser.username}</a>`);
41   }
42 }
```

Example challenge

The screenshot shows a dark-themed user interface for a training or learning platform. At the top, there is a navigation bar with the following items from left to right: a shield icon, Home, Tournaments, Training (which is highlighted in orange), Courses, Assessments, Sensei LABS, Resources, Metrics, Administration, Help, and a user profile icon.

The main content area displays a grid of challenge categories:

- C# (.NET) Core**: 396 Challenges
- C# (.NET) MVC**: 1383 Challenges
- C# (.NET) Basic**: 120 Challenges
- C# (.NET) Web API**: 141 Challenges
- C# (.NET) Web Forms** (ACTIVE): 1143 Challenges
- CloudFormation Basic**: 108 Challenges
- C**
- C# (.NET) Core**
- CloudFormation Basic**

A yellow "Get Started" button with a "4" badge is located at the bottom left. The "C# (.NET) Web Forms" category is highlighted with a yellow border and has the word "ACTIVE" above it. The "1143 Challenges" text in this category is also highlighted with a yellow border.

[Home](#)[Tournaments](#)[Training](#) ▾[Courses](#) ▾[Assessments](#)[Sensei](#) ▾[Resources](#)[Metrics](#) ▾[Administration](#)[Help](#) ▾

C# (.NET) Web Forms



Select a level to play. Each level will have a different set of quests to complete.

OWASP Web Top 10 2017

Learn the ropes or hone your skills in secure programming here. This set of levels will focus on individual vulnerability categories so that you can practise finding and fixing certain types of issues.

1

OWASP A1-A2

ACTIVE

Let's start with the most critical application weaknesses. These challenges get you the foundations of 1: Injection Flaws and 2: Broken Authentication vulnerabilities

2

OWASP A3-A4

Let's continue with some other very common application weaknesses. This set of levels will focus on 3: Sensitive Data Exposure and 4: XXE vulnerabilities

3

OWASP A5-A7

Learn the ropes or hone your skills in secure programming here. These challenges will give you an understanding of 5: Broken Access Control, 6: Security Misconfiguration and 7: XSS vulnerabilities

4

OWASP A8-A10

Last but not least. these set challenges consist of 8: Insecure deserialisation, 9: Using Components with Known Vulnerabilities, 10: Insufficient Logging and Monitoring

[Get Started](#)

Training Ground

Learn the ropes or hone your skills in secure programming here. This set of levels will focus on individual vulnerability categories so that you can practise finding and fixing certain types of issues.

[Home](#)[Tournaments](#)[Training](#) ▾[Courses](#) ▾[Assessments](#)[LABS](#)[Sensei](#) ▾[Resources](#)[Metrics](#) ▾[Administration](#)[Help](#) ▾[C# \(.NET\) Web Forms](#)

Level OWASP A1-A2

Accuracy Security Maturity

1 points

Active Missions

A1 - Injection: A Hacker from 🇫🇷 France is attacking the C# (.NET) Web Forms Code Snippets application. [VIEW](#)

A2 - Injection: A state-sponsored adversary from 🇨🇩 Dem. Rep. Congo is attacking the C# (.NET) Web Forms Code Snippets application. [View](#)

This map is based on public domain map data available from JVectorMap and Natural Earth

A-Team Leaderboard

Developer names have been anonymised by your company administrator

Rank	Name	points
11	Klaudia Zabek	391
12	Adamantium Halcyon	386
13	Chunky Greathornedowl	344
14	Allstar Treecreeper	245
15	Odorful Seamonkey	222
16	Proacademic Hamadryas	211
17	Selfassured Stegosaurus	185
18	Anthropoidal Mamba	165
19	Unspecialized Tapaculo	101
20	Couped Bordercollie	62



Home

Tournaments

Training

Courses

Assessments

LABS
Sensei

Resources

Metrics

Administration

Help

C# (.NET) Web Forms



1

2



Locate Vulnerability

Identify Solution

Challenge Complete

Locate Vulnerability

Identify and select the code blocks that cause the vulnerability listed below by clicking the next to the line numbers in the code viewer.

Files containing selectable code blocks have been marked with .

Vulnerability Category

Injection Flaws - OS Command Injection

Submit Your Answer

There is **1** vulnerable block in the source code that you need to locate.

You have selected **0** code blocks

[Skip](#)[Hint](#)[Next](#)[Get Started](#) 4

Quickswitch



Jump



images
Default.aspx
Default.aspx.cs
Web.config

```
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15         //Do nothing on page load
16     }
17
18     protected void CheckIfExists_Click(object sender, Eve
19     {
20         //Collect user input
21         string fileName = FileNameInput.Text.Trim();
22
23         // create and start the process
24         var process = new System.Diagnostics.Process { StartI
25         process.Start();
26
27         //Collect command output
28         string fileExists = "We are sorry, it appears your fi
29         if (process.StandardOutput.ReadToEnd().Contains("true
30             fileExists = "Alright! It looks like we received
31
32         //Wait for command output
33         process.WaitForExit();
34
35         //Return output to the page
36         Result.Text = "<pre>" + fileExists + "</pre>";
37         Result.Visible = true;
38     }
39
40     protected System.Diagnostics.ProcessStartInfo getProcess
```

[Home](#)[Tournaments](#)[Training](#) ▾[Courses](#) ▾[Assessments](#)[LABS](#)
Sensei ▾[Resources](#)[Metrics](#) ▾[Administration](#)[Help](#) ▾

C# (.NET) Web Forms



Locate Vulnerability



Identify Solution



Challenge Complete

Identify Solution

Determine the correct fix from a number of different proposed solutions for the vulnerability listed below. These solutions will be full code repositories, where completely different approaches may have been taken to address the problem.

Vulnerability Category

Injection Flaws - OS Command Injection

[Skip](#)[Hint](#)[View Solutions](#)[Accept](#)[Get Started](#) 4

Quickswitch



Jump



images
Default.aspx
Default.aspx.cs
Web.config

28 string fileExists = "we are sorry, it appears your request
29 if (process.StandardOutput.ReadToEnd().Contains("true"))
30 fileExists = "Alright! It looks like we received your
31 //Wait for command output
32 //Return output to the page
33 Result.Text = "<pre>" + fileExists + "</pre>";
34 Result.Visible = true;
35 }
36 //Build command
37 string cmd = "if exist \\" + AppDomain.CurrentDomain.
38 // create the process start info
39 return new System.Diagnostics.ProcessStartInfo("cmd",
40 {
41 // redirect the output
42 RedirectStandardOutput = true,
43 // do not use system shell
44 UseShellExecute = false,
45 // do not create any window
46 CreateNoWindow = true
47 };
48 }
49 }
50 }

[VIEW SOLUTIONS](#)



Home

Tournaments

Training

Courses

Assessments

LABS

Sensei

Resources

Metrics

Administration

Help

C# (.NET) Web Forms



Quickswitch

Jump

Vulnerable Code

Solution 3

Accept

Hint



- images
- Default.aspx
- *Default.aspx.cs
- Web.config

```
19  {
20      //Collect user input
21      string fileName = FileNameInput.Text.Trim();
22
23      // create and start the process
24      var process = new System.Diagnostics.Process { StartInfo =
25          process.Start();
26
27      //Collect command output
28      string fileExists = "We are sorry, it appears your file
29      if (process.StandardOutput.ReadToEnd().Contains("true"))
30          fileExists = "Alright! It looks like we received yo
31
32      //Wait for command output
33      process.WaitForExit();
34
35      //Return output to the page
36      Result.Text = "<pre>" + fileExists + "</pre>";
37      Result.Visible = true;
38  }
39
40  protected System.Diagnostics.ProcessStartInfo getProcessSta
41  {
42      //Build command
43      string cmd = "if exist \\\" + AppDomain.CurrentDomain.Ba
44
45      // create the process start info
46      return new System.Diagnostics.ProcessStartInfo("cmd", "/c
47      {
48          // redirect the output

```

```
25
26      // create the process start info
27      var procStartInfo = new System.Diagnostics.ProcessStart
28      {
29          // redirect the output
30          RedirectStandardOutput = true,
31          // do not use system shell
32          UseShellExecute = false,
33
34          // do not create any window
35          CreateNoWindow = true
36      };
37
38      // create and start the process
39      var process = new System.Diagnostics.Process { StartInfo =
40          process.Start();
41
42      //Collect command output
43      string fileExists = "We are sorry, it appears your file
44      if (process.StandardOutput.ReadToEnd().Contains("true"))
45          fileExists = "Alright! It looks like we received yo
46
47      //Wait for command output
48      process.WaitForExit();
49
50      //Return output to the page
51      Result.Text = "<pre>" + fileExists + "</pre>";
52      Result.Visible = true;
53  }
54 }
```

Get Started 4



Home

Tournaments

Training

Courses

Assessments

LABS
Sensei

Resources

Metrics

Administration

Help

C# (.NET) Web Forms



Locate Vulnerability

Identify Solution

Challenge Complete



🕒 01:59

Solution 3



That was incorrect. Give it another go!

In this case the developer thought HTML decoding the users input should be sufficient to mitigate the vulnerability. Unfortunately HTML decoding doesn't encode special characters and therefore the vulnerability still exists in this code snippet.



Get Started 4

Skip

Retry

Reveal Answer

Quickswitch ▲ ▼

Jump ⏪ ⏩

images
Default.aspx
*Default.aspx.cs
Web.config

```
13 protected void Page_Load(object sender, EventArgs e)
14 {
15     //Do nothing on page load
16 }
17
18 protected void CheckIfExists_Click(object sender, Eve
19 {
20     //Collect user input
21     string fileName = FileNameInput.Text.Trim();
22
23     // create and start the process
24     var process = new System.Diagnostics.Process { StartI
25     process.Start();
26
27     //Collect command output
28     string fileExists = "We are sorry, it appears your fi
29     if (process.StandardOutput.ReadToEnd().Contains("true
30         fileExists = "Alright! It looks like we received
31
32     //Wait for command output
33     process.WaitForExit();
34
35     //Return output to the page
36     Result.Text = "<pre>" + fileExists + "</pre>";
37     Result.Visible = true;
38 }
39
40 protected System.Diagnostics.ProcessStartInfo getProcessS
41 {
42     //Build command
43     string cmd = "if exist \\" + AppDomain.CurrentDomain
```



Home

Tournaments

Training

Courses

Assessments

LABS
Sensei

Resources

Metrics

Administration

Help

C# (.NET) Web Forms



01:59

Solution 2



.NET has many components to perform almost any operation available to the command line. There is usually no reason to choose a command shell over the native .NET components.



Get Started 4

Review Solutions

Continue

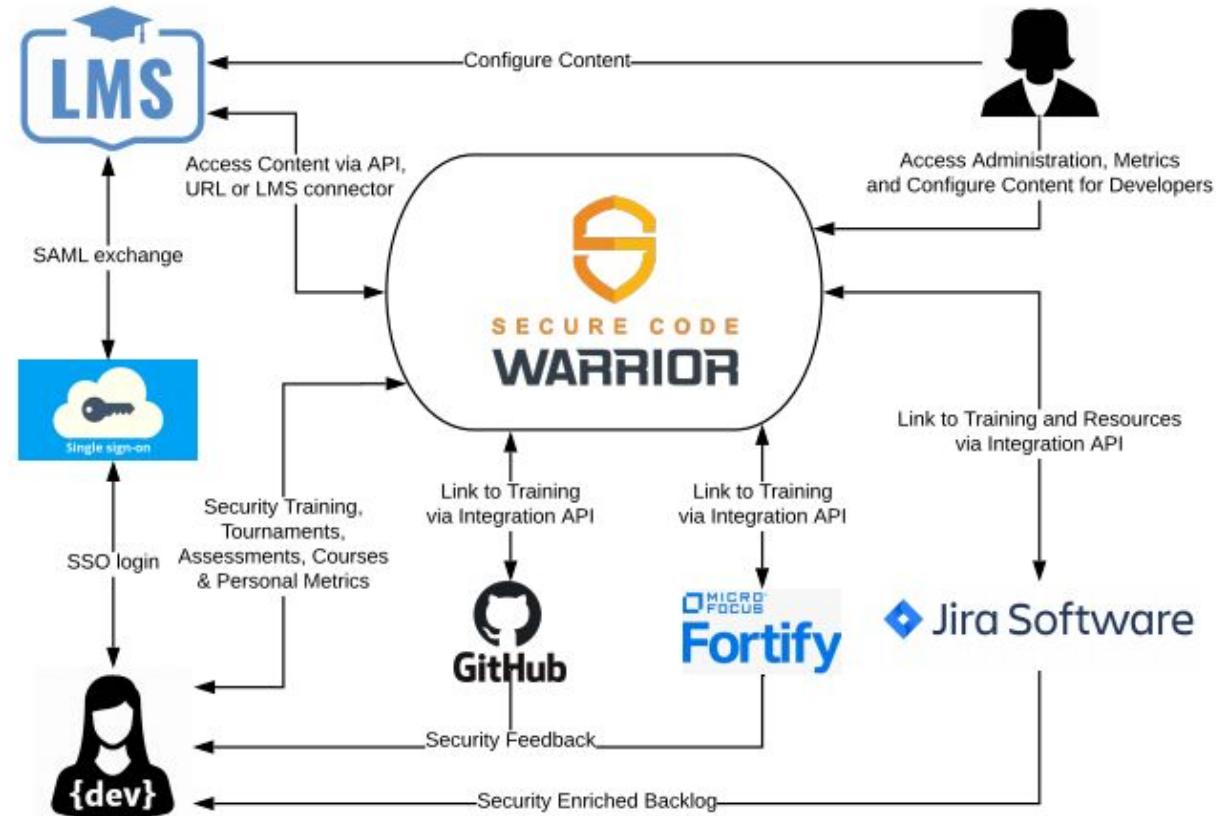
Quickswitch Jump

images
Default.aspx
***Default.aspx.cs**
Web.config

```
16    }  
17  
18    protected void CheckIfExists_Click(object sender, Eve  
19    {  
20        //Collect user input  
21        string fileName = FileNameInput.Text.Trim();  
22  
23        //create and start the process  
24        var process = new System.Diagnostics.Process { StartI  
25        process.Start();  
26  
27        //Collect command output  
28        string fileExists = "We are sorry, it appears your fi  
29        if (process.StandardOutput.ReadToEnd().Contains("true  
30        fileExists = "Alright! It looks like we received  
31  
32        //Wait for command output  
33        process.WaitForExit();  
34  
35        //Return output to the page  
36        Result.Text = "<pre>" + fileExists + "</pre>";  
37        Result.Visible = true;  
38    }  
39  
40    protected System.Diagnostics.ProcessStartInfo getProcess  
41    {  
42        //Build command  
43        string cmd = "if exist \\" + AppDomain.CurrentDomain.  
44  
45        // create the process start info  
return new System.Diagnostics.ProcessStartInfo("cmd")
```

Embed security training into dev day-to-day

- **Contextual learning theory** - effective learning only takes place when we process new information or new knowledge in such a way that it makes sense to us within our individual frame of reference.
- Integrate contextual micro learning challenges into tools like Jira, GitHub etc. to enable practical training based around specific vulnerabilities





Treadstone

TREAD board

Backlog

Active sprints

Releases

Reports

Issues

Components

PROJECT SHORTCUTS

Add a link to useful information for your whole team to see.

+ Add link



Treadstone / TREAD-10

SQL Injection (CWE 89) found in the latest commit for the user reports component

Edit

Comment

Assign

More ▾

To Do

In Progress

Done

Admin ▾

Export ▾

Details

Type:	Story	Status:	IN PROGRESS (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	Version 2.0
Labels:	CWE_89		
Sprint:	Sample Sprint 2		
SCW	C# (.NET):MVC		
Language/Framework:			

People

Assignee:	cw@scw.io
Reporter:	cw@scw.io
Votes:	0
Watchers:	Start watching this issue

Dates

Created:	22/Dec/19 4:02 PM
Updated:	Just now

Agile

Active Sprint:	Sample Sprint 2 ends 05/Jan/20
View on Board	

Hipchat discussions

Secure Code Warrior



SQL Injection

This is probably one of the two most exploited vulnerabilities in web applications and has led to a number of high profile company breaches. It occurs when an application fails to sanitize or validate input before using it to dynamically construct a statement. An attacker that exploits this vulnerability will be able to gain access to the underlying database and view or modify data without permission.

Level-up your secure coding prowess with language and framework specific gamified training.

[Train Now](#)

Thank you for participating in Secure Code Warrior Private Labs. Labs is where our more courageous warriors can play around with early releases of our new and exciting features and offer feedback directly to the team who develops them.

[Provide feedback](#)

Project settings

<>

Recommended Roll Out Approach

Engage your team and build a pathway of achievement



Don't Take My Word For It.....



One of the worlds largest **INTERNATIONAL BANKS** beginning with “H”

Tools:

- MPT
- SAST
- DAST
- IAST
- **8000+ DEVELOPERS**



80% REDUCTION in VULNERABILITIES in just **2 YEARS** by implementing Secure Code Warrior.



SECURE CODE
WARRIOR

Thank you!

Klaudia Zabek

kzabek@securecodewarrior.com

Partnerships Manager