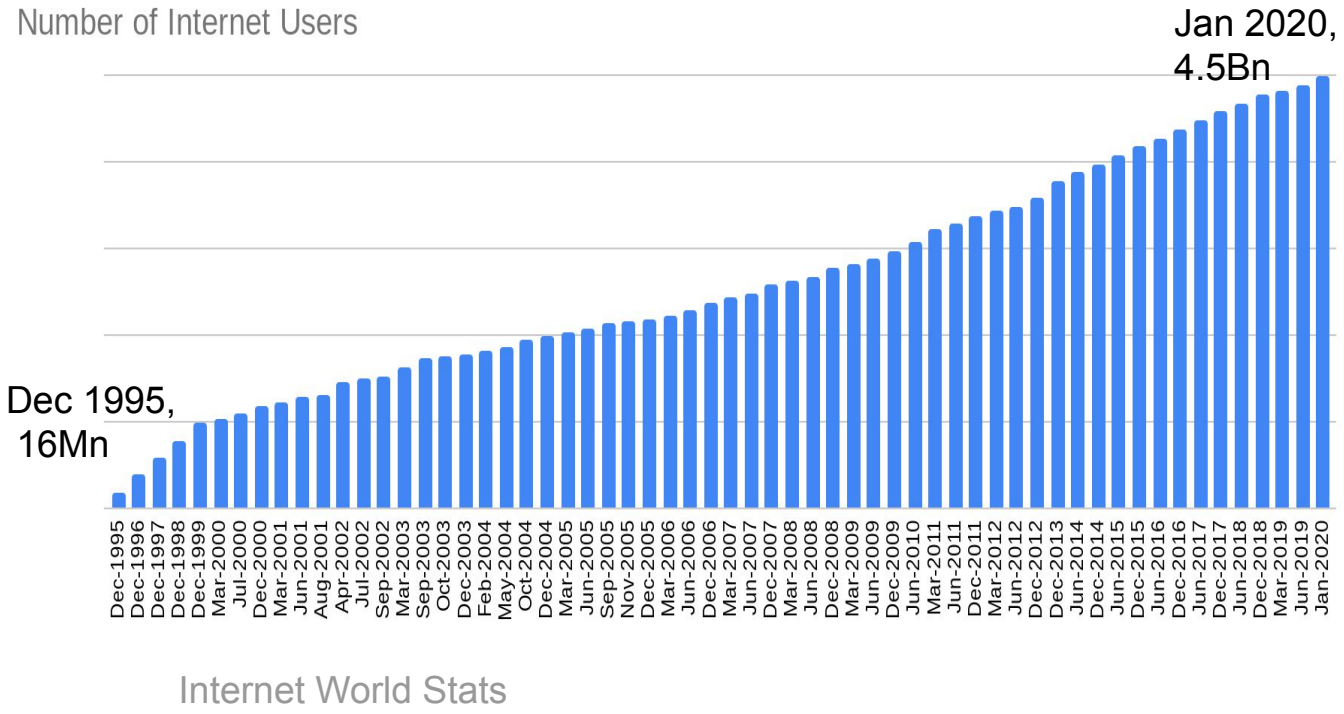
Make your developers the first line of defence for application security

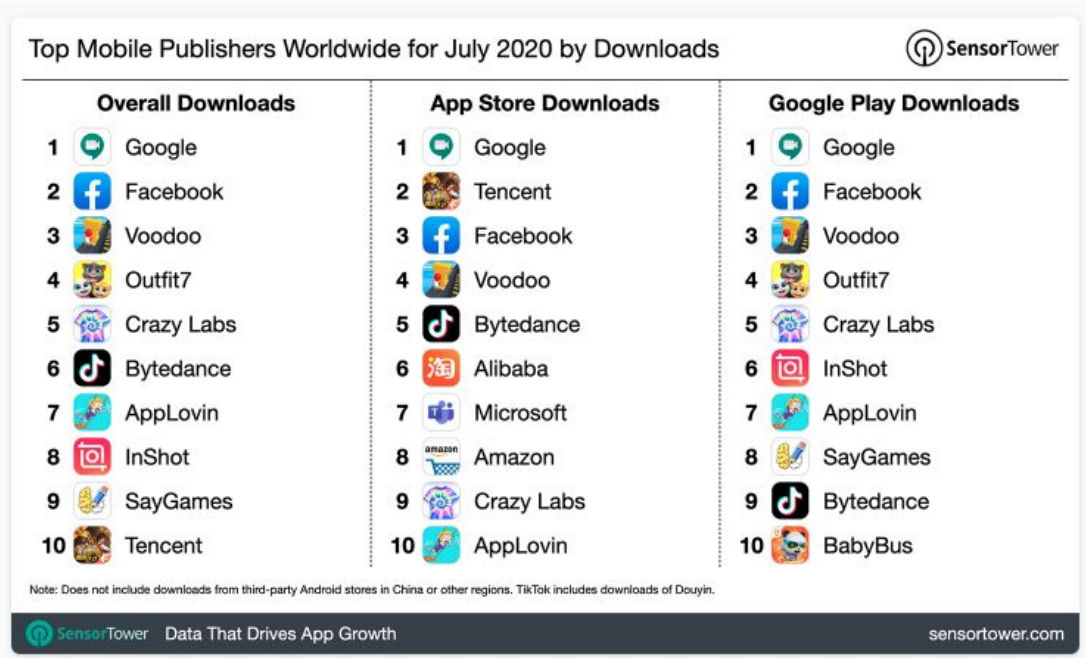
Klaudia Zabek

kzabek@securecodewarrior.com

Partnerships Manager

Digital Landscape is growing





Why Application Security Training?

SECURE CODE WARRIOR

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.

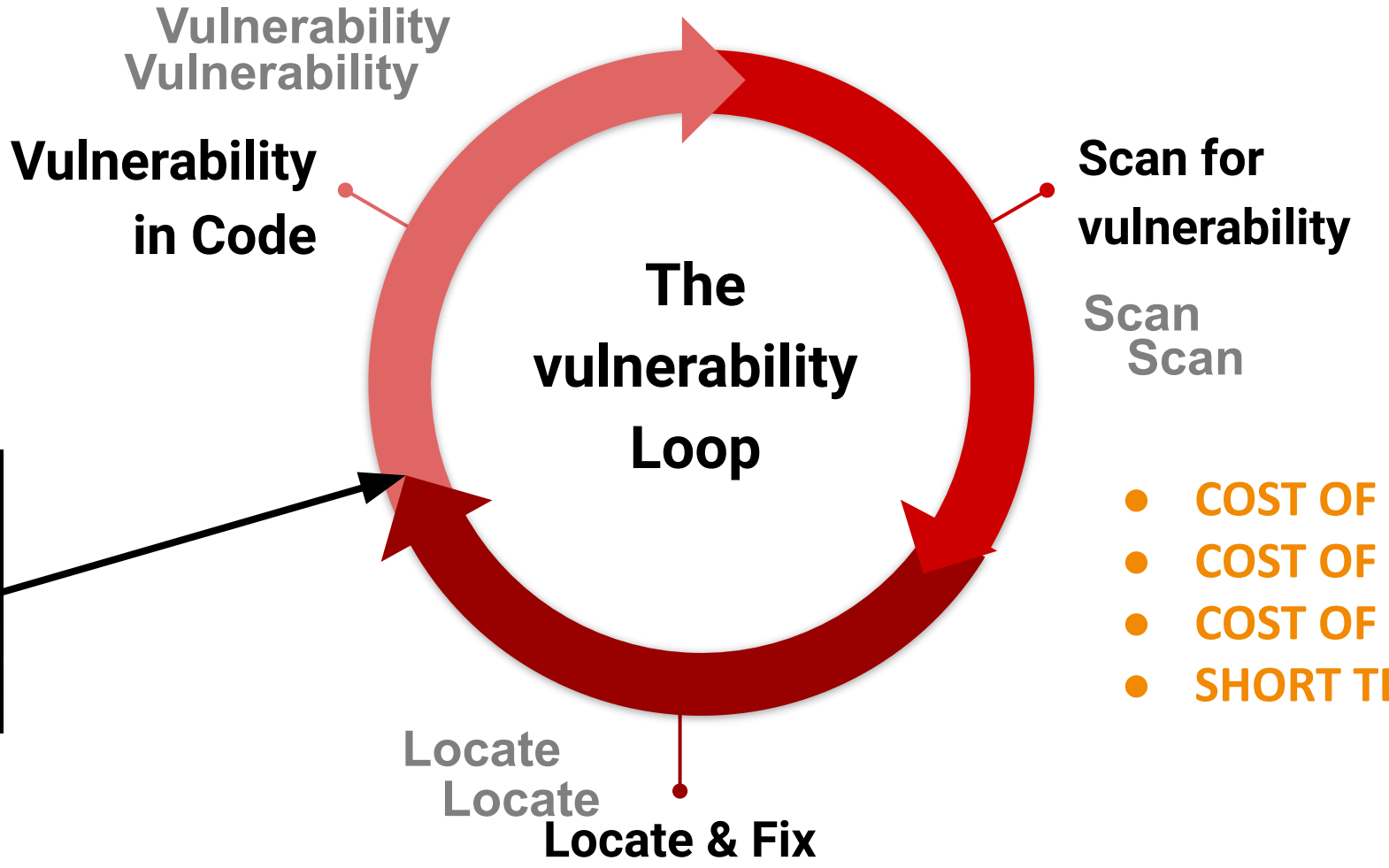
20,000: Number of times the average web app was attacked, January and February 2020

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

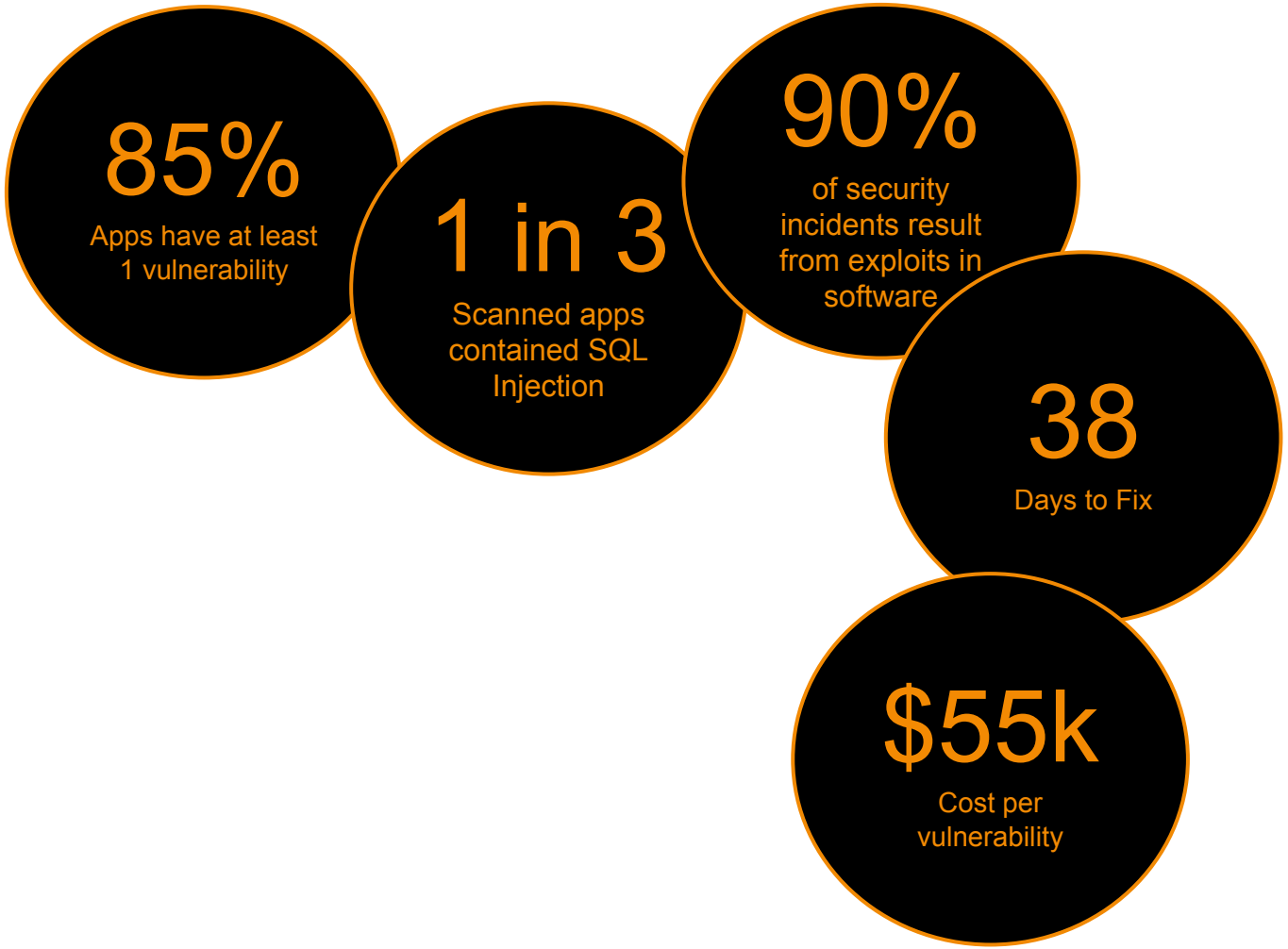
The Vulnerability Loop



- COST OF DELAYS
- COST OF SCANS & TESTS
- COST OF REMEDIATION
- SHORT TERM FIX

Break the Cycle

Why Application Security Training?



TESTING ALONE IS NOT THE ANSWER...

SOFTWARE NEEDS TO BE CODED SECURELY FROM THE START

Starting Left In the SDLC

SECURE CODE WARRIOR

Developers → Code Writing → Code Repository → Application Build → Application Deploy → Production

SECURE CODE WARRIOR

> Tournaments
> Training
> Courses
> Metrics
> Assessments

SCA: sonatype, snyk, BLACKDUCK, WhiteSource

SAST: SYNOPSYS, CHECKMARX, VERACODE, kiuwan, Fortify Application Security (MICRO FOCUS)

IAST: CONTRAST SECURITY

RASP: IMMUNIO, BRAKEMANPRO

DAST: PREVOTY, bugcrowd, hackerone, Outpost24, Synack, Cobalt

Prevent

Detect & React

Cost to fix a security bug

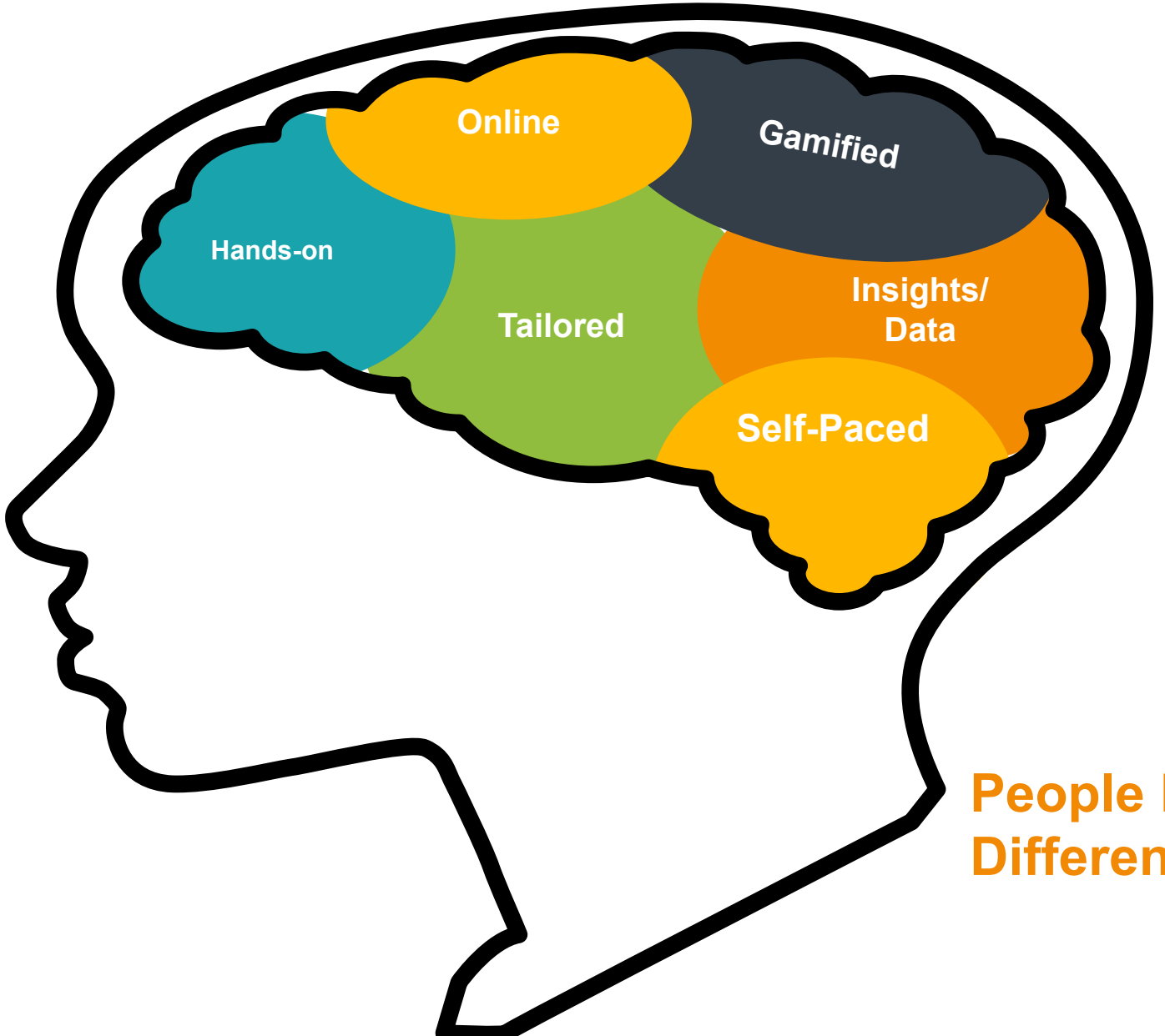
SECURE CODE WARRIOR

SHIFTING FROM REACTION TO PREVENTION WITH ONE POWERFUL SECURE CODING PLATFORM

A DIFFERENT APPROACH

SECURE CODE WARRIOR

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.

Tournaments

Create awareness and drive continuous engagement for secure coding

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

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.

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.

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.

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.

Courses

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

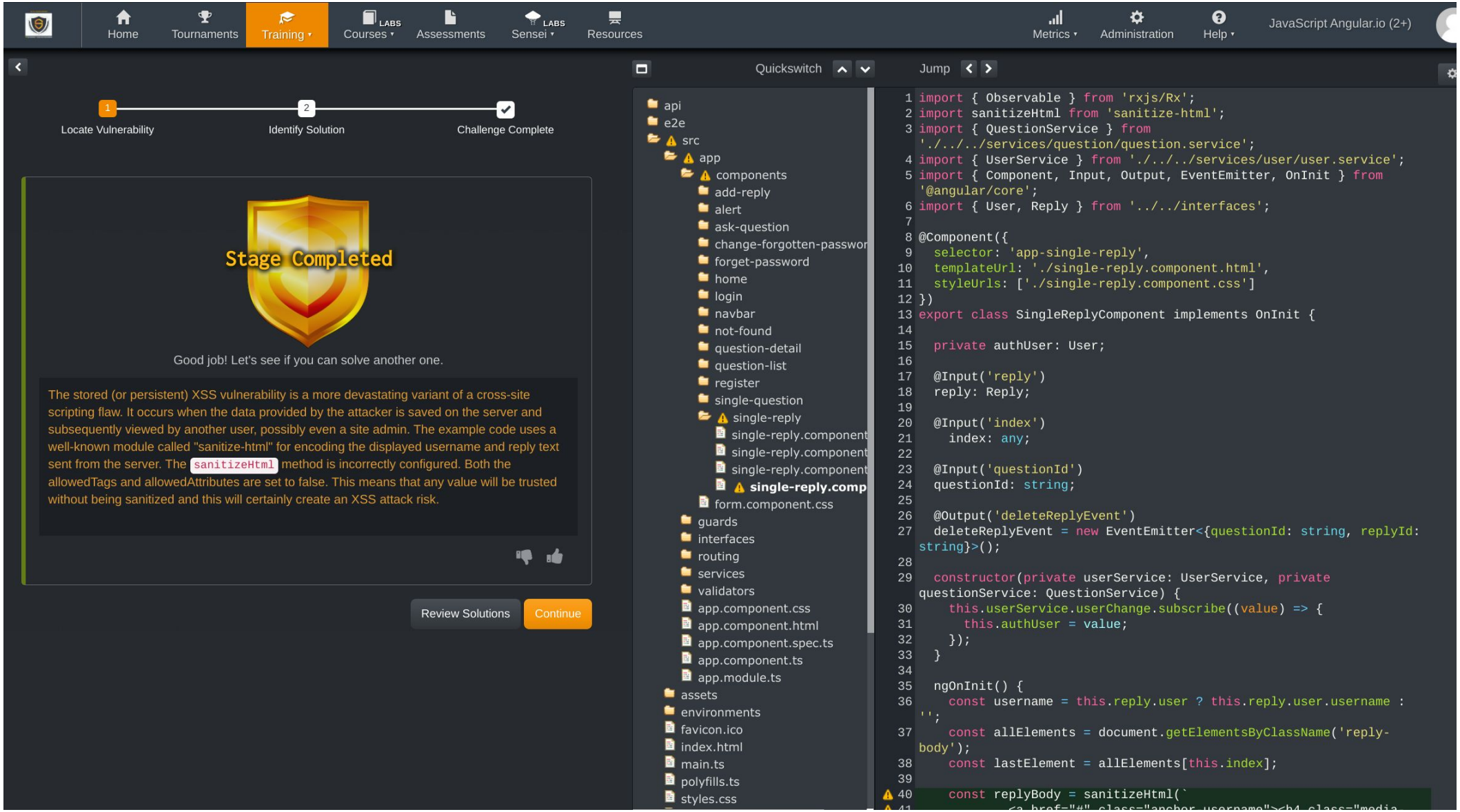
Configure and assign training activities to assist in achieving compliance requirements, like NIST and PCI-DSS or target specific skill gaps.

Integrations (APIs)

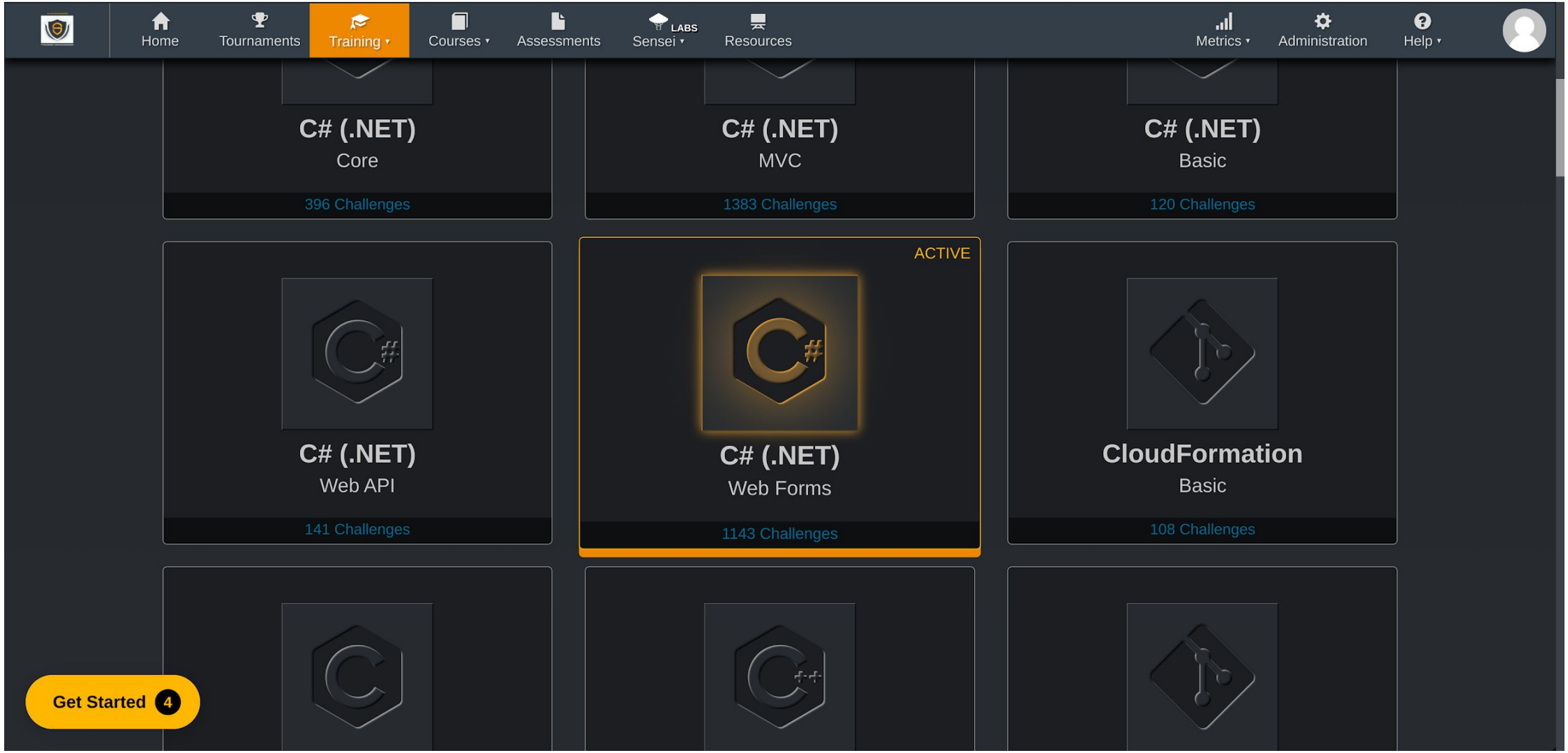
Connect with your core business systems to streamline your workflow

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

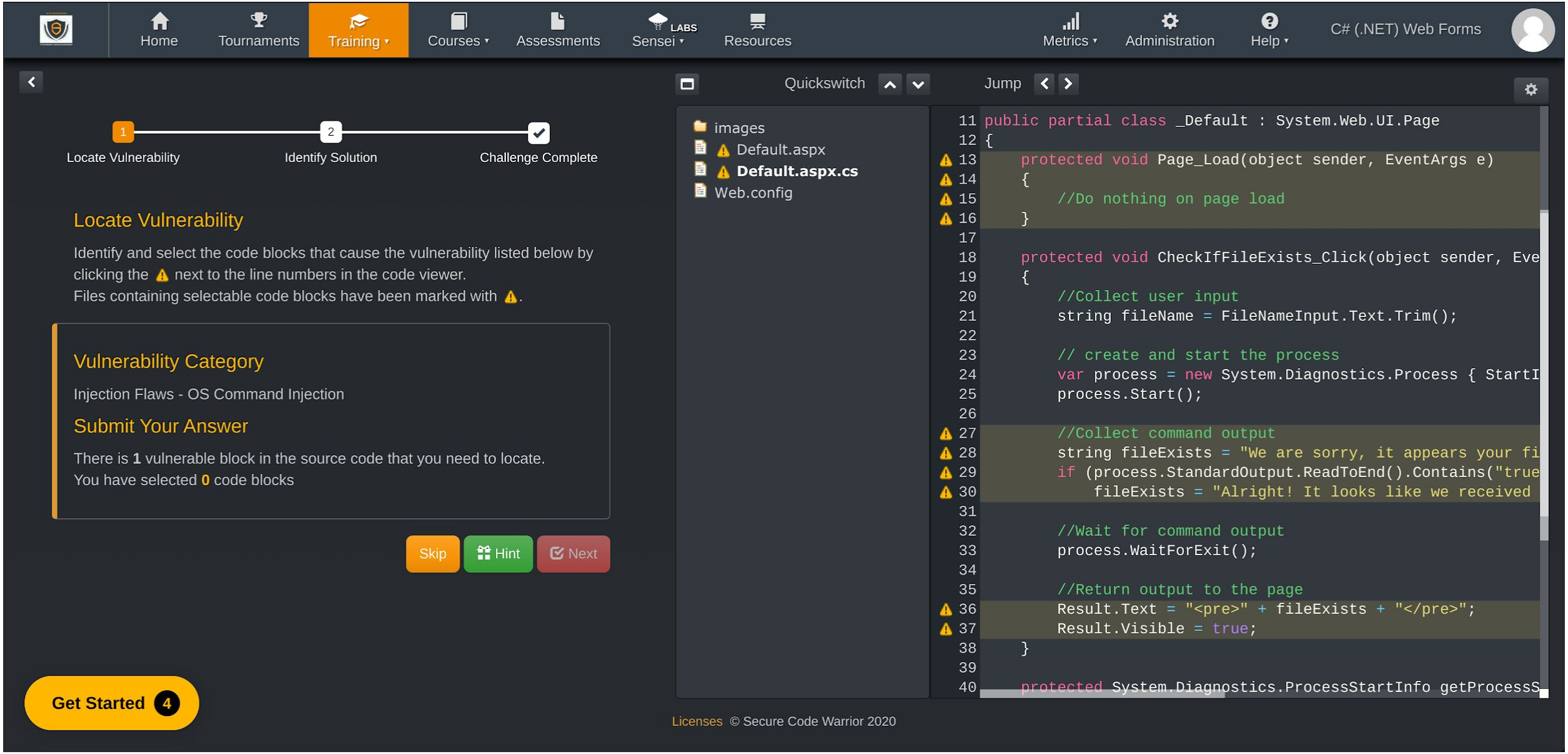


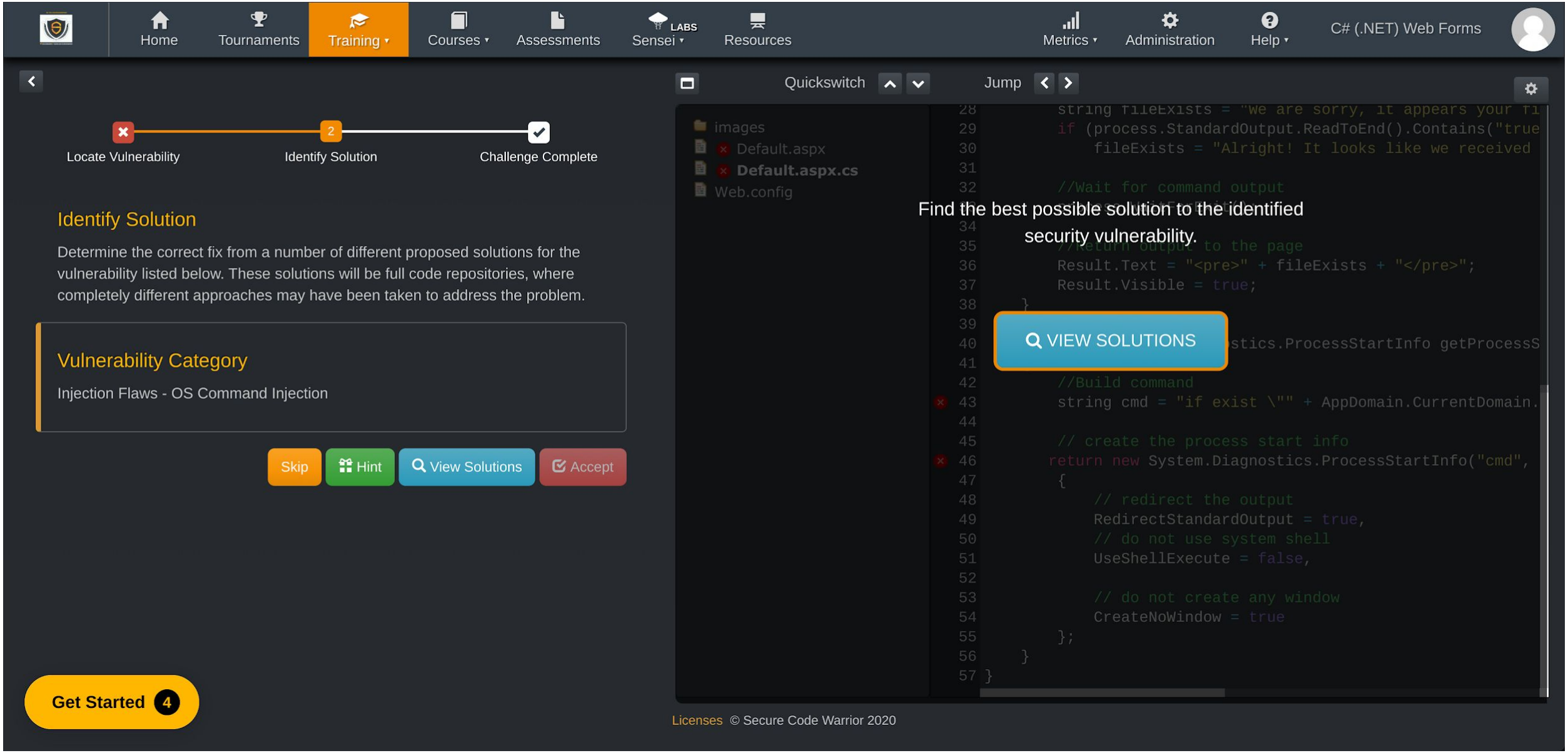
Example challenge

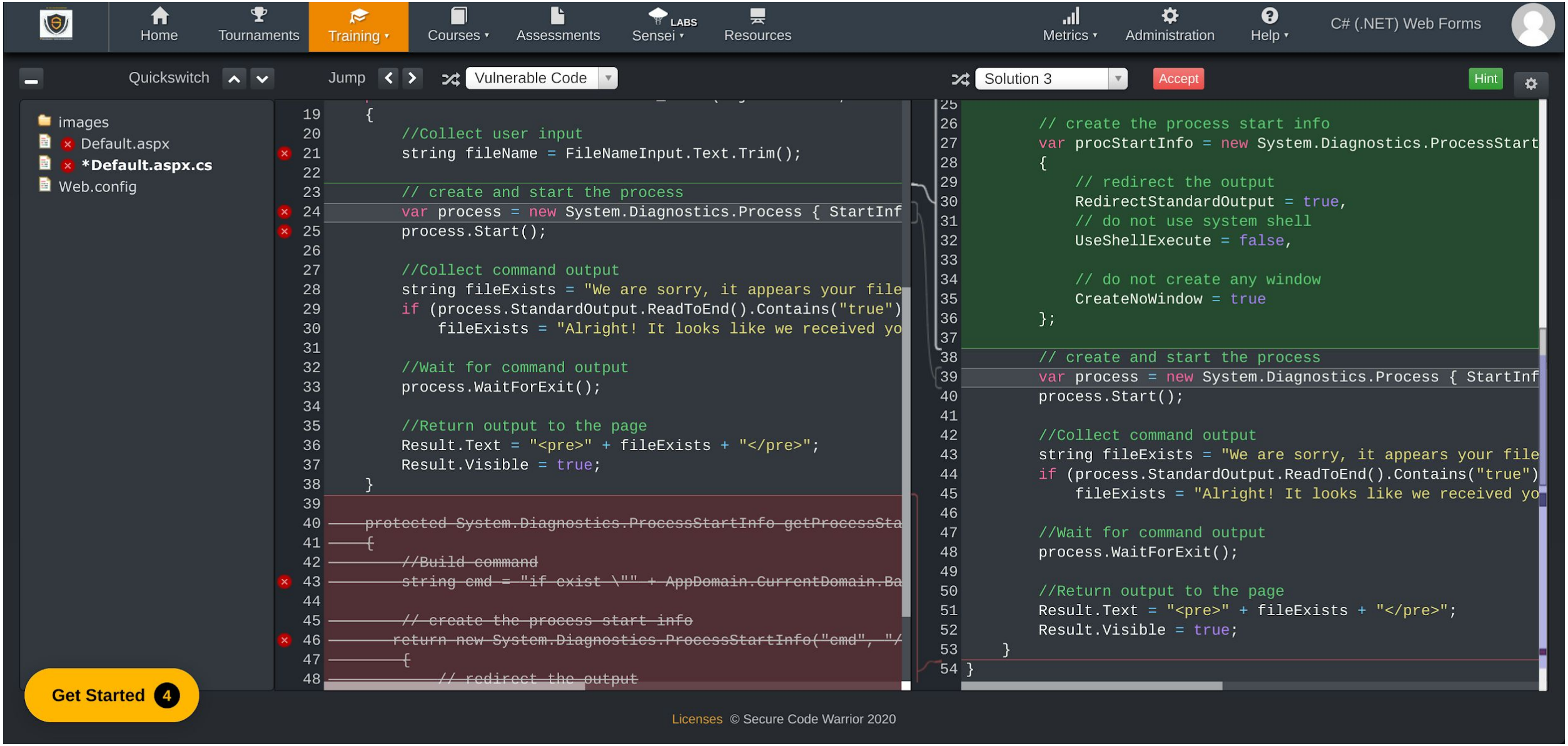


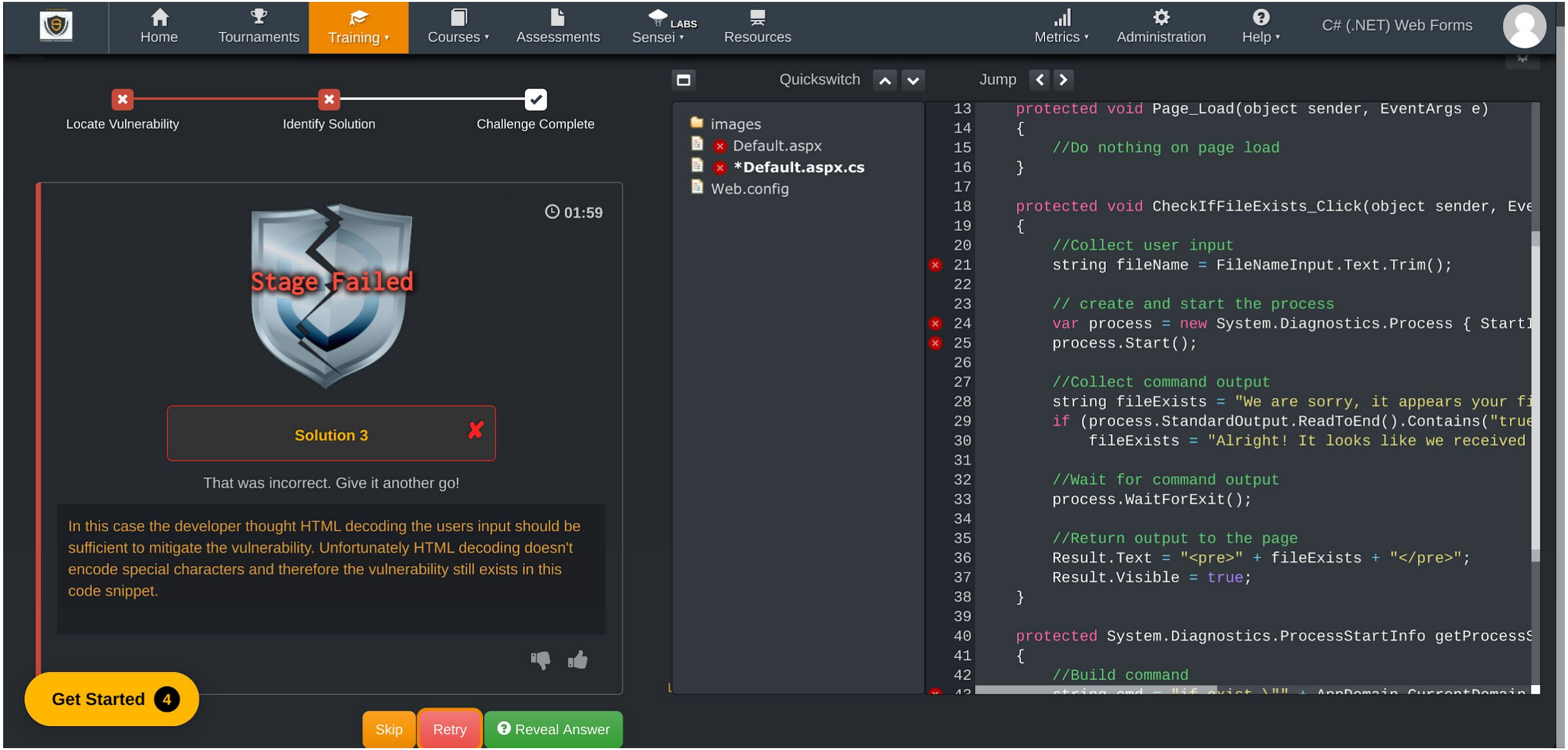

Home
Tournaments
Training
Courses
Assessments
LABS
Sensei
Resources
Metrics
Administration
Help
C# (.NET) Web Forms
Select a level to play. Each level will have a different set of quests to complete.
OWASP Web Top 10 2017
Learn the ropes or hone your skills in secure programming here. This set of levels will focus on individual vulnerability categories so that you can practise finding and fixing certain types of issues.
ACTIVE
1
OWASP A1-A2
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
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.
Get Started
4

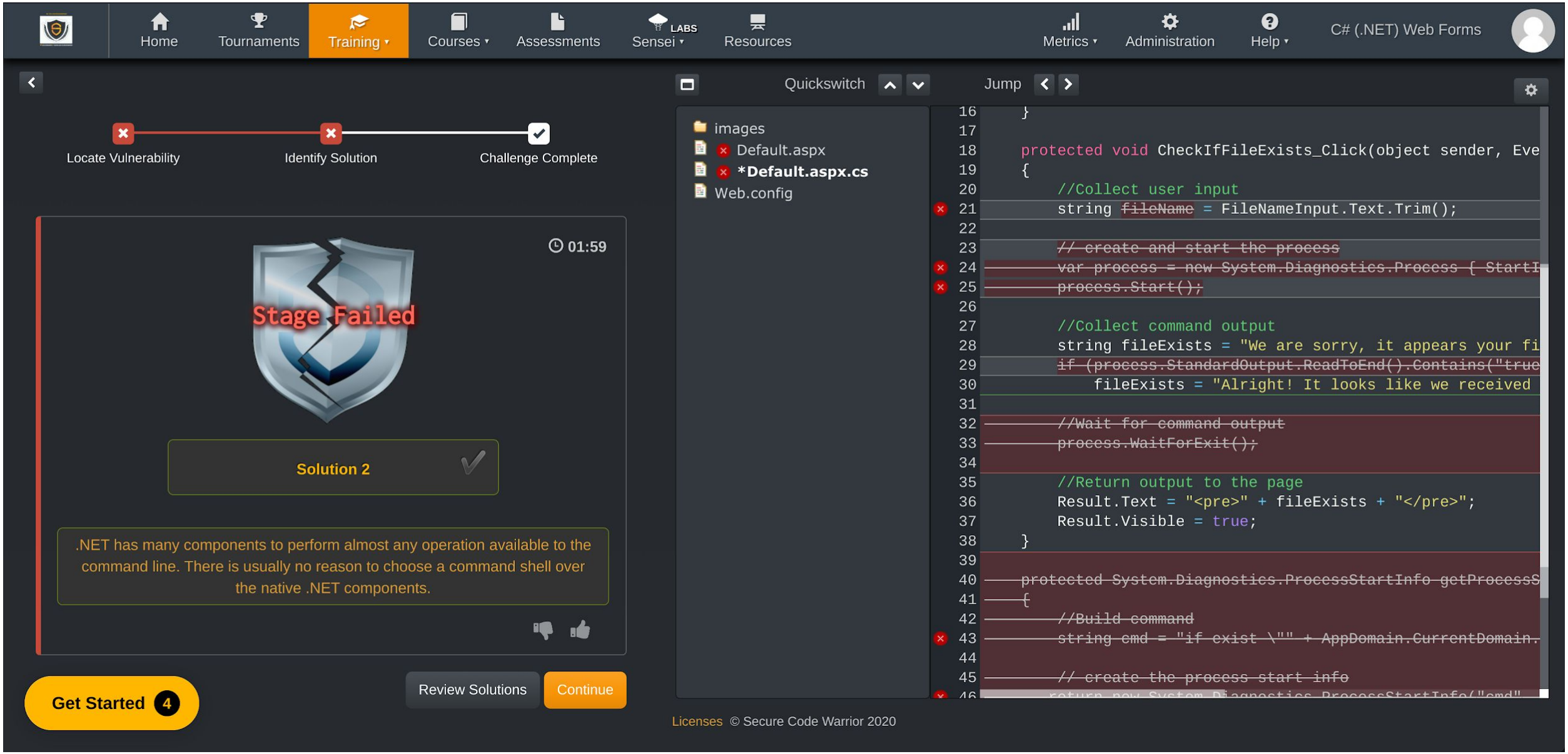

Home
Tournaments
Training
Courses
Assessments
LABS
Sensei
Resources
Metrics
Administration
Help
C# (.NET) Web Forms
Level
1
points
OWASP A1-A2
Accuracy
Security Maturity
A-Team Leaderboard
Developer names have been anonymised by your company administrator
Rank
Name
points
11 Klaudia Zabek 391
12 Adamantium 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
Active Missions
A1 - Injection: A Hacker from France is attacking the C# (.NET) Web Forms Code
VIEW
Get Started 4
state-sponsored adversary from Dem. Rep. Congo is
View
Web Forms Code Snippets application
is based on public domain map data available from jVectorMap and Natural Earth


Home
Tournaments
Training
Courses
Assessments
LABS
Sensei
Resources
Metrics
Administration
Help
C# (.NET) Web Forms
1
Locate Vulnerability
2
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
public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
//Do nothing on page load
}
protected void CheckIfFileExists_Click(object sender, Eve
{
//Collect user input
string fileName = FileNameInput.Text.Trim();
// create and start the process
var process = new System.Diagnostics.Process { StartI
process.Start();
//Collect command output
string fileExists = "We are sorry, it appears your fi
if (process.StandardOutput.ReadToEnd().Contains("true
fileExists = "Alright! It looks like we received
//Wait for command output
process.WaitForExit();
//Return output to the page
Result.Text = "<pre>" + fileExists + "</pre>";
Result.Visible = true;
}
protected System.Diagnostics.ProcessStartInfo getProcessS
Licenses © Secure Code Warrior 2020




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
Get Started 4
Licenses © Secure Code Warrior 2020

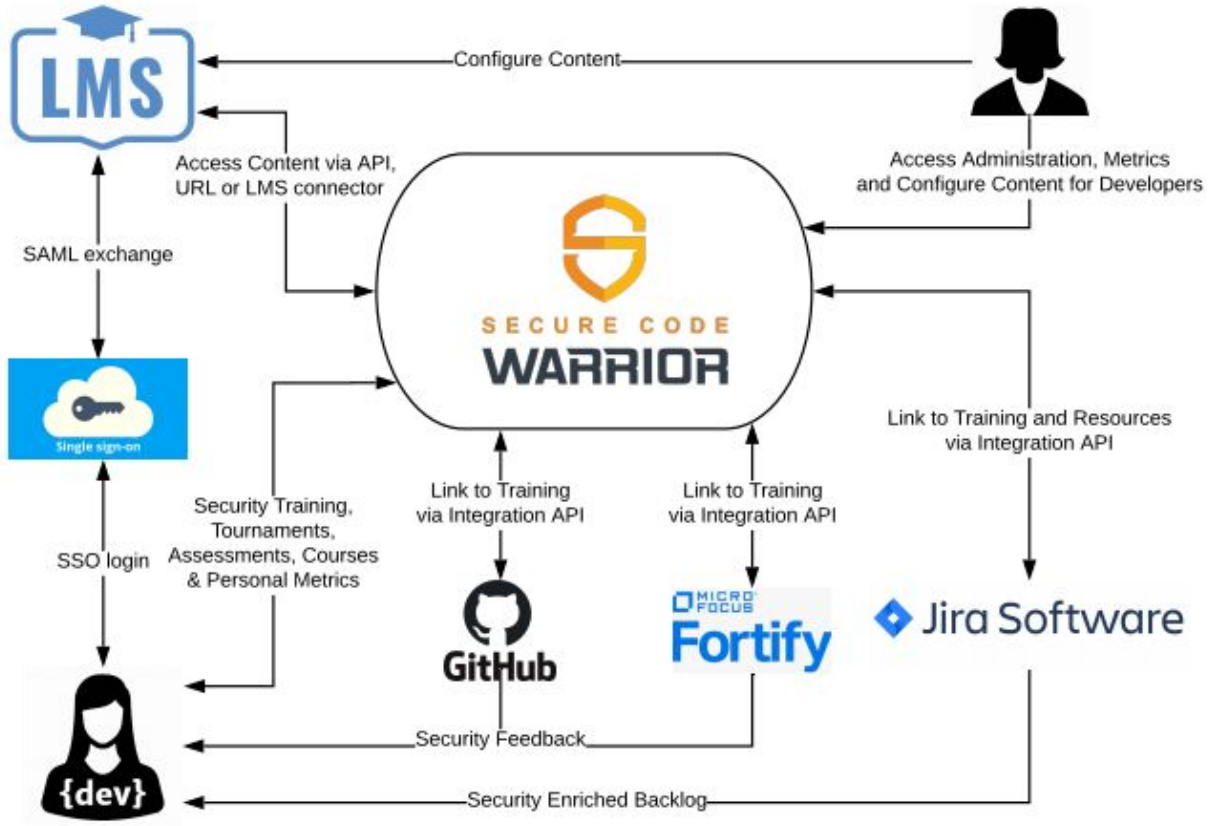

Home
Tournaments
Training
Courses
Assessments
LABS
Sensei
Resources
Metrics
Administration
Help
C# (.NET) Web Forms
Locate Vulnerability
Identify Solution
Challenge Complete
01:59
Stage Failed
Solution 3
That was incorrect. Give it another go!
In this case the developer thought HTML decoding the users 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
protected void Page_Load(object sender, EventArgs e)
{
//Do nothing on page load
}
protected void CheckIfFileExists_Click(object sender, Eve
{
//Collect user input
string fileName = FileNameInput.Text.Trim();
// create and start the process
var process = new System.Diagnostics.Process { StartI
process.Start();
//Collect command output
string fileExists = "We are sorry, it appears your fi
if (process.StandardOutput.ReadToEnd().Contains("true
fileExists = "Alright! It looks like we received
//Wait for command output
process.WaitForExit();
//Return output to the page
Result.Text = "<pre>" + fileExists + "</pre>";
Result.Visible = true;
}
protected System.Diagnostics.ProcessStartInfo getProcessS
{
//Build command

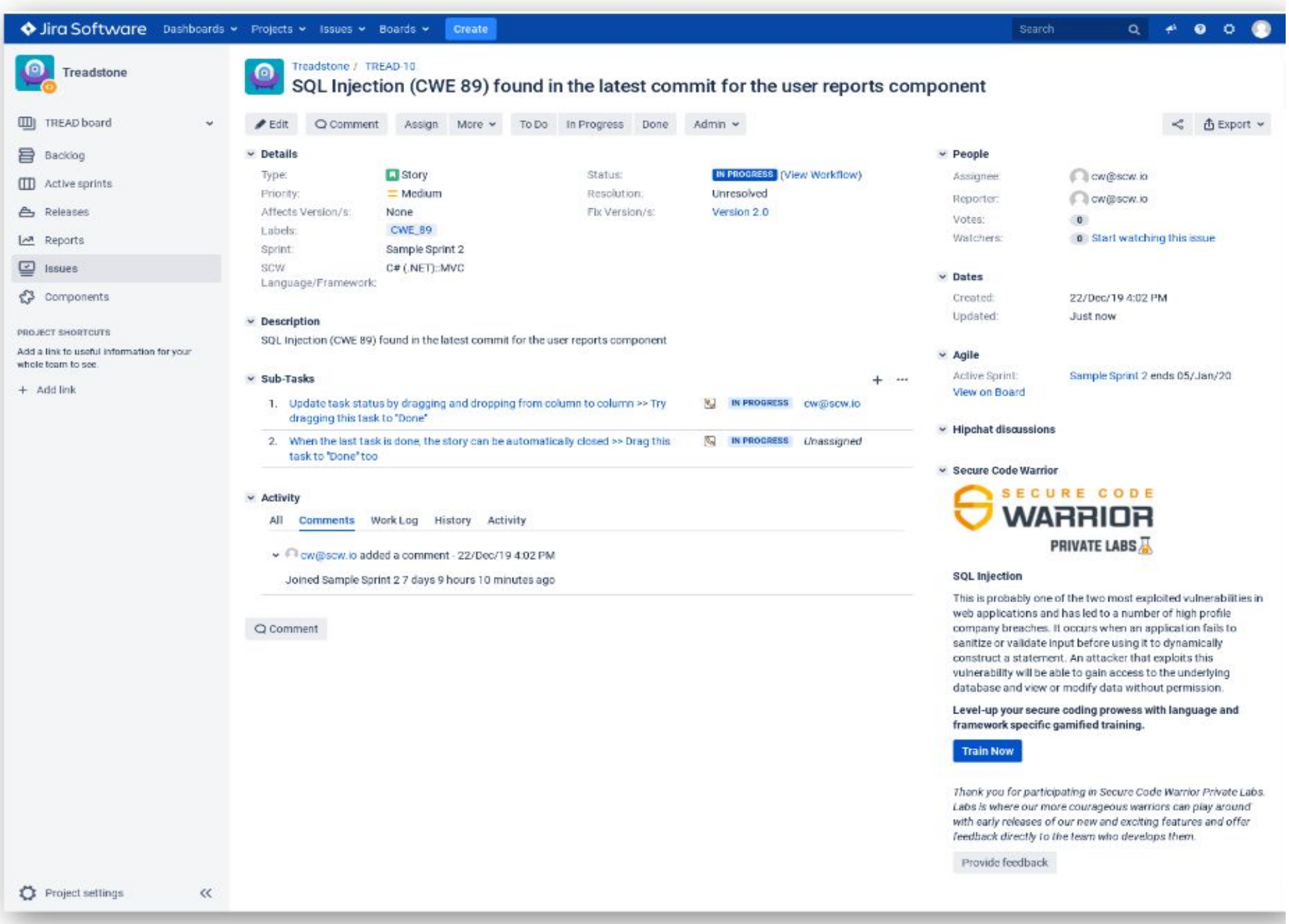

Home
Tournaments
Training
Courses
Assessments
LABS
Sensei
Resources
Metrics
Administration
Help
C# (.NET) Web Forms
Locate Vulnerability
Identify Solution
Challenge Complete
01:59
Stage Failed
Solution 2
.NET has many components to perform almost any operation available to the command line. There is 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 CheckIfFileExists_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.
44
45 // create the process start info
Licenses © Secure Code Warrior 2020

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




Jira Software
Dashboards
Projects
Issues
Boards
Create
Search
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
Project settings
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
Priority: Medium
Affects Version/s: None
Labels: CWE_89
Sprint: Sample Sprint 2
SCW Language/Framework: C# (.NET)::MVC
Status: IN PROGRESS (View Workflow)
Resolution: Unresolved
Fix Version/s: Version 2.0
Description
SQL Injection (CWE 89) found in the latest commit for the user reports component
Sub-Tasks
1. Update task status by dragging and dropping from column to column >> Try dragging this task to "Done" IN PROGRESS cw@scw.io
2. When the last task is done, the story can be automatically closed >> Drag this task to "Done" too IN PROGRESS Unassigned
Activity
All
Comments
Work Log
History
Activity
cw@scw.io added a comment - 22/Dec/19 4:02 PM
Joined Sample Sprint 2 7 days 9 hours 10 minutes ago
Comment
People
Assignee: cw@scw.io
Reporter: cw@scw.io
Votes: 0
Watchers: 0 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
SECURE CODE WARRIOR
PRIVATE LABS
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

Recommended Roll Out Approach

Engage your team and build a pathway of achievement

ICON KEY

- TOURNAMENTS
- RESOURCES
- TRAINING
- ASSESSMENTS
- CERTIFICATION

PHASE 1

AWARENESS

- Introduction tournament
- Bring security front of mind
- Encourage user on-boarding
- Identify weaknesses in developer community to incorporate into future training
- Identify "Security Champions" amongst your developers
- Provide feedback to developers on their own strengths and weaknesses

PHASE 2

IDENTIFY & ADDRESS KEY WEAKNESSES

- Focus on top 3 vulnerabilities identified by static code analysis tools, penetration testing and tournaments

- Train and certify all engineers on top company vulnerabilities

PHASE 3

ESTABLISH A COMPANY BASELINE

- Establish a company baseline that determines the minimum security skills requirement

- Train and certify all engineers

PHASE 4

CERTIFICATION PROGRAM

- Establish a learning path that focuses on both developer personal skills development and contributions to helping others
- All developers reach level 1 baseline certification

- Aim for a diverse mix of skills in each team

Don’t Take My Word For It……

SECURE CODE WARRIOR

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.

Thank you!

Klaudia Zabek

kzabek@securecodewarrior.com

Partnerships Manager