SEC552: **Bug Bounties and Responsible Disclosure**

2	12	Laptop
Day Course	CPEs	Required

You Will Be Able To

- Learn the art of translating app features to attack vectors
- Find complex and tricky security bugs in real-life apps
- Draw on real-life bug bounty stories discovered by talented researchers
- Gain a deep understanding of the root cause of security bugs in modern apps
- Correlate security bugs with defenses and understand how to bypass weak app defense controls by participating in labs based on real-life applications and using professional tools (Burp Suite Professional)
- Properly modify insecure defenses

Author Statement

"During my journey working in bug bounty programs, it was always challenging to catch security bugs. The bugs had to be risky, unique, and tricky so that they wouldn't be considered duplicate by other researchers. This course is inspired by real-life case studies and is designed to help you catch and fix tricky security bugs using logic techniques and professional tools."

— Hassan El Hadary

Pen testers and security researchers face the challenge of discovering and weaponizing complicated vulnerabilities in order to properly perform security assessments for applications. Modern applications are enriched with advanced and complex features that increase the attack surface. Every application has its own unique logic that requires the pen tester to deeply understand how the app functions before beginning a security assessment. Discovering and exploiting tricky security bugs in these assessments requires the art of mixing manual and automated techniques.

Bug bounty programs are put in place so that the security community can help vendors discover application security flaws that are difficult to discover and exploit. The scope of such programs includes security bugs for web apps, mobile apps, APIs, and more. Large IT companies, such as Google, Facebook, Twitter, and PayPal, have participated in such programs. Security researchers who follow the responsible disclosure policy of bug bounty programs are rewarded and acknowledged, since such programs improve and secure applications.

SEC552 is inspired from case studies found in various bug bounty programs, drawing on recent real-life examples of web and mobile app attacks. The experiences of different researchers yield ideas for pen testers and developers about unconventional attack techniques and mindsets. Each section of the course is influenced by bug bounty stories that are examined through the following structure:

- Attack concept: The idea, concept, and root cause of the attack.
- Test technique: How to test and discover the application security flaw manually and automatically.
- Attack exercise: This lab uses tools such as Burp Professional to analyze code samples from the vulnerable applications.
- Related bug bounty case study: Analysis of several bug bounty stories that are related to the attack.
- Defense techniques: The best security practices to defend from the attack and mitigate the application security flaws.

Here are just a few considerations when organizations are implementing bug bounty programs:

- Regardless of whether a company has a bug bounty program, attackers and researchers are assessing their Internet-facing and cloud applications. Security teams within companies, as well as consulting teams that provide security services for customers, need to understand how to assess Internet-facing applications.
- Companies rely on single sign-on (SSO) with third parties such as Dropbox. Authentication and session management shared between these sites offer opportunities for attackers.
- Most companies have cloud applications, many of which have weak APIs, weak single-factor authentication, poor session management, and other issues that can result in data exposure or remote code execution

Available Training Formats

Live Training

Live Events
sans.org/information-security-training/by-location/all

Summit Events
sans.org/cyber-security-summit

Section Descriptions

SECTION 1: App Analysis, Logic, and SQL Attacks

Section 1 begins by introducing you to bug bounty programs, the scope of testing, and common program rules. Understanding an app's functionality can open attack ideas and facilitate catching tricky app security bugs. You will learn and practice mapping the app logic and features into HTTP requests of real-life apps. You will learn different techniques inspired from real-life case studies in order to perform authentication bypass and account takeover. You will discover and exploit real-life bugs manually in an authentication bypass exercise. We'll inspect source code to understand the root cause of the bug, and all exercises will be performed on real-life apps using a trial license for Burp Suite Professional. You'll be hunting security bugs like professionals. Tricky logic bugs are some of the hardest to discover and catch in complex apps. You will learn different tricks to conduct logic and authorization bypass attacks while walking through real-life cases in bug bounty programs. An authorization bypass lab will enable you to practice catching tricky logic bugs. Finally, you will learn about various methods to perform SQL injection attacks in different contexts inspired by real-life bug bounty case studies.

TOPICS: Introduction and HTTP basics; Understanding the app; Hunting for authentication and session flaws; Logic attacks and authorization bypass; SQL injection

SECTION 2: Cross-Site Request Forgery, Client-Side and Mobile API Attacks

Section 2 continues covering various attack techniques for different security bugs such as Open Redirect, Server-Side Request Forgery (SSRF), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).The attack techniques covered will draw on real-life bug bounty stories that give different attack ideas for discovery, filter bypass, and exploitation. You will learn attack techniques on modern apps that are rich with client-side code and API calls. You will also learn how to chain different bugs to cause a greater security impact. The section is filled with exercises that will walk you through real-life apps. During the exercises, you'll learn how to discover the bug manually, how to inspect the root cause of the bug from the source code, and how to fix the bug. Finally, you will learn how to deliver quality app security bug reports with proper descriptions and evidence.

TOPICS: Open redirect; Server-side request forgery; Cross-site request forgery; Cross-site scripting; Client-Side code and APIs; Combining web attacks; Reporting and responsible disclosure

Who Should Attend

- Penetration testers: the course will enrich the skills of pen testers through real-life stories and practical labs covering the most popular web and mobile app attacks.
- Software developers and architects: the course will help developers link attack and defense techniques while discovering security bugs in the source code before making the app public.
- Security engineers: the course will help attendees who are managing a bug bounty program or planning to implement one by enabling them to practice the techniques used by security researchers to report security bugs, and to verify if the bugs are valid or false positives.
- Network/system engineers: the course will help attendees fill the gap of application security and get started in the field.

Hands-on Training

In SEC552, students will perform labs on real-world applications using professional tools to practice hunting genuine security bugs. We will then examine web application defenses and extra code review exercises to close the loop on the attacks covered. Finally, we'll look at reporting and responsible disclosure, ensuring delivery of quality app security bug reports with proper description, evidence, and recommendations. Bug bounty stories are full of ideas and clever tactics from which much can be learned about mixing manual and automated techniques. This course will teach you how to apply modern attack techniques to discover and disclose tricky, logic-based application flaws that automated scanning tools will not reveal.

The labs cover:

- App mapping
- Authentication bypass
- Logic attacks
- SQL injection - Boolean
- XSS bypassing filters
- API attacks
- Chaining logic attacks
- Reporting
- Extensive use of Burp Suite Pro