YOUR ULTIMATE GUIDE

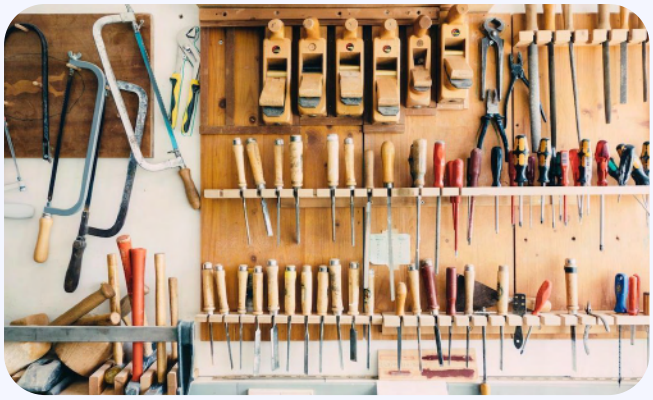
TO APPLICATION SECURITY TESTING

Mayhem

TABLE OF CONTENTS

WHAT IS APPLICATION SECURITY TESTING?

There are a lot of options for software security testing tools. How do you know which ones are right for you? Some types of tools, such as **SCA tools, are made to find vulnerabilities in existing code,** while others, such as **DAST tools, are more useful for finding vulnerabilities in your own code.** Some tools only find potential vulnerabilities, while others find confirmed vulnerabilities.



SCA, SBOM, Vulnerability Management, SAST, or DAST Tools: Which Is Best for Your Team?

READ

mayhem.security/4eEW5n1

WHAT IS APPLICATION SECURITY TESTING?

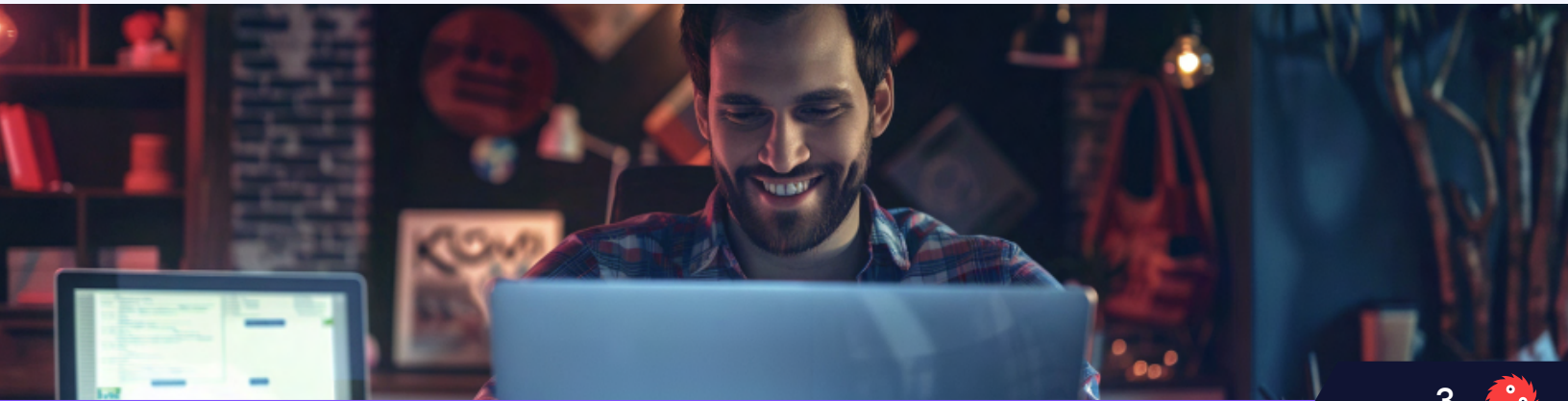
False Positives in Testing

In a perfect world, your software testing strategy would surface all of the security risks that exist inside your environment, and nothing more. But we don't live in a perfect world. Sometimes, the security issues that software testing tools flag turn out to be false positives. That means that they're not actually problems, even though the software security testing process identified them as such. **False positives create distractions that make it harder for security teams to detect and address actual security risks.** Why do false positives occur in software testing, and what can teams do about them? This article discusses those questions by explaining common causes of false positives and how to mitigate them.

4 Common Causes of False Positives in Software Security Testing

READ

mayhem.security/BNWaal



WHAT IS APPLICATION SECURITY TESTING?

Why Increase Test Coverage

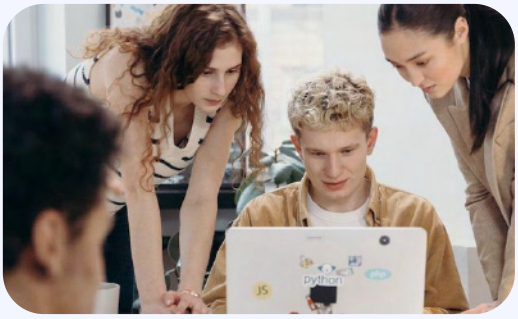
As software development becomes increasingly complex, ensuring the quality of the software is essential. One critical aspect of quality assurance is test coverage, which refers to the percentage of the code covered by automated tests. **The higher the test coverage, the more confidence we have in the software's functionality and reliability.** In this post, we will explore how to increase test coverage in your API with Mayhem in four easy steps.

How to Increase Test Coverage (And Confidence!) With Mayhem in 4 Easy Steps

READ mayhem.security/3xcNlDI

Performing Regression Testing

You knew that your application was secure when you scanned it for vulnerabilities prior to deploying it into production. But was it also secure when you applied an update or made a configuration change within the production environment. Unless you've performed regression testing, you don't know. **Regression testing is the only way to ensure that your software remains secure after you make changes.** This is especially important if you use modern software development practices, such as CI/CD, which involve making regular updates to applications.



3 Reasons Your Security Testing Tool Needs To Do Regression Testing

READ

mayhem.security/mkW8hH

WHY PRACTICE DEVSECOPS?

DevSecOPs Best Practices

As software development teams move towards a DevOps culture, security is becoming an increasingly important aspect of the development process. DevSecOps is a practice that integrates security into the DevOps workflow. The aim is to **build secure, reliable and compliant applications from the outset of the development process**, rather than addressing security as an afterthought.

This blog post explores the DevSecOps best practices that development teams can use to ensure that security is ingrained in the development process, leading to **better products with reduced security risks** and faster time-to-market.

7 Essential DevSecOps Best Practices Every Development Team Should Implement

READ

mayhem.security/t234ln

WHY PRACTICE DEVSECOPS?

“Shift Left”

DevSecOps has transformed the software development landscape, embedding security practices at each stage of the development and delivery pipeline. While the DevSecOps approach has (rightfully) been lauded for helping teams produce safer software, it has come with its own set of problems. **With this “shift left” has come a slew of new processes and tools that have become the responsibility of development teams to learn, follow and use.** This raises the question: Do the benefits of “shift left” justify the extra workload placed on development teams?

Who Shift Left Really Benefits: 4 Responsibilities DevSecOps Shifts Onto Developers

READ mayhem.security/X55gny

WHY PRACTICE DEVSECOPS?

Security Breaches

Historically, security has been bolted on at the end of the development cycle, often resulting in software riddled with vulnerabilities. This leaves the door open for security breaches that can lead to serious financial and reputational damage. According to the 2022 cost of a data breach report by IBM, **the average cost of a data breach in the United States is $9,440,000.** To mitigate these risks, organizations are increasingly turning to DevSecOps, a methodology that integrates security into the software development process from the very beginning, with the goal of delivering safer applications, faster.

The DevSecOps Lifecycle: How to Automate Security in Software Development

READ

mayhem.security/JbamNKv

WHY PRACTICE DEVSECOPS?

Becoming the Standard

From humble beginnings in basic IT configuration automation, DevOps has become **the de facto standard for organizations looking to ship software faster.** "Shift left" approaches combined development processes and methodologies with traditional operations tasks, putting more work on development teams in exchange for freedom from fire drills and production fixes. It wasn't long before security followed, with DevSecOps now shorthand for modern application security—and everything from SAST, DAST and SCA shoehorned into developers' toolchains and workflows.

DevOps vs. DevSecOps Process: How to Ensure Your Organization Has a Security Mindset

READ

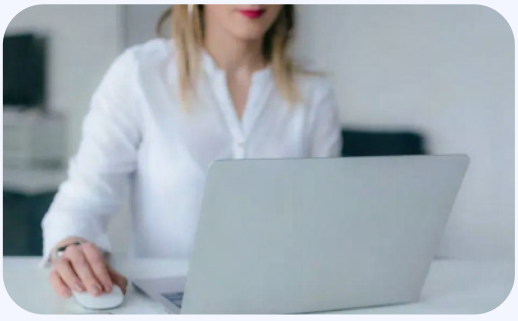
mayhem.security/1GkLgd

ENHANCE SECURITY WITH HELP FROM HACKERS

Ethical Hacking

The word "hacker" is all too often associated with criminal activities—"The hacker who broke into the systems at …" This association, however, does a disservice to the legitimately curious people, including students, academics and researchers—"Researchers worked with Microsoft to patch the vulnerability before it became known."

What people don't often realize is that these "researchers" are hackers. **Really, hacking, by itself, is not a crime. The word "hack" simply means to take something apart.**



Why Is Hacking Good? Ethical Hacking is a Skill, Not a Crime

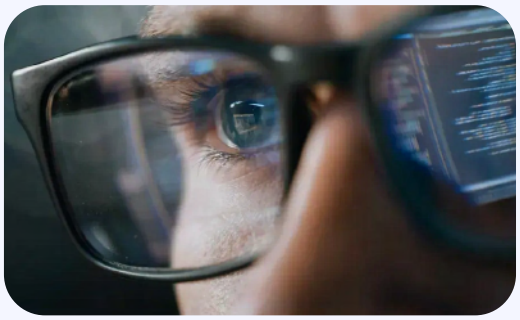
READ

mayhem.security/w8VFQA

ENHANCE SECURITY WITH HELP FROM HACKERS

Criminal Hacking

On the other hand, *criminal* hacking has become a major threat to today's organizations. According to a Deloitte Center for Controllership poll, **"During the past 12 months, 34.5% of polled executives report that their organizations' accounting and financial data were targeted by cyber adversaries."** And, "Nearly half (48.8%) of C-suite and other executives expect the number and size of cyber events targeting their organizations' accounting and financial data to increase in the year ahead." By understanding the methods that criminal hackers commonly use, organizations can take proactive measures to safeguard their systems and protect their data.



Common Techniques Hackers Use to Penetrate Systems and How to Protect Your Organization

READ mayhem.security/WHMqWt

ENHANCE SECURITY WITH HELP FROM HACKERS

Hacking through the Years

Hacking has gone through several eras over the years, each with its own unique characteristics and motivations. **Understanding the history of computer hacking is important for understanding its impact on technology and society, the current state of cybersecurity**, and for developing effective strategies for protecting against cyber threats.

History of Computer Hacking and Cybersecurity Threats: From the 50s to Today

READ

mayhem.security/NXd3qt

BEST PRACTICES FOR LANGUAGES AND FRAMEWORKS

Best Practices for Rust

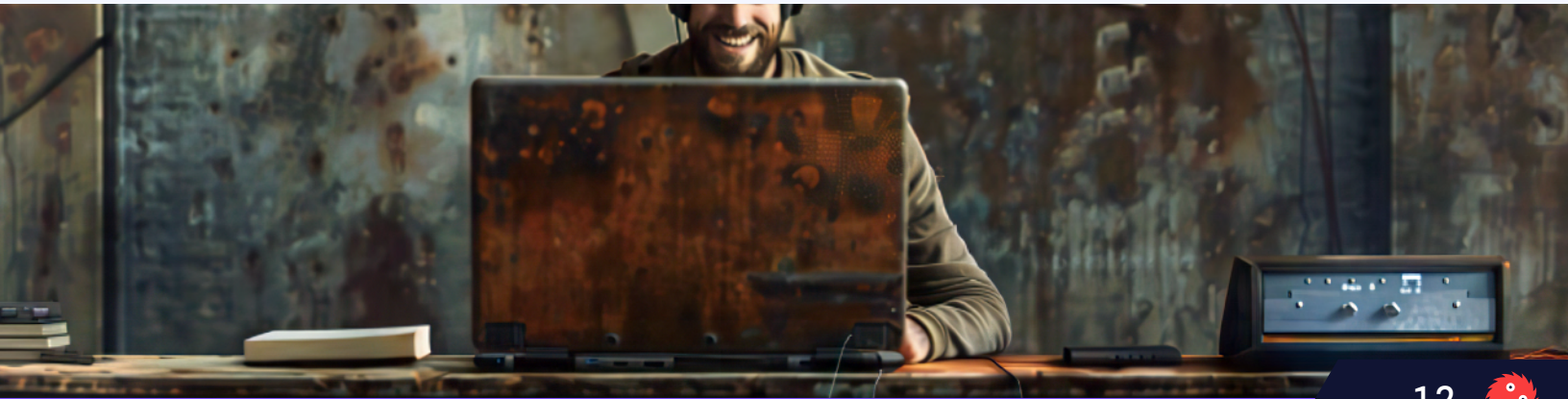
Rust is a modern programming language that is known for its safety and security features. As a Rust developer, you understand the importance of writing secure code. **Rust's memory safety and type system help prevent entire classes of vulnerabilities**, but that doesn't mean Rust's code is impervious to security issues. There are still risks from logic errors, improper handling of edge cases, and malicious inputs that you must consider.

Best Practices for Secure Programming in Rust

READ

mayhem.security/QqNIWV

Thus, the objective of this article is to provide you—Rust developers—with some best practices and recommendations for secure application development. These best practices will enable you to take advantage of the range of security possibilities and features that Rust has to offer.



BEST PRACTICES FOR LANGUAGES AND FRAMEWORKS

Best Practices for C++

Despite the introduction of multiple programming languages over the past few years, C++ still remains one of the most powerful and widely used programming languages among developers. **It's widely known for its efficiency and performance, which allows developers to create reliable and high-performing applications.** However, like any other programming language, C++ faces security vulnerabilities.

As a developer, secure programming should be among your top priorities during development. Secure programming ensures that you follow all the best practices available to maintain the integrity of the applications you're developing. Whether you are developing small utility applications or working on complex systems, ensuring the security of your code is really important, as this will help protect user data while preventing issues like unauthorized access.

Best Practices for Secure Programming in C++

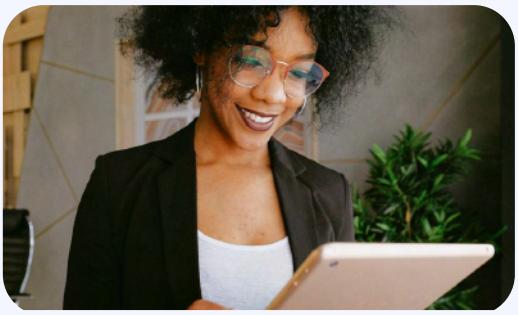
READ

mayhem.security/c3jc8R

BEST PRACTICES FOR LANGUAGES AND FRAMEWORKS

Securing APIs

API security testing has always been critical for any organization that relies on APIs to connect its applications. But securing APIs is now more important than ever, given that API security attacks have surged by an astounding rate of 400 percent in recent months.



Securing a REST API: 4 Tips to Make Sure Your REST API Is Secure

READ

mayhem.security/55hniA

To help provide guidance on protecting APIs, this article walks through the essentials of REST API security. It explains why REST APIs can be vulnerable to attack, which types of harm REST API security breaches can cause, and best practices that developers should adopt to keep their REST APIs safe.

SECURE YOUR ORGANIZATION

Want to learn how to best secure your organization?

Add some Mayhem to your security strategy and rest assured that your systems are guarded against cyber threats.

SCHEDULE A DEMO

mayhem.security/demo