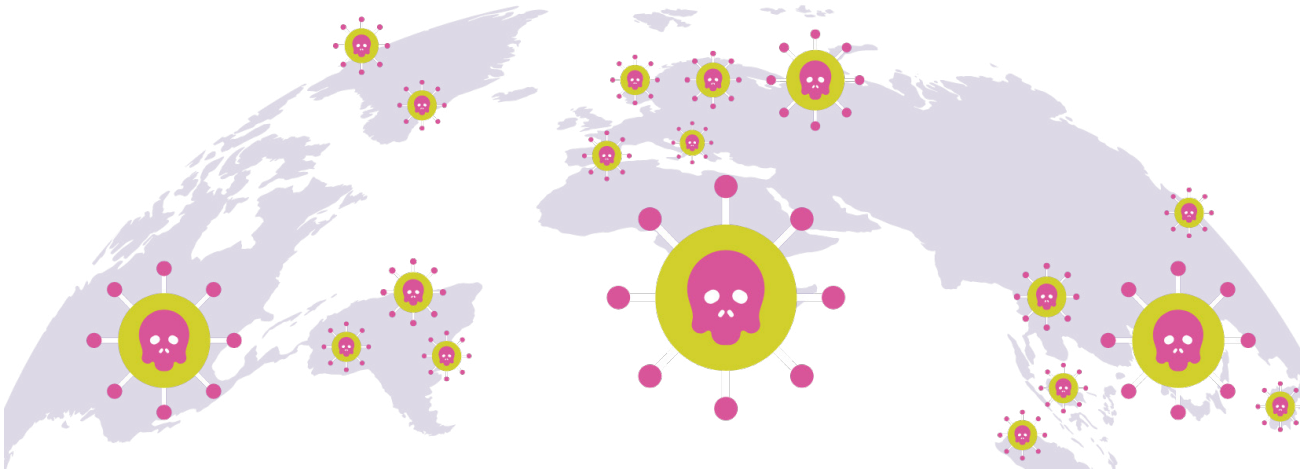
A Security Guide for

ActionScript Developers

kiuwan
An Idera, Inc. Company

Overview

Application security is becoming increasingly important as the world of programming and development grows. Now more than ever, developers must ensure that their applications are robust, secure, and reliable. This is because malicious actors are finding new ways to exploit applications, take advantage of vulnerabilities, and steal sensitive data. In the third quarter of 2022, there were **15 million data breaches worldwide** – a 167 percent increase from the prior quarter.



As ActionScript becomes an essential language for application development, developers must become aware of the security considerations that come with ActionScript programming.

ActionScript is an object-oriented programming language used to create interactive web applications and is found in Adobe products such as Flash and Flex. Developers can also use ActionScript to develop desktop applications, mobile apps, games, and webpages. Due to its simple syntax, powerful features, and scalability, this programming language has attracted the attention of both experienced and new developers.

Unfortunately, just like any other programming language, ActionScript applications are vulnerable to various risks, and organizations need to recognize the implications and proactively protect ActionScript applications.

This guide will provide an overview of ActionScript application security and discuss the specific risks that ActionScript developers should be aware of. We'll also discuss the proactive steps organizations can take to protect ActionScript applications from attacks.

How Do You *Secure* Applications?

In general, application security combines two approaches: proactive and reactive.

Proactive

The proactive approach involves implementing measures at the development level to ensure applications are secure. This means in following safe coding practices, using security frameworks and libraries, and incorporating automated testing into the development process.

Reactive

The reactive approach focuses on monitoring and responding to potential threats once they have been identified. This involves scanning applications for vulnerabilities and detecting any malicious activities. Developers should also create a system for quickly responding to any attacks.

ActionScript in *Detail*

ActionScript is a programming language developed by Adobe Systems, Inc., primarily used for web application development. ActionScript has its roots in JavaScript and ActionScript 3, the latest version of ActionScript, is an object-oriented language with powerful features for creating dynamic web applications.

This programming language is unique in its ability to allow developers to create visually appealing applications with powerful features. ActionScript applications are typically developed using Adobe Flash, a platform that enables developers to create, modify, and deploy ActionScript applications easily.

ActionScript is a relatively young language stacked against other programming languages such as Java and C++. However, ActionScript's popularity among developers has steadily increased as they discover its various features and benefits.

One of ActionScript's main advantages is its development environment, which is designed to make it easy for developers to create, modify, and deploy ActionScript applications quickly. ActionScript's scalability also allows developers to create applications that can grow and evolve with their organizations.

In addition to ActionScript's development environment, ActionScript applications are also appealing because they can be deployed across the internet. This allows ActionScript applications to be accessible to many users, regardless of location.

Top *5 Risks* for Developing in ActionScript

While ActionScript is an easy-to-learn and powerful language, ActionScript applications are susceptible to a variety of risks. Here, we will cover the top five risks ActionScript developers should consider when developing applications.

1. Security Vulnerabilities

ActionScript applications are vulnerable to several security risks that can compromise the integrity and functionality of the application. These security risks include cross-site scripting (XSS) attacks, SQL injection attacks, and misuse of exposed APIs.

Cross-Site Scripting (XSS)

ActionScript is vulnerable to XSS attacks, which are a type of attack that injects malicious code into ActionScript applications. XSS can occur if ActionScript code is not correctly sanitized to prevent malicious code from being injected into ActionScript applications.

Hackers use XSS attacks to steal sensitive data, hijack user sessions, and execute malicious code on ActionScript applications.

Cross-Site Flashing (XSF)

Another threat for ActionScript developers to consider is cross-site flashing (XSF). XSF attacks are similar to XSS attacks, except they entail forcibly making a damaged small web format (SWF) load an external malicious Flash file. If successful, this attack could lead to XSS or the illicit modification of the graphical user interface (GUI) so that a user enters credentials on a counterfeit Flash form.

A scenario where this type of attack can occur includes ActionScript applications that allow Flash content to be embedded from external sources or ActionScript applications that load and execute ActionScript from URLs from external sources.

Attackers can exploit XSS vulnerabilities in a Flash application to launch various cross-site scripting attacks, including DOM-based XSS, reflected (non-persistent) XSS, or stored (persistent) XSS. They can achieve this by:

- Misuse of the GetURL() function
- HTML in text fields
- Other methods (loadMovie(), asfunction())

SQL Injection

SQL injection is an attack in which malicious code is injected into ActionScript applications to manipulate or exploit a given database. ActionScript applications are particularly susceptible to this type of attack since ActionScript developers often use ActionScript to access databases. These attacks can reveal sensitive information, modify ActionScript data, or even delete ActionScript databases.

The attacker tricks the system by inputting data containing SQL commands, which are then executed in the control plane. This exploit is possible because, to SQL, there is no difference between the control and data planes.

Broken Access Control

OWASP lists this vulnerability among its top 10 most common web application security risks. Broken access controls occur when developers fail to manage the access control mechanisms built into their applications properly. This can lead to unauthorized users bypassing security measures or even gaining access to confidential data.

ActionScript applications are vulnerable to broken access control risks since their software security measures are often applied client-side rather than server-side. Furthermore, ActionScript developers may not always properly configure their access control mechanisms, leaving them vulnerable to attack.

Cryptographic Failures

ActionScript applications may also be susceptible to cryptographic failures. Cryptography is used in ActionScript applications to protect the data stored within them and prevent unauthorized access or modification of that data. However, if the cryptography is not implemented or appropriately configured, ActionScript applications can be vulnerable to attack.

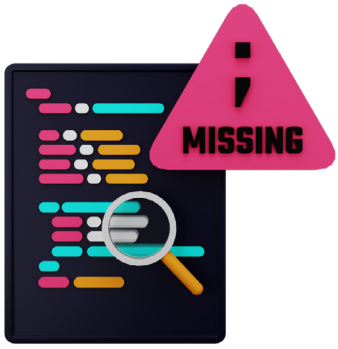
Cryptographic failures can range from weak encryption algorithms to improper key management, allowing attackers to gain access to sensitive data that has been encrypted. Furthermore, if cryptographic keys are not correctly secured or managed, attackers may also be able to reverse-engineer them, allowing them to gain access to the data.

2. Inadequate ActionScript Resources

ActionScript is a relatively newer programming language, and thus, ActionScript developers often find themselves lacking access to ActionScript resources.

Compared to other programming languages, ActionScript still needs to mature in terms of documentation and tutorials. ActionScript is also a more niche language, which can make it hard to find ActionScript developers when ActionScript applications need to be scaled up.

3. ActionScript Debugging Difficulties



Debugging ActionScript applications can be challenging due to ActionScript's lack of debugging tools and its complexity. ActionScript developers often have to manually debug ActionScript code, which can be a difficult and time-consuming process. ActionScript developers also have to take additional steps for debugging ActionScript applications that rely on external data sources, such as databases.

ActionScript does not have as many debugging tools as other programming languages, such as Java and C++. This can make it difficult for ActionScript developers to identify and fix ActionScript code errors, leading to issues such as application crashes and software security vulnerabilities.

4. ActionScript Performance and Optimization

ActionScript applications often require additional optimization and performance tuning due to ActionScript's limited language features. ActionScript developers often have to optimize ActionScript code manually, which can be a difficult and time-consuming process. ActionScript developers also have to take additional measures to ensure that ActionScript applications are optimized for different platforms and devices, such as mobile devices.

ActionScript has limited support for optimization techniques such as parallel processing and threading compared to other programming languages. This can make ActionScript applications more prone to performance issues, leading to application crashes and security vulnerabilities.

5. ActionScript Versioning

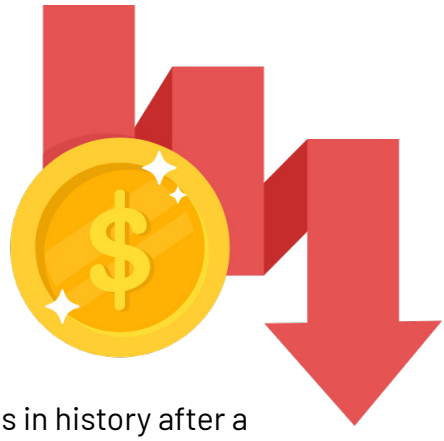
ActionScript applications often have to be updated and tested for compatibility with different ActionScript versions. ActionScript developers often have to ensure that their applications are compatible with multiple ActionScript versions, which can be difficult and time-consuming. ActionScript applications can also be vulnerable to versioning issues, such as ActionScript code that only works with specific ActionScript versions.

In comparison to other programming languages, ActionScript has a more dynamic versioning system that can be difficult for ActionScript developers to manage. ActionScript also has limited version control tools, which can make it difficult for ActionScript developers to identify and fix ActionScript code errors related to ActionScript versioning issues. This can lead to ActionScript applications being vulnerable to security issues and application crashes, which can be a significant risk when developing ActionScript applications.

As ActionScript is a programming language, the risks associated with ActionScript development are similar to those related to other development languages such as Java and C++. However, ActionScript has unique risks associated with it, such as ActionScript versioning and ActionScript debugging difficulties. Therefore, ActionScript developers need to take additional steps to ensure that ActionScript applications are secure and optimized to mitigate these risks.

The *Cost* of Not Securing ActionScript Applications

The consequences of insecure ActionScript applications can be costly in terms of time and money. ActionScript application vulnerabilities can lead to data breaches, crashes, and other issues impacting an organization's reputation and bottom line.



Data Breaches

In 2021, Android experienced one of the most significant data breaches in history after a misconfiguration of cloud services occurred. The error exposed databases containing over 100 million users' personal information, including names, email addresses, dates of birth, chat messages, location, gender, passwords, photos, payment information, phone numbers, and push notifications.

A report by IBM revealed that the average cost of a data breach in the US is $9.44 million, which is $5.09 million more than the global average. Databases are a common target for malicious actors, and ActionScript applications can be vulnerable to data breaches if data security measures are not taken. It is, therefore, essential for ActionScript developers to ensure that ActionScript applications are secure and regularly updated with the latest security patches.

Application Crashes

Malicious attacks can lead to applications crashing, which can be damaging to businesses in terms of money and reputation. In 2022, 70 Ukrainian government websites crashed due to malicious cyber-attacks designed by Russian hackers. Diia was one of the main websites compromised in the attack, which offers government services like storing personal vaccination records and certificates (and also allowed Ukrainian citizens to report the movement of Russian troops).

Such crashes cost time and money to repair, adversely affecting an institution's bottom line. ActionScript applications can be vulnerable to attacks if the ActionScript code is faulty or outdated. This is why ActionScript developers should ensure that their applications are regularly tested for errors and optimized for performance to reduce the risk of application crashes.

Loss of Customer Trust

Data breaches and application crashes can lead to a bad reputation for organizations, which can be damaging in the long term. Customers may be reluctant to use services from companies that have experienced data breaches, which can impact the customer base and decrease sales. For example, in 2010, Farmville, a popular Facebook game developed by Zynga, experienced a data breach that led to the personal information of over 200 million users being exposed. The company faced backlash from customers and experienced a decrease in profits due to the breach.

How Organizations *Can Secure* ActionScript Applications

Organizations can take certain proactive measures to secure ActionScript applications and reduce the risk of data breaches and application crashes, including:

Adopting Industry Security *Best Practices*

Organizations should adopt security best practices such as using secure coding techniques, encrypting data, and regularly patching ActionScript applications with the latest security patches to reduce the risk of data breaches and application crashes.

Data Encryption

Cryptographic failures are one of the most common data security risks for ActionScript applications. These attacks occur when an attacker can decrypt sensitive data stored within an ActionScript application.

Data encryption is a process by which developers can ensure that any secure or sensitive data stored within an ActionScript application is protected from potential attackers. By encrypting the data, ActionScript developers can ensure that any sensitive data is kept secure and safe from malicious attacks. Developers can encrypt data in the following ways:

- **Symmetric encryption:** This type of encryption uses a single key to encrypt and decrypt data.
- **Asymmetric encryption:** This type of encryption requires two different keys for encrypting and decrypting data.
- **Hashing algorithms:** This type of encryption uses a mathematical algorithm to create a hash value for the data, which is then used to store and transfer the data securely.

Input Validation

As discussed earlier, injection attacks are one of the most common security risks of which ActionScript developers need to be aware. These attacks occur when malicious code is injected into an ActionScript application, allowing hackers to access the application and sensitive data.

Input validation effectively prevents injection attacks by ensuring that user input is sanitized before the application processes it. This process works by checking the user input in an ActionScript application to ensure it is valid and secure, thus protecting it against malicious attacks.

Patch Management

Patch management aims to efficiently and promptly distribute updates that fix software errors, commonly known as vulnerabilities or bugs. When bugs arise, organizations should take the necessary steps to apply the latest security patches to reduce the risk of data breaches and application crashes. The process involves continuous monitoring of ActionScript applications for errors, followed by prompt patching with the latest software security updates.

Endpoint Security

In addition, organizations should also employ endpoint security measures such as firewalls and antivirus software. Firewalls help protect against malicious traffic and intrusions, while antivirus software helps detect and remove malicious code from ActionScript applications.

Backups

Organizations should also create regular backups of their ActionScript applications to restore them in the event of data breaches or application crashes. By creating backups, organizations can quickly restore their ActionScript applications and ensure minimal downtime in case of system failure.

Regular *Testing*

Various tools are available for application testing, such as static application security testing (SAST), software composition analysis (SCA), and Add-Ons.

SAST

SAST is a critical component of the software development life cycle, helping teams identify applications' vulnerabilities before deployment. Teams can code, test, and revise in this stage to ensure production products meet expectations.

Kiuwan's SAST solution can help ActionScript developers assess applications and eliminate potential risks before they are released. Developers can use SAST to scan...

...ActionScript code for security vulnerabilities and errors, identify and fix ActionScript code issues, and reduce the risk of application crashes and data breaches.

SCA

Open-source solutions come with many security, dependency, and license compatibility issues that developers must consider before launching applications.

SCA is a tool that developers can use to scan ActionScript code for known open-source vulnerabilities and license violations. This tool scans ActionScript code and compares it to a database of known open-source components. If any issues are found, developers can work to identify and address any potential risks.

Kiuwan SCA allows developers to test the security of applications. This tool can help identify any open-source components that could be vulnerable to malicious attacks and work to address any issues before launching ActionScript applications into production.

Add-Ons

Add-Ons are third-party components developers can add to applications to extend the application's functionality. They add features and functionalities to ActionScript applications to enhance the user experience.

However, ActionScript developers should be aware that these third-party components can introduce potential security risks to ActionScript applications, as they may contain malicious code that hackers can use to gain access. ActionScript developers should always ensure that any Add-Ons they use are obtained from reliable sources and that ActionScript applications are regularly tested for security vulnerabilities.

Kiuwan's Add-Ons allow organizations to *improve security* by identifying code defects quickly and managing remediation efforts. Kiuwan Code Analysis (QA) integrates directly into the development infrastructure, providing decision-support indicators for changes. This tool is a multifaceted addition to the Kiuwan platform that goes beyond code analysis to secure your codebase entirely from start to finish.

Benefits of Securing ActionScript Applications

As is often the case with software development, securing ActionScript applications is a complex process. However, taking proactive steps to secure these applications can provide organizations numerous benefits:

1. **Improved user experience:** Secure ActionScript applications can provide users with a better and safer user experience, as they are less likely to encounter issues and malicious attacks.

2. **Reduced risk of data breaches:** By taking the proper steps to secure ActionScript applications, organizations can reduce the risk of data breaches, as any malicious code is less likely to gain access to the application.

3. **Better compliance:** Organizations can also benefit from improved compliance with industry regulations such as HIPAA and GDPR, as ActionScript applications that are securely developed and maintained can help organizations remain compliant.

4. **Save time and money:** Securing ActionScript applications can help organizations save time by taking proactive steps to reduce the risk of application crashes or data breaches. Additionally, taking the proper steps to secure ActionScript applications can help organizations save money, as they are less likely to spend resources addressing security issues or data breaches.

5. **Increased customer trust:** Finally, secure ActionScript applications can also help organizations build customer trust, as customers are more likely to trust and use applications that they know are secure.

Secure Your ActionScript Applications

Security is a critical component of any successful software development project. Taking the proper steps to secure your ActionScript applications can help ensure that they remain safe and secure and provide numerous benefits for organizations.

ActionScript developers are responsible for ensuring that the applications they create are secure. Kiuwan is the best solution for ActionScript developers to use to test and secure their applications.

Kiuwan is a powerful application security tool that can help ActionScript developers identify any open-source components that could be vulnerable to malicious attacks and work to address any issues before launching ActionScript applications into production. Additionally, Kiuwan's Add-Ons allow organizations to improve security by identifying code defects quickly and managing remediation efforts.

Explore Kiuwan further with a free trial, and ensure that your ActionScript applications are properly protected and provide users with a safe and secure user experience.

*YOU KNOW **CODE**, WE KNOW **SECURITY!***

GET IN TOUCH:

Headquarters
2950 N Loop Freeway W, Ste 700
Houston, TX 77092, USA

United States **+1 732 895 9870**
Asia-Pacific, Europe, Middle East and Africa **+44 1628 684407**
contact@kiuwan.com
Partnerships: **partners@kiuwan.com**