A Risk Analysis for United States IT Organizations

kiuwan
An Idera, Inc. Company

INDEX

Overview

The 21st century is heavily influenced by technology, due to which more and more organizations are using automated informational technology solutions to meet their goals and develop long-lasting customer relationships.

Unfortunately, there's still a significant gap between the use and security of these tech tools.

The gap is creating problems for information technology organizations worldwide, especially in terms of how they use IT solutions. In the United States, the effect is even more pronounced. Even though IT teams are pushing hard to create effective tech solutions and engaging tools, the gap between usage and security needs bridging.

This guide goes into depth about information technology security and explains its importance for organizations. Plus, we explore the role of the IT security industry in the U.S. tech space.

What Is **Risk Analysis**?

Risk analysis in an enterprise refers to the process of identifying and prioritizing risks that may harm the organization over time. Risk analysis aims to help organizations find ways to manage these risks more effectively and efficiently.

This must be done before any negative consequences can occur, such as decreased productivity or financial losses. A strong risk strategy means an effective risk analysis plan.

Risk analysis requires a thorough examination of the potential threats and dangers, along with their probability of occurring. This helps organizations take immediate action on issues that are highly likely to affect them in the future.

In an IT organization, risk analysis has a lot to do with app cybersecurity, application security, code security, and data security. It is imperative for information technology organizations that develop applications or software used by a large audience.

Types of **Risks Analysis**?

Many factors lead to risks in U.S. IT organizations, such as using insecure third party code, inefficiencies in the development operations, insufficient IT app cybersecurity, and poor application security.

Here are some types of risk analyses that cover all these issues in the IT security industry:

Qualitative. Qualitative risk analysis refers to the non-numeric measurement of risk. It is a process that involves assessing and ranking risks by categories such as likelihood and impact based on subjective judgment.

Quantitative. Quantitative risk analysis refers to the numeric measurement of risk. It uses statistical models, such as Monte Carlo simulations, to represent uncertainty about how variables will affect outcomes. For example, a quantitative model might crunch data about the number of malicious code attacks over the Microsoft Windows operating system, how many succeeded, and what impact each had to come up with an average cost per successful attack.

Site-Specific. A site-specific risk analysis for a specific location and activity is carried out. It's perfectly relevant for the industry in which it is being conducted and effectively controls or eliminates risks.

Steps in a Risk Analysis Process

Before you can manage any risk, it's important to determine the hazard at hand, the impact on stakeholders, and the regulations you must follow to mitigate or resolve it.

The process can be divided into ***five steps***.

Step **1**

Hazard Identification

The first step is to assess the nature and probability of a material risk. It's important to determine who might be harmed and why, and what damage they might suffer if harmed.

For instance, if the risk is cross-site scripting, this step ensures a clear overview of who will be affected, how much they will be affected, and how to identify the presence of an XSS risk.

Step **2**

Risk Characterization

This step involves describing the risk in numbers or tables. It provides an understanding of how much damage could occur due to the risk you have identified. For instance, Statista data shows that in 2020, the average loss businesses suffered in the U.S. due to data breaches was $8.64 million.

Step **3**

Risk Evaluation

The next step is to evaluate the risk and compare it with the impact. How likely is it for the risk to become a hazard? How severe will the consequences be?

For instance, if an organization has identified two or three risks, risk evaluation will help them prioritize the risk they need to reduce or tackle first.

Step **4**

Risk Treatment

These are actions you should take to reduce or avoid the risk as much as possible. They will vary depending on the type of risk and your organization's readiness to deal with it.

Step **5**

Risk Communication

Once an effective risk mitigation strategy is created, other departments will be affected by it. For example, IT security should know about the steps taken to avoid risks that involve cybersecurity. A meeting with them is in order to explain how mitigation strategies will affect their duties.

Communication also means recording the findings. In case there's a similar risk in the future, a record will facilitate a response in a shorter time span.

Risk *Associated* With Insecure Mobile Applications

Vulnerabilities in **Android apps**

Vulnerabilities in **iOS apps**

Insecure data storage in **Mobile apps**

According to PT security's report, high-risk vulnerabilities were found in 43% of Android and 38% of iOS mobile applications. The most common issue on both platforms is insecure data storage. 76% of mobile applications are affected by this problem, which means the financial information, personal data, and passwords of the users are at risk.

Alarmingly, hackers don't need to physically access the smartphone to steal the users' data. Rather, they can exploit 89% of vulnerabilities through malware. The lack of mobile app security is one of the major challenges the IT sector is facing in the US. A study showed that more than 10,573 malicious mobile apps were blocked daily in 2018.

With the advancement of technology, it has become easy to build apps and deploy them. At the same time, it has become quite simple for nefarious individuals to crack an app's security since the code developers are writing is insecure.

Kiuwan is a helpful tool that performs code analysis, identifies code defects, and ensures code security, especially for third party code. It has an easy and instant setup that starts scanning for code vulnerabilities in just minutes.

Plus, you can integrate Kiuwan with your CI/CD/DevOps pipeline to automate your security processes.

Speaking of security processes, here are the top risks associated with application security in the U.S.:

Insecure Communication

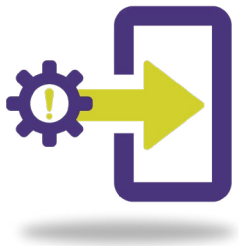
Data in a mobile app is exchanged in a client-server manner. When transmitting the data, the app traverses through the mobile app user's carrier network and the internet.

Hackers and attackers can exploit that weak point to steal or tamper with the data. Moreover, since the apps use HTTP for communication, hackers can read the transmitted data if they are able to intercept it during its transfer.

Unavailable Input Validations

Input validation refers to the process of filtering the input received by an application. It is especially critical for apps that accept user inputs to take specific actions or modify data in some way.

Poor input validation techniques can lead to security issues, such as SQL injection, which enables hackers to override internal variables and execute malicious queries on a database. Additionally, this puts sensitive information at risk of being stolen or manipulated.



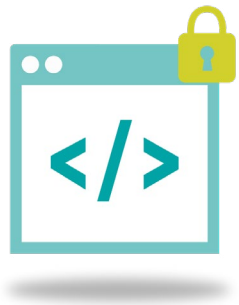
Insecure Data Storage

As mentioned earlier, insecure data storage is one of the most common mobile app security issues. Developers often store sensitive data such as passwords, credit card numbers, and contact details in an unencrypted form on the device.

This practice makes it easy for hackers to steal the data if they gain access to the smartphone. Hackers can do this by exploiting other vulnerabilities or through malware.

Insecure data storage results in the following:

- **Loss of Intellectual Property.** Sensitive company data can be stolen and sold to the competition, or it can simply be published online for all to see. Likewise, individuals can also lose their intellectual property, especially if the app is built for business or artwork creation
- **Financial Loss.** Credit card numbers, bank account details, and other financial information that is stored in an unencrypted form can be easily accessed and used by criminals.
- **Reputational Damage.** If confidential customer data is stolen, it can lead to the brand's reputation being damaged.
- **Compromised Compliance.** Sensitive data stored on mobile devices is at risk of breaching compliance with industry regulations such as HIPAA, PCI-DSS, and GDPR.
- **Disclosure of Personal Information.** If hackers gain access to any unencrypted personal information, they could use it for identity theft.
- **Privacy Policy Noncompliance.** If security protocols are not in place, it can violate the company's privacy policy and lead to civil actions.



Code Security

Code security issues are some of the most complicated issues that developers face. Hackers can also exploit this issue to gain access to any app or its server.

Poorly written code is often vulnerable to SQL injection, cross-site scripting attacks, and other vulnerabilities. Hackers can use automated tools to detect coding errors in an application built using open-source software.

Although you can use automated tools to perform static analysis or fuzzing to identify weak encryption and insecure data storage, some level of manual effort still goes into finding security threats in the code.

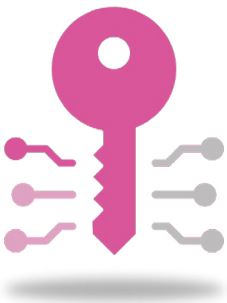
Kiuwan is a global organization providing an end-to-end application security platform with specialized tools that help your team identify vulnerabilities in your application code security. You can simply integrate Kiuwan with your CI/CD/DevOps pipeline to automate your security processes.

Insufficient Authorization and Authentication Controls

Poor or missing authentication schemes enable hackers to easily bypass the authentication process in order to gain access to private or secure data.

In mobile apps, authentication requirements are different from web applications. Users don't have to be online continuously throughout their sessions. Some mobile apps have uptime requirements for offline authentication.

However, this offline authentication method poses a security risk to developers.



Poor Encryption

Encryption refers to the process of converting data into a readable code to ensure its confidentiality and integrity. There are many ways that developers can use encryption in mobile apps, such as using HTTPS for communication with backend servers, encrypting sensitive data on the device by using keystores, and hashing and salting passwords before storing them in an unencrypted form in the database.

Poor encryption leads to the following:

- **Data Breach.** If the encryption is weak, it can be easily cracked by hackers, which can lead to a data breach.
- **Loss of Confidential Information.** Unencrypted data is at risk of being stolen by hackers or other malicious individuals.
- **Compromised Security.** Encrypted data that is not properly protected can be easily decrypted by hackers.

Vulnerable Interfaces and APIs

Several mobile apps connect to one or more external applications and content providers to provide a rich user experience. However, this connection poses the following security risks:

- **Unprotected Data Exchange.** Unencrypted data can be accessed by any application that connects to these external systems.
- **Vulnerable APIs.** Unsecured APIs and backend databases can be used by hackers to bypass authentication and gain access to any application or its server.
- **Weak Authorization and Authentication Controls.** Lack of proper security controls while connecting to external applications can lead to vulnerabilities.

Most Common Web Security Vulnerabilities

The Open Web Security Project is an ambitious effort to identify and explain web vulnerabilities that are capable of compromising security.

The project supports individuals, organizations, and companies by providing the public with information about what makes up secure Web applications. The OWASP project is run by a non-profit charitable organization.

OWASP publishes the top web security vulnerabilities based on data collected from different security organizations. The vulnerabilities are prioritized on the basis of:

- **Exploitability.** How easy is it to exploit the vulnerability? What does the attacker need for exploration? The highest exploitability is when the attacker only needs the web browser. The lowest is when tools and advanced programming are required.
- **Detectability.** How easy is it to detect the vulnerability? For example, does the application crash or return an error message when exploited? The highest detectability is when the information is displayed as an error message. The lowest is when it is shown in the source code.
- **Impact.** How much damage does the user sustain if the vulnerability is attacked or exposed? The lowest is, of course, when there is no damage. The highest is when the whole system crashes.

Here are the top ten risks identified by the organization and ways to mitigate them.

1 SQL Injection

An injection is a type of security vulnerability that can occur when user input is not properly sanitized. It allows an attacker to execute malicious SQL statements against a database.

The injection takes place when the user input is not properly filtered or validated before it is passed to the database. Here are the implications of SQL injection:

- An attacker may inject malicious content into the vulnerable fields.
- Someone may read sensitive information, like passwords and usernames, from the database.
- An attacker may update, delete, insert, or make other changes to the database.
- An attacker may execute administration operations on the database.

The vulnerable objects, in this case, are the URLs that interact with the database and the input fields.

How to Mitigate?

Organizations can take the following steps to prevent SQL attacks:

- **Validate User Inputs.** Ensure that all user inputs are properly sanitized and filtered before they are passed to the database.
- **Use Parameterized Queries.** These queries, bind input parameters to a query string. Thus, the user input is treated as data and not as part of the SQL command.
- **Use Prepared Statements.** This technique is similar to parameterized queries, but it requires more coding effort.
- **Limit Privileges.** Limit the privileges of the users who have access to the database and make sure that they are authorized for what they need to do in the application itself.
- **Disable Unused Features and Functions.** Many applications have unused features and functions that can be exploited by attackers. Disable these features and functions until they are needed.
- **Ensure Proper Security Measures.** Implement proper security measures such as firewalls, intrusion detection/prevention systems, and proper access controls.
- **Patch Applications and Operating Systems.** Make sure that you regularly patch your applications and operating systems with the latest security updates.

2 Cross-Site Scripting

Cross-site scripting or XSS refers to a vulnerability that allows an attacker to inject malicious code into a web page. The code is executed when the page is viewed by a victim.

Attackers use this to execute malicious scripts. As the browser cannot differentiate between trustworthy and untrustworthy scripts, it executes the script.

The attacker then hijacks the session cookies and can also redirect the user to a malicious or unwanted website. Here are the implications of XSS:

- An attacker may inject malicious content into the vulnerable fields.
- Someone may read sensitive information, like passwords and usernames, from the database.
- An attacker may update, delete, insert, or make other changes to the database.
- An attacker may execute administration operations on the database.

The vulnerable objects, in this case, are the URLs and input fields.

How to Mitigate?

You can mitigate XSS through content security policy, which defines how incoming data should be handled by the browser. Here are two other ways:

- **Input Validation.** Ensure that user inputs are filtered at runtime to check for executable code, meta-characters, and other malicious content before it is processed by the application.
- **Encode Output.** Encode your output before displaying it on the web page, so any malicious scripts are rendered harmless.

3 Cross-Site Request Forgery

CSRF or cross-site request forgery takes place when an attacker tricks the victim's browser to perform actions on a website without the user's knowledge. The attackers may trick the users into clicking on malicious links that contain forged requests. The attacks are generally carried out through email, instant messaging, social media, or other online services.

The requests must be sent from a trusted source, so an attacker will either have to trick the victim into clicking the URL or use social engineering to take control of their accounts. The latter is more difficult but not impossible.

Here are the implications of CSRF:

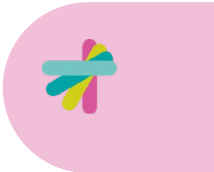
- An attacker can perform actions on a website without the user's knowledge.
- They can use this vulnerability to change the user's profile information, create a new user account, or change status.

The vulnerable objects are the user profile page, business transaction page, and user account forms.

How to Mitigate?

You can mitigate CSRF attacks by using the following techniques:

- **Cookies and Token.** Use cookies and tokens to authenticate requests from the user's browser. Doing so ensures that only the authorized user can make requests to the website.
- **Form Authentication.** Use form authentication to authenticate requests that target business transactions.
- **Input Validation.** Perform input validation on all user inputs and accept only known, safe values such as numbers and characters.



4 Broken Authentication and Session Management

Websites typically create a session ID and cookie for every valid session. The cookies contain sensitive information, like the user's name and email address.

The session ID is used to identify the user's session and to keep track of their activities on the website. However, if the session ID or cookie is compromised, the attacker can gain access to the user's account and perform actions on their behalf.

Session ID vulnerabilities can be exploited by an attacker who manages to recover a session ID. The attacker can then use the stolen ID to access the user's account and carry out transactions, or to change the user's profile details.

Here are the implications of this vulnerability:

- An attacker can gain access to a system that lets them modify information and hijack a session.
- An attacker can hijack a session through sessions or cookies via XSS.

The vulnerable objects in this regard are session IDs and an application's authenticated parts.

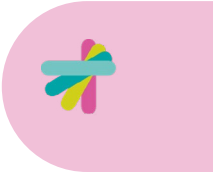
How to Mitigate?

Organizations can secure their development operations and mitigate these risks by:

- **Protecting Credentials.** Ensure that all credentials are encrypted and securely stored.
- **Enforcing Session Management.** Ensure that the session ID is unique for each user and can't be reused or shared with other users.
- **Implementing a Secure Password Policy.** Include a minimum length, special characters, required number of upper and lower-case letters, numbers, and symbols in a password.
- **Using Session Fixation Protection.** Protect the session ID from being hijacked by using a secret key or token that is known only to the server and the client.

5 Insecure Direct Object References

An insecure direct object reference develops when a developer uses a direct object reference in their code without checking for existing objects. The vulnerable objects in this regard are transfers and the removal of money from an account.



These attacks can be used to:

- Change the quantity on an order.
- Transfer funds from one user's account to another, or remove them altogether.
- Create a new account with increased privileges.

How to Mitigate?

Organizations can mitigate these risks and ensure information technology security by:

- **Restricting Access.** Restricting the user's access to specific pages and objects on the website. Doing this will help prevent them from accessing or modifying sensitive data.
- **Checking for Existing Objects.** Checking for existing objects prior to creating a new one will allow you to spot and block any illicit activity that may arise.
- **Input Validation.** Input validation ensures that only the right data is submitted by users. It reduces the risk of code injection, which can lead to server-side attacks such as SQL injection and cross-site scripting (XSS).

6 Security Misconfiguration

Security configuration refers to the process of hardening a system to make it more resistant to attack.

Organizations often overlook security during the software development process, which can lead to vulnerabilities that can be exploited by attackers. Security misconfiguration is one of the most common types of vulnerabilities found in U.S. organizations today.

A 2017 Threat Stack study showed 73% of companies have at least one security misconfiguration.

To make matters worse, a Data Breach Investigations Report by Verizon showed that it takes just minutes for attackers to compromise an organization's system. Only 3% of organizations discover the compromise quickly while the rest take too long, giving cybercriminals plenty of time to steal an organization's or its users' data.

Misconfigured systems are easy prey for attackers who are looking for an easy way into the network.

They can find these systems by scanning the network for open ports and services that are running on them. Once they've found a system that is vulnerable, they can exploit it to gain access to the network and sensitive data.

Here are the implications of security misconfiguration:

- It can lead to a data breach.
- An attacker can exploit the vulnerability to access an organization's system.
- An attacker can gain access to sensitive information or to systems that are running on the network.

The vulnerable objects in this regard are servers, web applications, and databases.

How to Mitigate?

Organizations can mitigate these risks through application security testing, perfecting IT app cybersecurity, and doing the following:

- **Configuring Systems Securely.** Apply the latest security patches and configure systems to run only the services and applications that are needed.
- **Restricting Access.** Restrict access to sensitive data and systems only to authorized users.
- **Monitoring and Logging.** Track all activity on systems and applications and review logs regularly.
- **Incident Response Plan.** Have a plan in place to quickly respond to any security incidents.
- **Monitoring Systems.** Configure systems to send alerts when they are accessed.

7 Insecure Cryptographic Storage

One of the main concerns in the IT security industry is insecure cryptographic storage, which is a threat to digital identities and transactions.

When storing passwords, the website encrypts the password using an encryption algorithm before it is stored in a database. The problem arises when the website stores the encrypted data rather than the raw password value.

The sensitive information can be found inside a system's configuration files or other types of databases. An attacker can extract it from these sources and decrypt the data.

An attacker can view, edit, or delete data on a system.

The vulnerable objects in this regard are passwords and encryption keys used to encrypt/decrypt data.

How to Mitigate?

IT organizations can secure their development operations by securing cryptographic storage in these ways:

- **Store Passwords in a Hashed Format**. When storing passwords, hash them using a secure hashing algorithm. Doing so makes it difficult for an attacker to extract the cleartext password from the hashed value.
- **Use Strong Encryption Algorithms** like AES to encrypt data.
- **Secure Cryptographic Storage.** Ensure that sensitive data is adequately protected from unauthorized access.

8 Security Logging and Monitoring Failures

Another common concern in the IT security industry is the prevention of data breaches. Security logging and monitoring can help identify when these breaches occur and allow for a quicker response. However, these systems must be properly configured and monitored to be fully effective.

One common mistake is to focus on the detection of malicious activity without also looking for signs of accidental or unauthorized access. A security incident can still occur even if no malicious code is found, so it is important to have a comprehensive security monitoring solution in place.

Kiuwan offers security solutions for your DevOps processes. You can scan your code and identify vulnerabilities. The solutions are compliant with stringent security standards, including SANS, CERT, PCI, and OWASP.

How to Mitigate?

Here's how to mitigate security logging and monitoring failures:

- **Involve Security Professionals.** Make sure security professionals are involved early in the development lifecycle. They can help you identify weaknesses before they are built into your products. It will improve security and reduce costs associated with remediation later on. You should also communicate to all of your developers that their code is being reviewed for vulnerabilities at every stage of its development lifecycle.
- **Create a Security Professional Group.** It is often very difficult for one person or department to know all of the tools and techniques involved in securing an application. A group of security professionals, each with their own specialty, can help you evaluate your products early on.

- **Establish Secure Software Development Policies.** Establish policies that set secure coding standards to help ensure that your developers are using current and effective methods for writing secure software.
- **Use Standard Development Tools.** Use development tools that automate the build process to reduce the amount of time developers spend entering the constant strings needed for encryption, hashing, or encoding operations. Doing this also makes it much easier to notice when something is typed incorrectly.

9 Lack of Transport Layer Protection

Transport layer protection is a security measure that helps protect information as it is transmitted between systems. Without transport layer protection, attackers can eavesdrop on communications and steal data.

To secure communications between systems, IT organizations should use strong encryption algorithms, like AES, and ensure that sensitive data is adequately protected from unauthorized access. Additionally, it is important to remember to activate transport layer protection when transmitting sensitive data.

The minimum impact of this vulnerability is the privacy violation of the users. The violation of any user's confidentiality leads to identity theft, fraud, and reputational damage.

How to Mitigate?

OWASP recommends the following mitigation steps for the IT security industry.

- **Apply SSL/TLS.** Apply TLS or SSL to transport channels enforcing encryption of data in transit.
- **Do not Use RC4.** Remove or disable support for cipher suites that use the RC4 algorithm.
- **Use Strong Cryptographic Ciphers.** Ensure that all cryptographic algorithms and protocols, such as AES and SSL/TLS, used to protect sensitive data operate using only strong cryptographic ciphers with sufficiently long keys.
- **Limit Attack Surface.** Restricting external access to ports 22 (SSH) and 5357 (MQTT), both on the server-side as well as at the IoT device level, can be important to prevent direct attacks from outside an organization's network perimeter.
- **Require SSL Chain Verification.** Ensure you are using a Certificate Authority (CA) that performs OCSP/CRL verification since this provides strong security and availability guarantees.

10

Insecure Design

OWASP's report showed that insecure design emerged as a new security risk in 2021, as compared to the report from 2017. Insecure design represents several weaknesses, such as insecure coding, insecure communications, and platforms.

One reason for the increase in insecure design is the increasing use of third-party components in software development. These components may be from unknown or untrusted sources, which can introduce vulnerabilities to the software.

Insecure design can also arise when developers do not properly understand the security risks associated with their code. For example, they may not properly vet input data that could be used to exploit a vulnerability.

How to Mitigate?

To mitigate the risk of insecure design, organizations should ensure that their developers have a good understanding of secure coding practices. They should also perform regular security reviews of their software products to identify any vulnerabilities.

OWASP recommends the following to prevent insecure design:

- **Establish Secure Development Lifecycles.** Work with AppSec professionals to develop a secure SDLC that includes security reviews and testing throughout the software development process.
- **Require Security Training.** Developers should be required to complete security training courses that teach them how to write code securely.
- **Use OWASP Resources.** OWASP provides a variety of resources that developers can use to learn about secure coding practices. These resources include the OWASP Top 10, the OWASP Application Security Verification Standard (ASVS), and the OWASP Secure Coding Practices Quick Reference Guide.
- **Perform Code Reviews.** Security professionals should perform code reviews to identify insecure coding practices and vulnerabilities.
- **Use Automated Tools.** Automated tools, such as Kiuwan, can help identify vulnerabilities in code. These tools identify defects in your code as per industry standard characteristics, such as reliability, efficiency, portability, and maintainability.

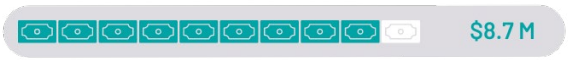
Challenges in the IT Security Industry

According to Ponemon's 2017 Cost of a Data Breach Study, it takes 206 days on average for organizations to identify a security breach in the U.S. A breach identified in 100 days cost $5.99 million to companies, while a breach that took longer to be identified cost up to $8.7 million.

Apart from late identification of security risks, the IT security industry also faces some other challenges.

A Breach Identified in **(100 Days) cost:**

A Breach Identified in **(100 + Days) cost:**



Adaption to Remote Workforce. Company data is under threat by internal employees who may have bad intentions towards their employers' data or who may accidentally release an important file due to their carelessness. At the same time, the COVID-19 pandemic has led to a large number of employees working remotely for the first time. This can be a challenge for companies who are not used to this way of working and who have not put the necessary security protocols in place.

IoT Attacks. IoT attacks are also a huge concern. In a recent study by Kaspersky, it was found that the number of attacks targeting IoT devices reached 1.51 billion breaches from January to June in 2021.

Emerging 5G Applications. 5G will increase the responsiveness and speed of data transmission. This has many positive consequences for companies in terms of faster communication and more efficient work processes. At the same time, this will also have consequences for cyber security installed in the network infrastructure, as 5G will enable new applications to access company networks.

Crypto and Blockchain Attacks. The crypto world is growing. With this growth, the IT security industry will also have to focus on protecting against security breaches and potential threats in the blockchain space, besides code security and application security testing.

Final Words

Wrapping up, it's no secret that the IT security industry is facing certain challenges that will only become bigger with the rise of blockchain, 5G, and other technologies. IT Organizations in the U.S. should implement effective risk analysis and mitigation practices to deal with these problems and empower the teams carrying out development operations.

In this regard, Kiuwan is a helpful tool that performs code analysis, identifies defects in your code, and ensures code security, especially for third party code. Moreover, the tool also audits your software vdeliveries from internal and external providers, defining checkpoints and comparing modifications.

Most importantly, you can manage your application portfolio across languages and understand all business risks completely. Kiuwan also helps immensely in objective decision-making since it offers features for predictive analysis.

REQUEST A TRIAL AT KIUWAN.COM/REQUEST-A-TRIAL
LEARN MORE AT KIUWAN.COM

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**