

No-Code Security Review

What should I review in applications without codes?

WHO AM I

Inaae Kim

 [@superhuman4891](https://twitter.com/superhuman4891)

Lead Security Engineer @ Unqork

Application Security Engineer

Software Engineer

DISCLAIMER

Opinions expressed are solely my own and do not express the views or opinions of my employer.

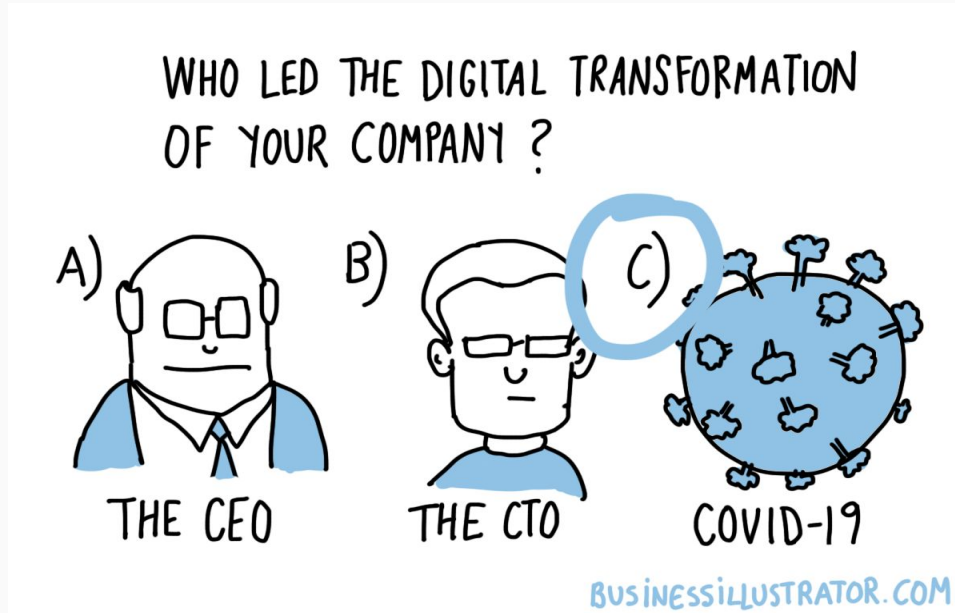
I do not promote or endorse any providers.

KEY TOPICS

1. What is Low-Code/No-Code application?
2. What are security risks and How to prevent them?
3. How to conduct security reviews?

SOFTWARE DEVELOPMENT TRENDS IN 2022

Increasing popularity of Low-Code/No-Code Platforms



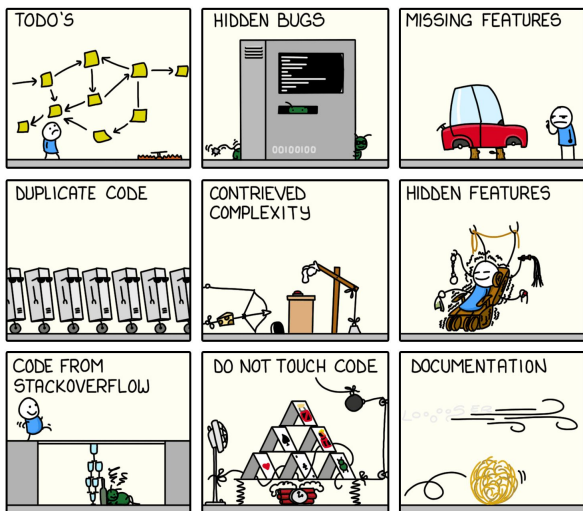
In 2020, 25% Use

By 2025,  70% Use

Recent research from
Gartner

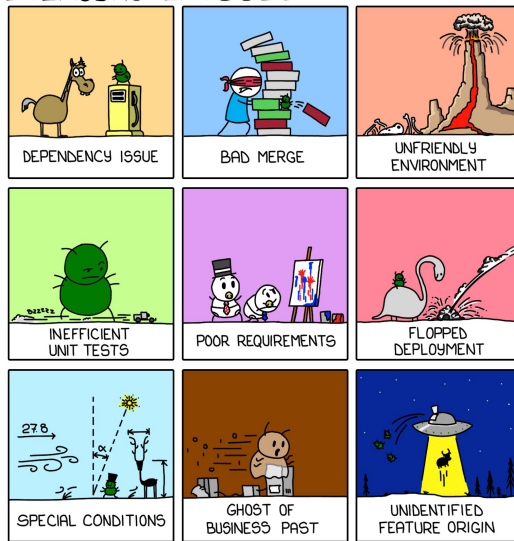
DIFFICULTIES - TRADITIONAL DEVELOPMENT

POSSIBLE CODE CONTENTS



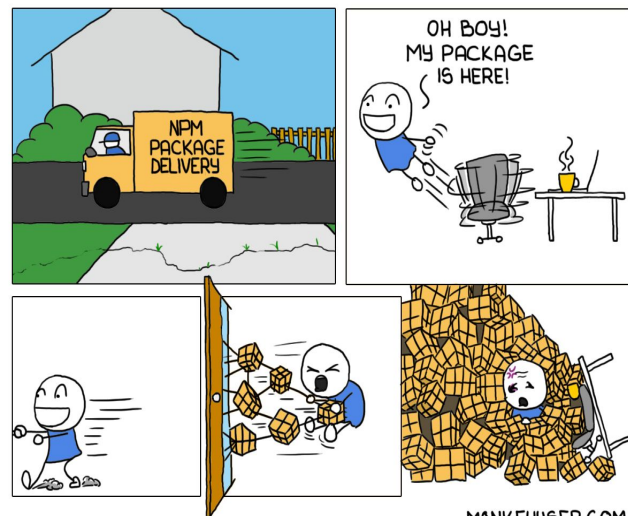
MONKEYUSER.COM

EVERYDAY EXCUSES



MONKEYUSER.COM

NPM DELIVERY



MONKEYUSER.COM

BENEFITS - LOW-CODE/NO-CODE DEVELOPMENT



Anyone can build

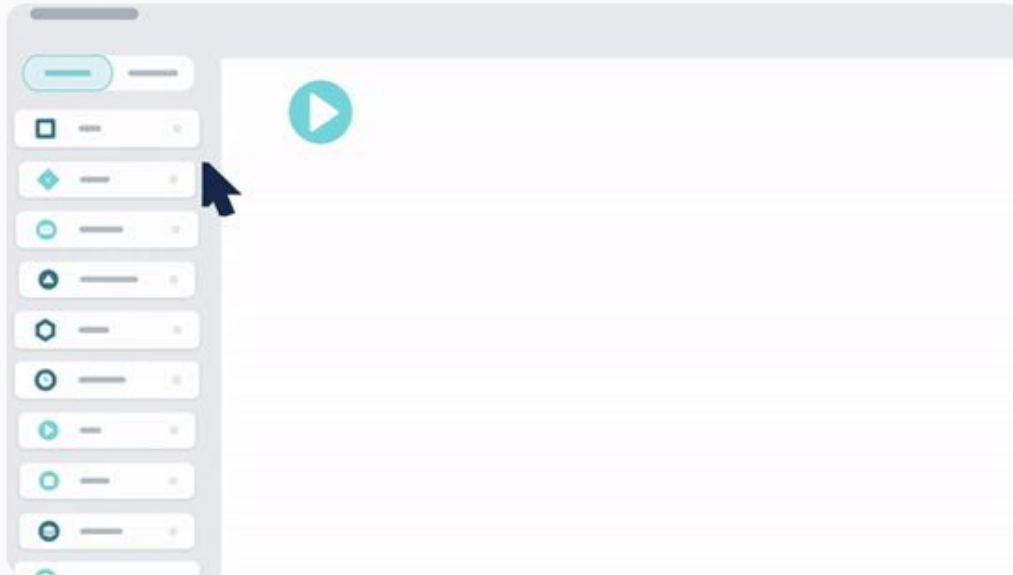


Rapid Application
Delivery



Cost Saving

KEY FEATURES



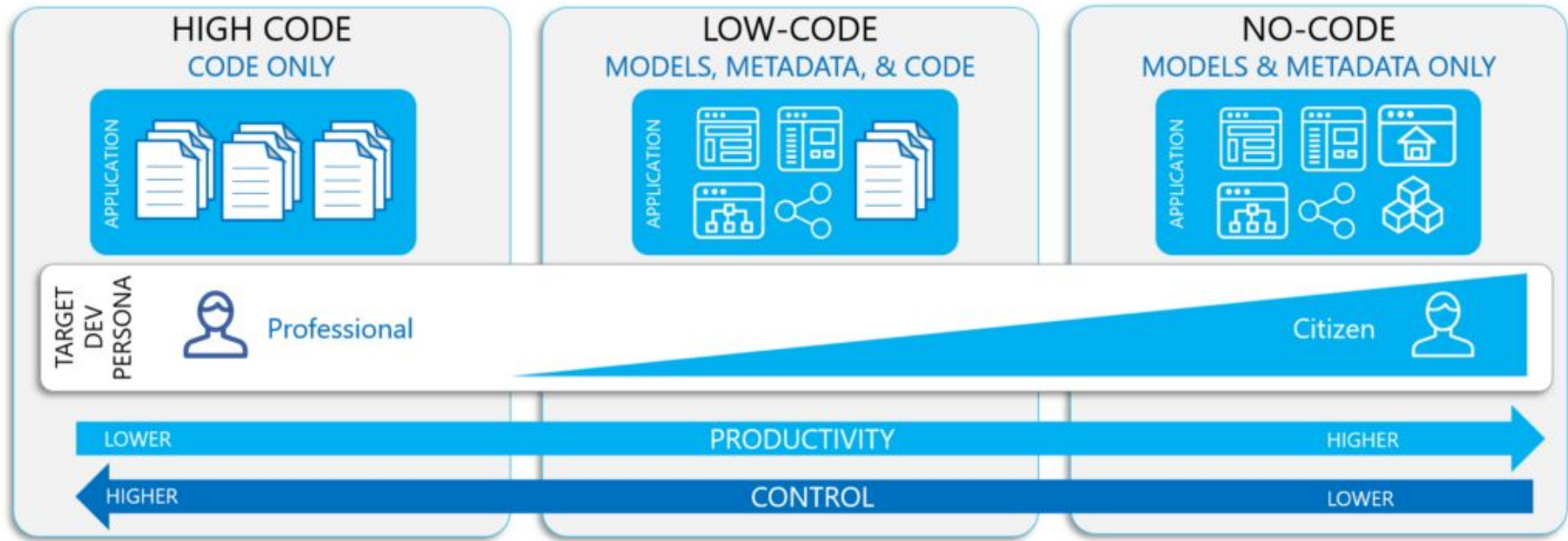
Drag and Drop
Interface

Visual Modeling

Reusability

Security

TRADITIONAL vs LOW-CODE vs NO-CODE DEVELOPMENT



SHARED RESPONSIBILITY MODEL

On-Premise You manage it	IaaS Infrastructure as a Service	PaaS platform as a Service	LCNC Platform	SaaS Software as a Service
Application	Application	Application	Application	Application
Data	Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware	Middleware
Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking	Networking
You Manage	Managed for you	Depends		

SHARED RESPONSIBILITY (SECURITY)

Customer

RBAC

Configuration

API

Security
Review

Compliance

Custom
code *

APPLICATION & DATA

LCNC Platform

Platform Software

Platform Vulnerability
Management

Platform Support

Operating
System & IAM

Compliance

Data
Encryption

Network
Traffic
Protection

Infrastructure
Deployment

** If the user uploads custom codes*

** Responsibilities may be different by vendors or each client's SLAs.*

SECURITY SURVEY

According to Dark Reading Survey, What security concerns do you have?



32%

Data
Governance



26%

Trust
platforms



26%

Vulnerability
Management



25%

Software
Inventory

Note: multiple response allowed. Data: Dark Reading survey of 136 IT, cybersecurity and application professional, January 2022, <https://www.darkreading.com/tech-trends/low-code-no-code-tools-are-popular-but-untrusted>

SECURITY CONCERNS

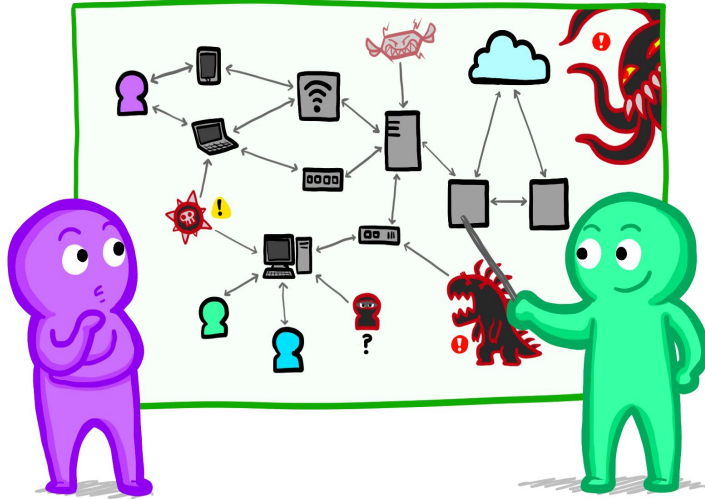
Reduce the security visibility and controls on applications

- Application inherits platform vulnerabilities
- Limit access to vendor systems

SECURITY CONCERNS

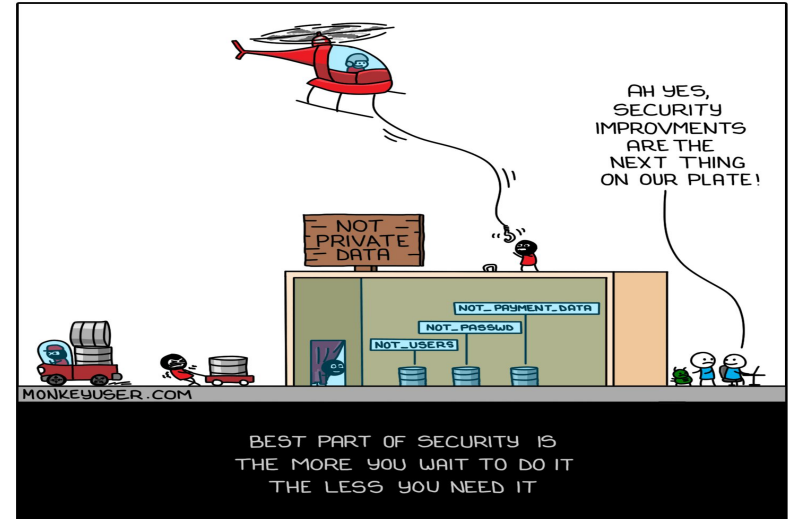
Lack of security knowledges

Lack of security training



Business Logic mistakes leak data

S3CUR1TY



OWASP LOW-CODE/NO-CODE SECURITY RISKS

LCNC-SEC-01: Account Impersonation

LCNC-SEC-02: Authorization Misuse

LCNC-SEC-03: Data leakage and Unexpected consequence

LCNC-SEC-04: Authentication and Secure Communication Failure

LCNC-SEC-05: Security Misconfiguration

LCNC-SEC-06: Injection handling Failures

LCNC-SEC-07: Vulnerable and Untrusted Components

LCNC-SEC-08: Data and Secret Handling Failures

LCNC-SEC-09: Asset Management Failures

LCNC-SEC-10: Security Logging and Monitoring Failures

NO CODE  **NO
VULNERABILITIES**

SECURITY CONCERNS

NO CODE  **NO SECURITY**

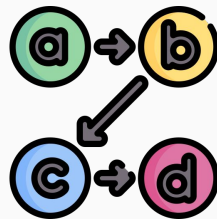
MITIGATIONS

1. Buy the platform from trusted vendors
2. Acquisition of Software bill of Materials (SBOM) from a vendor.
3. Security training for citizen developers
4. Integrate the security into low-code/no-code application development.

SECURITY REVIEWS

Application Security Engineer will review

Business Logics/Workflows
Application Configurations



Instead of

Source Codes



SECURITY REVIEWS

Application Security Engineer will train

Citizen developers/business users



Instead of

Professional developers



SOFTWARE DEVELOPMENT LIFE CYCLE



Plan

- Gather requirements
- Assess the platform



Build

- Security best practices
- Security controls



Verify

- Security testing



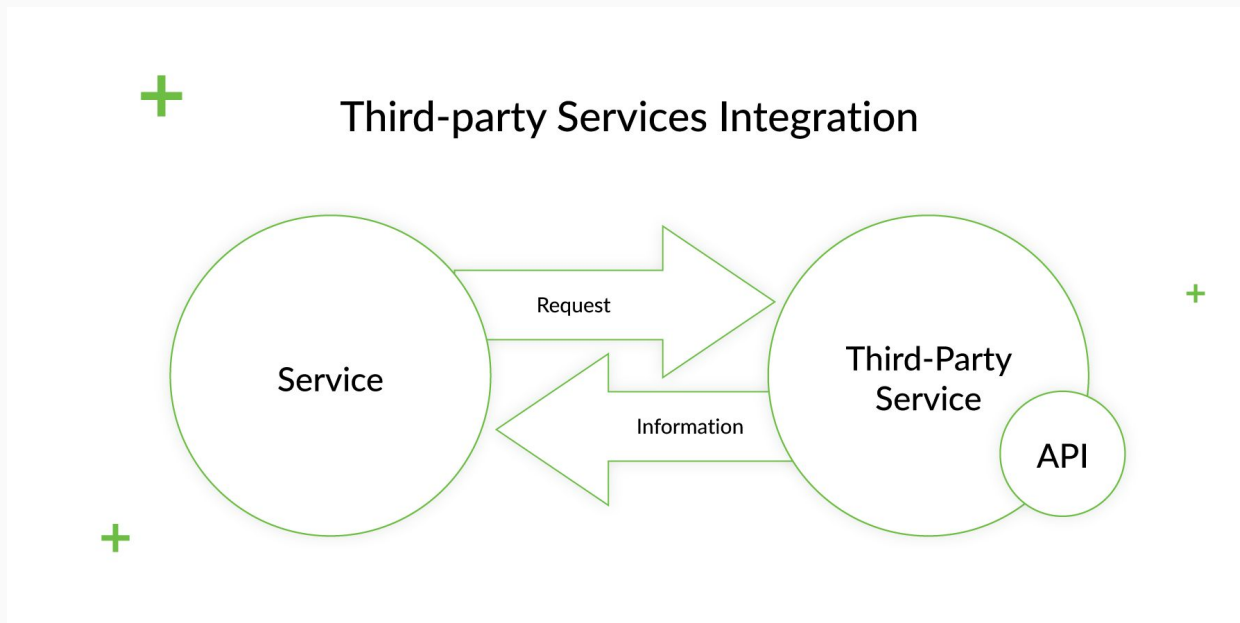
Run

- Version Controls
- Monitor Platform & Applications

EXAMPLE: SECURITY REVIEW 1

Service Integrations

Build an application that the users can see their own data only from the third party service.



EXAMPLE: SECURITY REVIEW 1

Developer creates a service

Service Info

Service title *

AppSecDemo

Service protocol + host *

https://appsecdemo.io/api/v2/

Service name *

This is permanent and cannot be reversed

AppSecDemo



Allow service execution server side only

Manage Access

Share to ⓘ



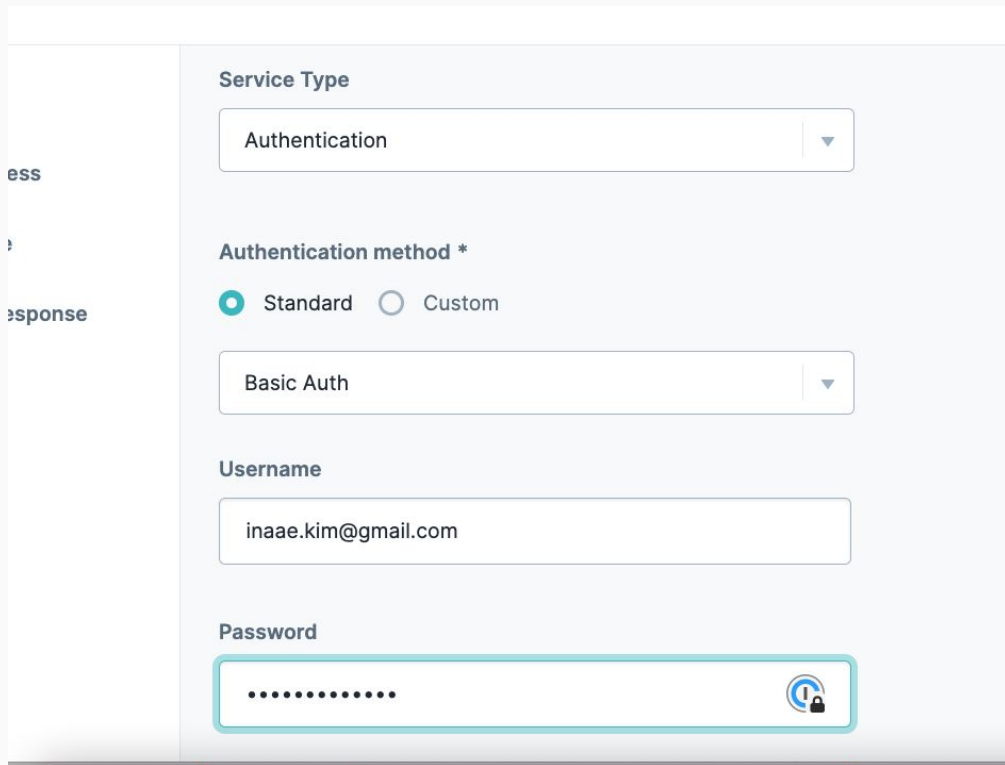
Environment



Workspaces

EXAMPLE: SECURITY REVIEW 1

Create the service user with **personal account**.



The image shows a screenshot of a web form for creating a service user. The form is titled "Service Type" and includes the following fields and options:

- Service Type:** A dropdown menu with "Authentication" selected.
- Authentication method *:** Two radio buttons: "Standard" (selected) and "Custom".
- Basic Auth:** A dropdown menu with "Basic Auth" selected.
- Username:** A text input field containing "inaae.kim@gmail.com".
- Password:** A password input field with a blue border and a lock icon on the right.

On the left side of the form, there is a vertical sidebar with the following text: "ess", ":", and "esponse".

EXAMPLE: SECURITY REVIEW 1

Add the service and **URL** to the application

Service Type

Internal External

External Services ?

AppSecDemo ▼

Data Source URL ?

https://appsecdemo.io/api/v2/query

Request Type ?

Get Patch Post Put Delete

INPUTS ?

VULNERABILITIES

- Account Impersonation - Personal Account
- Privilege Escalation - Personal Account, Disable SSE only config, No filter query URL
- Authorization Misuse - Global Access to the service
- Data Leakage - Third party integration

REMEDIATION STEP 1

Service Info

Service title *

AppSecDemo

Service protocol + host *

https://appsecdemo.io/api/v2/

Service name *

This is permanent and cannot be reversed

AppSecDemo

☒ Allow service execution server side only

Prevent the app users
from accessing APIs

Manage Access

Share to ⓘ

☐ Environment ☒ Workspaces

Select workspaces to share this service with * ⓘ

Removing a workspace will no longer give it access to this service and may break the application.

AppSec x

Grant access to
resources that needed
to perform the job

REMIEDIATION STEP 2

Data

Service Type

☐ Internal ☒ External

External Services ?

AppSecDemo

Data Source URL ?

https://appsecdemo.io/api/v2/query?userid={data.userid}

Request Type ?

☒ Get ☐ Patch ☐ Post ☐ Put ☐ Delete

filter userid

SECURITY CONTROLS

- Add only approved services to the platform.
- Enforce the least privileges principle.
- Monitor data traffic both the platform and the third party service.
- Train developers on the risks of insecure third-party service.

EXAMPLE: SECURITY REVIEW 2

Add **secret data**.

Property ID: plugGetSessionToken Field Tags: + add a tag [Settings](#)

Make a multipart API call ☐

Promote object/array to top level ☐

INPUTS

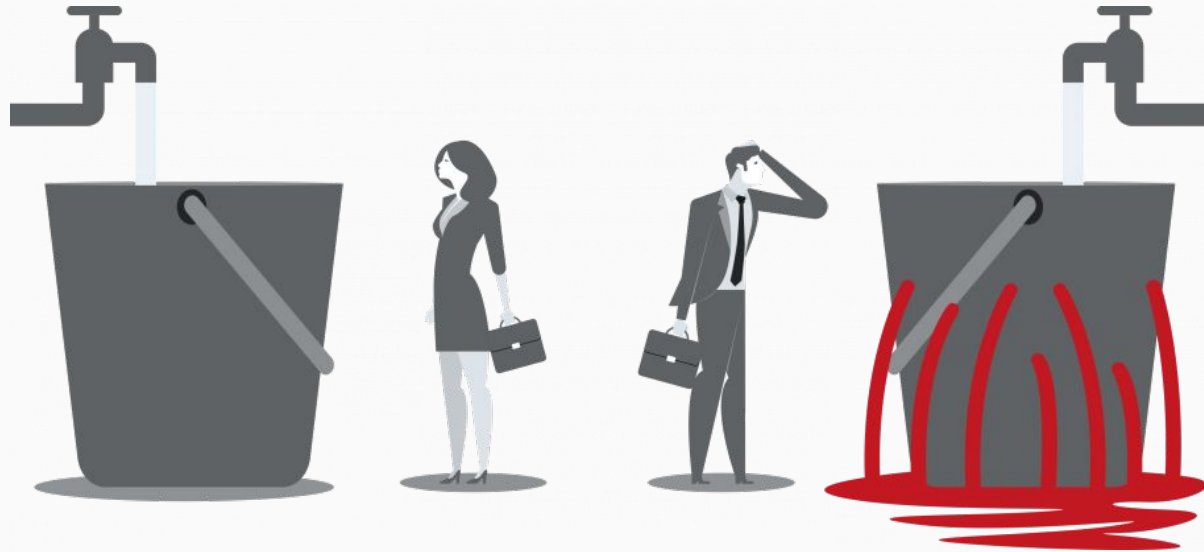
Add secrets in insecure place

Property ID	Mapping	Required	Exclude	Header	Resolve
'fdf36	de68... client_id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'236	f24... client_secret	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'session'	type	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
userId	client_user_id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'christopher.jones@ungar.com'	client_user_email	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OUTPUTS

VULNERABILITIES

Data Leakage - Security Misconfiguration



REMEDIATION STEP

Move secrets to secure location

Property ID: plugGetSessionToken Field Tags: + add a tag [Settings](#)

Make a multipart API call ☐

Promote object/array to top level ☐

INPUTS ? **Add secrets in insecure place**

Property ID	Mapping	Required	Exclude	Header	Resolve
'fdf36	de68... ▼ client_id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'234	124... ▼ client_secret	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'session'	▼ type	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
userId	▼ client_user_id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
'christopher.igusa@usnook.com'	▼ client_user_email	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OUTPUTS ?



Service Type

Service Type

Authentication

Authentication method *

☒ Standard ☐ Custom

OAuth2 Client Credentials Grant

Access Token URL

test

Client ID

fdf36ac

Client Secret

.....

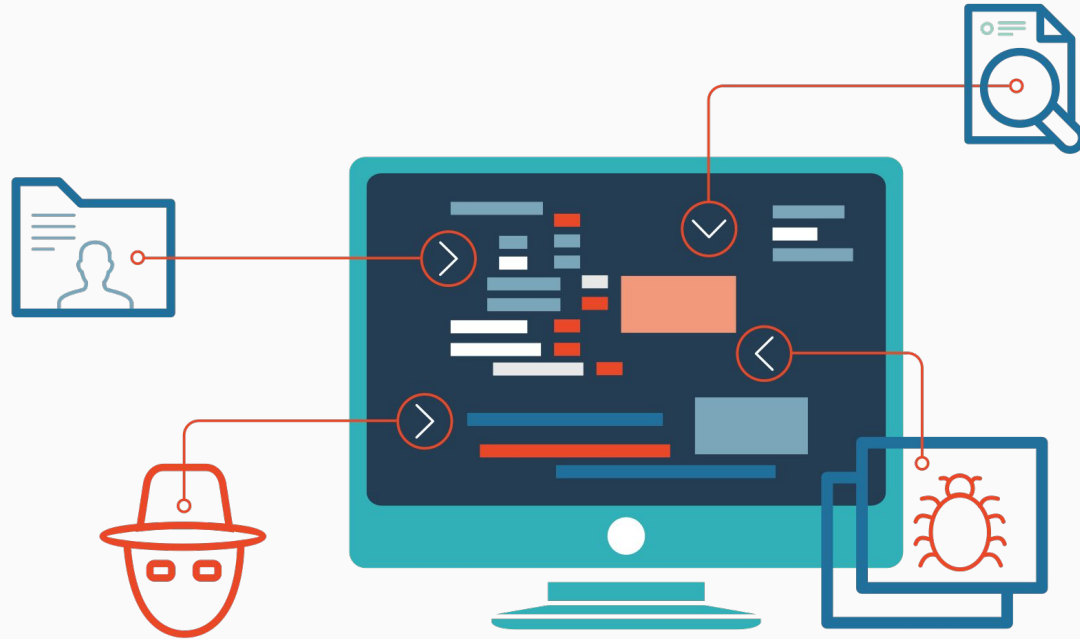
Scope

SECURITY CONTROLS

- Review the application configuration
- Establish Secure by default
- Follow the industry security best practices

DEMO - SECURITY REVIEW

Static Application Security Testing (SAST) - Configuration Review



APPENDIX

1. Talk presentation and resources

<https://github.com/inaaekim/LCNCSecurity>

2. Useful resources

<https://www.codelessarchitecture.org/>

https://cdn2.assets-servd.host/quaint-admiral/production/files/no_code_low_code_platforms_bridge_legacy_digital.pdf

<https://owasp.org/www-project-top-10-low-code-no-code-security-risks/>



END