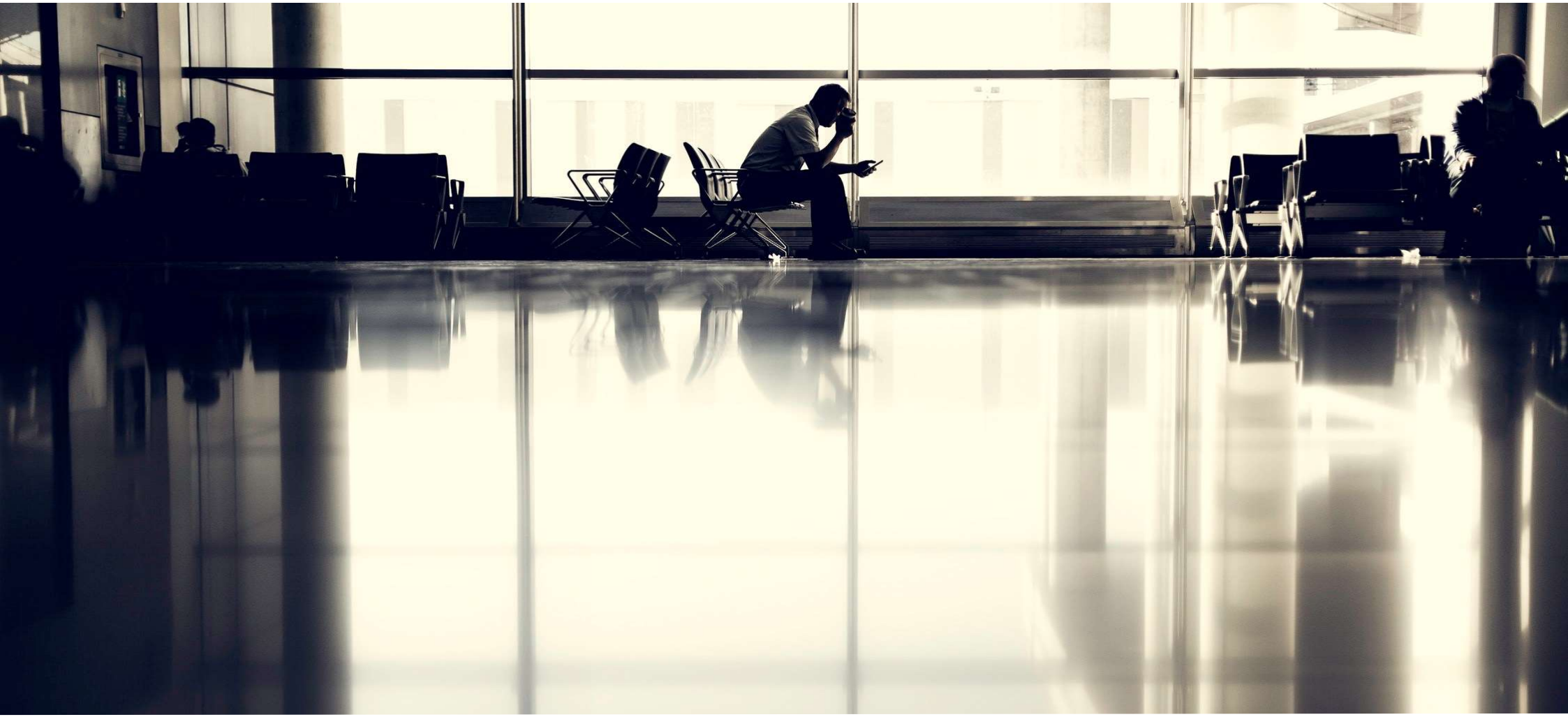


Low-code Application and Security

Celine Blandin – Security Consultant

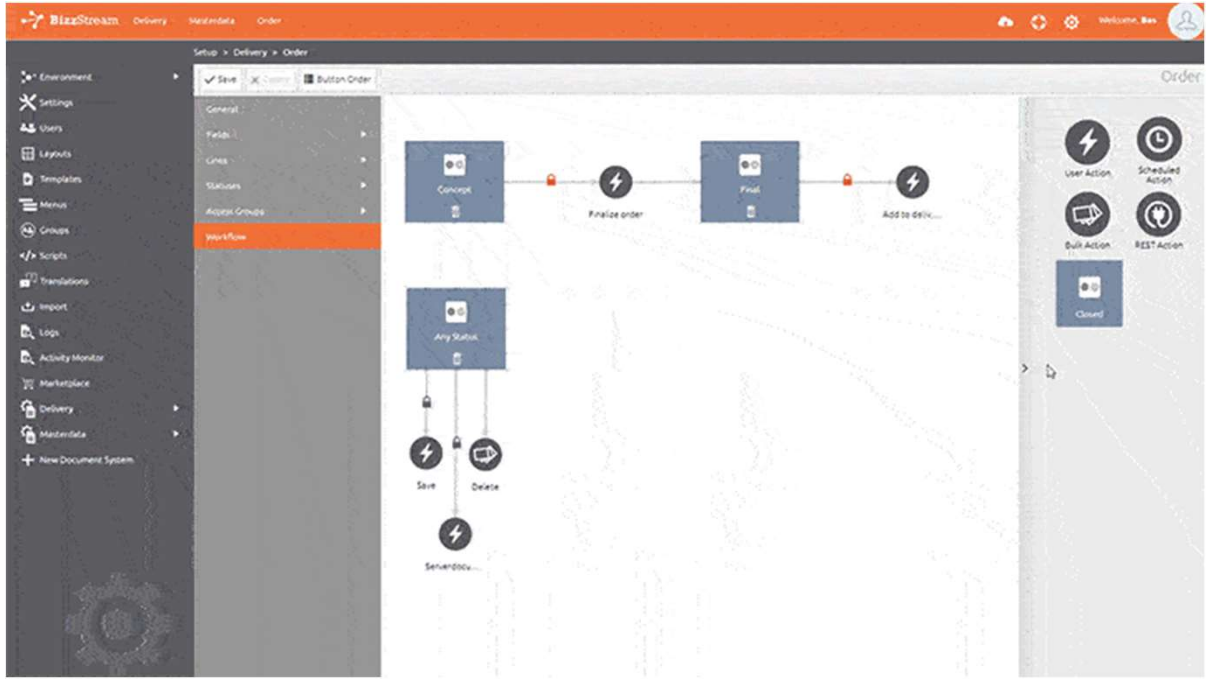
September 2023

Agenda

- What is Low-code Application
- Security Audit of Low-code application : How?
- Specificity of low-code application pentest
    - OWASP Top10 Low-Code/No-code Applications
    - Enumeration Risks
    - Data Leak Risks
    - Business Logic Analysis

What is Low-code Application

A low-code development platform (LCDP) provides a development environment used to create application software through a graphical user interface.



- Visual environment, easy to understand and use
- Faster development
- Low maintenance cost

What is Low-code Application

Forecasts provided by Gartner predict that low-code platforms will account for **65%** of the IT and software development market before **2024**.

Gartner predicts that by **2026**, developers outside formal IT departments will account for at least **80% of the user base** for low-code development tools, up from 60% in 2021.

Important companies already started to use Low-Code platform:

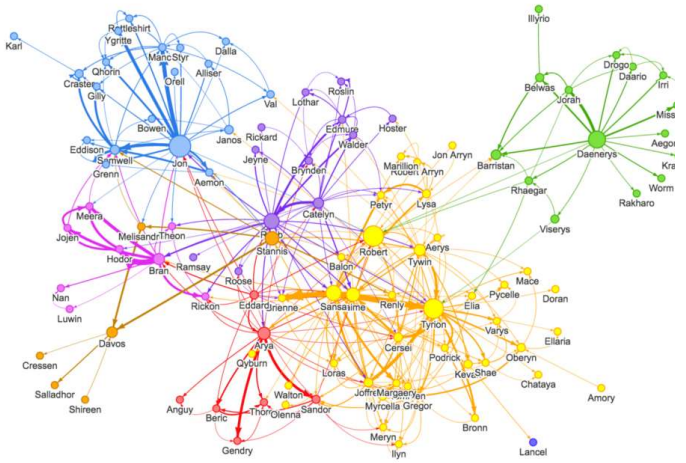
- Santander --> Outsystems
- Bendigo Bank --> Appian
- Washington Federal Bank --> Mendix
- Al Baraka Bank --> Mendix
- BNP Paribas --> Creatio

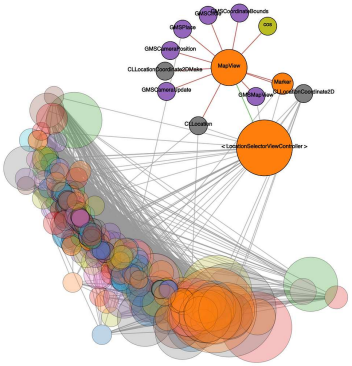
Security Audit of Low-code application : How?

The larger the application, the more visibility you lose .

Low-code platform is equivalent to :

- ➔ No access or difficulty to access the source code – code source size increase
- ➔ Lost track of what exactly is build:
  - inventory mapping,
  - shadow IT risk,
  - user privileges track
  - ...





➔ Important to cover all the ends points

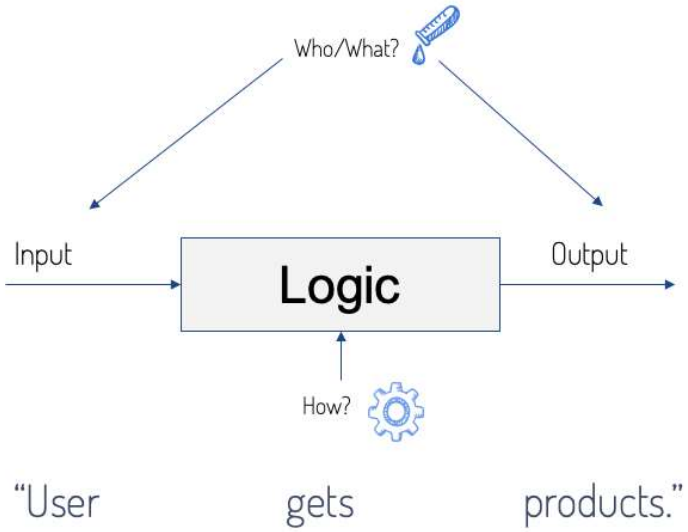
➔ **WHITE BOX Pentest**

Security Audit of Low-code application : How?

Low-Code Application have already automatic static scan (depends of the provider maturity)

- Low probability of injection vulnerability
- Most of vulnerabilities: **Human error, cannot be found by automated audit**

MANUAL pentest



Security Audit of Low-code application : How?

Low-code platform:

- Lack of Control Over the Underlying Code and Infrastructure
- Too Much Dependency on Platform Providers for Support and Update

Check Low-Code Platform **PROVIDER reliability**

- Provider vulnerability, cve?
- Reliability source code scanner used (SonarQube, PV5-Studio…)
- Other static scan?

Specificity of Low-Code application pentest

OWASP has identified the risk of the low-code platforms expansion. Accordingly, they wrote the Top 10, specific to the Low-Code / No-Code applications

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

OWASP Low-Code/No-Code Top 10

Main | Join | Contributors

Stars 53 | slack nocode | group nocode

Overview

Low-Code/No-Code development platforms provide a development environment used to create application software through a graphical user interface instead of traditional hand-coded computer programming. Such platforms reduce the amount of traditional hand-coding, enabling accelerated delivery of business applications.

As Low-Code/No-Code platforms proliferate and become widely used by organizations, there is a clear and immediate need to create awareness around security and privacy risks related to applications developed on such platforms.

The primary goal of the "OWASP Low-Code/No-Code Top 10" document is to provide assistance and education for organizations looking to adopt and develop Low-Code/No-Code applications. The guide provides information about what the most prominent security risks are for such applications, the challenges involved, and how to overcome them.

The List

1. LCNC-SEC-01: Account Impersonation
2. LCNC-SEC-02: Authorization Misuse
3. LCNC-SEC-03: Data Leakage and Unexpected Consequences
4. LCNC-SEC-04: Authentication and Secure Communication Failures
5. LCNC-SEC-05: Security Misconfiguration
6. LCNC-SEC-06: Injection Handling Failures
7. LCNC-SEC-07: Vulnerable and Untrusted Components
8. LCNC-SEC-08: Data and Secret Handling Failures
9. LCNC-SEC-09: Asset Management Failures
10. LCNC-SEC-10: Security Logging and Monitoring Failures

OWASP Top10 Low-Code/No-Code Applications

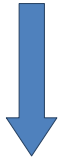
OWASP TOP 10 Web Applications 2021	OWASP TOP 10 Low-Code/No-Code
A01:2021-Broken Access Control	LCNC-SEC-01: Account Impersonation
A02:2021-Cryptographic Failures	LCNC-SEC-02: Authorization Misuse
A03:2021-Injection	LCNC-SEC-03: Data Leakage and Unexpected Consequences
A04:2021-Insecure Design	LCNC-SEC-04: Authentication and Secure Communication Failures
A05:2021-Security Misconfiguration	LCNC-SEC-05: Security Misconfiguration
A06:2021-Vulnerable and Outdated Components	LCNC-SEC-06: Injection Handling Failures
A07:2021-Identification and Authentication Failures	LCNC-SEC-07: Vulnerable and Untrusted Components
A08:2021-Software and Data Integrity Failures	LCNC-SEC-08: Data and Secret Handling Failures
A09:2021-Security Logging and Monitoring Failures	LCNC-SEC-09: Asset Management Failures
A10:2021-Server-Side Request Forgery	LCNC-SEC-10: Security Logging and Monitoring Failures

Enumerations Risks

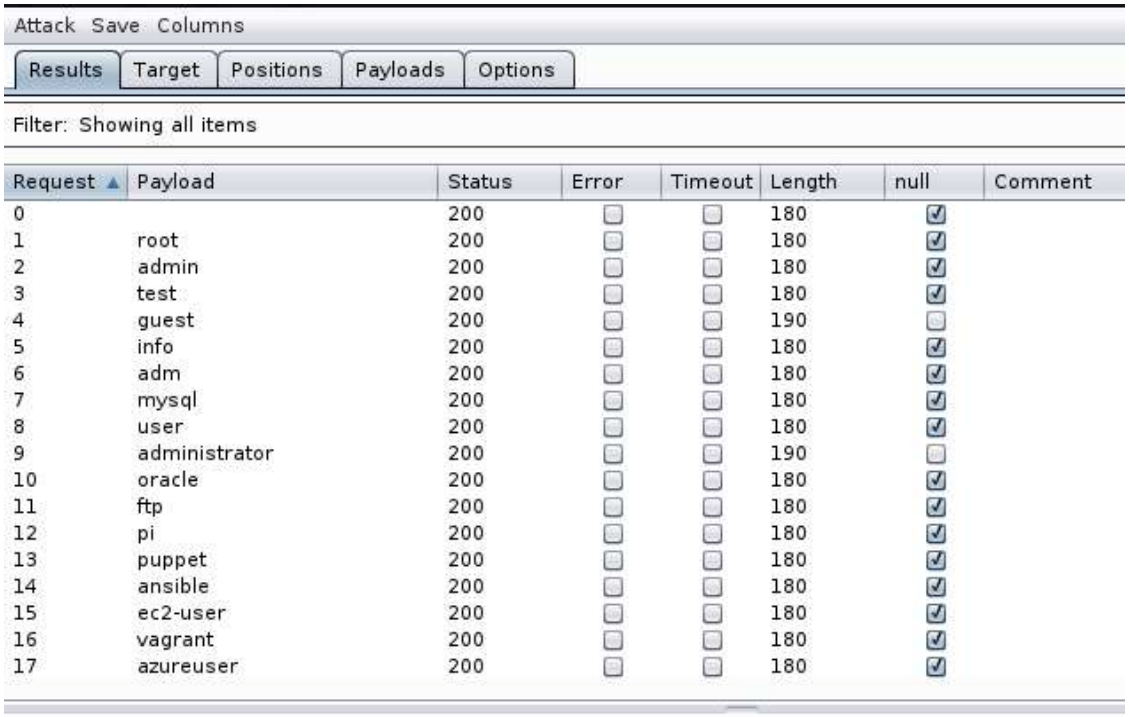
Different vulnerability focus:

- **Enumeration**

Audit needs to check every input for enumeration



Try Extract database



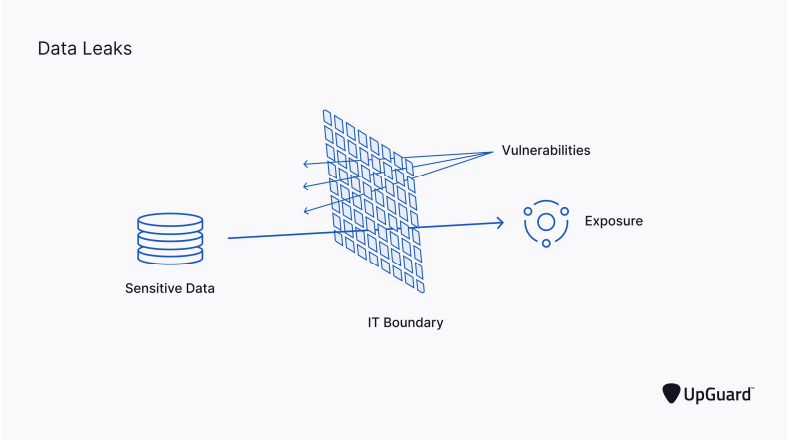
Request	Payload	Status	Error	Timeout	Length	null	Comment
0		200	☐	☐	180	☑	
1	root	200	☐	☐	180	☑	
2	admin	200	☐	☐	180	☑	
3	test	200	☐	☐	180	☑	
4	guest	200	☐	☐	190	☐	
5	info	200	☐	☐	180	☑	
6	adm	200	☐	☐	180	☑	
7	mysql	200	☐	☐	180	☑	
8	user	200	☐	☐	180	☑	
9	administrator	200	☐	☐	190	☐	
10	oracle	200	☐	☐	180	☑	
11	ftp	200	☐	☐	180	☑	
12	pi	200	☐	☐	180	☑	
13	puppet	200	☐	☐	180	☑	
14	ansible	200	☐	☐	180	☑	
15	ec2-user	200	☐	☐	180	☑	
16	vagrant	200	☐	☐	180	☑	
17	azureuser	200	☐	☐	180	☑	

Data Leak Risks

Different vulnerability focus:

- **Data Leak**
  - Full mapping of the application
  - Check sensible data published by mistake
  - Check error messages

WhiteBox pentest → Test the entire application footprint



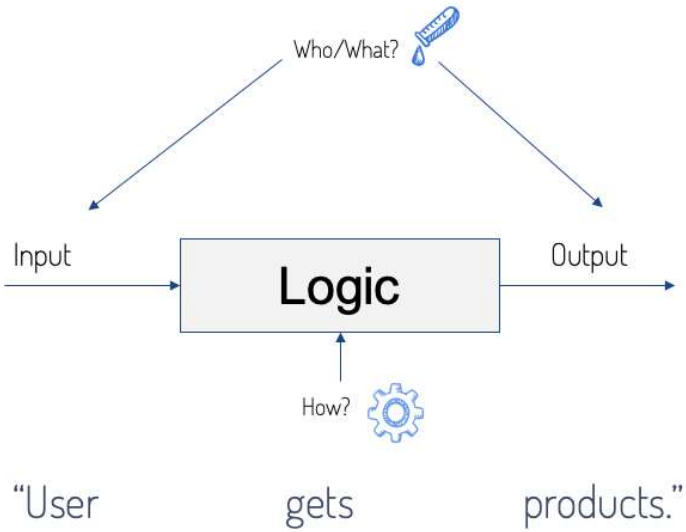
Business Logic Analysis

Different vulnerability focus:

- **Business Logic**: play with the application, and try to force unwanted behavior

Check permission to end user:

User by group category ——— Access to data other user



→ Cannot be found by automated audits

→ Manual pentest

Conclusion

Low-code/no-code applications → future **NORM**

different security audit focus

→

- Security Audit as **White Box**?
- Security Audit as **Manual Pentest**?
- Security Audit check **OWASP Low-Code/No-Code Top 10**?
- Security Audit check for **Enumeration** risk?
- Security Audit check **Data Leak** risk?
- Security Audit follows **Business Logic** test?

References

https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023

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

https://www.outsystems.com/case-studies/santander-consumer-digital-transformation/

https://appian.com/about/explore/customers/all-customers/bendigo-bank.html

https://www.mendix.com/customer-stories/delivering-great-online-customer-experiences-in-retail-banking/

https://www.classicinformatics.com/low-code-development-guide#challengessolvedusecases