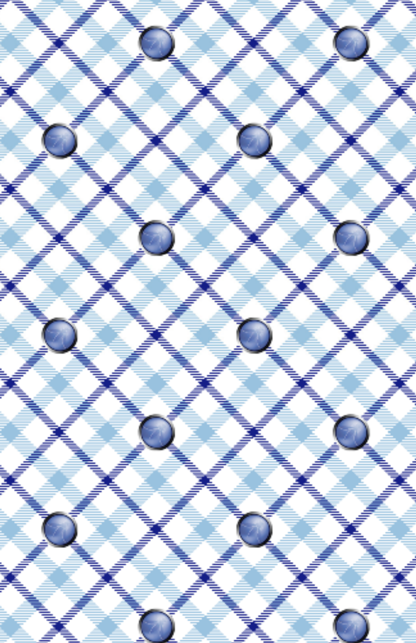
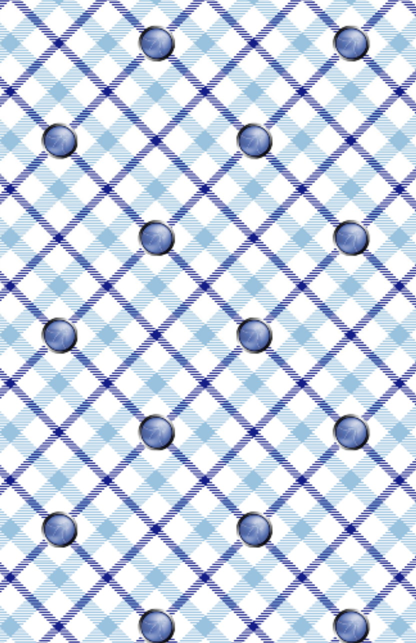


AUTHENTICATION

A

You have invented a new attack against Authentication

Read more about this topic in OWASP's free Authentication Cheat Sheet

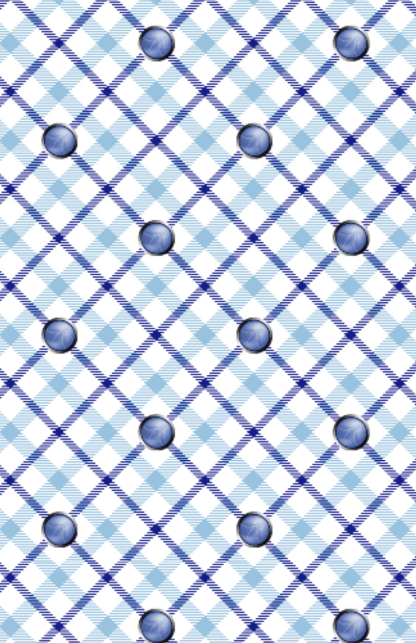


AUTHENTICATION

2

James can undertake authentication functions (e.g. attempt to log in, log in with stolen credentials, reset the password) without the real user ever being aware this has occurred

OWASP SCP	47, 52
OWASP ASVS	2.12
OWASP AppSensor	UT1
CAPEC	-
SAFECode	28



AUTHENTICATION

3

Muhammad can obtain a user's password or other secrets such as security questions, by observation during entry, or from a local cache, or in transit, or by reading it from some unprotected location, or because it is widely known, or because it never expires, or because the user cannot change her own password

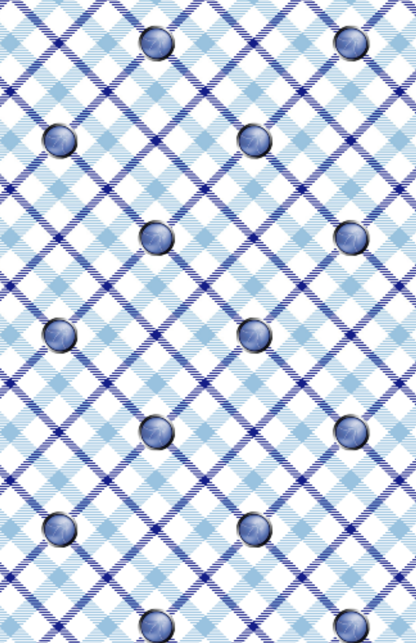
OWASP SCP
36-7, 40, 43, 48, 51, 119, 139-40, 146

OWASP ASVS
2.2, 2.8, 2.10, 8.10, 9.1

OWASP AppSensor
-

CAPEC
37

SAFECODE
28



AUTHENTICATION

4

Sebastien can easily identify user names or can enumerate them

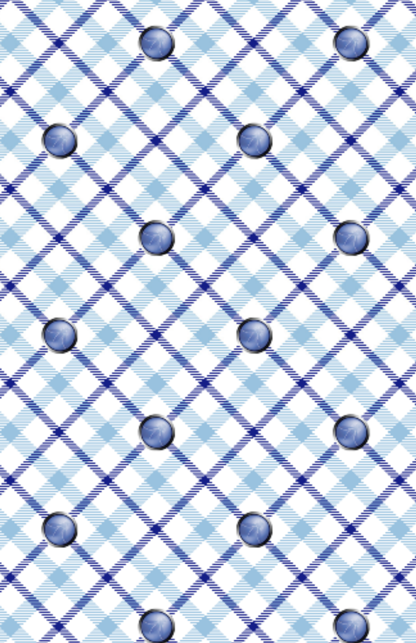
OWASP SCP
33, 53

OWASP ASVS
-

OWASP AppSensor
AE1

CAPEC
383

SAFECODE
28

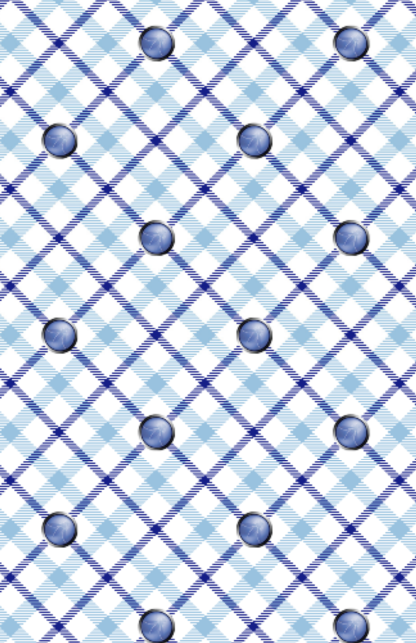


AUTHENTICATION

5

Javier can use default, test or easily guessable credentials to authenticate, or can use an old account or an account not necessary for the application

OWASP SCP	54, 175, 178
OWASP ASVS	-
OWASP AppSensor	AE12, HT3
CAPEC	70
SAFECODE	28



AUTHENTICATION

6

Sven can reuse a temporary password because the user does not have to change it on first use, or it has too long or no expiry

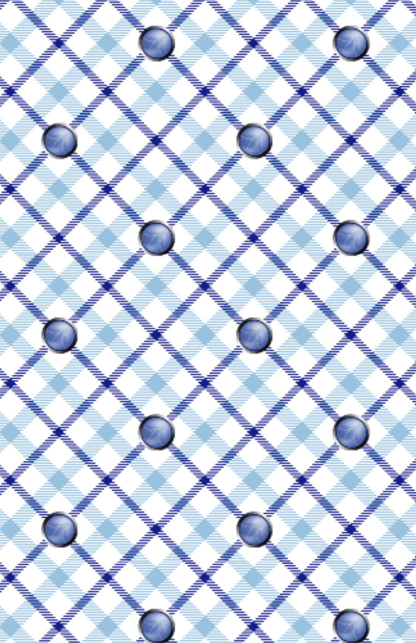
OWASP SCP
37, 45, 46, 178

OWASP ASVS
-

OWASP AppSensor
-

CAPEC
50

SAFECODE
28



AUTHENTICATION

7

Cecilia can use brute force and dictionary attacks against one or many accounts without limit, or these attacks are simplified due to insufficient complexity, length, expiration and re-use requirements for passwords

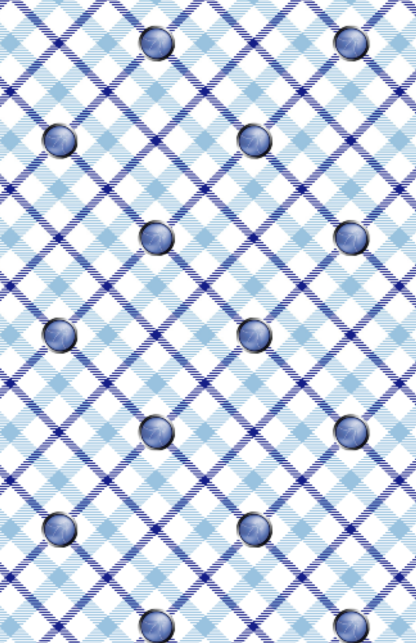
OWASP SCP
33, 38, 39, 41, 50, 53

OWASP ASVS
2.3

OWASP AppSensor
AE2, AE3

CAPEC
2, 16

SAFECODE
27

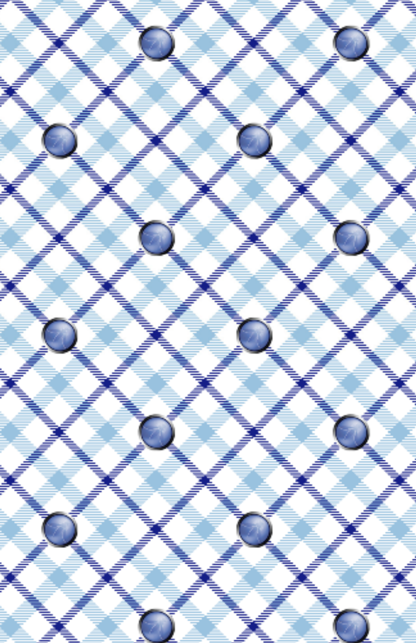


AUTHENTICATION

8

Kate can by bypass authentication because it does not fail secure (i.e. it defaults to allowing access)

OWASP SCP	28
OWASP ASVS	2.5
OWASP AppSensor	-
CAPEC	115
SAFECode	28

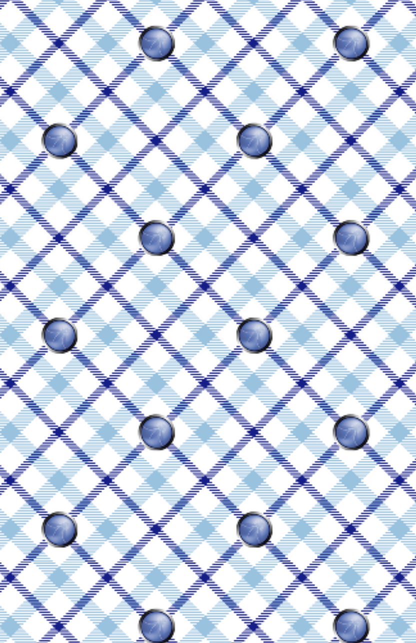


AUTHENTICATION

9

Claudia can undertake more critical functions because authentication requirements are too weak, or there is no requirement to re-authenticate for these

OWASP SCP	55, 56
OWASP ASVS	2.6, 2.9
OWASP AppSensor	-
CAPEC	21
SAFECODE	14, 28



AUTHENTICATION

10

Pravin can bypass authentication controls because a centralized standard, tested and approved authentication module/framework/service, separate to the resource being requested, is not being used

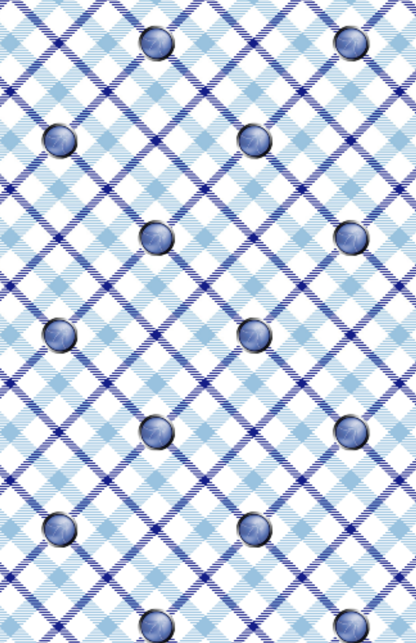
OWASP SCP
25, 26,27

OWASP ASVS
2.11

OWASP AppSensor
-

CAPEC
90, 115

SAFECODE
14, 28



AUTHENTICATION

J

Mark can access resources or services because there is no authentication requirement, or it was assumed authentication would be undertaken by some other system, or was performed in some previous action

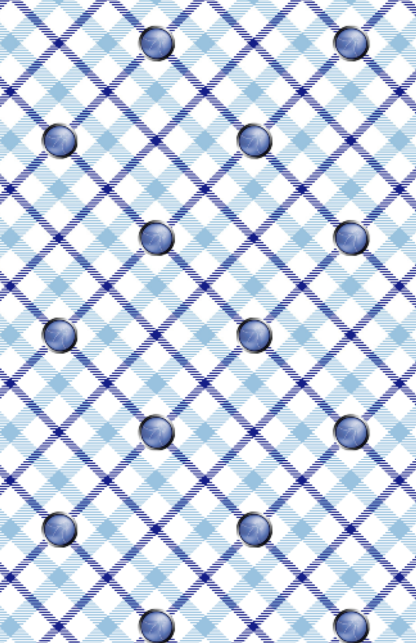
OWASP SCP
23, 32, 34

OWASP ASVS
2.1

OWASP AppSensor
-

CAPEC
115

SAFECODE
14, 28



AUTHENTICATION

Q

Jaime can bypass authentication because it is not enforced comprehensively across all entry points, modules, functions, content and other data, or is not applied with equal rigor for all types of authentication functionality (e.g. register, password change, password change, log out, administration)

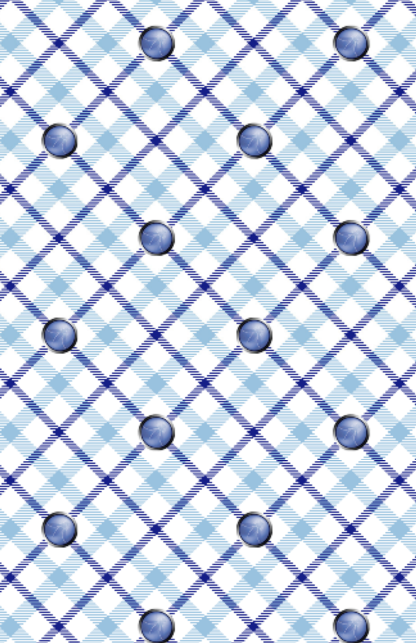
OWASP SCP
23, 29, 42, 49

OWASP ASVS
2.1, 2.7

OWASP AppSensor
-

CAPEC
36, 50, 115, 121, 179

SAFECODE
14, 28



AUTHENTICATION

K

Olga can influence or alter authentication code/routines so they can be bypassed

OWASP SCP
24

OWASP ASVS
2.4

OWASP AppSensor
-

CAPEC
115, 207

SAFECODE
14, 28