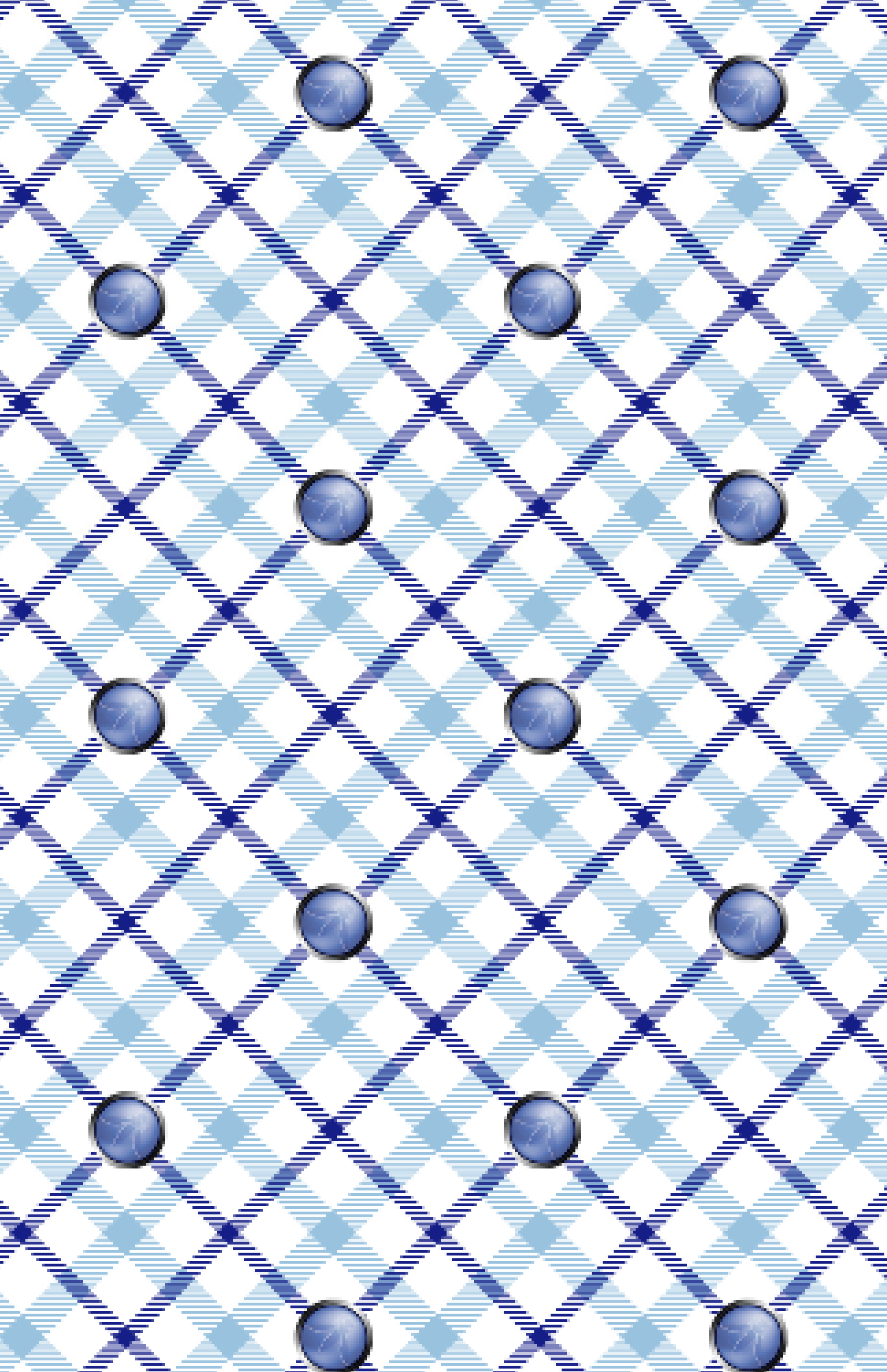
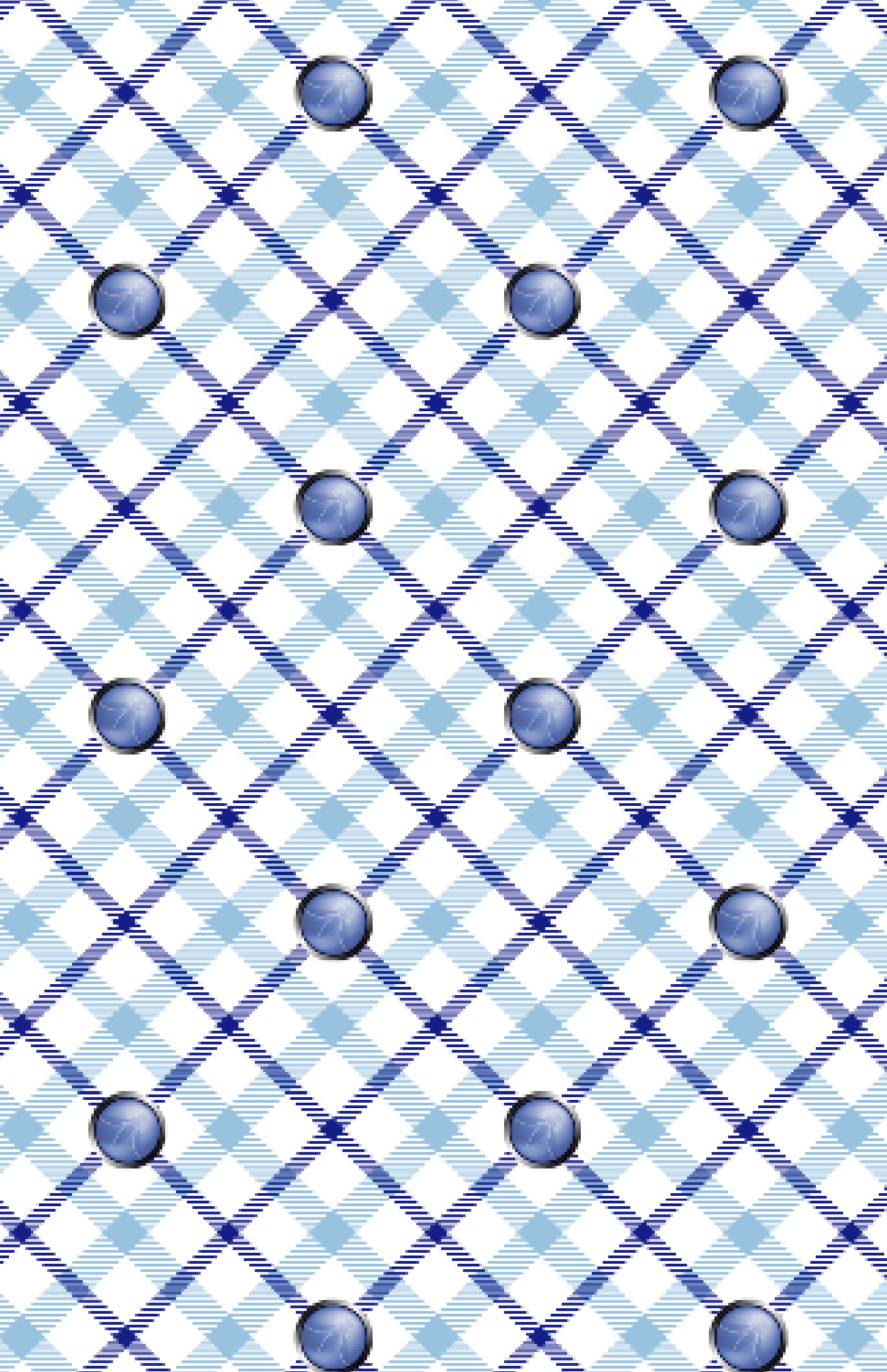


A

You have invented a new attack against Session Management

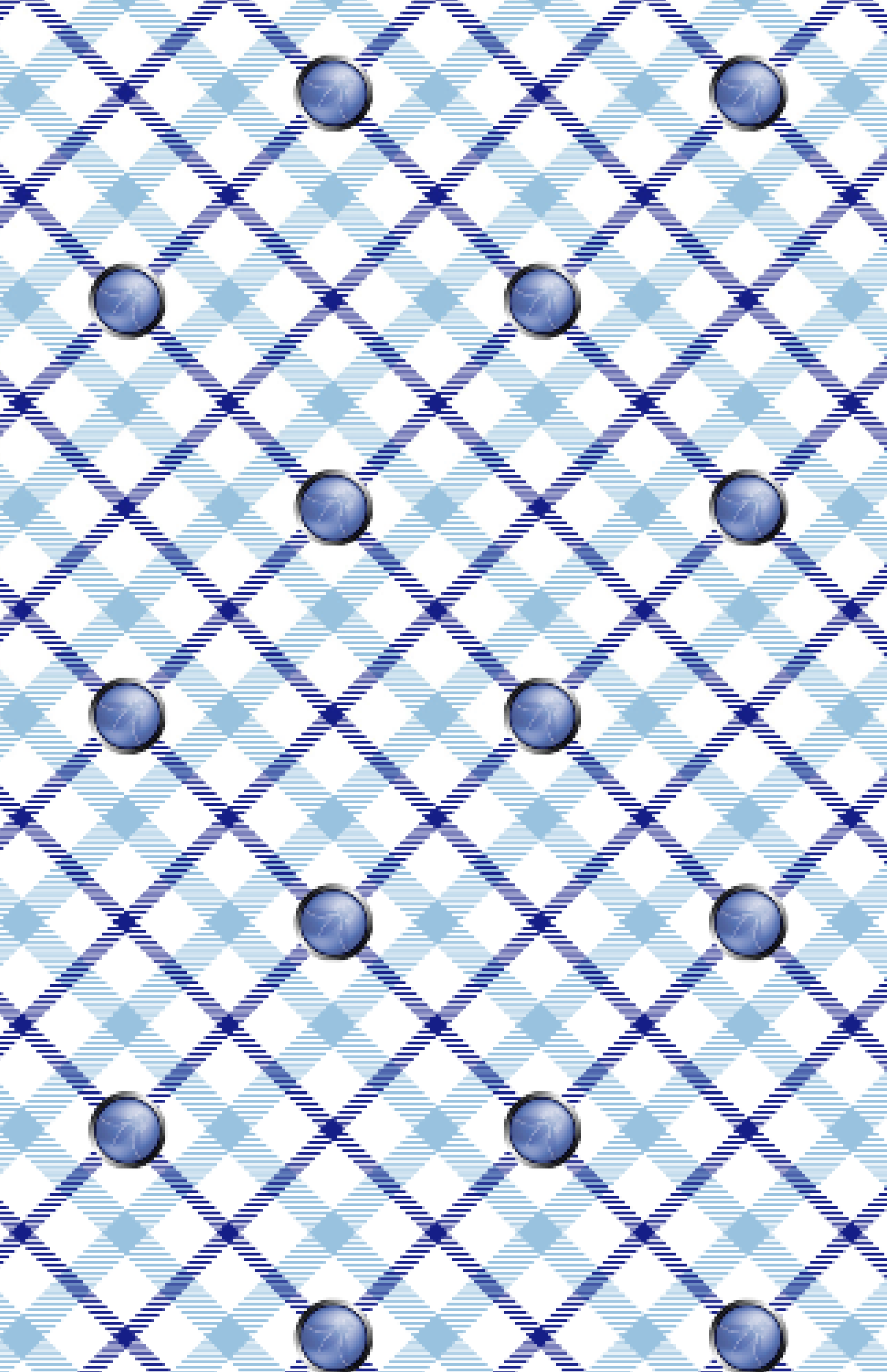
Read more about this topic in OWASP's free Cheat Sheets on Session Management, and Cross Site Request Forgery (CSRF) Prevention



2

William has control over the generation of session identifiers

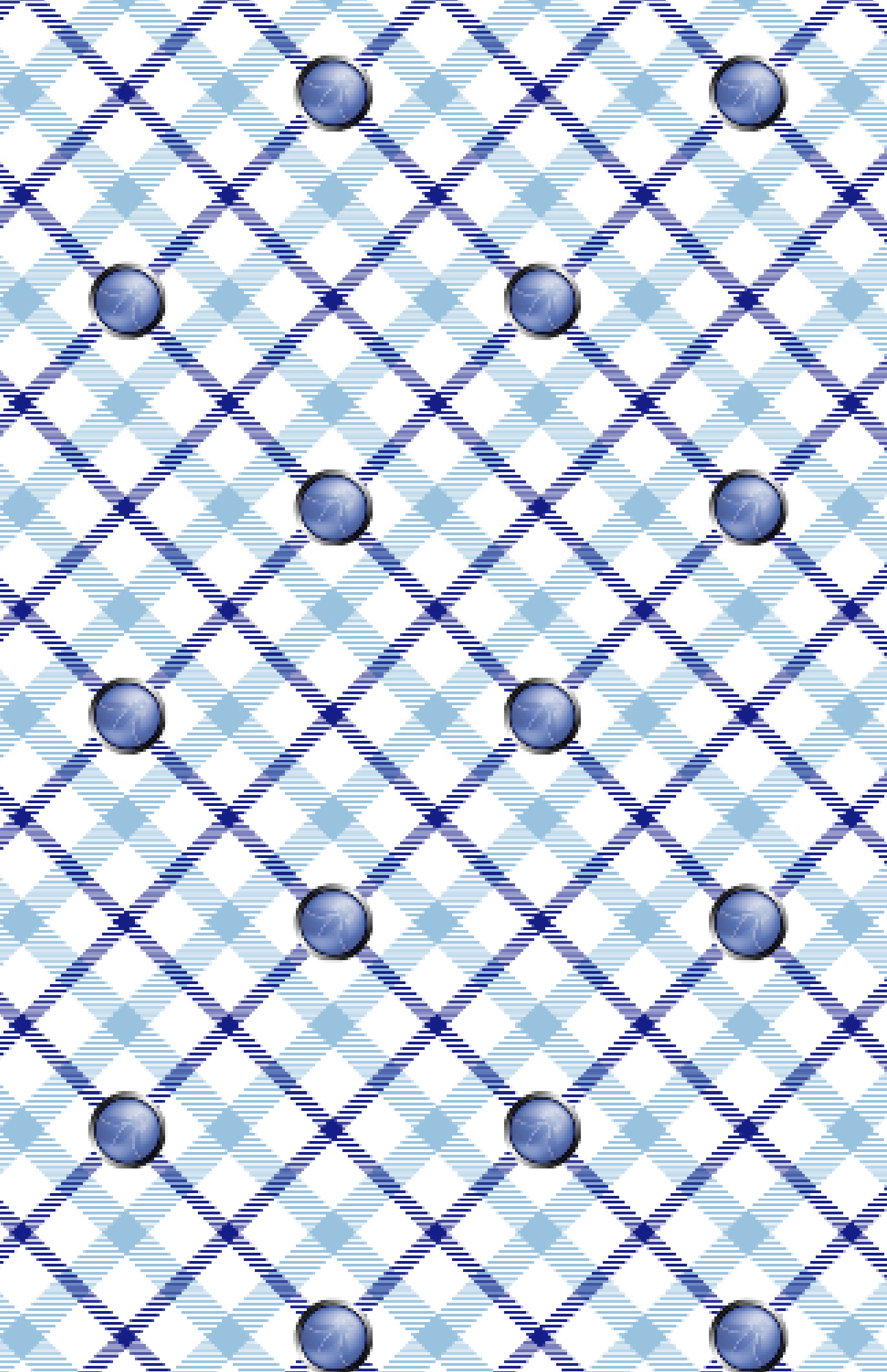
OWASP SCP	59
OWASP ASVS	3.9
OWASP AppSensor	SE2
CAPEC	31, 60, 61
SAFECODE	28



3

Ryan can use a single account in parallel since concurrent sessions are allowed

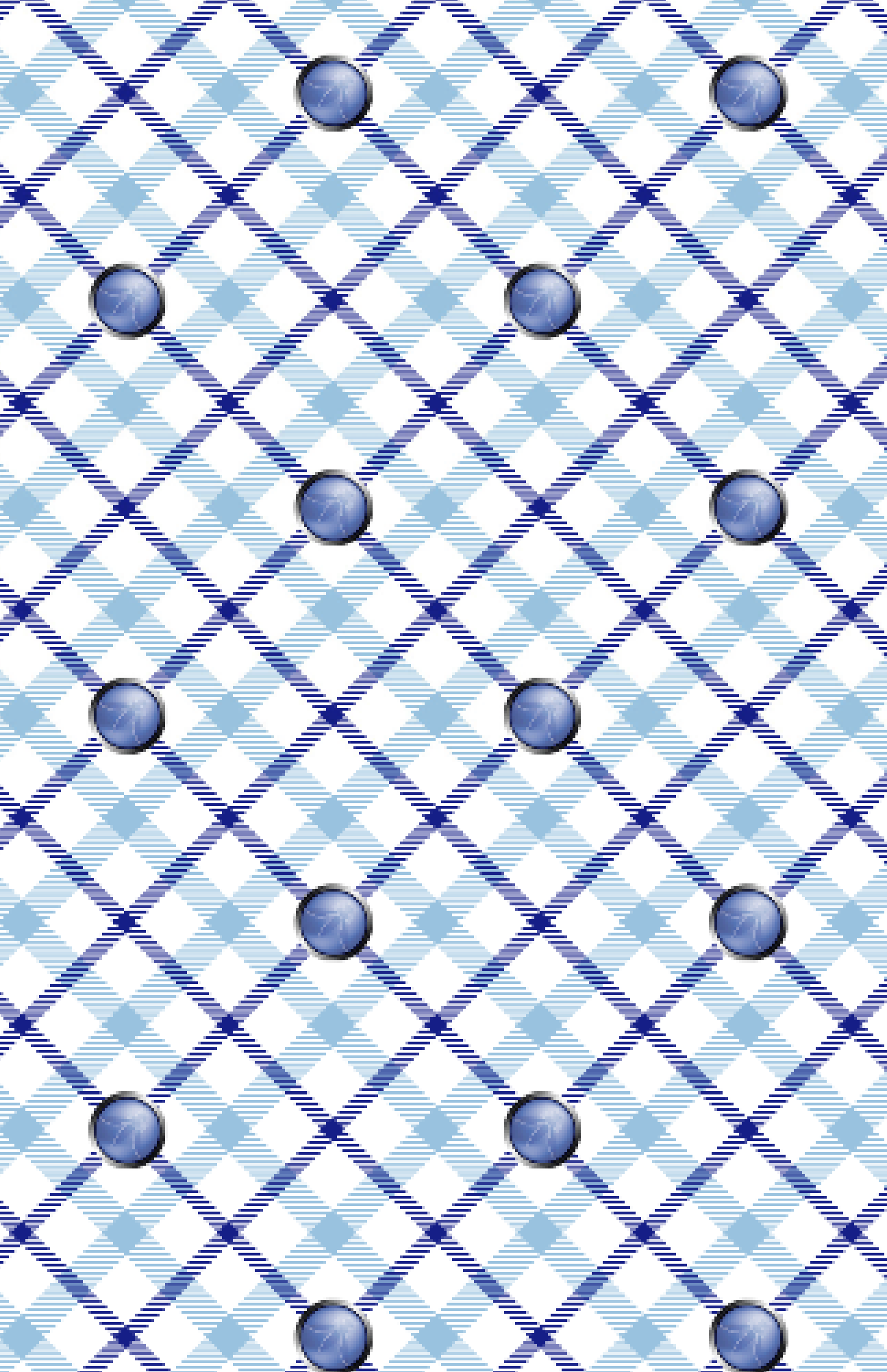
OWASP SCP 68
OWASP ASVS -
OWASP AppSensor -
CAPEC -
SAFECODE 28



4

Alison can set session identification cookies on another web application because the domain and path are not restricted sufficiently

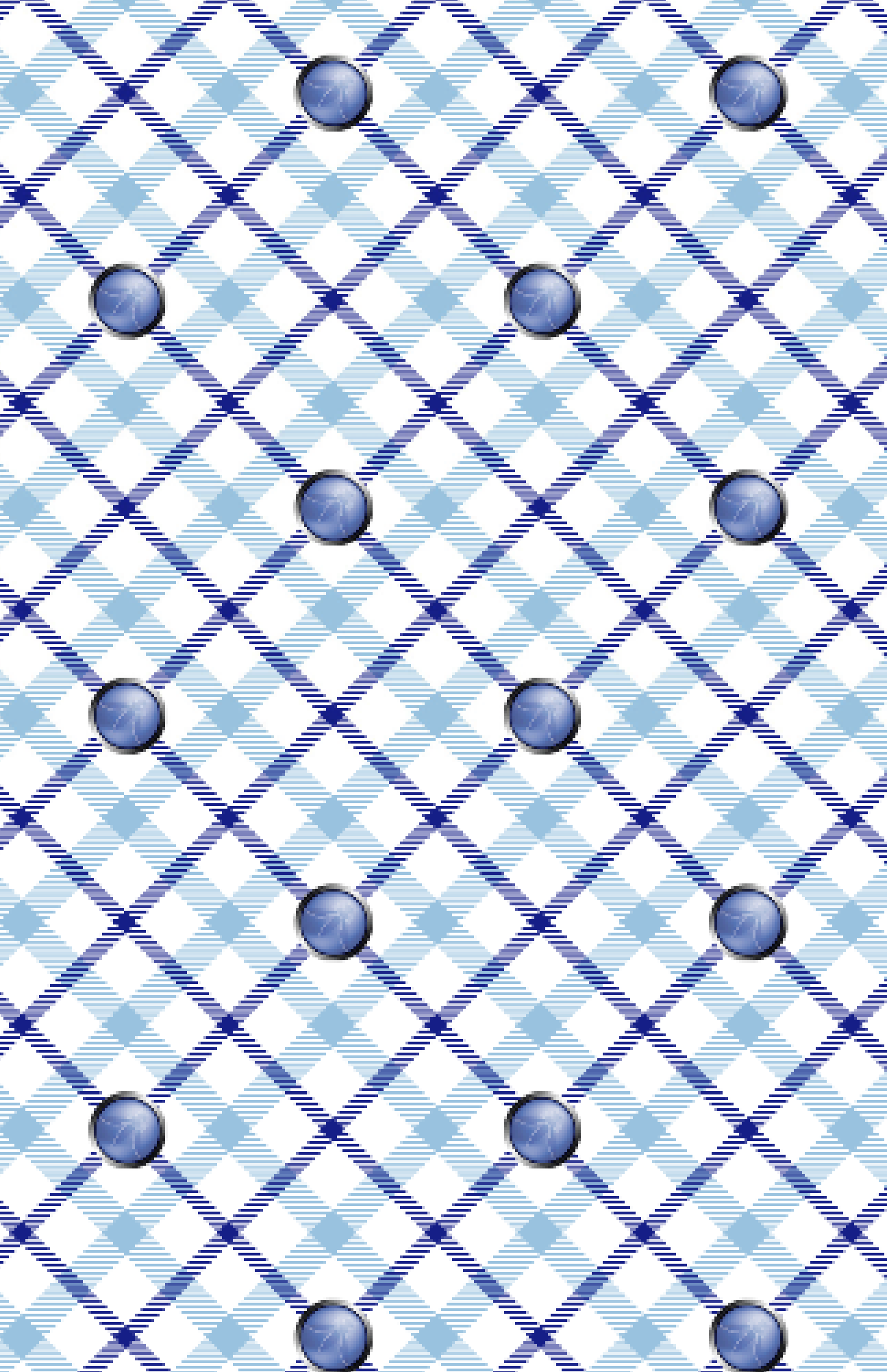
OWASP SCP 59, 61
OWASP ASVS 3.12
OWASP AppSensor SE2
CAPEC 31, 61
SAFECODE 28



5

John can predict or guess session identifiers because they are not changed when the user's role alters (e.g. pre and post authentication) and when switching between non-encrypted and encrypted communications, or are not sufficiently long and random, or are not changed periodically

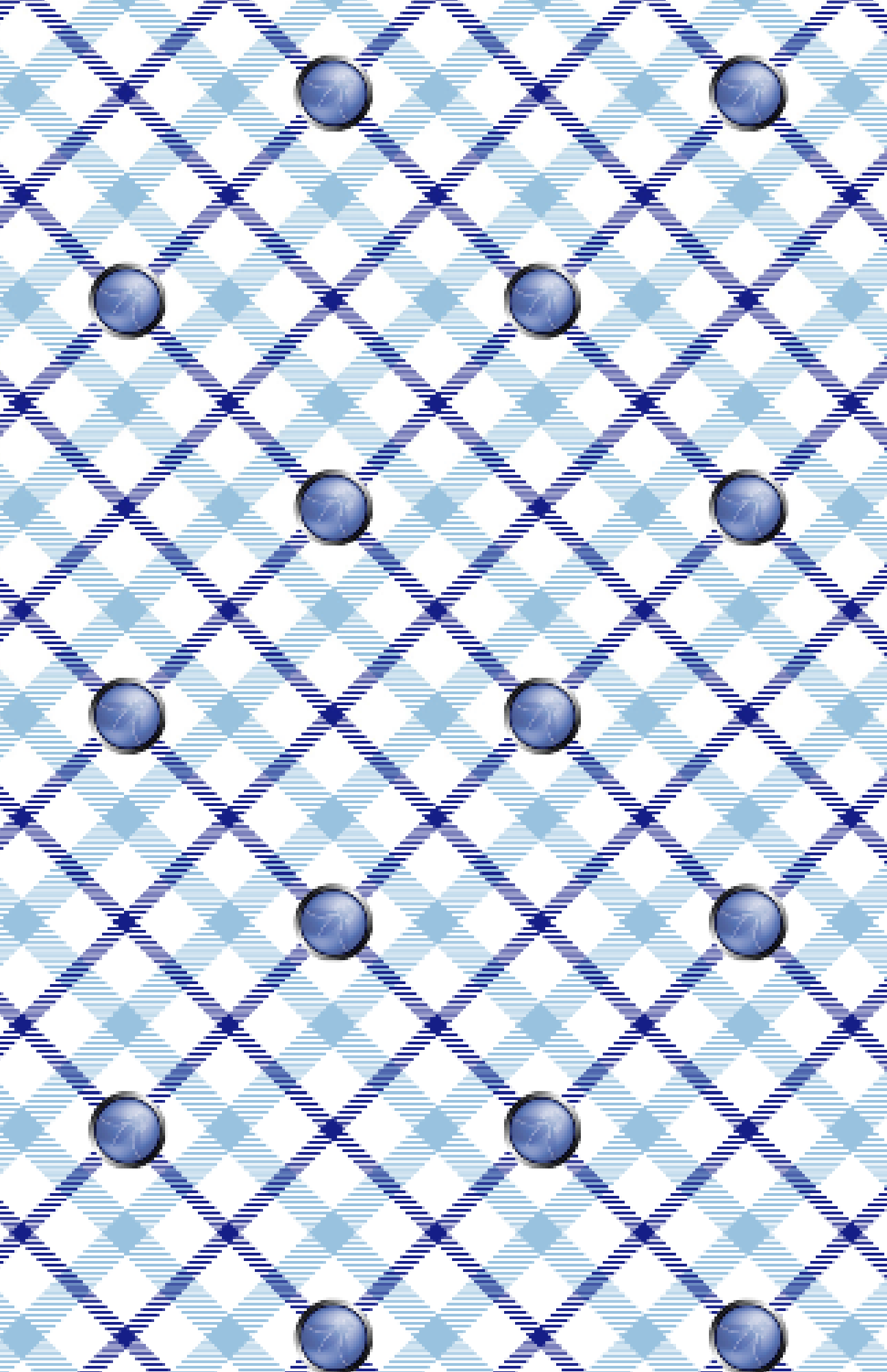
OWASP SCP 66, 67, 71, 72
OWASP ASVS 3.6, 3.7, 3.8, 3.11
OWASP AppSensor SE4-6
CAPEC 31
SAFECODE 28



6

Gary can take over a user's session because there is a long or no inactivity timeout, or a long or no overall session time limit, or the same session can be used from more than one device/location

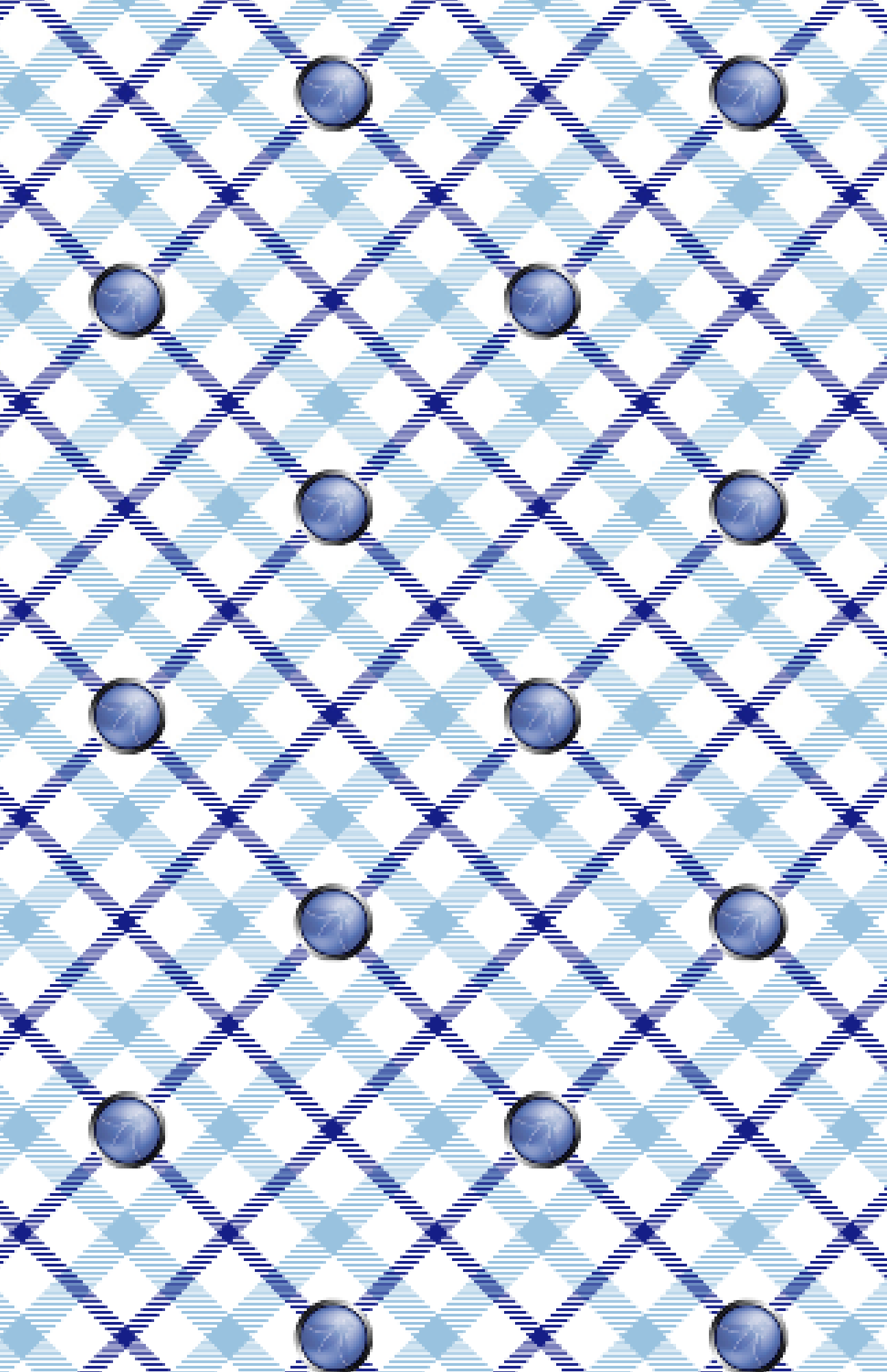
OWASP SCP 64, 65
OWASP ASVS 3.3, 3.10
OWASP AppSensor SE5, SE6
CAPEC 21
SAFECODE 28



7

Casey can utilize Adam's session after he has finished, because there is no log out function, or he cannot easily log out, or log out does not properly terminate the session

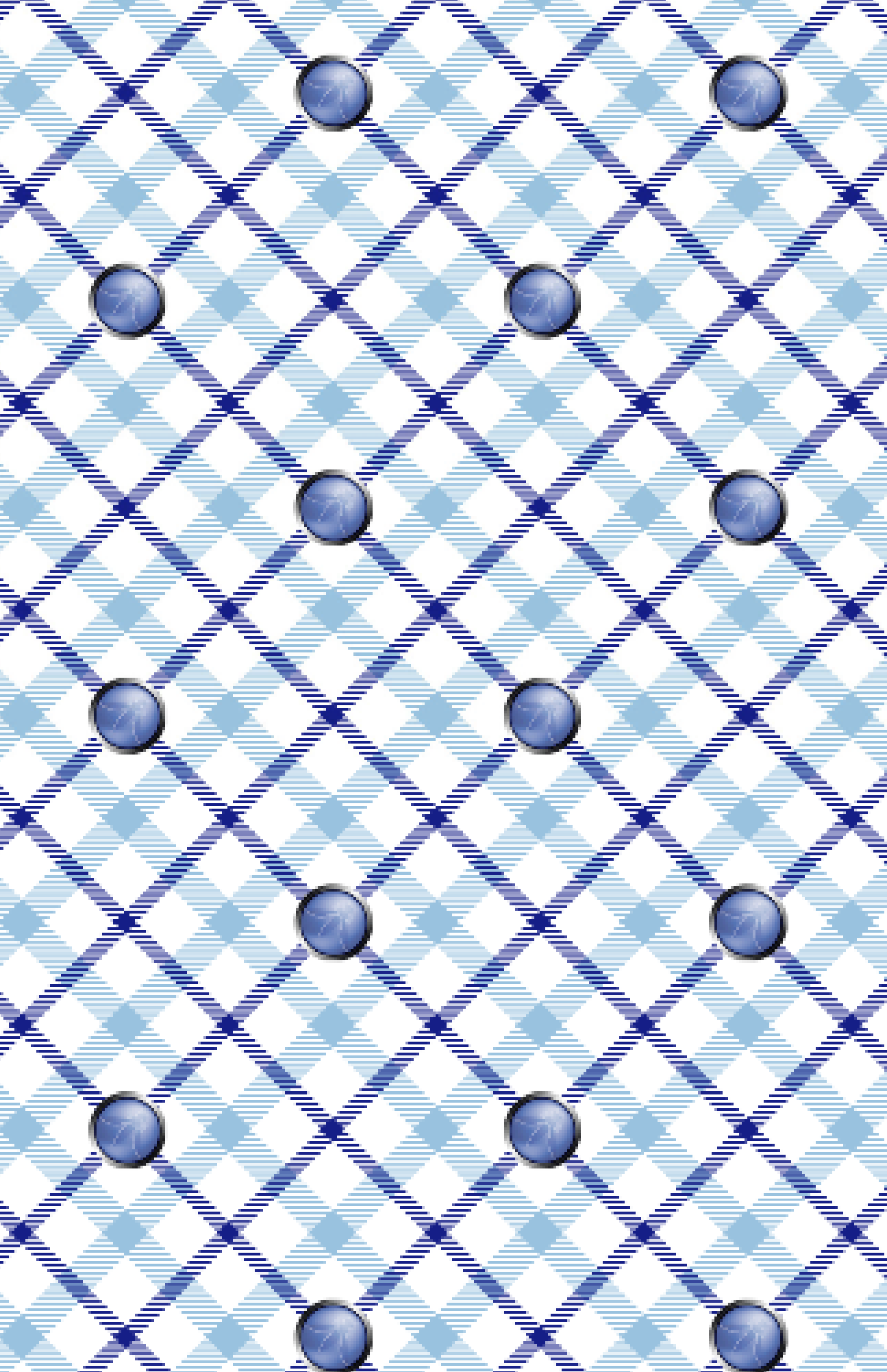
OWASP SCP 62, 63
OWASP ASVS 3.2, 3.4, 3.8
OWASP AppSensor -
CAPEC 21
SAFECODE 28



8

Matt can abuse long sessions because the application does not require periodic re-authentication to check if privileges have changed

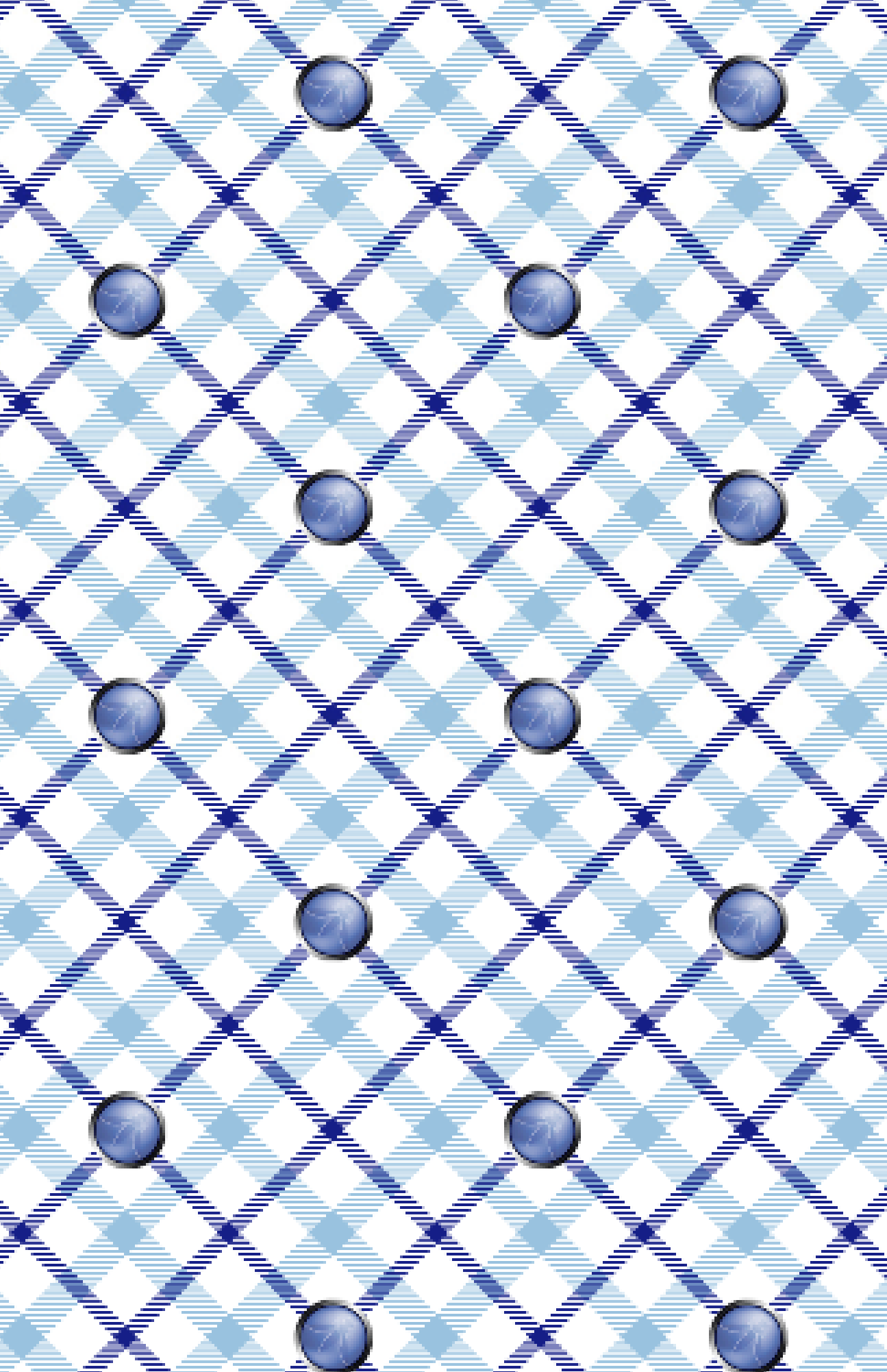
OWASP SCP 96
OWASP ASVS -
OWASP AppSensor -
CAPEC 21
SAFECODE 28



9

Ivan can steal session identifiers because they are sent over insecure channels, or are logged, or are revealed in error messages, or are included in URLs, or are accessible un-necessarily by code which the attacker can influence or alter

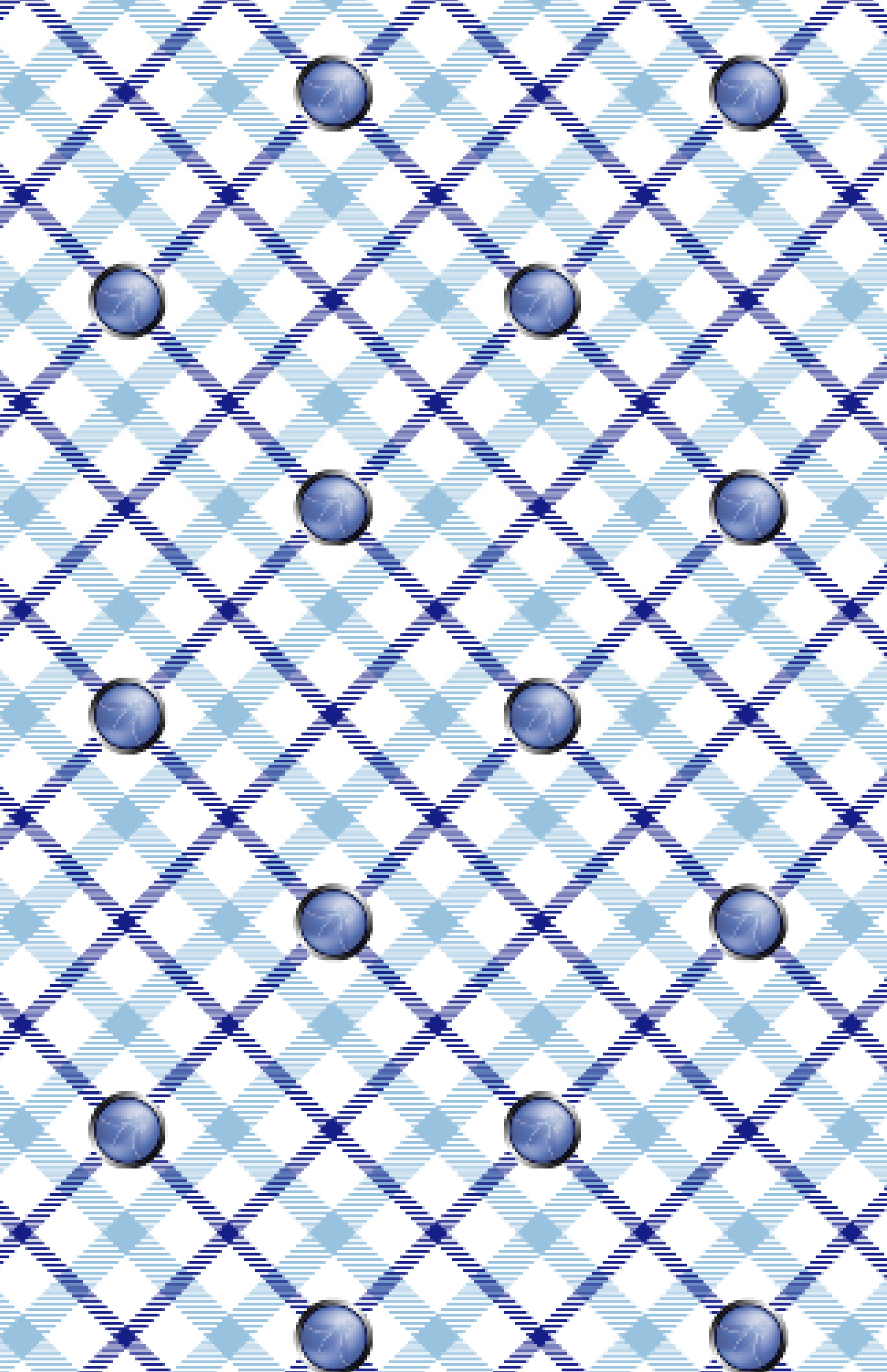
OWASP SCP 69, 75, 76, 119, 138
OWASP ASVS 3.5, 8.10, 11.4
OWASP AppSensor SE4-6
CAPEC 31, 60
SAFECODE 28



10

Marce can forge requests because per-session, or per-request for more critical actions, strong random tokens or similar are not being used for actions that change state

OWASP SCP 73, 74
OWASP ASVS 11.7
OWASP AppSensor IE4
CAPEC 62, 111
SAFECODE 18



J

Jeff can resend an identical interaction (e.g. HTTP request, signal, button press) and it is accepted, not rejected

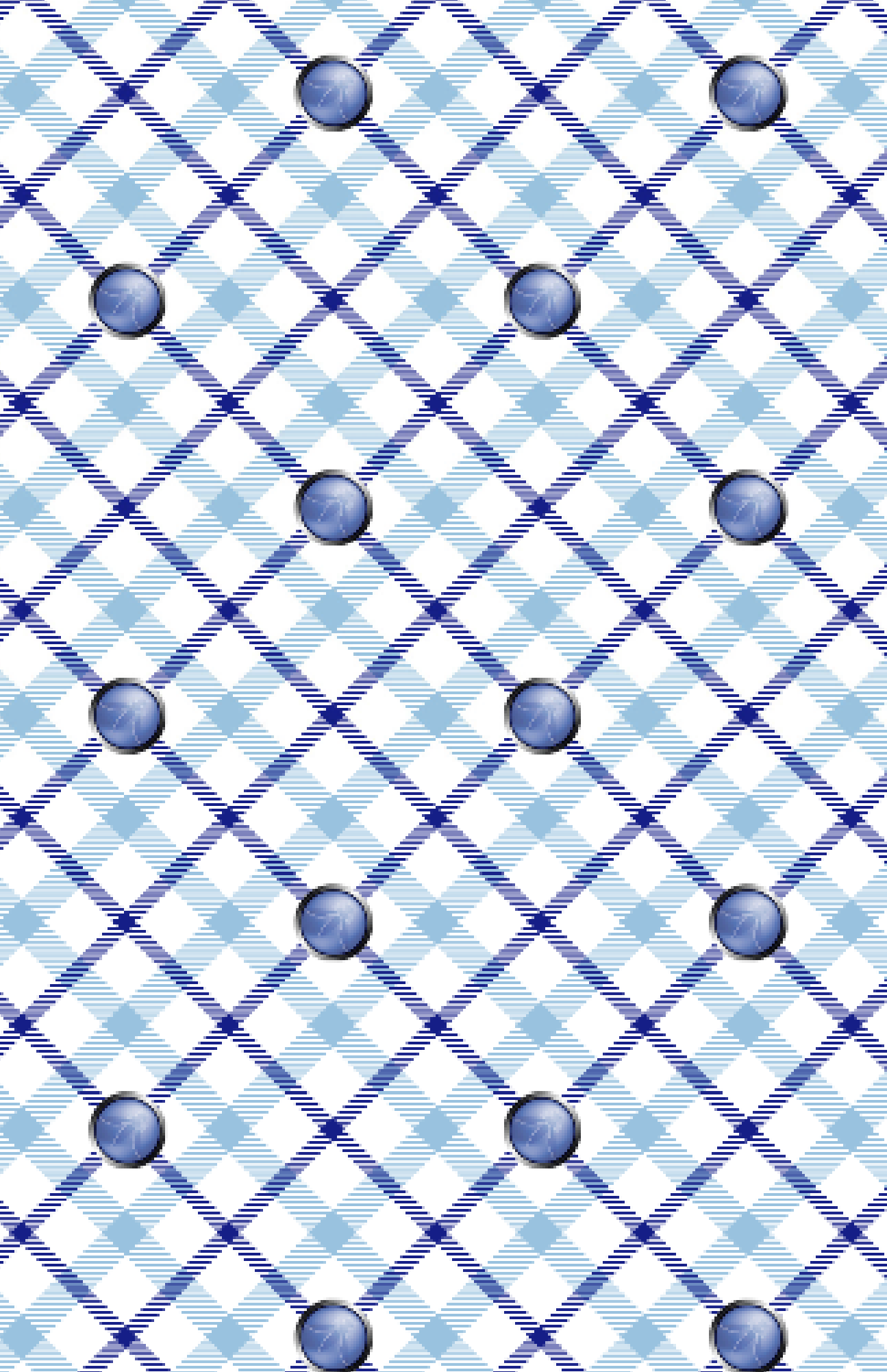
OWASP SCP
-

OWASP ASVS
-

OWASP AppSensor
IE5

CAPEC
60

SAFECODE
12, 14



Q

Salim can bypass session management because it is not applied comprehensively and consistently across the application

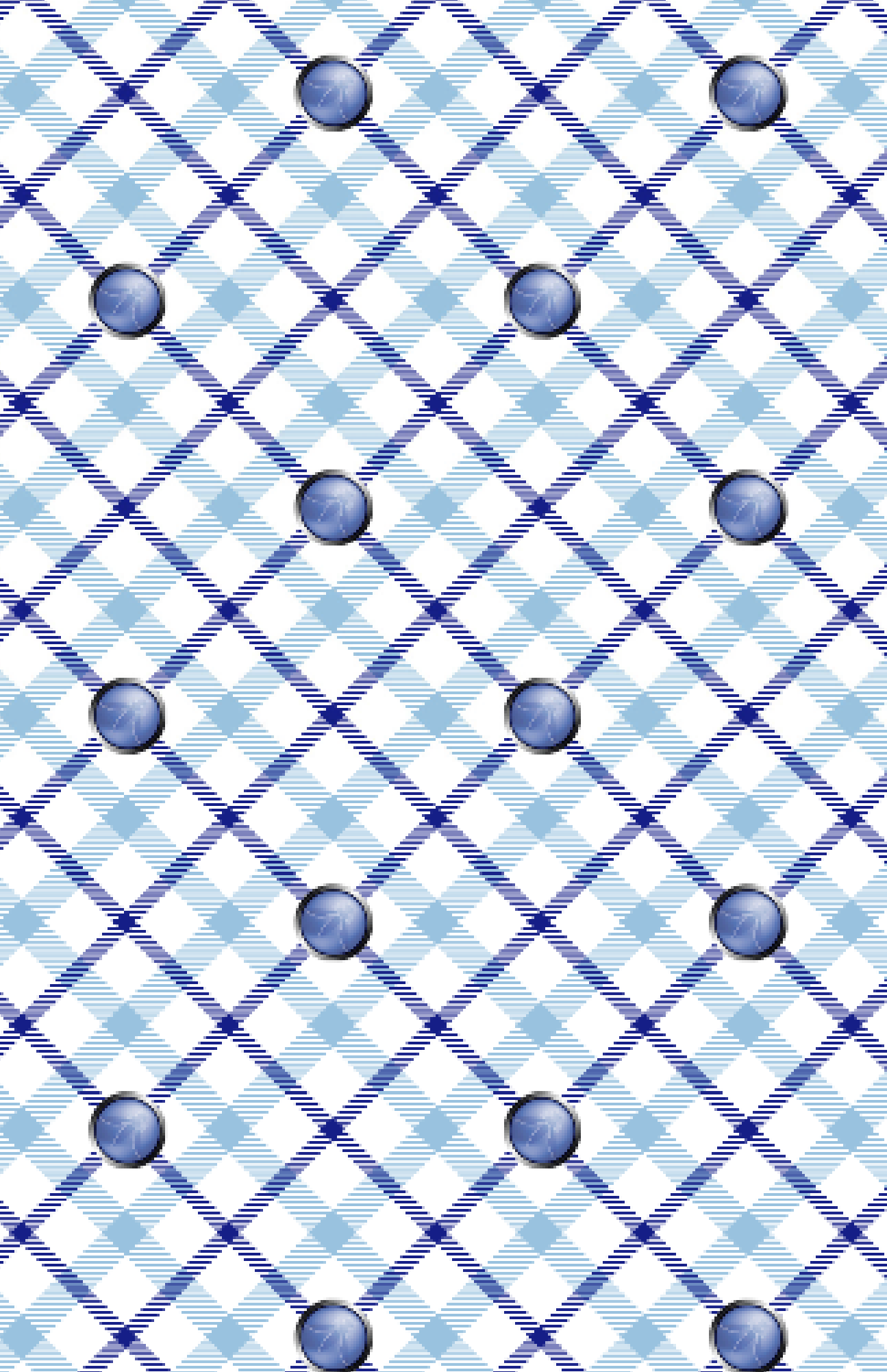
OWASP SCP
58

OWASP ASVS
3.1

OWASP AppSensor
-

CAPEC
21

SAFECODE
14, 28



K

Peter can bypass the session management controls because they have been self-built and/or are weak, instead of using a standard framework or approved tested module

OWASP SCP 58, 60
OWASP ASVS 3.1
OWASP AppSensor -
CAPEC 21
SAFECODE 14, 28