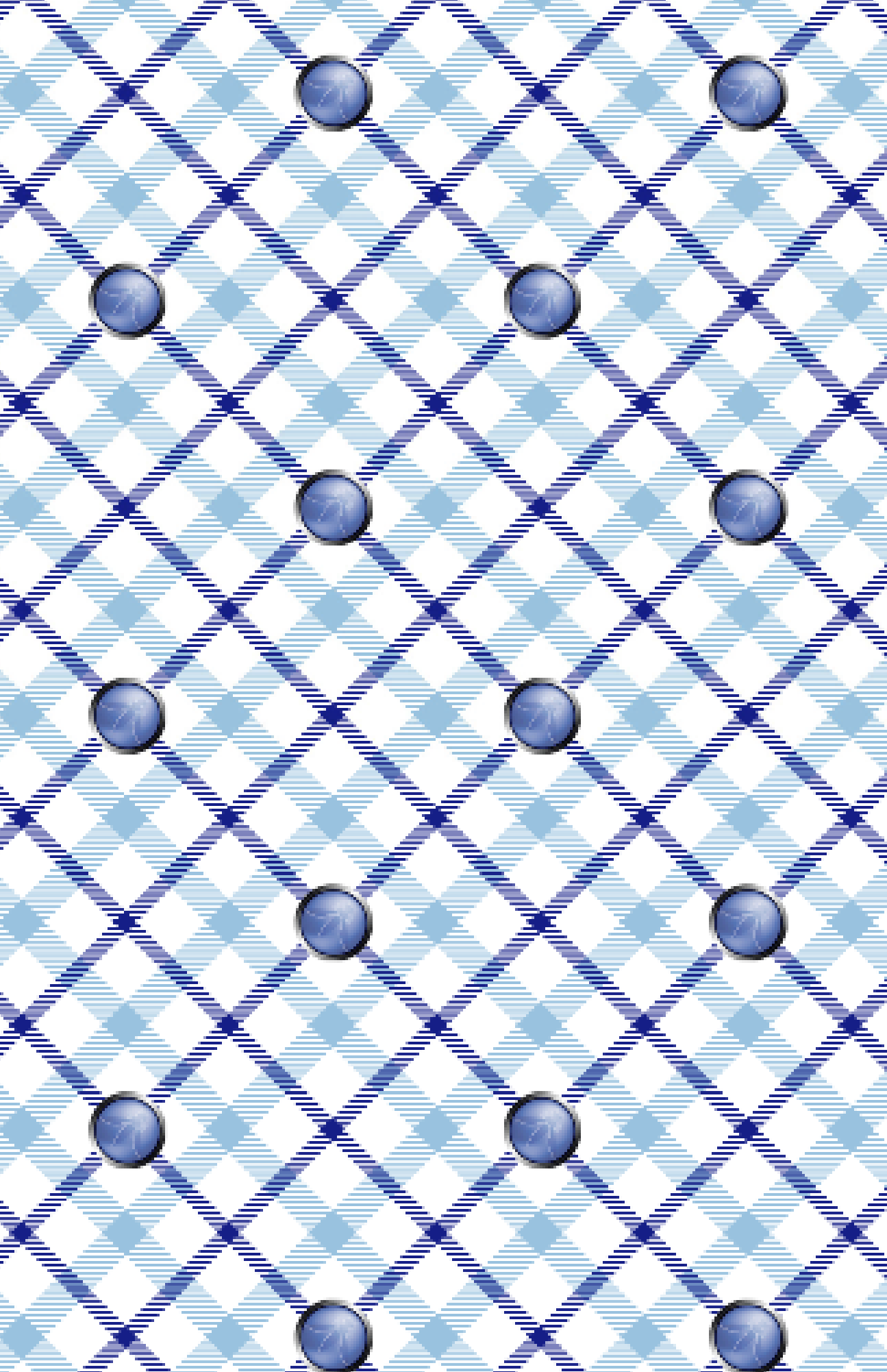
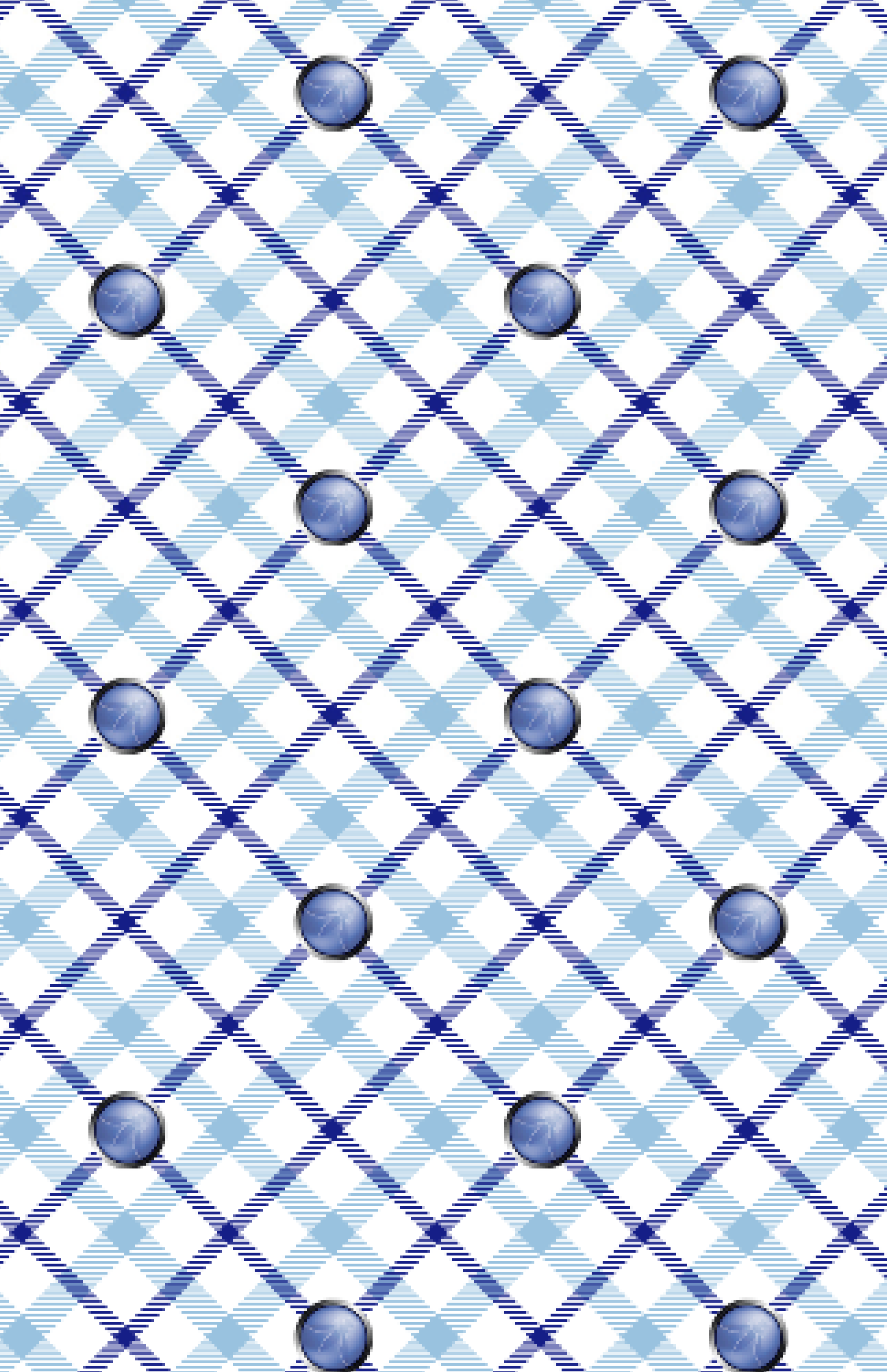


AUTHORIZATION

A

You have invented a new attack against Authorization

Read more about this topic in OWASP's Development and Testing Guides



AUTHORIZATION

2

Tim can influence where data is sent or forwarded to

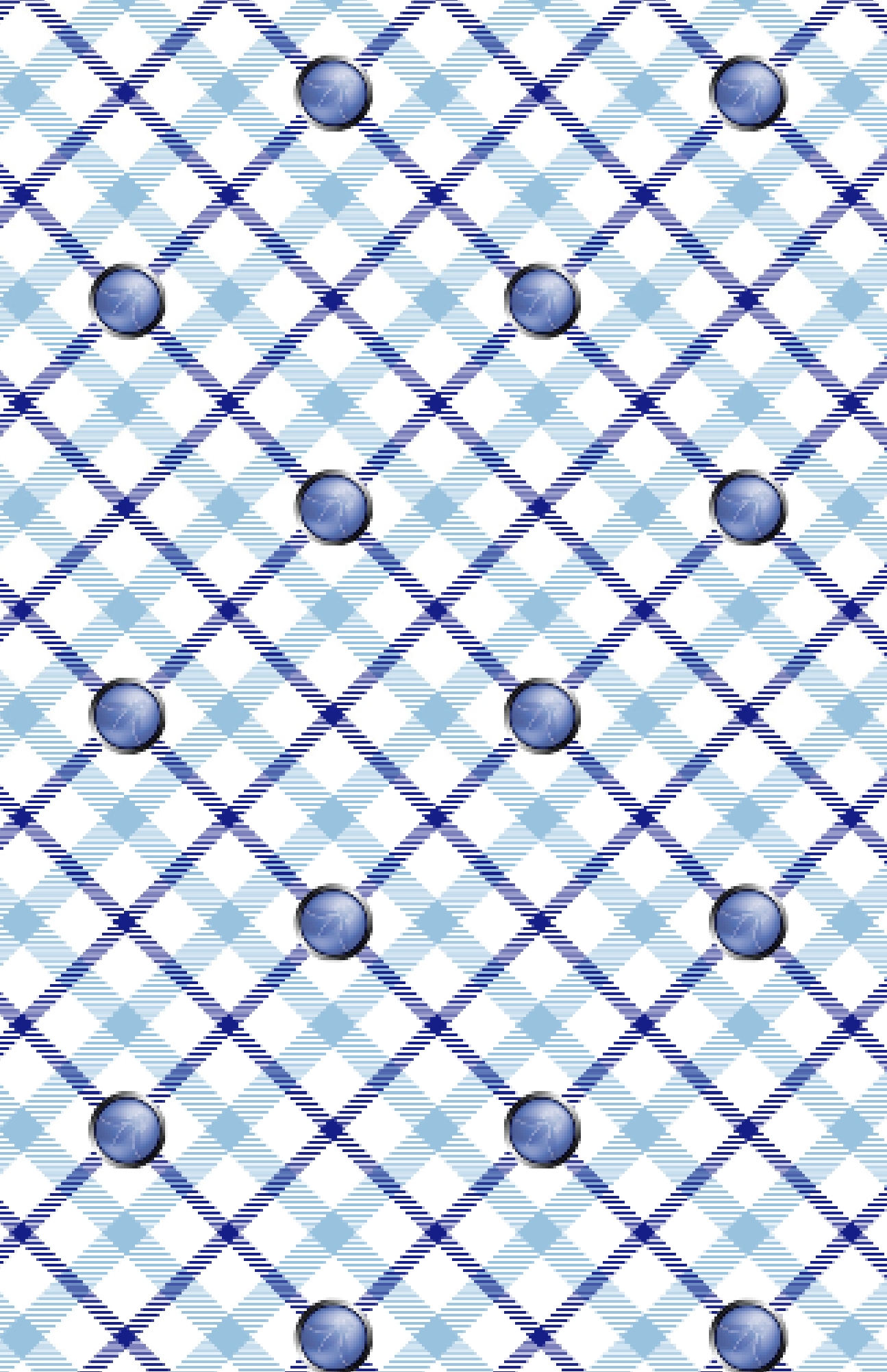
OWASP SCP
44

OWASP ASVS
4.1, 4.2, 4.3, 4.4, 4.6

OWASP AppSensor
-

CAPEC
153

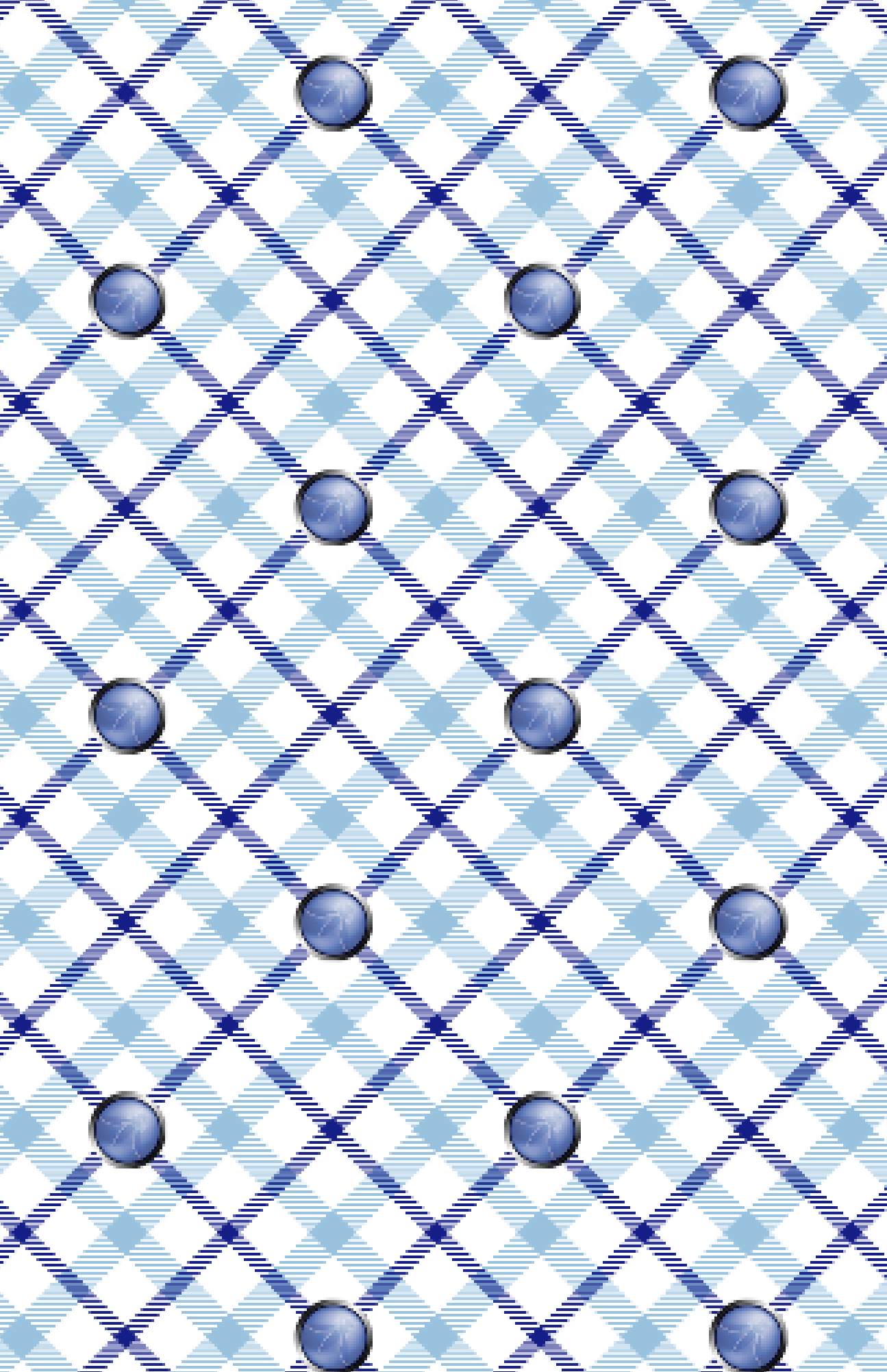
SAFECODE
8, 10, 11



3

Christian can access (read, write, update or delete) information, which they should not have permission to, through another mechanism that does have permission (e.g. search indexer, logger, reporting), or because it is cached, or other information leakage

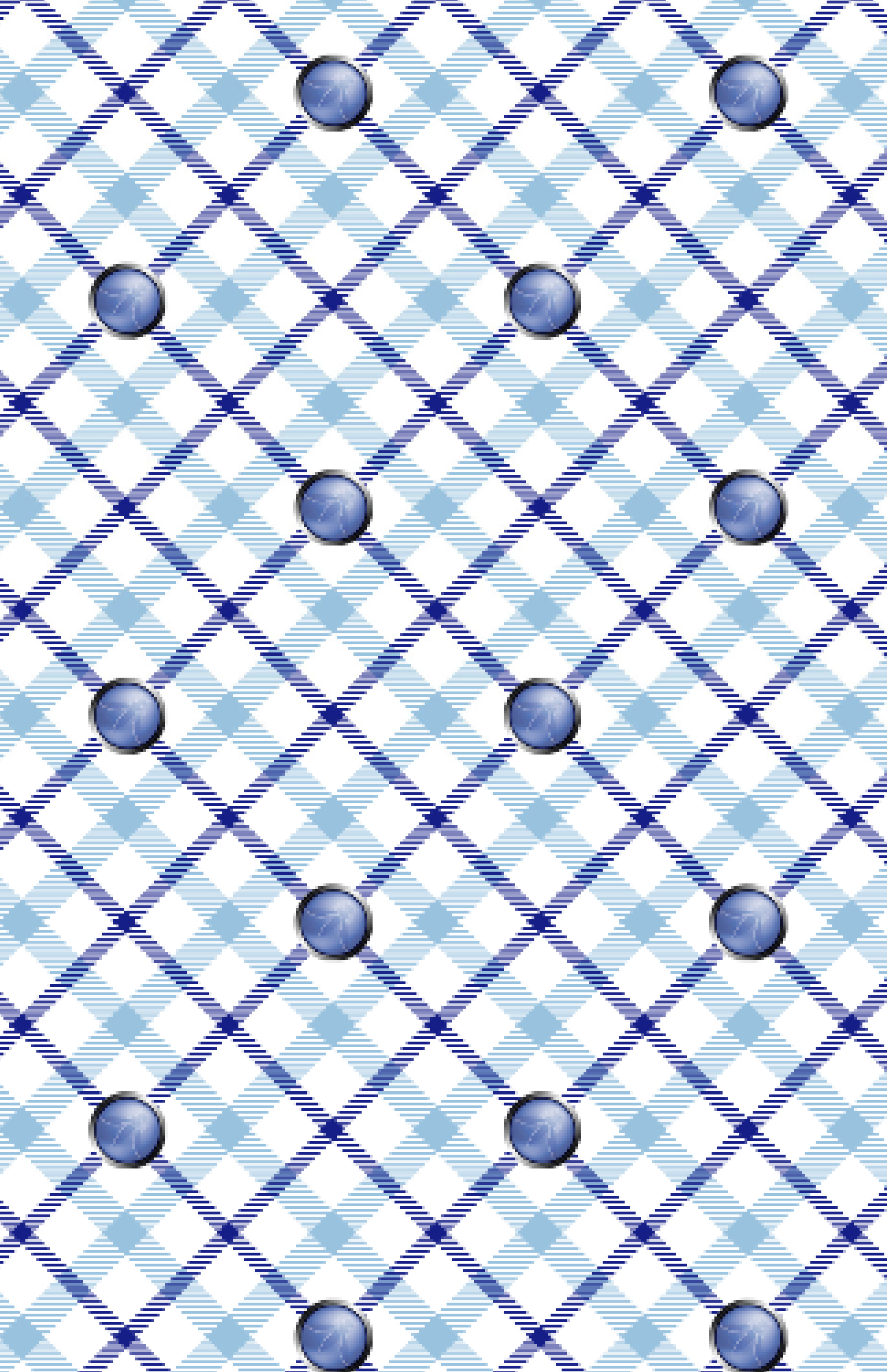
OWASP SCP 51, 139, 140, 150
OWASP ASVS 4.1, 8.7, 9.1, 9.2, 9.3, 9.4, 9.5
OWASP AppSensor -
CAPEC 69, 213
SAFECODE 8, 10, 11



4

Kelly can bypass authorization controls because they do not fail securely (i.e. they default to allowing access)

OWASP SCP 79, 80
OWASP ASVS 4.8
OWASP AppSensor -
CAPEC 122
SAFECODE 8, 10, 11

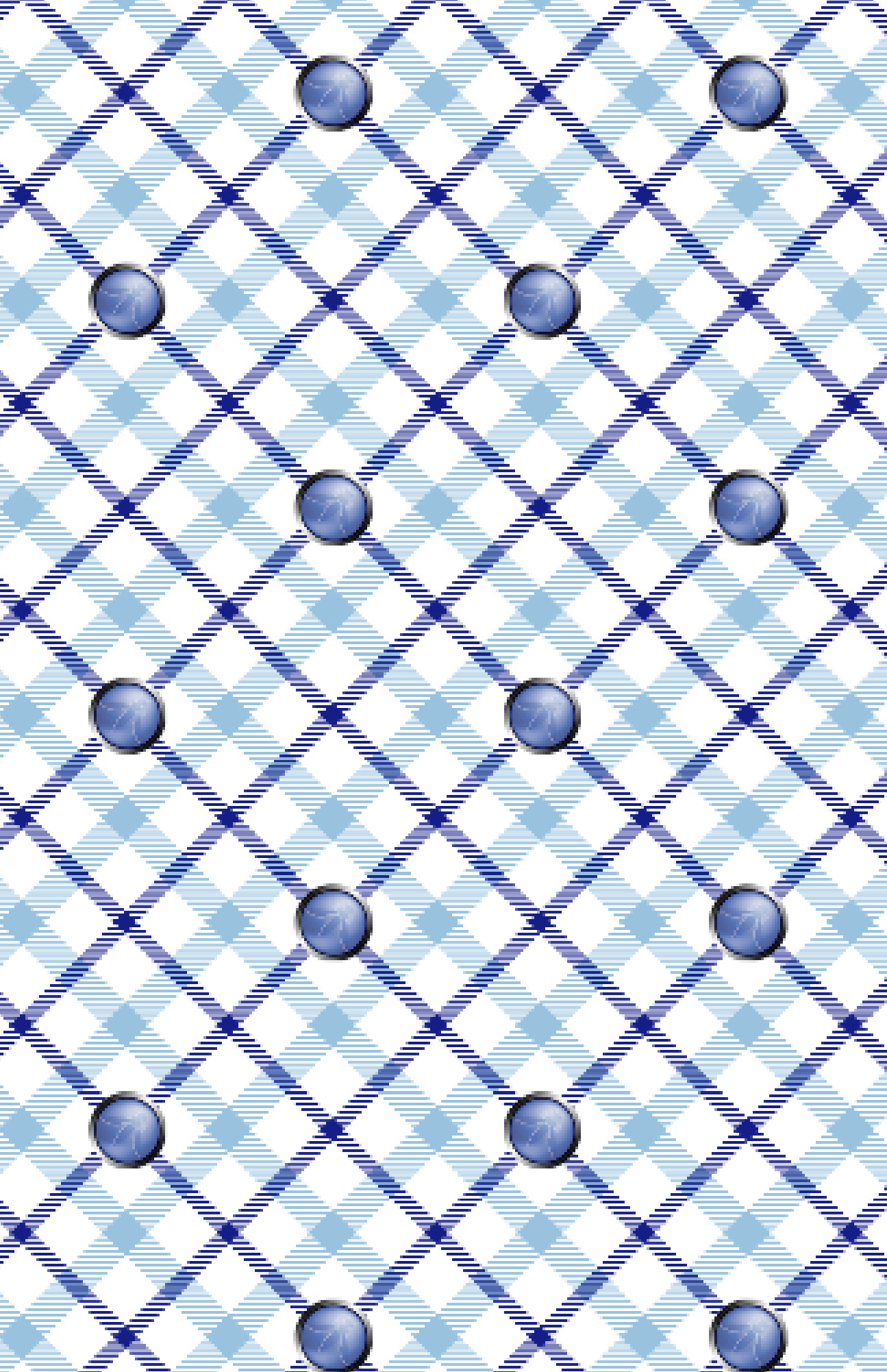


AUTHORIZATION

5

Chad can access resources (including services, processes, AJAX, Flash, video, images, documents, temporary files, session data, system properties, configuration data, registry settings, logs) he should not be able to due to missing authorization, or due to excessive privileges (e.g. not using the principle of least privilege)

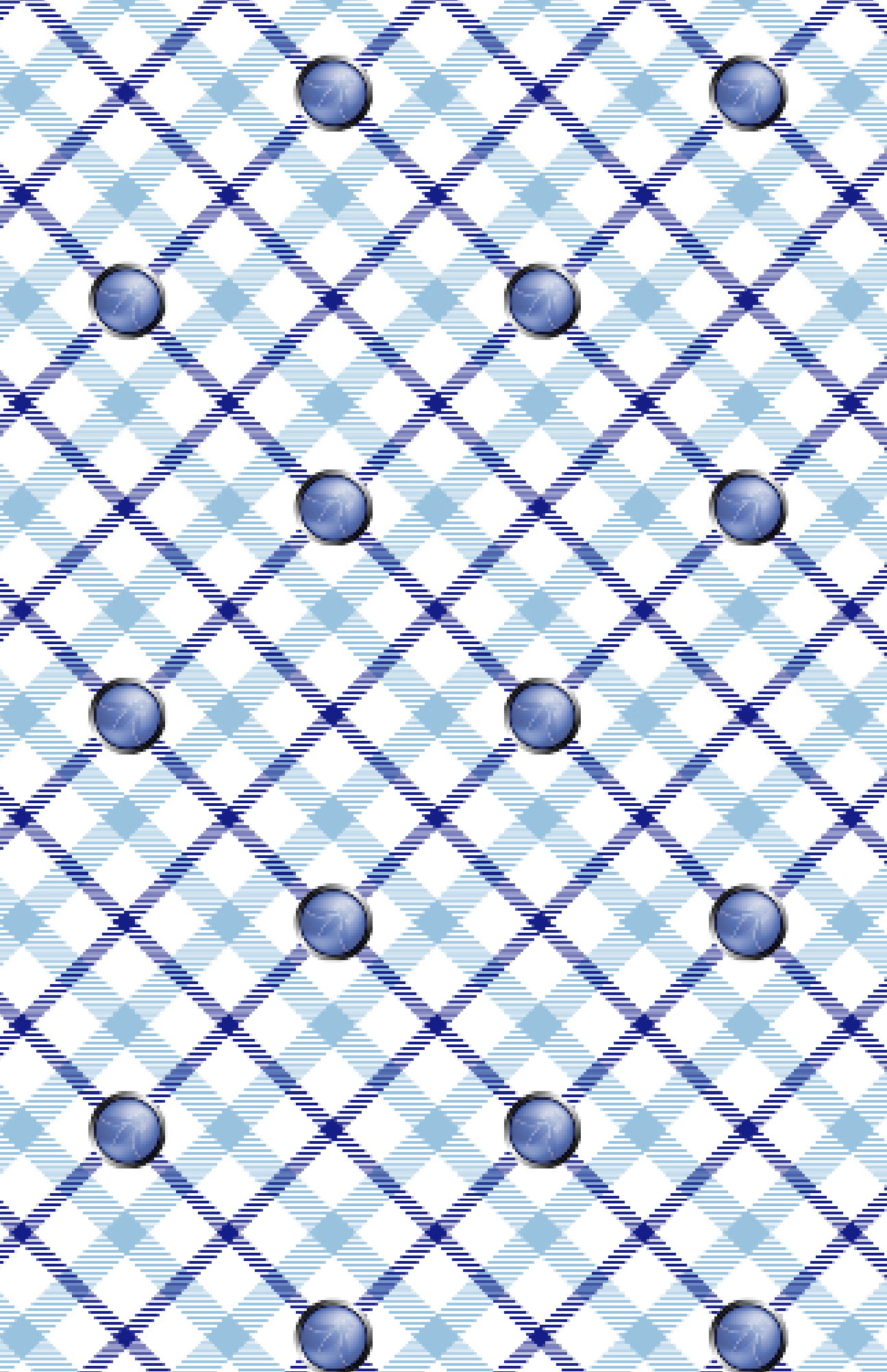
OWASP SCP 30,70,81,83-4,87-9, 99,117,131-2,142,154,170,179,190-2
OWASP ASVS 4.1, 4.3, 4.4, 4.6, 8.7, 10.7
OWASP AppSensor ACE1-4, HT2
CAPEC 75, 87, 95, 126, 149, 155, 203, 213, 264-5
SAFECODE 8, 10, 11, 13



6

Eduardo can access data he does not have permission to, even though he has permission to the form/page/URL/entry point

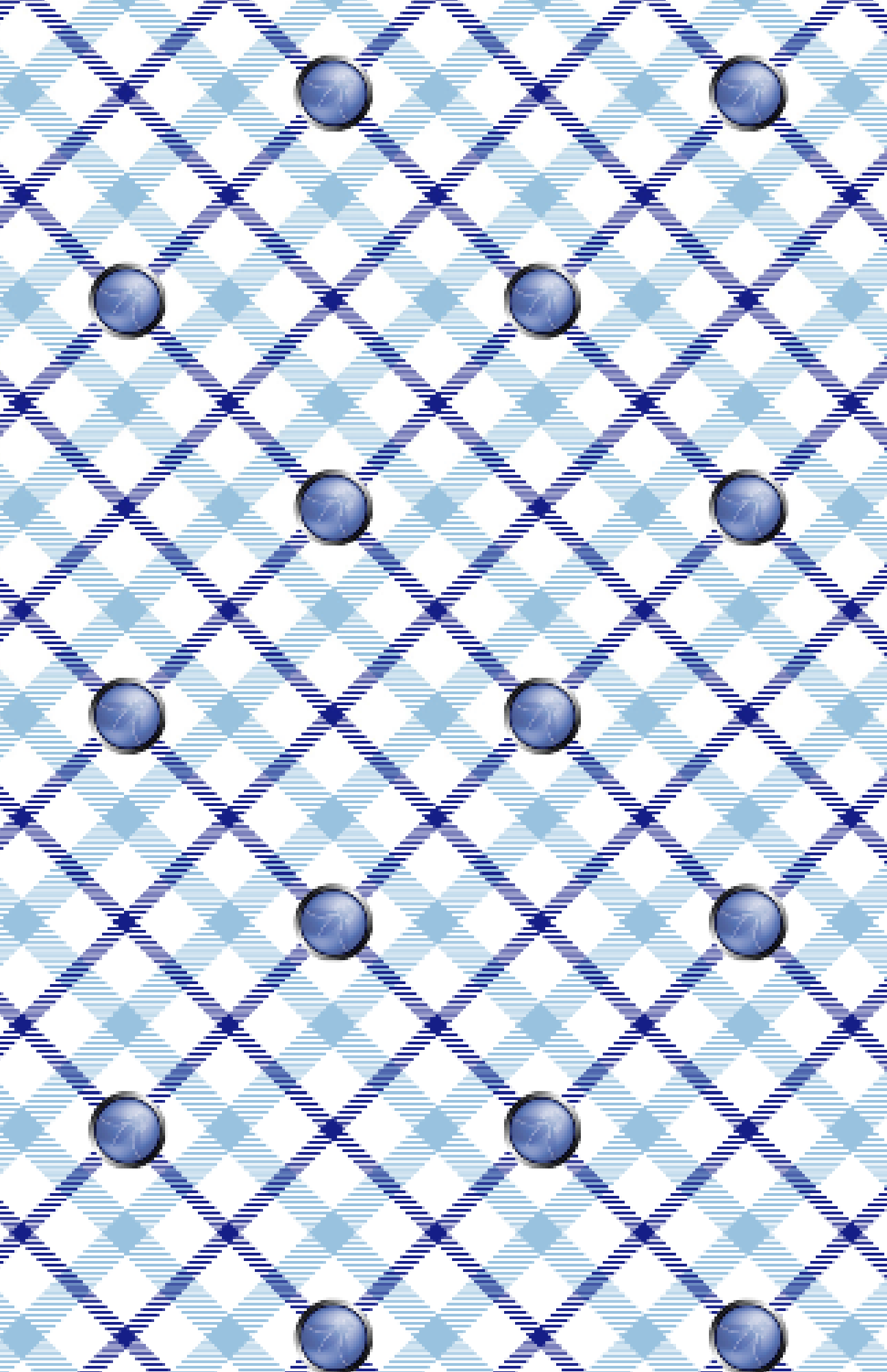
OWASP SCP 81
OWASP ASVS 4.1, 4.2, 4.3, 4.4, 4.6
OWASP AppSensor ACE1-4
CAPEC 122
SAFECODE 8, 10, 11



7

Yuanjing can access application functions, objects, or properties he is not authorized to access

OWASP SCP 81, 85, 86
OWASP ASVS 4.1, 4.2, 4.3, 4.4, 4.6
OWASP AppSensor ACE1-4
CAPEC 122
SAFECODE 8, 10, 11

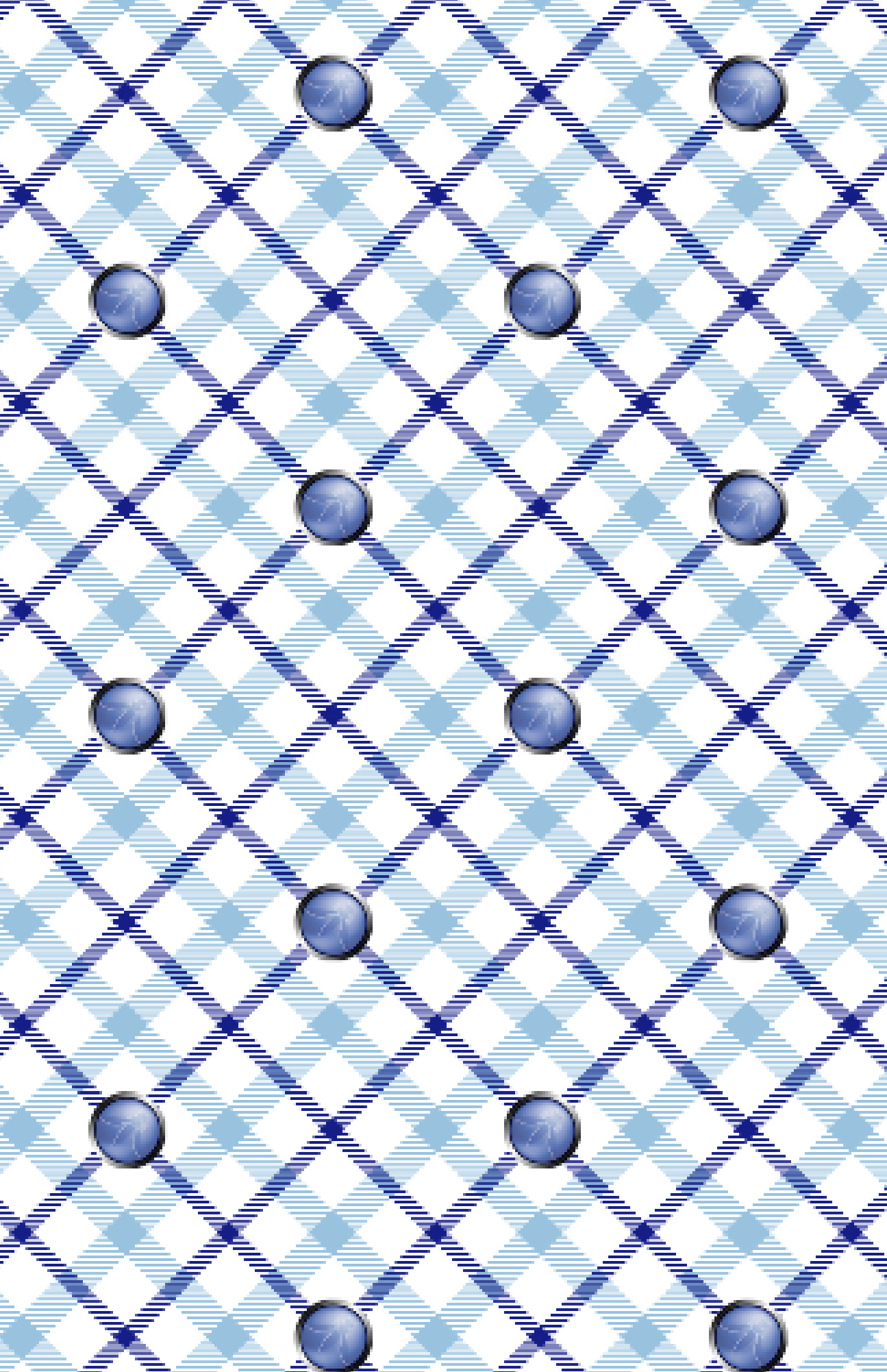


AUTHORIZATION

8

Tom can bypass business rules by altering the usual process sequence or flow, or by undertaking the process in the incorrect order, or by manipulating date and time values used by the application, or by using valid features for unintended purposes, or by otherwise manipulating control data

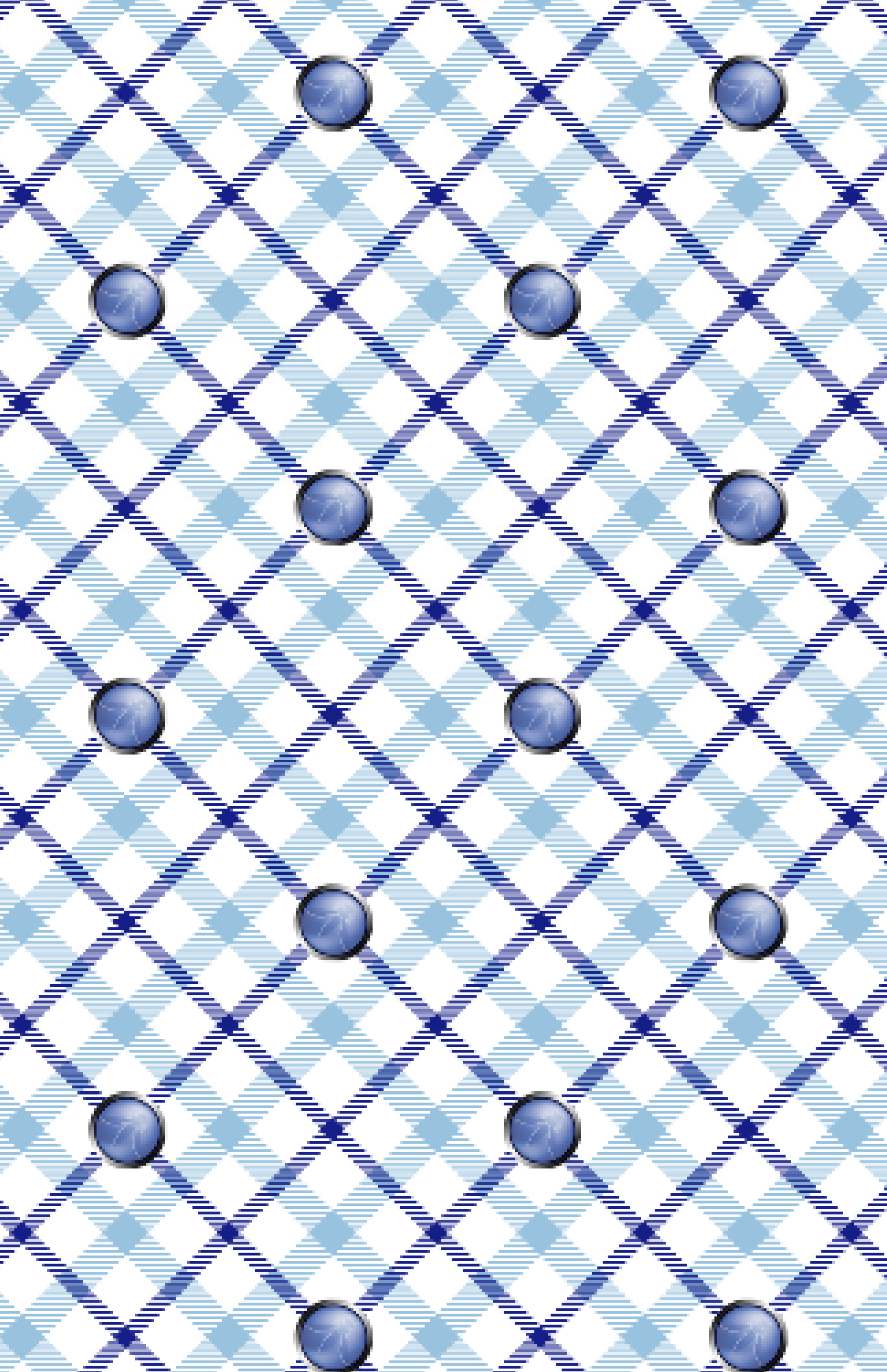
OWASP SCP 10, 32, 93, 94, 189
OWASP ASVS 4.1, 4.2, 4.3, 4.4, 4.6, 4.12
OWASP AppSensor ACE3
CAPEC 25, 39, 74, 162, 166, 207
SAFECODE 8, 10, 11, 12



9

Mike can misuse an application by using a valid feature too fast, or too frequently, or other way that is not intended, or consumes the application's resources, or causes race conditions, or over-utilizes a feature

OWASP SCP 94
OWASP ASVS 4.12
OWASP AppSensor AE3, FIO1-2, UT2-4, STE1-3
CAPEC 26, 29, 119, 261
SAFECODE 1, 35

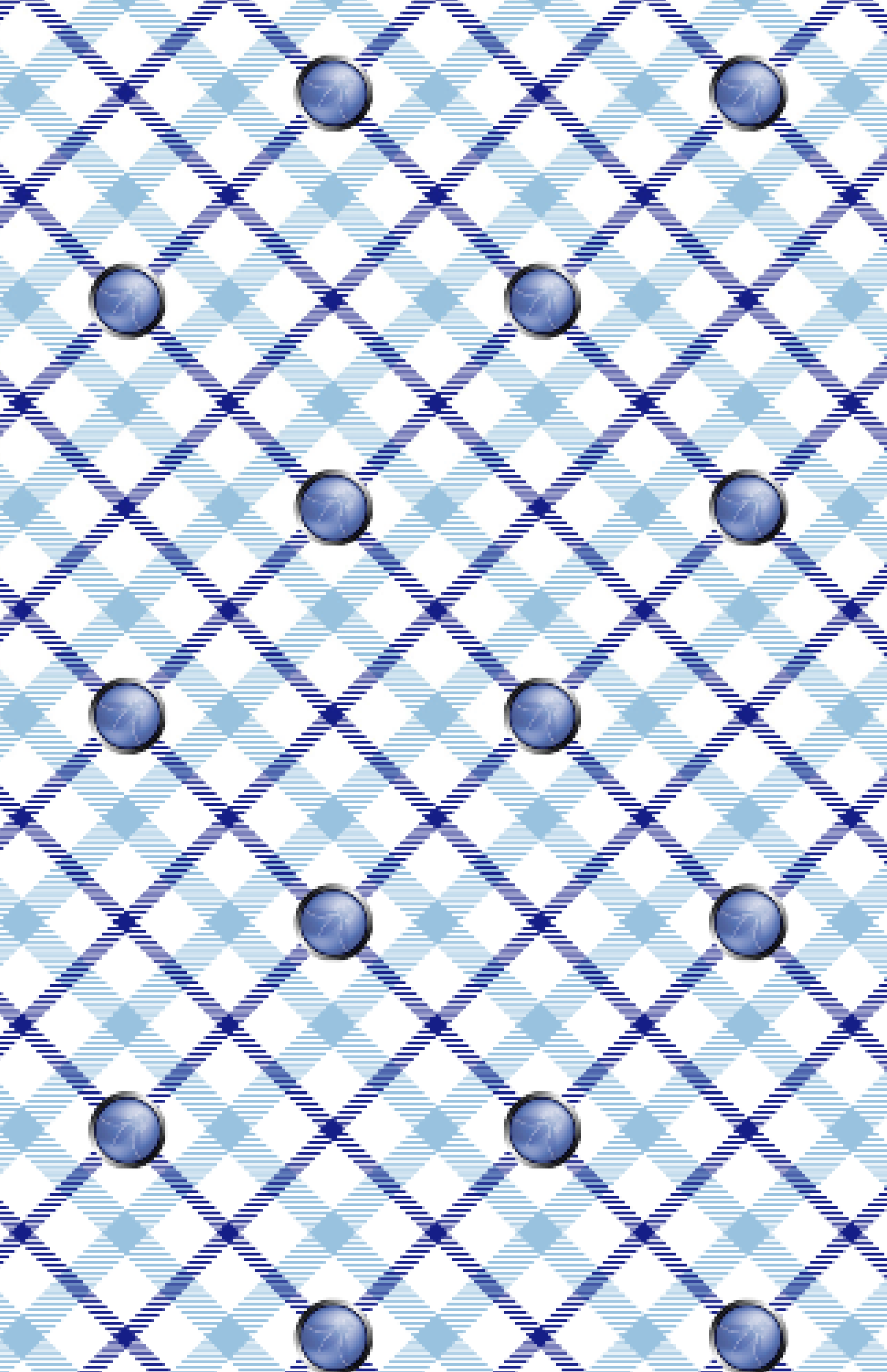


AUTHORIZATION

10

Richard can bypass the centralized authorization controls since they are not being used comprehensively on all interactions

OWASP SCP 78, 91
OWASP ASVS 4.13, 4.14
OWASP AppSensor ACE1-4
CAPEC 36, 95, 121, 179
SAFECODE 8, 10, 11



AUTHORIZATION

J

Dinis can access security configuration information, or access control lists

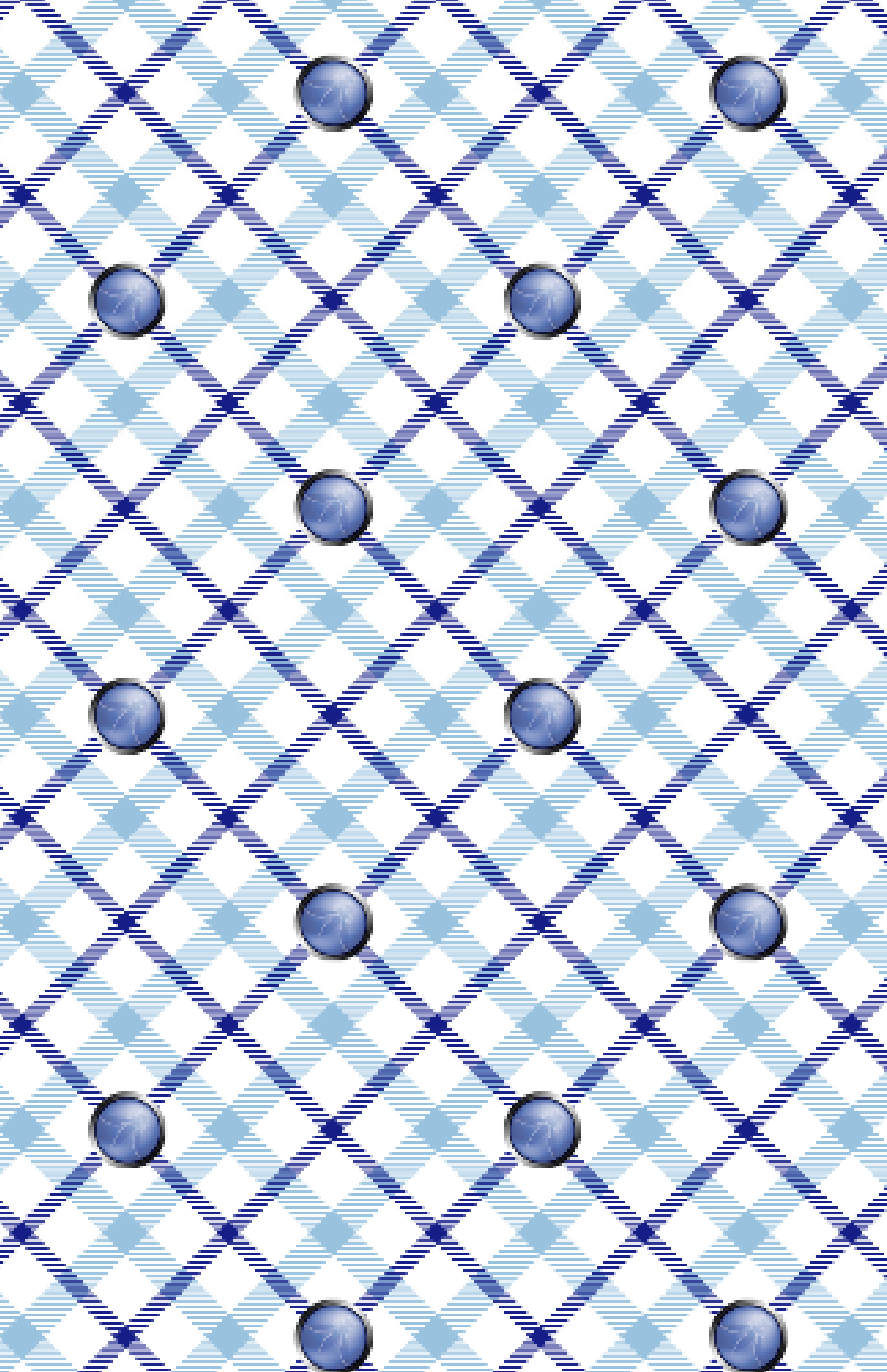
OWASP SCP
89, 90

OWASP ASVS
12.1

OWASP AppSensor
-

CAPEC
75, 133, 203

SAFECODE
8, 10, 11



AUTHORIZATION

Q

Christopher can inject a command that the application will run at a higher privilege level

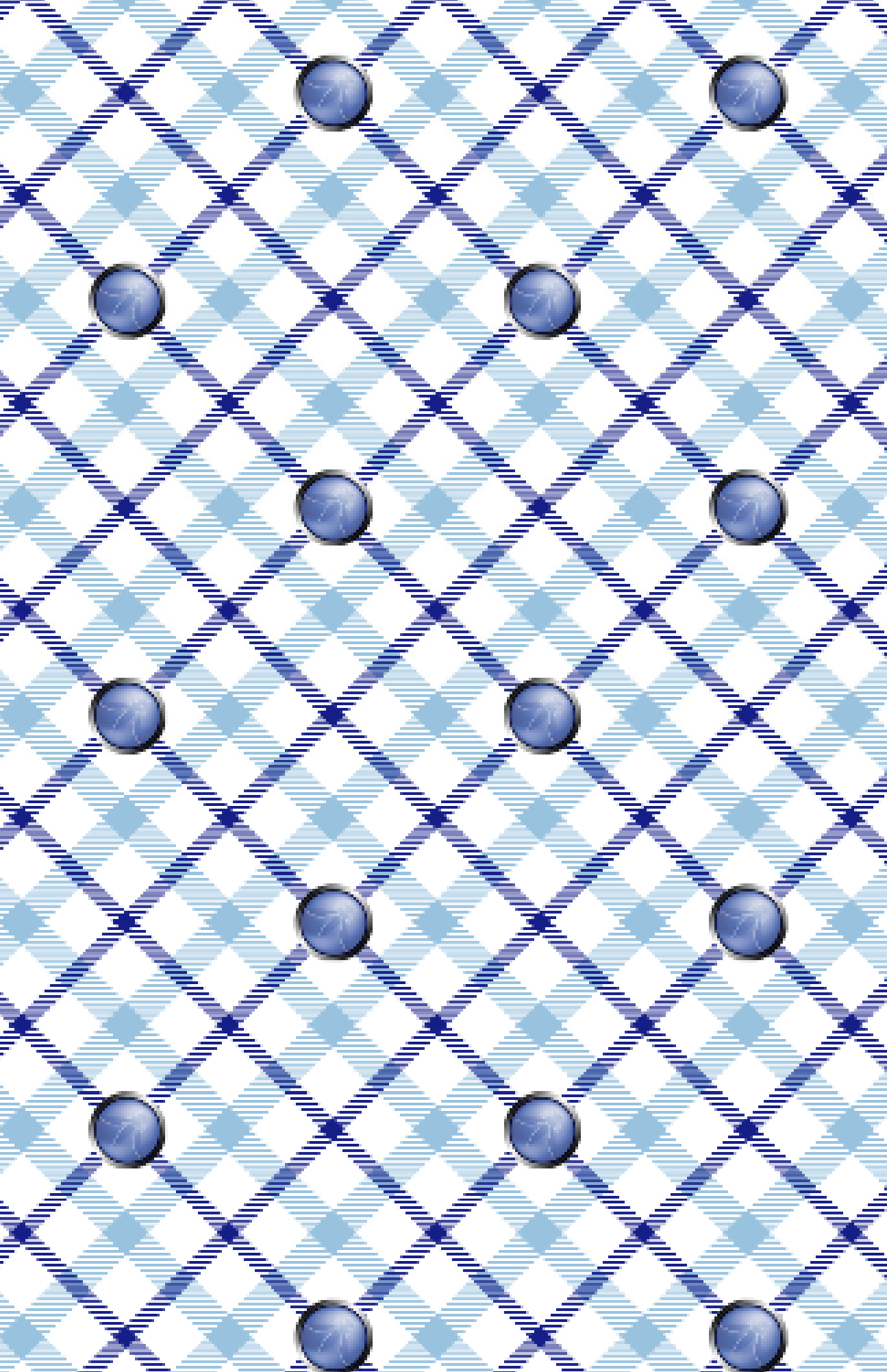
OWASP SCP
208

OWASP ASVS
4.1, 4.6

OWASP AppSensor
-

CAPEC
17, 30, 69, 234

SAFECODE
8, 10, 11



AUTHORIZATION

K

Ryan can influence or alter authorization controls and permissions, and can therefore bypass them

OWASP SCP
77, 91

OWASP ASVS
4.9, 4.10, 4.11

OWASP AppSensor
-

CAPEC
56, 207, 211

SAFECODE
8, 10, 11