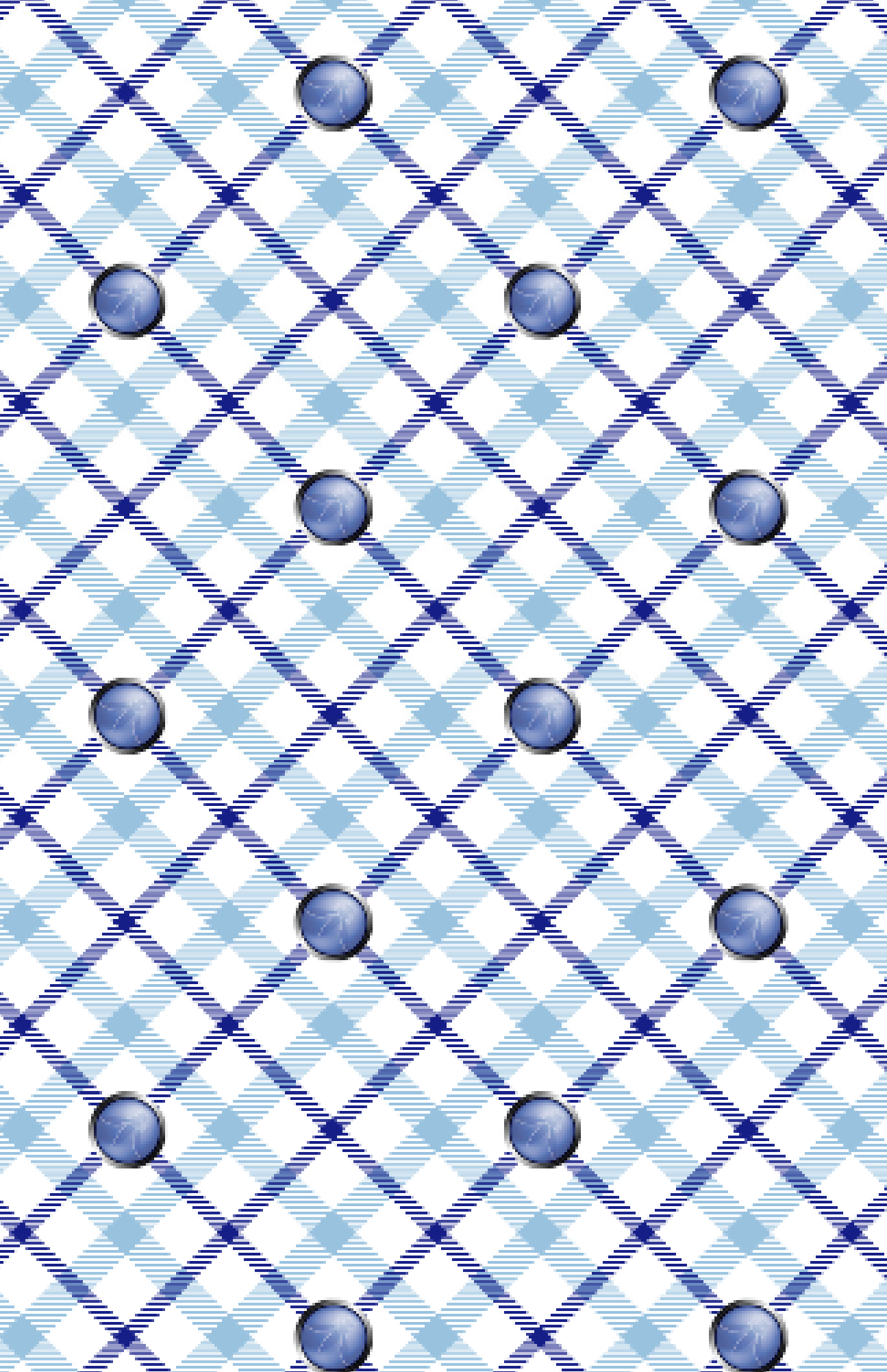
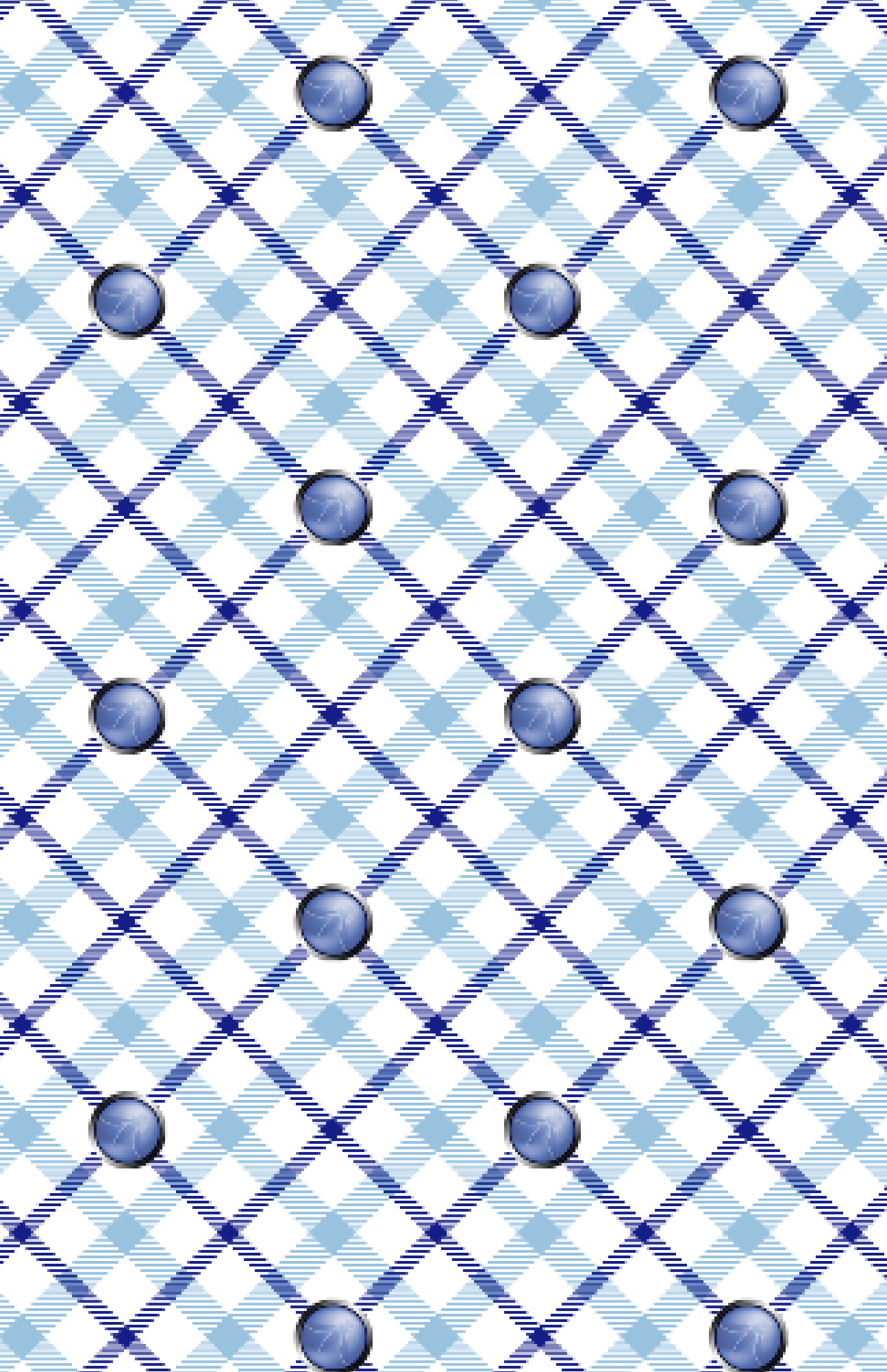


Data Validation & Encoding

A

You have invented a new attack against Data Validation and Encoding

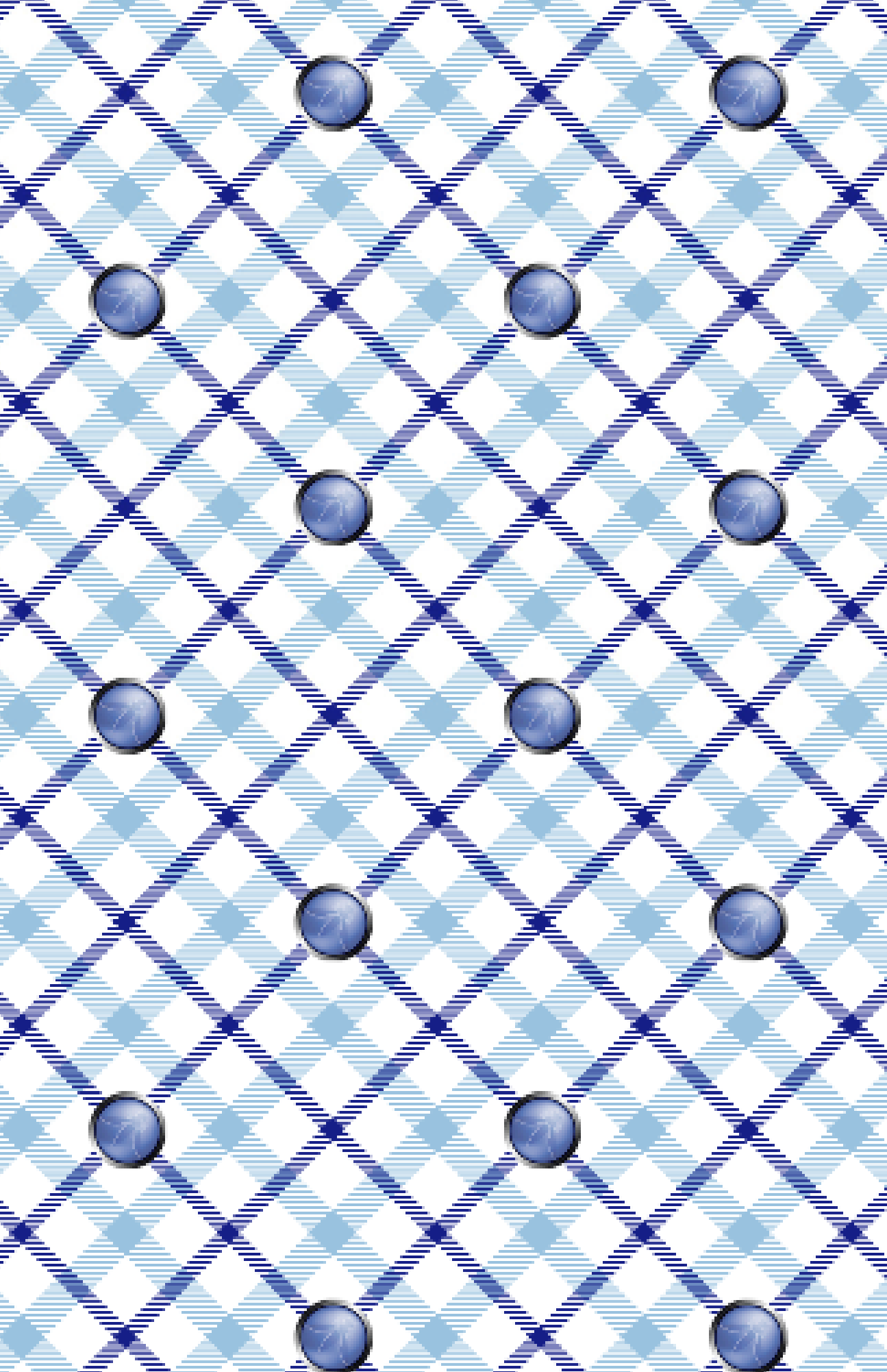
Read more about this topic in OWASP's free Cheat Sheets on Input Validation, XSS Prevention, DOM-based XSS Prevention, SQL Injection Prevention, and Query Parameterization



2

Brian can gather information about the underlying configurations, schemas, logic, code, software, services and infrastructure due to the content of error messages, or due to poor configuration, or due to the presence of default installation files or old, test, backup or copies of resources, or exposure of source code

OWASP SCP 69, 107-109, 136, 137, 153, 156, 158, 162
OWASP ASVS 4.5, 8.1, 8.2
OWASP AppSensor HT1-3
CAPEC 54, 224
SAFECODE 4, 23

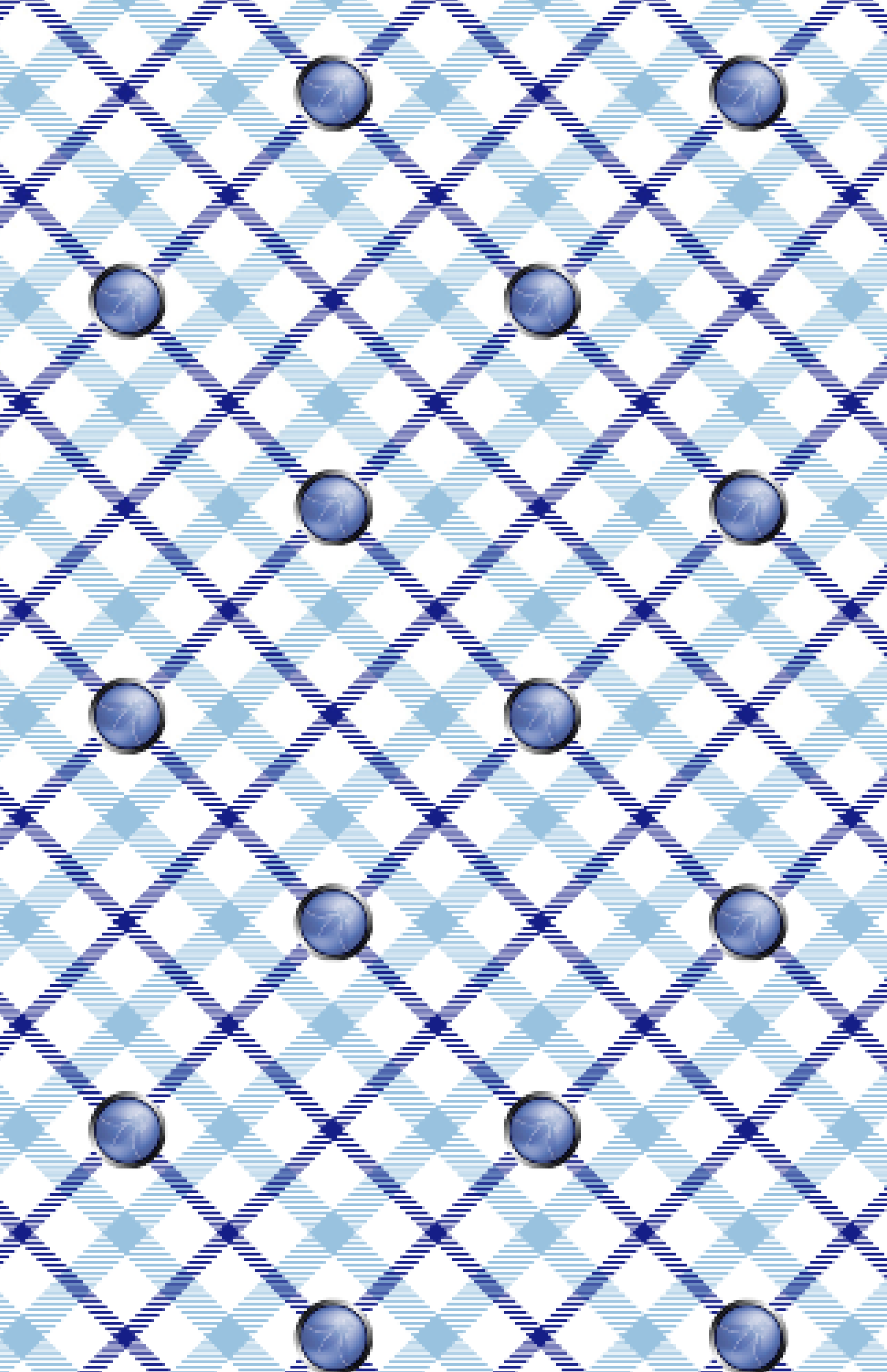


Data Validation & Encoding

3

Robert can input malicious structured or unstructured data because the allowed protocol format is not being checked, or the structure is not being verified, or the individual data elements are not being validated for format, type, range, length and a whitelist of allowed characters or formats

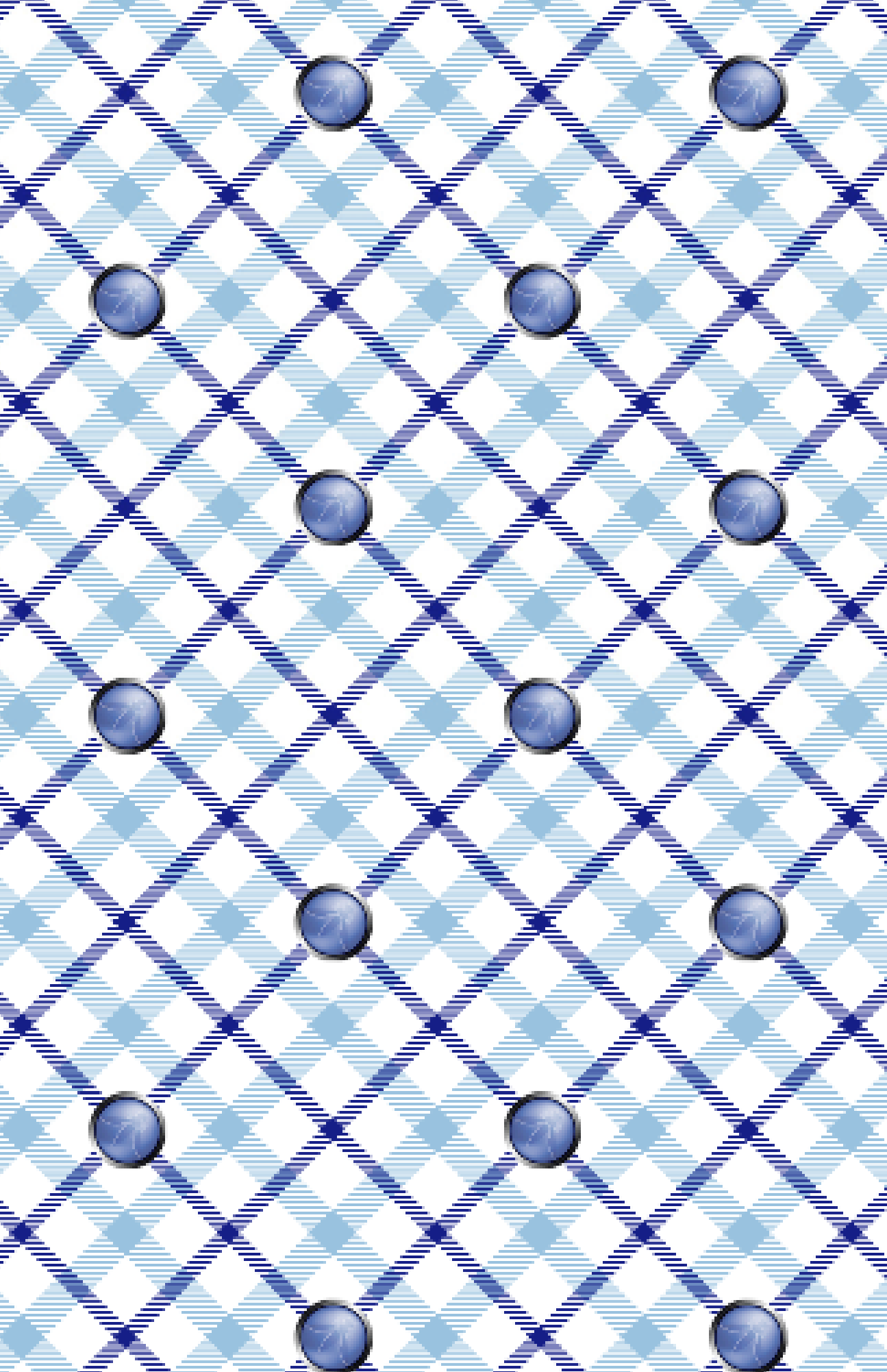
OWASP SCP 8, 9, 11-14, 16, 159, 190, 191
OWASP ASVS 5.2
OWASP AppSensor RE7-8, AE4-7, IE2-3,CIE1,CIE3-4,HT1-3
CAPEC 28,48,126,165,213,220,221,257,261,271,272
SAFECode 3, 16, 24, 35



4

Dave can input malicious data because it is not being checked within the context of the current user and process

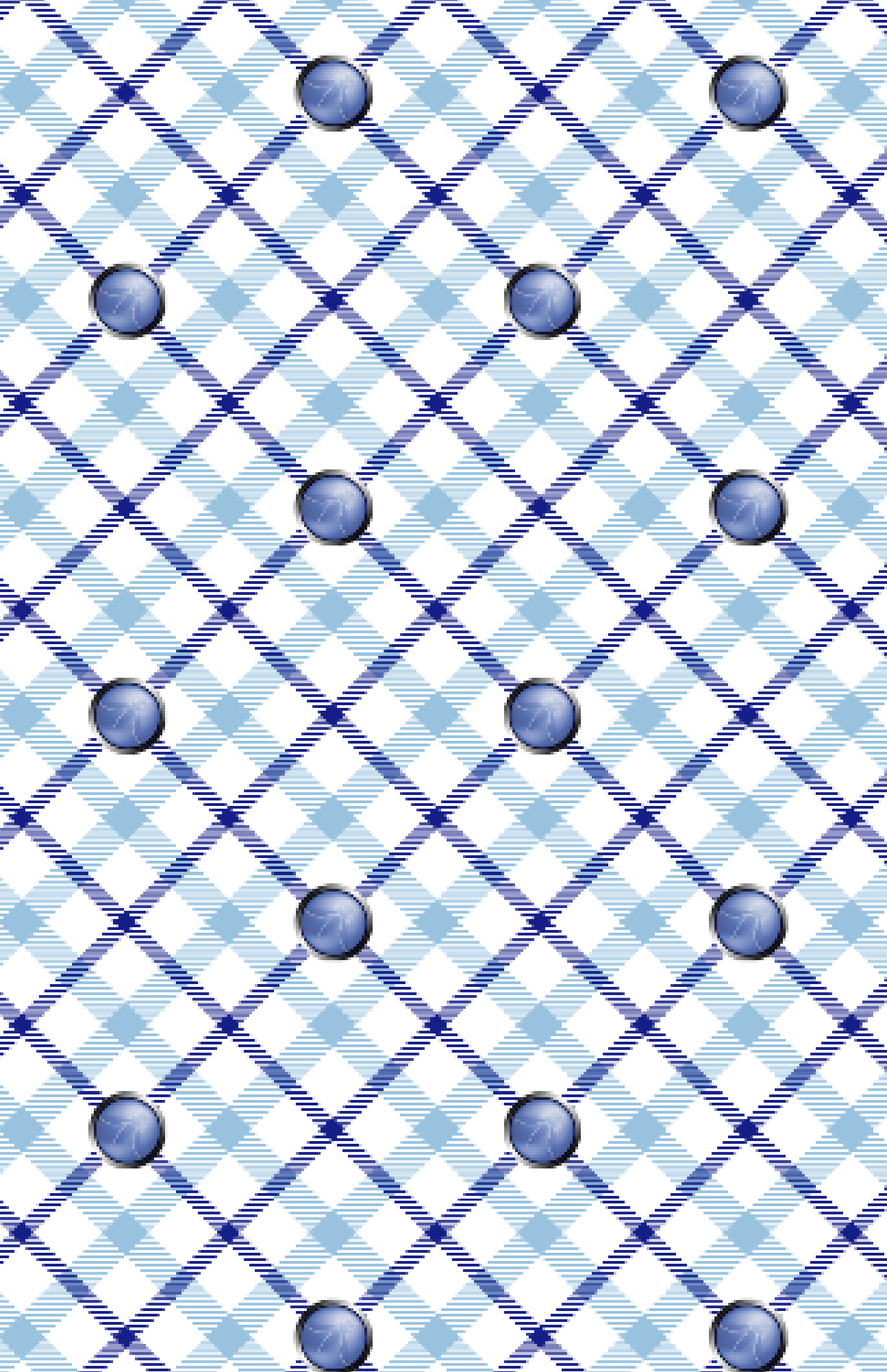
OWASP SCP 8, 10, 183
OWASP ASVS 5.2, 11.1
OWASP AppSensor RE3-6,AE8-11,SE1,3-6,IE2-4,HT1-3
CAPEC 28, 31, 48, 126, 162, 165, 213, 220, 221,261
SAFECODE 24, 35



5

Jee can bypass the centralized encoding routines since they are not being used comprehensively, or the wrong encodings are being used for the context

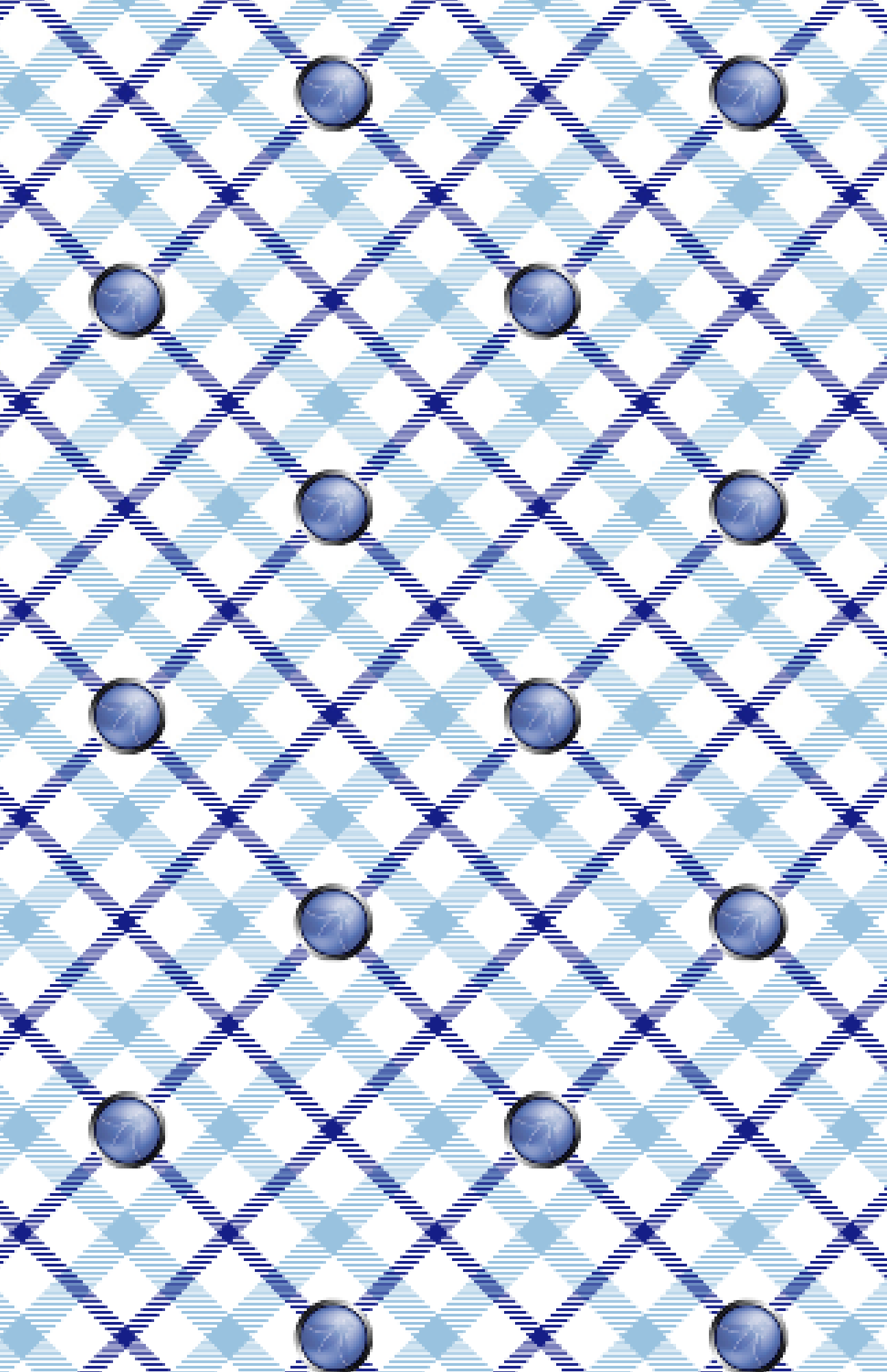
OWASP SCP 3, 15, 18, 19, 168
OWASP ASVS 6.9
OWASP AppSensor -
CAPEC 28, 31, 152, 160, 468
SAFECODE 2, 17



6

Jason can bypass the centralized validation routines since they are not being used comprehensively on all inputs

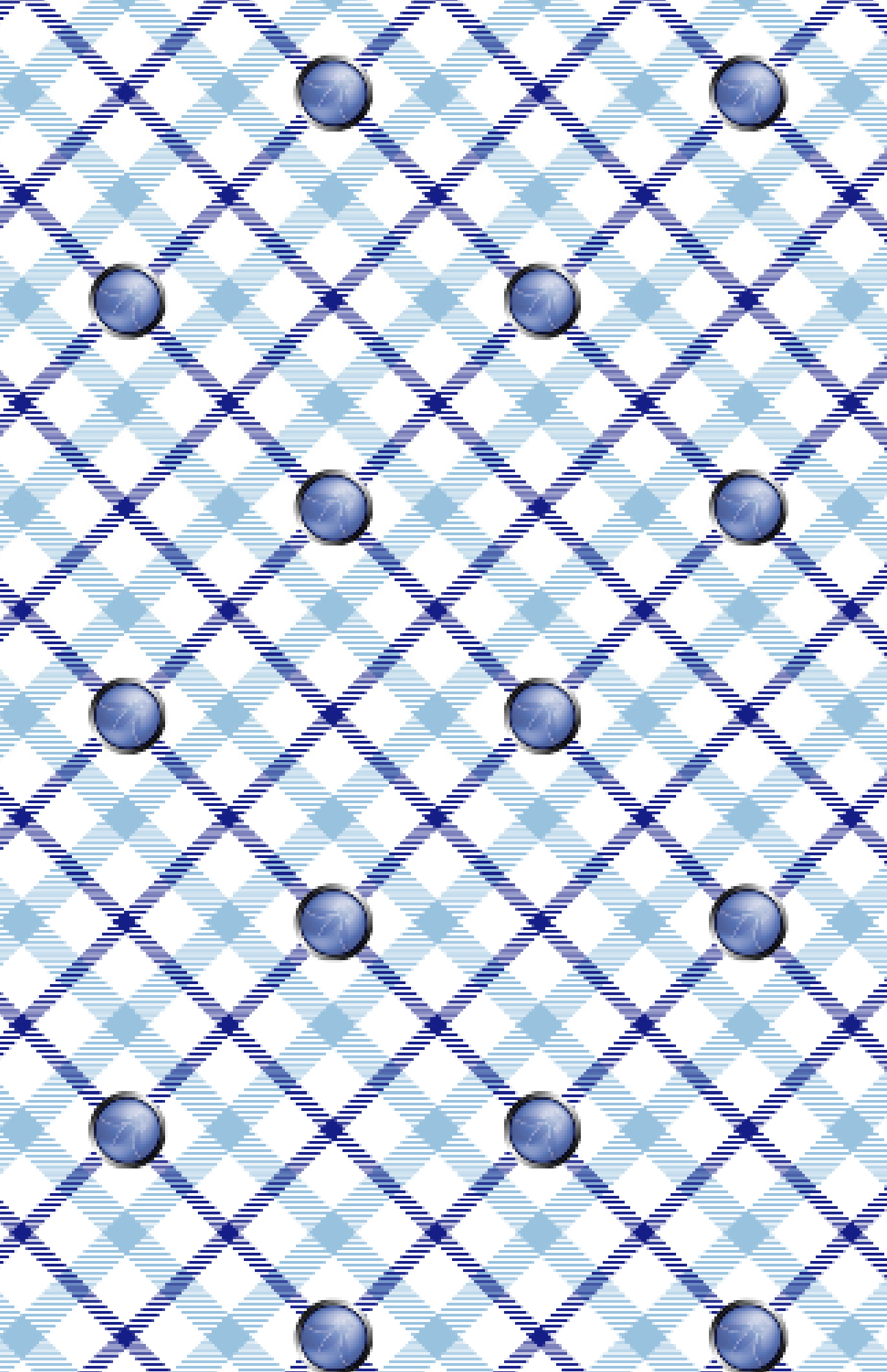
OWASP SCP 3, 168
OWASP ASVS 5.2, 5.6, 6.9
OWASP AppSensor IE2-3
CAPEC 28
SAFECODE 3, 16, 24



7

Jan can craft special payloads to foil input validation because the character set is not specified/enforced, or the data is encoded multiple times, or the data is not fully converted into the same format the application uses (e.g. canonicalization) before being validated, or variables are not strongly typed

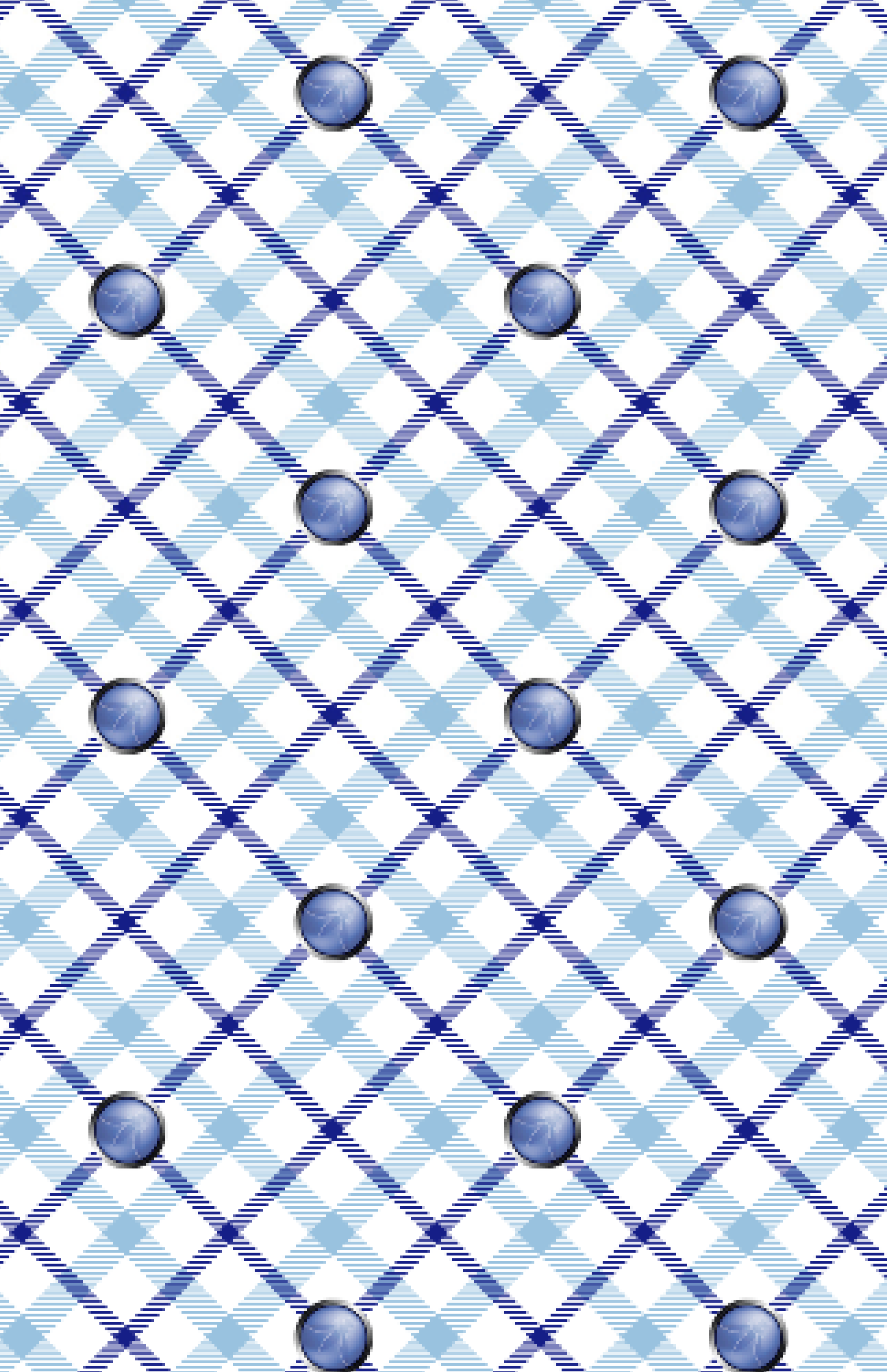
OWASP SCP 4, 5, 7, 150
OWASP ASVS 5.4, 5.8, 10.9
OWASP AppSensor IE2-3, EE1-2
CAPEC 28, 153, 165
SAFECODE 3, 16, 24



8

Sarah can bypass the centralized sanitization routines since they are not being used comprehensively

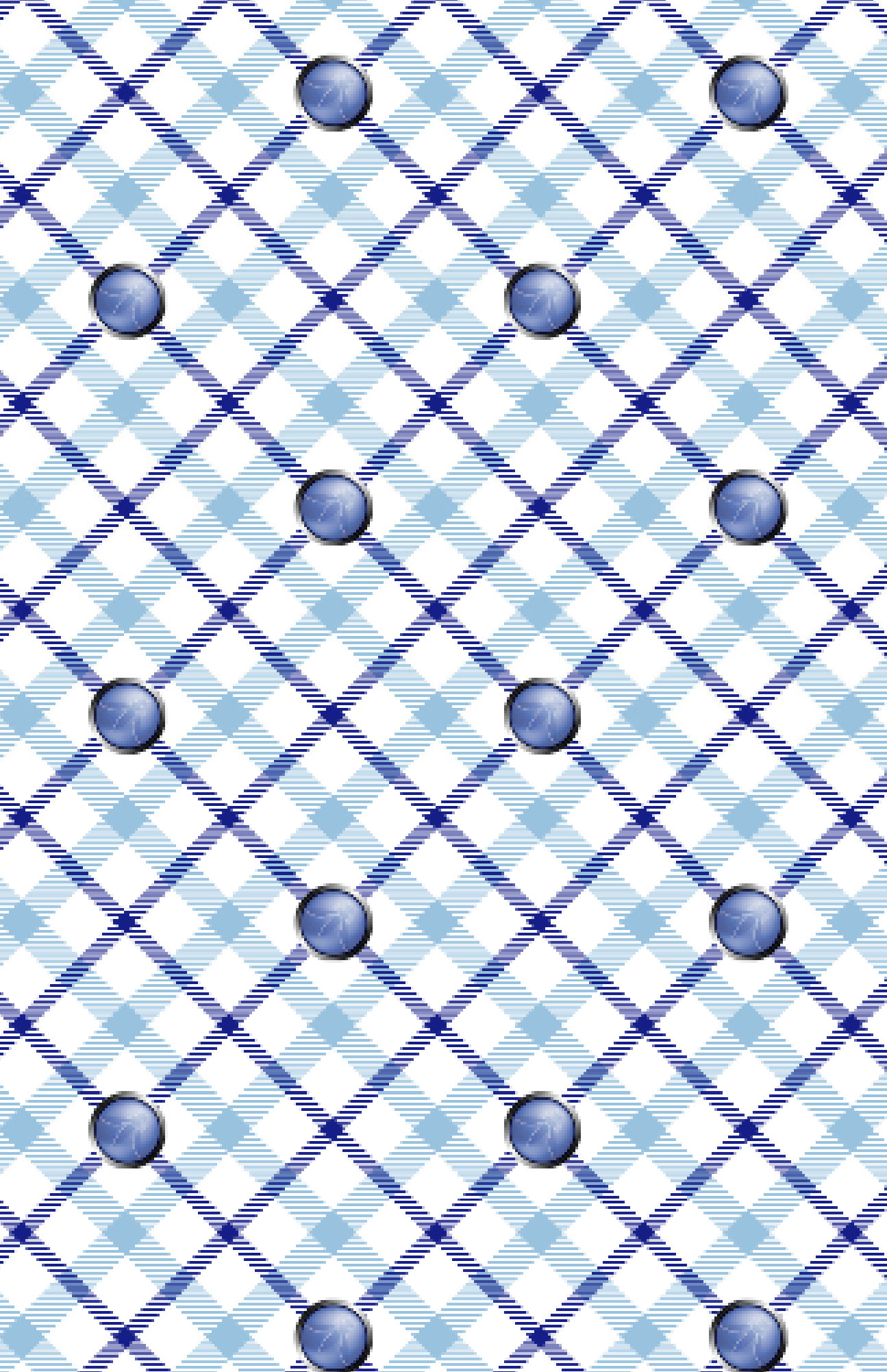
OWASP SCP 15, 169
OWASP ASVS 6.9, 8.7
OWASP AppSensor -
CAPEC 28, 31, 152, 160, 468
SAFECODE 2, 17



9

Shamun can bypass input validation or output validation checks because validation failures are not rejected or sanitized

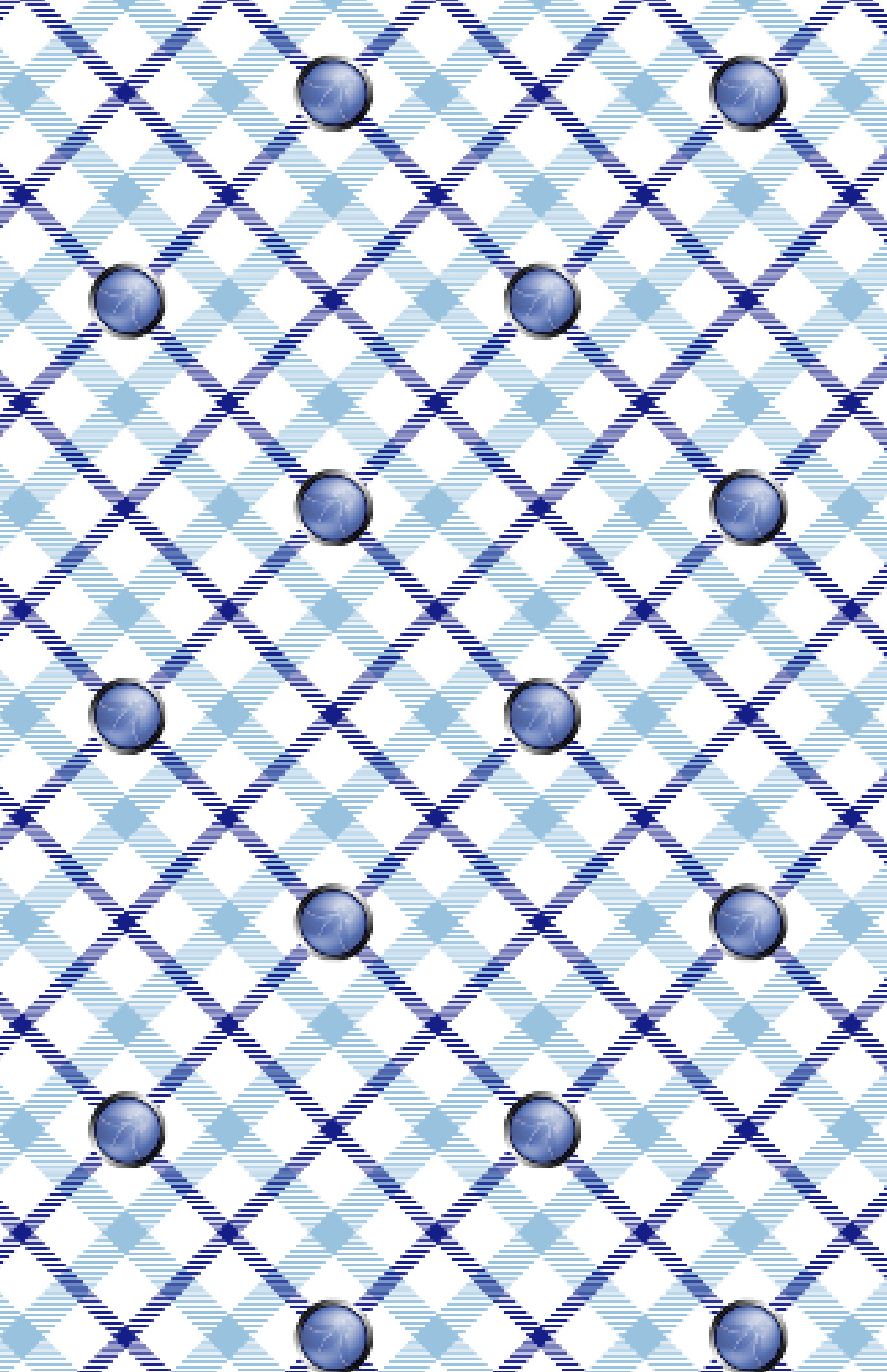
OWASP SCP 6, 168
OWASP ASVS 5.3
OWASP AppSensor IE2-3
CAPEC 28
SAFECODE 3, 16, 24



10

Jerry can exploit the trust the application places in a source of data (e.g. user-definable data, manipulation of locally stored data, alteration to state data on a client device, lack of verification of identity such as Jerry can pretend to be Colin)

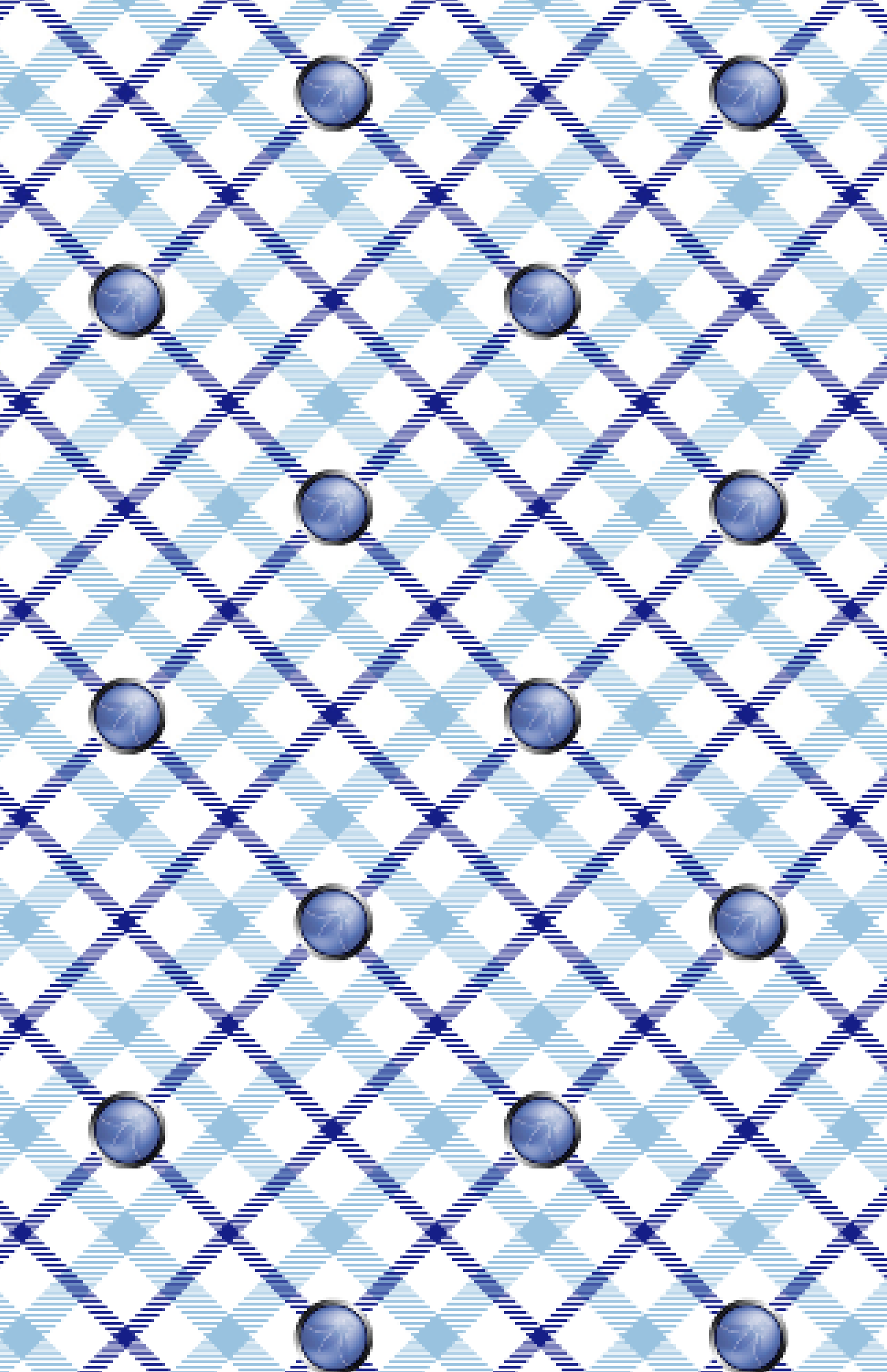
OWASP SCP 2, 19, 92, 95, 180
OWASP ASVS 10.6
OWASP AppSensor IE4, IE5
CAPEC 12, 51, 57, 90,111,145,194,195,202,218,463
SAFECODE 14



J

Dennis has control over input validation, output validation or output encoding code/routines so they can be bypassed

OWASP SCP 1, 17
OWASP ASVS 5.5, 6.2
OWASP AppSensor RE3, RE4
CAPEC 56, 87, 207
SAFECODE 2, 17

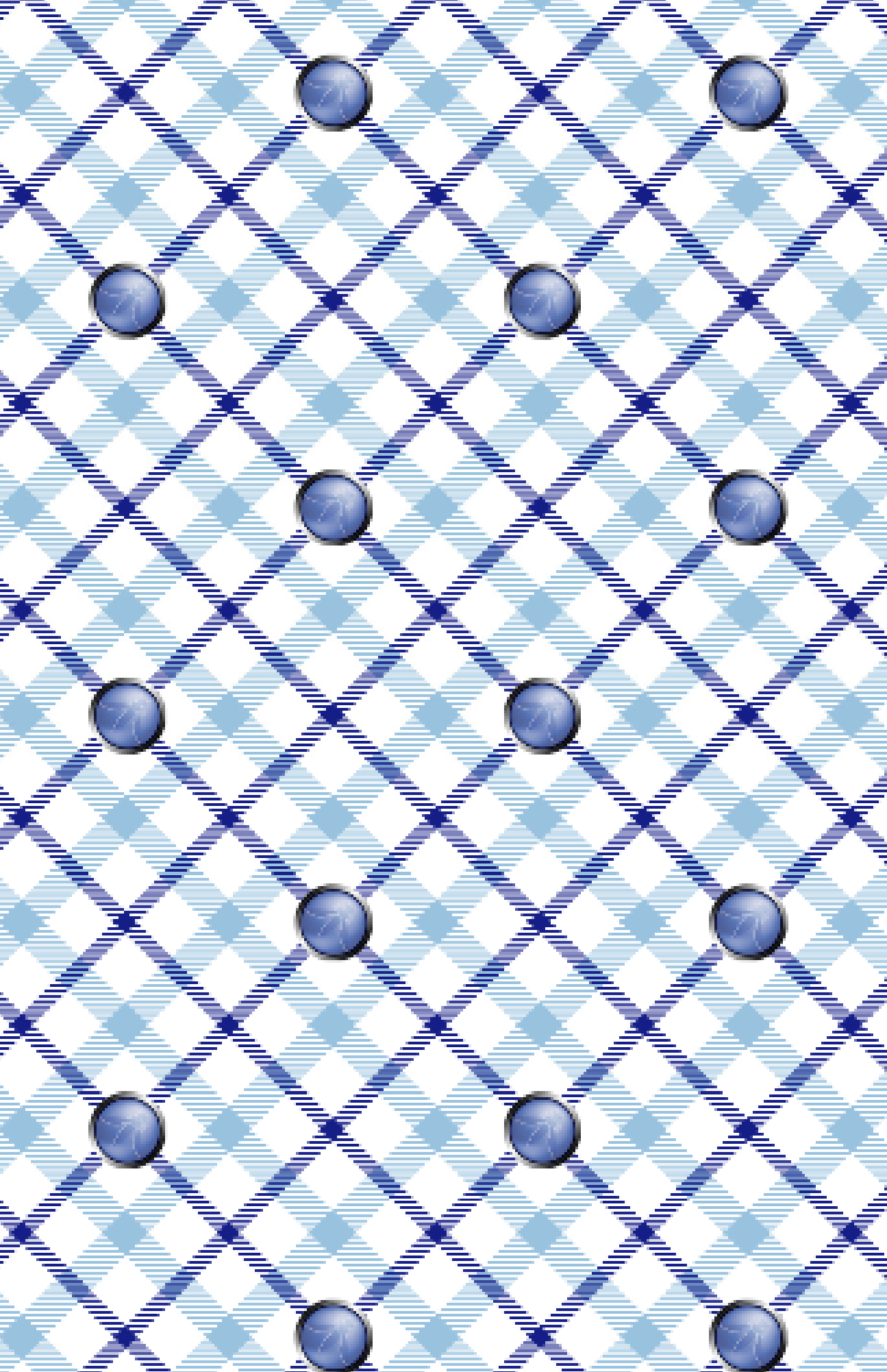


Data Validation & Encoding

Q

Geoff can inject data into a client or device interpreter because a parameterised interface is not being used, or has not been implemented correctly, or the data has not been encoded correctly for the context, or there is no restrictive policy on code or data includes

OWASP SCP 10, 15, 16, 19, 20
OWASP ASVS 6.1, 6.3, 6.8
OWASP AppSensor IE1, RP3
CAPEC 28, 31, 152, 160, 468
SAFECode 2, 17



Data Validation & Encoding

K

Gabe can inject data into an server-side interpreter (e.g. SQL, OS commands, Xpath, Server JavaScript, SMTP) because a strongly typed parameterised interface is not being used or has not been implemented correctly

OWASP SCP
15, 19-22, 167, 180, 203, 210, 211

OWASP ASVS
6.3, 6.4, 6.5, 6.6, 6.7, 6.8

OWASP AppSensor
CIE1-2

CAPEC
23, 28, 76, 152, 160, 261

SAFECode
2, 19, 20