Sparrow 분석 보고서

Sparrow Cloud

분석 요약 정보

프로젝트 이름	Demo
분석 ID	723
분석 유형	파일
분석 시작 일시	2025-06-11 10:07:09
분석 완료 일시	2025-06-11 10:13:52
분석 시간	6분 42초
총 이슈 수	273
출력된 이슈 수	273

위험도별 이슈 수

매우 높음	높음	보통	낮음	매우 낮음
19	94	157	3	0

■ 레퍼런스별 이슈 수

레퍼런스 이름	총 이슈 수
.NET framework design guideline	0
CWE 658 4.14	0
CWE 658 4.7	0
CWE 659 4.14	0
CWE 659 4.7	0
CWE 660 4.14	125
CWE 660 4.7	68
Code conventions for the Java Programming Language(Oracle)	0
JavaScript 시큐어코딩 가이드 2022	0
MISRA-C 2004	0
MISRA-C 2012	0
MISRA-C 2012 Amendment 2	0
MISRA-C 2012 Amendment 3	0
MISRA-C++ 2008	0
OWASP 2017	7
OWASP 2021	29
Python 시큐어코딩 가이드 2022	0
Rust ANSSI guide v1.0	0
무기체계 소프트웨어 보안약점 점검 목록	227
방위사업청 코딩규칙	0
소프트웨어 보안약점 진단가이드 2021	236
주요정보통신기반시설 취약점 분석·평가 기준	0

● .NET framework design guideline

레퍼런스 항목 이름	이슈 수
System.Xml 사용법	0
구조체 디자인	0
네임스페이스의 이름	0
리소스 이름 지정	0
매개변수 이름 지정	0
멤버 오버로드	0
보호된 멤버	0

봉인	0
예외 throw	0
예외 및 성능	0
인터페이스 디자인	0
일반 명명 규칙	0
컬렉션	0
클래스와 구조체 간의 선택	0
표준 예외 형식 사용	0

CWE 658 4.14

레퍼런스 항목 이름	이슈 수
119 - Improper Restriction of Operations within the Bounds of a Memory Buffer	0
120 - Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	0
121 - Stack-based Buffer Overflow	0
122 - Heap-based Buffer Overflow	0
123 - Write-what-where Condition	0
124 - Buffer Underwrite ('Buffer Underflow')	0
125 - Out-of-bounds Read	0
126 - Buffer Over-read	0
127 - Buffer Under-read	0
128 - Wrap-around Error	0
129 - Improper Validation of Array Index	0
131 - Incorrect Calculation of Buffer Size	0
1325 - Improperly Controlled Sequential Memory Allocation	0
1335 - Incorrect Bitwise Shift of Integer	0
134 - Use of Externally-Controlled Format String	0
1341 - Multiple Releases of Same Resource or Handle	0
135 - Incorrect Calculation of Multi-Byte String Length	0
14 - Compiler Removal of Code to Clear Buffers	0
170 - Improper Null Termination	0
188 - Reliance on Data/Memory Layout	0
191 - Integer Underflow (Wrap or Wraparound)	0
192 - Integer Coercion Error	0
194 - Unexpected Sign Extension	0
195 - Signed to Unsigned Conversion Error	0

196 - Unsigned to Signed Conversion Error	0
197 - Numeric Truncation Error	0
242 - Use of Inherently Dangerous Function	0
243 - Creation of chroot Jail Without Changing Working Directory	0
244 - Improper Clearing of Heap Memory Before Release ('Heap Inspection')	0
362 - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	0
364 - Signal Handler Race Condition	0
366 - Race Condition within a Thread	0
375 - Returning a Mutable Object to an Untrusted Caller	0
401 - Missing Release of Memory after Effective Lifetime	0
415 - Double Free	0
416 - Use After Free	0
457 - Use of Uninitialized Variable	0
462 - Duplicate Key in Associative List (Alist)	0
463 - Deletion of Data Structure Sentinel	0
464 - Addition of Data Structure Sentinel	0
467 - Use of sizeof() on a Pointer Type	0
468 - Incorrect Pointer Scaling	0
469 - Use of Pointer Subtraction to Determine Size	0
476 - NULL Pointer Dereference	0
478 - Missing Default Case in Multiple Condition Expression	0
479 - Signal Handler Use of a Non-reentrant Function	0
480 - Use of Incorrect Operator	0
481 - Assigning instead of Comparing	0
482 - Comparing instead of Assigning	0
483 - Incorrect Block Delimitation	0
484 - Omitted Break Statement in Switch	0
558 - Use of getlogin() in Multithreaded Application	0
560 - Use of umask() with chmod-style Argument	0
562 - Return of Stack Variable Address	0
587 - Assignment of a Fixed Address to a Pointer	0
676 - Use of Potentially Dangerous Function	0
685 - Function Call With Incorrect Number of Arguments	0
690 - Unchecked Return Value to NULL Pointer Dereference	0
704 - Incorrect Type Conversion or Cast	0

733 - Compiler Optimization Removal or Modification of Security-critical Code	0
762 - Mismatched Memory Management Routines	0
783 - Operator Precedence Logic Error	0
785 - Use of Path Manipulation Function without Maximum-sized Buffer	0
787 - Out-of-bounds Write	0
789 - Memory Allocation with Excessive Size Value	0
805 - Buffer Access with Incorrect Length Value	0
806 - Buffer Access Using Size of Source Buffer	0
839 - Numeric Range Comparison Without Minimum Check	0
843 - Access of Resource Using Incompatible Type ('Type Confusion')	0
910 - Use of Expired File Descriptor	0

CWE 658 4.7

레퍼런스 항목 이름	이슈 수
Access of Resource Using Incompatible Type ('Type Confusion') - (843)	0
Addition of Data Structure Sentinel - (464)	0
Assigning instead of Comparing - (481)	0
Assignment of a Fixed Address to a Pointer - (587)	0
Buffer Access Using Size of Source Buffer - (806)	0
Buffer Access with Incorrect Length Value - (805)	0
Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') - (120)	0
Buffer Over-read - (126)	0
Buffer Under-read - (127)	0
Buffer Underwrite ('Buffer Underflow') - (124)	0
Comparing instead of Assigning - (482)	0
Compiler Optimization Removal or Modification of Security-critical Code - (733)	0
Compiler Removal of Code to Clear Buffers - (14)	0
Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') - (362)	0
Creation of chroot Jail Without Changing Working Directory - (243)	0
Deletion of Data Structure Sentinel - (463)	0
Double Free - (415)	0
Duplicate Key in Associative List (Alist) - (462)	0
Function Call With Incorrect Number of Arguments - (685)	0
Function Call With Incorrect Variable or Reference as Argument - (688)	0

Heap-based Buffer Overflow - (122)	0
Improper Cleanup on Thrown Exception - (460)	0
Improper Clearing of Heap Memory Before Release ('Heap Inspection') - (244)	0
Improper Handling of Length Parameter Inconsistency - (130)	0
Improper Null Termination - (170)	0
Improper Restriction of Operations within the Bounds of a Memory Buffer - (119)	0
Improper Update of Reference Count - (911)	0
Improper Validation of Array Index - (129)	0
Incorrect Block Delimitation - (483)	0
Incorrect Calculation of Buffer Size - (131)	0
Incorrect Calculation of Multi-Byte String Length - (135)	0
Incorrect Pointer Scaling - (468)	0
Incorrect Type Conversion or Cast - (704)	0
Integer Coercion Error - (192)	0
Integer Underflow (Wrap or Wraparound) - (191)	0
Mismatched Memory Management Routines - (762)	0
Missing Default Case in Switch Statement - (478)	0
NULL Pointer Dereference - (476)	0
Numeric Range Comparison Without Minimum Check - (839)	0
Numeric Truncation Error - (197)	0
Omitted Break Statement in Switch - (484)	0
Operator Precedence Logic Error - (783)	0
Out-of-bounds Read - (125)	0
Out-of-bounds Write - (787)	0
Race Condition within a Thread - (366)	0
Reliance on Data/Memory Layout - (188)	0
Return of Pointer Value Outside of Expected Range - (466)	0
Return of Stack Variable Address - (562)	0
Signal Handler Race Condition - (364)	0
Signal Handler Use of a Non-reentrant Function - (479)	0
Signed to Unsigned Conversion Error - (195)	0
Stack-based Buffer Overflow - (121)	0
Unexpected Sign Extension - (194)	0
Unsigned to Signed Conversion Error - (196)	0
Use After Free - (416)	0
Use of Expired File Descriptor - (910)	0

Use of Externally-Controlled Format String - (134)	0
Use of Incorrect Operator - (480)	0
Use of Inherently Dangerous Function - (242)	0
Use of Pointer Subtraction to Determine Size - (469)	0
Use of Potentially Dangerous Function - (676)	0
Use of Uninitialized Variable - (457)	0
Use of getlogin() in Multithreaded Application - (558)	0
Use of sizeof() on a Pointer Type - (467)	0
Use of umask() with chmod-style Argument - (560)	0
Wrap-around Error - (128)	0
Write-what-where Condition - (123)	0

CWE 659 4.14

레퍼런스 항목 이름	이슈 수
119 - Improper Restriction of Operations within the Bounds of a Memory Buffer	0
120 - Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	0
121 - Stack-based Buffer Overflow	0
122 - Heap-based Buffer Overflow	0
123 - Write-what-where Condition	0
124 - Buffer Underwrite ('Buffer Underflow')	0
125 - Out-of-bounds Read	0
126 - Buffer Over-read	0
127 - Buffer Under-read	0
128 - Wrap-around Error	0
129 - Improper Validation of Array Index	0
130 - Improper Handling of Length Parameter Inconsistency	0
131 - Incorrect Calculation of Buffer Size	0
1325 - Improperly Controlled Sequential Memory Allocation	0
1335 - Incorrect Bitwise Shift of Integer	0
134 - Use of Externally-Controlled Format String	0
1341 - Multiple Releases of Same Resource or Handle	0
135 - Incorrect Calculation of Multi-Byte String Length	0
14 - Compiler Removal of Code to Clear Buffers	0
170 - Improper Null Termination	0
188 - Reliance on Data/Memory Layout	0

191 - Integer Underflow (Wrap or Wraparound)	0
192 - Integer Coercion Error	0
194 - Unexpected Sign Extension	0
195 - Signed to Unsigned Conversion Error	0
196 - Unsigned to Signed Conversion Error	0
197 - Numeric Truncation Error	0
242 - Use of Inherently Dangerous Function	0
243 - Creation of chroot Jail Without Changing Working Directory	0
244 - Improper Clearing of Heap Memory Before Release ('Heap Inspection')	0
248 - Uncaught Exception	0
362 - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	0
364 - Signal Handler Race Condition	0
366 - Race Condition within a Thread	0
374 - Passing Mutable Objects to an Untrusted Method	0
375 - Returning a Mutable Object to an Untrusted Caller	0
396 - Declaration of Catch for Generic Exception	0
397 - Declaration of Throws for Generic Exception	0
401 - Missing Release of Memory after Effective Lifetime	0
415 - Double Free	0
416 - Use After Free	0
457 - Use of Uninitialized Variable	0
462 - Duplicate Key in Associative List (Alist)	0
463 - Deletion of Data Structure Sentinel	0
464 - Addition of Data Structure Sentinel	0
467 - Use of sizeof() on a Pointer Type	0
468 - Incorrect Pointer Scaling	0
469 - Use of Pointer Subtraction to Determine Size	0
476 - NULL Pointer Dereference	0
478 - Missing Default Case in Multiple Condition Expression	0
479 - Signal Handler Use of a Non-reentrant Function	0
480 - Use of Incorrect Operator	0
481 - Assigning instead of Comparing	0
482 - Comparing instead of Assigning	0
483 - Incorrect Block Delimitation	0
484 - Omitted Break Statement in Switch	0

493 - Critical Public Variable Without Final Modifier	0
495 - Private Data Structure Returned From A Public Method	0
496 - Public Data Assigned to Private Array-Typed Field	0
498 - Cloneable Class Containing Sensitive Information	0
500 - Public Static Field Not Marked Final	0
543 - Use of Singleton Pattern Without Synchronization in a Multithreaded Context	0
558 - Use of getlogin() in Multithreaded Application	0
562 - Return of Stack Variable Address	0
587 - Assignment of a Fixed Address to a Pointer	0
676 - Use of Potentially Dangerous Function	0
690 - Unchecked Return Value to NULL Pointer Dereference	0
704 - Incorrect Type Conversion or Cast	0
733 - Compiler Optimization Removal or Modification of Security-critical Code	0
762 - Mismatched Memory Management Routines	0
766 - Critical Data Element Declared Public	0
767 - Access to Critical Private Variable via Public Method	0
783 - Operator Precedence Logic Error	0
785 - Use of Path Manipulation Function without Maximum-sized Buffer	0
787 - Out-of-bounds Write	0
789 - Memory Allocation with Excessive Size Value	0
805 - Buffer Access with Incorrect Length Value	0
806 - Buffer Access Using Size of Source Buffer	0
839 - Numeric Range Comparison Without Minimum Check	0
843 - Access of Resource Using Incompatible Type ('Type Confusion')	0
910 - Use of Expired File Descriptor	0

CWE 659 4.7

레퍼런스 항목 이름	이슈 수
Access of Resource Using Incompatible Type ('Type Confusion') - (843)	0
Access to Critical Private Variable via Public Method - (767)	0
Addition of Data Structure Sentinel - (464)	0
Assigning instead of Comparing - (481)	0
Assignment of a Fixed Address to a Pointer - (587)	0
Buffer Access Using Size of Source Buffer - (806)	0
Buffer Access with Incorrect Length Value - (805)	0

Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') - (120)	0
Buffer Over-read - (126)	0
Buffer Under-read - (127)	0
Buffer Underwrite ('Buffer Underflow') - (124)	0
Comparing instead of Assigning - (482)	0
Compiler Optimization Removal or Modification of Security-critical Code - (733)	0
Compiler Removal of Code to Clear Buffers - (14)	0
Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') - (362)	0
Creation of chroot Jail Without Changing Working Directory - (243)	0
Critical Public Variable Without Final Modifier - (493)	0
Declaration of Catch for Generic Exception - (396)	0
Declaration of Throws for Generic Exception - (397)	0
Deletion of Data Structure Sentinel - (463)	0
Double Free - (415)	0
Duplicate Key in Associative List (Alist) - (462)	0
Heap-based Buffer Overflow - (122)	0
Improper Cleanup on Thrown Exception - (460)	0
Improper Clearing of Heap Memory Before Release ('Heap Inspection') - (244)	0
Improper Handling of Length Parameter Inconsistency - (130)	0
Improper Null Termination - (170)	0
Improper Restriction of Operations within the Bounds of a Memory Buffer - (119)	0
Improper Update of Reference Count - (911)	0
Improper Validation of Array Index - (129)	0
Incorrect Block Delimitation - (483)	0
Incorrect Calculation of Buffer Size - (131)	0
Incorrect Calculation of Multi-Byte String Length - (135)	0
Incorrect Pointer Scaling - (468)	0
Incorrect Type Conversion or Cast - (704)	0
Integer Coercion Error - (192)	0
Integer Underflow (Wrap or Wraparound) - (191)	0
Mismatched Memory Management Routines - (762)	0
Missing Default Case in Switch Statement - (478)	0
NULL Pointer Dereference - (476)	0
Numeric Range Comparison Without Minimum Check - (839)	0
Numeric Truncation Error - (197)	0

Omitted Break Statement in Switch - (484)	0
Operator Precedence Logic Error - (783)	0
Out-of-bounds Read - (125)	0
Out-of-bounds Write - (787)	0
Passing Mutable Objects to an Untrusted Method - (374)	0
Public Data Assigned to Private Array-Typed Field - (496)	0
Race Condition within a Thread - (366)	0
Reliance on Data/Memory Layout - (188)	0
Return of Pointer Value Outside of Expected Range - (466)	0
Return of Stack Variable Address - (562)	0
Returning a Mutable Object to an Untrusted Caller - (375)	0
Signal Handler Race Condition - (364)	0
Signal Handler Use of a Non-reentrant Function - (479)	0
Signed to Unsigned Conversion Error - (195)	0
Stack-based Buffer Overflow - (121)	0
Uncaught Exception - (248)	0
Unexpected Sign Extension - (194)	0
Unsigned to Signed Conversion Error - (196)	0
Use After Free - (416)	0
Use of Expired File Descriptor - (910)	0
Use of Externally-Controlled Format String - (134)	0
Use of Incorrect Operator - (480)	0
Use of Inherently Dangerous Function - (242)	0
Use of Pointer Subtraction to Determine Size - (469)	0
Use of Potentially Dangerous Function - (676)	0
Use of Uninitialized Variable - (457)	0
Use of getlogin() in Multithreaded Application - (558)	0
Use of sizeof() on a Pointer Type - (467)	0
Wrap-around Error - (128)	0
Write-what-where Condition - (123)	0

CWE 660 4.14

레퍼런스 항목 이름	이슈 수
102 - Struts: Duplicate Validation Forms	0
103 - Struts: Incomplete validate() Method Definition	0

104 - Struts: Form Bean Does Not Extend Validation Class	0
106 - Struts: Plug-in Framework not in Use	0
109 - Struts: Validator Turned Off	0
110 - Struts: Validator Without Form Field	0
111 - Direct Use of Unsafe JNI	0
1235 - Incorrect Use of Autoboxing and Unboxing for Performance Critical Operations	0
1335 - Incorrect Bitwise Shift of Integer	0
1336 - Improper Neutralization of Special Elements Used in a Template Engine	0
1341 - Multiple Releases of Same Resource or Handle	1
191 - Integer Underflow (Wrap or Wraparound)	1
192 - Integer Coercion Error	0
197 - Numeric Truncation Error	0
209 - Generation of Error Message Containing Sensitive Information	53
245 - J2EE Bad Practices: Direct Management of Connections	0
246 - J2EE Bad Practices: Direct Use of Sockets	0
248 - Uncaught Exception	0
362 - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	1
366 - Race Condition within a Thread	2
374 - Passing Mutable Objects to an Untrusted Method	0
375 - Returning a Mutable Object to an Untrusted Caller	0
382 - J2EE Bad Practices: Use of System.exit()	0
383 - J2EE Bad Practices: Direct Use of Threads	0
396 - Declaration of Catch for Generic Exception	0
397 - Declaration of Throws for Generic Exception	20
460 - Improper Cleanup on Thrown Exception	7
470 - Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection')	0
476 - NULL Pointer Dereference	34
478 - Missing Default Case in Multiple Condition Expression	0
481 - Assigning instead of Comparing	0
484 - Omitted Break Statement in Switch	0
486 - Comparison of Classes by Name	0
487 - Reliance on Package-level Scope	0
491 - Public cloneable() Method Without Final ('Object Hijack')	0

492 - Use of Inner Class Containing Sensitive Data	0
493 - Critical Public Variable Without Final Modifier	3
495 - Private Data Structure Returned From A Public Method	0
496 - Public Data Assigned to Private Array-Typed Field	0
498 - Cloneable Class Containing Sensitive Information	0
500 - Public Static Field Not Marked Final	3
502 - Deserialization of Untrusted Data	0
537 - Java Runtime Error Message Containing Sensitive Information	0
567 - Unsynchronized Access to Shared Data in a Multithreaded Context	0
568 - finalize() Method Without super.finalize()	0
572 - Call to Thread run() instead of start()	0
574 - EJB Bad Practices: Use of Synchronization Primitives	0
575 - EJB Bad Practices: Use of AWT Swing	0
576 - EJB Bad Practices: Use of Java I/O	0
577 - EJB Bad Practices: Use of Sockets	0
578 - EJB Bad Practices: Use of Class Loader	0
579 - J2EE Bad Practices: Non-serializable Object Stored in Session	0
580 - clone() Method Without super.clone()	0
581 - Object Model Violation: Just One of Equals and Hashcode Defined	0
582 - Array Declared Public, Final, and Static	0
583 - finalize() Method Declared Public	0
594 - J2EE Framework: Saving Unserializable Objects to Disk	0
595 - Comparison of Object References Instead of Object Contents	0
607 - Public Static Final Field References Mutable Object	0
608 - Struts: Non-private Field in ActionForm Class	0
609 - Double-Checked Locking	0
7 - J2EE Misconfiguration: Missing Custom Error Page	0
766 - Critical Data Element Declared Public	0
917 - Improper Neutralization of Special Elements used in an Expression Language Statement ('Expression Language Injection')	0
95 - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')	0

CWE 660 4.7

레퍼런스 항목 이름	이슈 수

Array Declared Public, Final, and Static - (582)	0
Assigning instead of Comparing - (481)	0
Call to Thread run() instead of start() - (572)	0
Cloneable Class Containing Sensitive Information - (498)	0
Comparison of Classes by Name - (486)	0
Comparison of Object References Instead of Object Contents - (595)	0
Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') - (362)	1
Critical Public Variable Without Final Modifier - (493)	3
Declaration of Catch for Generic Exception - (396)	0
Declaration of Throws for Generic Exception - (397)	20
Deserialization of Untrusted Data - (502)	0
Direct Use of Unsafe JNI - (111)	0
Double-Checked Locking - (609)	0
EJB Bad Practices: Use of AWT Swing - (575)	0
EJB Bad Practices: Use of Java I/O - (576)	0
EJB Bad Practices: Use of Sockets - (577)	0
Finalize() Method Without super.finalize() - (568)	0
Improper Cleanup on Thrown Exception - (460)	7
Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection') - (95)	0
J2EE Bad Practices: Direct Management of Connections - (245)	0
J2EE Bad Practices: Direct Use of Sockets - (246)	0
J2EE Bad Practices: Direct Use of Threads - (383)	0
J2EE Bad Practices: Use of System.exit() - (382)	0
NULL Pointer Dereference - (476)	34
Numeric Truncation Error - (197)	0
Object Model Violation: Just One of Equals and Hashcode Defined - (581)	0
Omitted Break Statement in Switch - (484)	0
Passing Mutable Objects to an Untrusted Method - (374)	0
Public Data Assigned to Private Array-Typed Field - (496)	0
Public Static Field Not Marked Final - (500)	3
Public Static Final Field References Mutable Object - (607)	0
Reliance on Package-level Scope - (487)	0
Returning a Mutable Object to an Untrusted Caller - (375)	0
Uncaught Exception - (248)	0

Use of Inner Class Containing Sensitive Data - (492)	0

Code conventions for the Java Programming Language(Oracle)

레퍼런스 항목 이름	이슈 수
04.1 Line Length	0
04.2 Wrapping Lines	0
05.1.1 Block Comments	0
05.1.2 Single-Line Comments	0
05.1.3 Trailing Comments	0
05.1.4 End-Of-Line Comments	0
05.2 Documentation Comments	0
06.1 Number Per Line	0
06.2 Initialization	0
06.3 Placement	0
06.4 Class and Interface Declarations	0
07.1 Simple Statements	0
07.2 Compound Statements	0
07.3 return Statements	0
07.4 if, if-else, if else-if else Statements	0
07.5 for Statements	0
07.6 while Statements	0
07.7 do-while Statements	0
07.8 switch Statements	0
07.9 try-catch Statements	0
08.1 Blank Lines	0
08.2 Blank Spaces	0
09.1 Package	0
09.2 Classes or Interface	0
09.3 Methods	0
09.4 Variables	0
09.5 Constants	0
10.1 Providing Access to Instance and Class Variables	0
10.2 Refferring to Class Variables and Methods	0
10.3 Constants	0
10.4 Variable Assignments	0

10.5.1 Parentheses	0
10.5.2 Returning Values	0
10.5.3 Expressions before '?' in the Conditional Operator	0

JavaScript 시큐어코딩 가이드 2022

레퍼런스 항목 이름	이슈 수
01.01. SQL 삽입	0
01.02. 코드 삽입	0
01.03. 경로 조작 및 자원 삽입	0
01.04. 크로스사이트 스크립트(XSS)	0
01.05. 운영체제 명령어 삽입	0
01.08. 부적절한 XML 외부 개체 참조	0
01.11. 크로스사이트 요청 위조(CSRF)	0
02.04. 취약한 암호화 알고리즘 사용	0
02.07. 충분하지 않은 키 길이 사용	0
02.08. 적절하지 않은 난수 값 사용	0
02.14. 솔트 없이 일방향 해쉬 함수 사용	0
03.01. 종료되지 않는 반복문 또는 재귀 함수	0
04.01. 오류 메시지 정보 노출	0
06.02. 제거되지 않고 남은 디버그 코드	0

MISRA-C 2004

레퍼런스 항목 이름	이슈 수
1.02 (Required) : No reliance shall be placed on undefined or unspecified behaviour.	0
1.04 (Required) : The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.	0
10.03 (Required) : The value of a complex expression of integer type may only be cast to a type that is narrower and of the same signedness as the underlying type of the expression.	0
10.04 (Required) : The value of a complex expression of floating type may only be cast to a narrower floating type.	0
10.05 (Required) : If the bitwise operators ~ and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand.	0
10.06 (Required) : A "U" suffix shall be applied to all constants of unsigned type.	0

11.01 (Required) : Conversions shall not be performed between a pointer to a function and any type other than an integral type.	0
11.02 (Required) : Conversions shall not be performed between a pointer to object and any type other than an integral type, another pointer to object type or a pointer to void.	0
11.03 (Advisory) : A cast should not be performed between a pointer type and an integral type.	0
11.04 (Advisory) : A cast should not be performed between a pointer to object type and a different pointer to object type.	0
11.05 (Required) : A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer.	0
12.01 (Advisory) : Limited dependence should be placed on C's operator precedence rules in expressions.	0
12.02 (Required) : The value of an expression shall be the same under any order of evaluation that the standard permits.	0
12.03 (Required) : The sizeof operator shall not be used on expressions that contain side effects.	0
12.04 (Required) : The right hand operand of a logical && or || operator shall not contain side effects.	0
12.05 (Required) : The operands of a logical && or || shall be primary-expressions.	0
12.06 (Advisory) : The operands of logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !).	0
12.07 (Required) : Bitwise operators shall not be applied to operands whose underlying type is signed.	0
12.08 (Required) : The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand.	0
12.09 (Required) : The unary minus operator shall not be applied to an expression whose underlying type is unsigned.	0
12.10 (Required) : The comma operator shall not be used.	0
12.11 (Advisory) : Evaluation of constant unsigned integer expressions should not lead to wrap-around.	0
12.12 (Required) : The underlying bit representations of floating-point values shall not be used.	0
12.13 (Advisory) : The increment (++) and decrement (--) operators should not be mixed with other operators in an expression.	0
13.01 (Required) : Assignment operators shall not be used in expressions that yield a Boolean value.	0

13.02 (Advisory) : Tests of a value against zero should be made explicit, unless the operand is effectively Boolean.	0
13.03 (Required) : Floating-point expressions shall not be tested for equality or inequality.	0
13.04 (Required) : The controlling expression of a for statement shall not contain any objects of floating type.	0
13.05 (Required) : The three expressions of a for statement shall be concerned only with loop control.	0
13.06 (Required) : Numeric variables being used within a for loop for iteration counting shall not be modified in the body of the loop.	0
13.07 (Required) : Boolean operations whose results are invariant shall not be permitted.	0
14.01 (Required) : There shall be no unreachable code.	0
14.02 (Required) : All non-null statements shall either : a) have at least one side-effect however executed, or b) cause control flow to change.	0
14.03 (Required) : Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is a white-space character.	0
14.04 (Required) : The goto statement shall not be used.	0
14.05 (Required) : The continue statement shall not be used.	0
14.06 (Required) : For any iteration statement there shall be at most one break statement used for loop termination.	0
14.07 (Required) : A function shall have a single point of exit at the end of the function.	0
14.08 (Required) : The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement.	0
14.09 (Required) : An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement.	0
14.10 (Required) : All if ... else if constructs shall be terminated with an else clause.	0
15.01 (Required) : A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement.	0
15.02 (Required) : An unconditional break statement shall terminate every non-empty switch clause.	0
15.03 (Required) : The final clause of a switch statement shall be the default clause.	0
15.04 (Required) : A switch expression shall not represent a value that is effectively Boolean.	0
15.05 (Required) : Every switch statement shall have at least one case clause.	0

16.01 (Required) : Functions shall not be defined with a variable number of arguments.	0
16.02 (Required) : Functions shall not call themselves, either directly or indirectly.	0
16.03 (Required) : Identifiers shall be given for all of the parameters in a function prototype declaration.	0
16.04 (Required) : The identifiers used in the declaration and definition of a function shall be identical.	0
16.05 (Required) : The identifiers used in the declaration and definition of a function shall be identical.	0
16.06 (Required) : The number of arguments passed to a function shall match the number of parameters.	0
16.07 (Advisory) : A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object.	0
16.08 (Required) : All exit paths from a function with non-void return type shall have an explicit return statement with an expression.	0
16.09 (Required) : A function identifier shall only be used with either a preceding &, or with a parenthesised parameter list, which may be empty.	0
16.10 (Required) : If a function returns error information, then that error information shall be tested.	0
17.01 (Required) : Pointer arithmetic shall only be applied to pointers that address an array or array element.	0
17.02 (Required) : Pointer subtraction shall only be applied to pointers that address elements of the same array.	0
17.03 (Required) : >, >=, <, <= shall not be applied to pointer types except where they point to the same array.	0
17.04 (Required) : Array indexing shall be the only allowed form of pointer arithmetic.	0
17.05 (Advisory) : The declaration of objects should contain no more than 2 levels of pointer indirection.	0
17.06 (Required) : The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist.	0
18.01 (Required) : All structure and union types shall be complete at the end of a translation unit.	0
18.02 (Required) : An object shall not be assigned to an overlapping object.	0
18.04 (Required) : Unions shall not be used.	0
19.01 (Advisory) : #include statements in a file should only be preceded by other preprocessor directives or comments.	0
19.02 (Advisory) : Non-standard characters should not occur in header file names in	

#include directives.	0
19.03 (Required) : The #include directive shall be followed by either a <filename> or "filename" sequence.	0
19.04 (Required) : C macros shall only expand to a braced initialiser, a constant, a parenthesised expression, a type qualifier, a storage class specifier, or a do-while-zero construct.	0
19.05 (Required) : Macros shall not be #define'd or #undef'd within a block.	0
19.06 (Required) : #undef shall not be used.	0
19.07 (Advisory) : A function should be used in preference to a function-like macro.	0
19.08 (Required) : A function-like macro shall not be invoked without all of its arguments.	0
19.09 (Required) : Arguments to a function-like macro shall not contain tokens that look like preprocessing directives.	0
19.10 (Required) : In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##.	0
19.11 (Required) : All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator.	0
19.12 (Required) : There shall be at most one occurrence of the # or ## operators in a single macro definition.	0
19.13 (Advisory) : The # and ## operators should not be used.	0
19.14 (Required) : The defined preprocessor operator shall only be used in one of the two standard forms.	0
19.15 (Required) : Precautions shall be taken in order to prevent the contents of a header file being included twice.	0
19.16 (Required) : Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor.	0
2.01 (Required) : Assembly language shall be encapsulated and isolated.	0
2.02 (Required) : Source code shall only use /* ... */ style comments.	0
2.03 (Required) : The character sequence /* shall not be used within a comment.	0
20.01 (Required) : Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined.	0
20.02 (Required) : The names of standard library macros, objects and functions shall not be reused.	0
20.04 (Required) : Dynamic heap memory allocation shall not be used.	0
20.05 (Required) : The error indicator errno shall not be used.	0

20.06 (Required) : The macro offsetof, in library <stddef.h>, shall not be used.	0
20.07 (Required) : The setjmp macro and the longjmp function shall not be used.	0
20.08 (Required) : The signal handling facilities of <signal.h> shall not be used.	0
20.09 (Required) : The input/output library <stdio.h> shall not be used in production code.	0
20.10 (Required) : The library functions atof, atoi and atol from library <stdlib.h> shall not be used.	0
20.11 (Required) : The library functions abort, exit, getenv and system from library <stdlib.h> shall not be used.	0
20.12 (Required) : The time handling functions of library <time.h> shall not be used.	0
21.1 (Required) : Minimisation of run-time failures shall be ensured by the use of at least one of a) static analysis tools/techniques; b) dynamic analysis tools/techniques; c) explicit coding of checks to handle run-time faults.	0
3.04 (Required) : All uses of the #pragma directive shall be documented and explained.	0
3.05 (Required) : If it is being relied upon, the implementation defined behaviour and packing of bitfields shall be documented.	0
4.01 (Required) : Only those escape sequences that are defined in the ISO C standard shall be used.	0
4.02 (Required) : Trigraphs shall not be used.	0
5.01 (Required) : Identifiers (internal and external) shall not rely on the significance of more than 31 characters.	0
5.02 (Required) : Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier.	0
5.03 (Required) : A typedef name shall be a unique identifier.	0
5.04 (Required) : A tag name shall be a unique identifier.	0
5.05 (Advisory) : No object or function identifier with static storage duration should be reused.	0
5.06 (Advisory) : No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names.	0
5.07 (Advisory) : No identifier name should be reused.	0
6.01 (Required) : The plain char type shall be used only for the storage and use of character values.	0
6.02 (Required) : Signed and unsigned char type shall be used only for the storage and use of numeric values.	0
6.03 (Advisory) : Typedefs that indicate size and signedness should be used in place of the basic types.	0

6.04 (Required) : Bit fields shall only be defined to be of type unsigned int or signed int.	0
6.05 (Required) : Bit fields of type signed int shall be at least 2 bits long.	0
7.01 (Required) : Octal constants (other than zero) and octal escape sequences shall not be used.	0
8.02 (Required) : Whenever an object or function is declared or defined, its type shall be explicitly stated.	0
8.03 (Required) : For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical.	0
8.04 (Required) : If objects or functions are declared more than once their types shall be compatible.	0
8.05 (Required) : There shall be no definitions of objects or functions in a header file.	0
8.06 (Required) : Functions shall be declared at file scope.	0
8.07 (Required) : Objects shall be defined at block scope if they are only accessed from within a single function.	0
8.08 (Required) : An external object or function shall be declared in one and only one file.	0
8.09 (Required) : An identifier with external linkage shall have exactly one external definition.	0
8.10 (Required) : All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.	0
8.11 (Required) : The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage.	0
8.12 (Required) : When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation.	0
9.01 (Required) : All automatic variables shall have been assigned a value before being used.	0
9.02 (Required) : Braces shall be used to indicate and match the structure in the non-zero initialisation of arrays and structures.	0
9.03 (Required) : In an enumerator list, the "=" construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised.	0

● MISRA-C 2012

레퍼런스 항목 이름	이슈 수
Directives 1.1	0

Directives 4.1	0
Directives 4.10	0
Directives 4.12	0
Directives 4.14	0
Directives 4.3	0
Directives 4.4	0
Directives 4.5	0
Directives 4.6	0
Directives 4.7	0
Directives 4.8	0
Directives 4.9	0
Rule 1.1	0
Rule 1.2	0
Rule 1.3	0
Rule 10.1	0
Rule 10.2	0
Rule 10.3	0
Rule 10.4	0
Rule 10.5	0
Rule 10.6	0
Rule 10.7	0
Rule 10.8	0
Rule 11.1	0
Rule 11.2	0
Rule 11.3	0
Rule 11.4	0
Rule 11.5	0
Rule 11.6	0
Rule 11.7	0
Rule 11.8	0
Rule 11.9	0
Rule 12.1	0
Rule 12.2	0
Rule 12.3	0
Rule 12.4	0
Rule 12.5	0

Rule 13.1	0
Rule 13.2	0
Rule 13.3	0
Rule 13.4	0
Rule 13.5	0
Rule 13.6	0
Rule 14.1	0
Rule 14.2	0
Rule 14.3	0
Rule 14.4	0
Rule 15.1	0
Rule 15.2	0
Rule 15.3	0
Rule 15.4	0
Rule 15.5	0
Rule 15.6	0
Rule 15.7	0
Rule 16.1	0
Rule 16.2	0
Rule 16.3	0
Rule 16.4	0
Rule 16.5	0
Rule 16.6	0
Rule 16.7	0
Rule 17.1	0
Rule 17.2	0
Rule 17.3	0
Rule 17.4	0
Rule 17.5	0
Rule 17.6	0
Rule 17.7	0
Rule 17.8	0
Rule 18.1	0
Rule 18.2	0
Rule 18.3	0
Rule 18.4	0

Rule 18.5	0
Rule 18.6	0
Rule 18.7	0
Rule 18.8	0
Rule 19.1	0
Rule 19.2	0
Rule 2.1	0
Rule 2.2	0
Rule 2.3	0
Rule 2.4	0
Rule 2.5	0
Rule 2.6	0
Rule 2.7	0
Rule 20.01	0
Rule 20.02	0
Rule 20.03	0
Rule 20.04	0
Rule 20.05	0
Rule 20.06	0
Rule 20.07	0
Rule 20.08	0
Rule 20.09	0
Rule 20.10	0
Rule 20.11	0
Rule 20.12	0
Rule 20.13	0
Rule 21.01	0
Rule 21.02	0
Rule 21.03	0
Rule 21.04	0
Rule 21.05	0
Rule 21.06	0
Rule 21.07	0
Rule 21.08	0
Rule 21.09	0
Rule 21.10	0

Rule 21.11	0
Rule 21.12	0
Rule 21.16	0
Rule 21.17	0
Rule 21.18	0
Rule 21.21	0
Rule 22.01	0
Rule 22.02	0
Rule 22.03	0
Rule 22.04	0
Rule 22.05	0
Rule 22.06	0
Rule 22.08	0
Rule 3.1	0
Rule 3.2	0
Rule 4.1	0
Rule 4.2	0
Rule 5.1	0
Rule 5.2	0
Rule 5.3	0
Rule 5.4	0
Rule 5.5	0
Rule 5.6	0
Rule 5.7	0
Rule 5.8	0
Rule 5.9	0
Rule 6.1	0
Rule 6.2	0
Rule 7.1	0
Rule 7.2	0
Rule 7.3	0
Rule 7.4	0
Rule 8.01	0
Rule 8.02	0
Rule 8.03	0
Rule 8.04	0

Rule 8.05	0
Rule 8.06	0
Rule 8.07	0
Rule 8.08	0
Rule 8.09	0
Rule 8.10	0
Rule 8.11	0
Rule 8.12	0
Rule 8.13	0
Rule 8.14	0
Rule 9.1	0
Rule 9.2	0
Rule 9.3	0
Rule 9.4	0
Rule 9.5	0

MISRA-C 2012 Amendment 2

레퍼런스 항목 이름	이슈 수
Directives 1.1	0
Directives 4.1	0
Directives 4.3	0
Directives 4.4	0
Directives 4.5	0
Directives 4.6	0
Directives 4.7	0
Directives 4.8	0
Directives 4.9	0
Rule 1.1	0
Rule 1.2	0
Rule 1.3	0
Rule 1.4	0
Rule 10.1	0
Rule 10.2	0
Rule 10.3	0
Rule 10.4	0

Rule 10.5	0
Rule 10.6	0
Rule 10.7	0
Rule 10.8	0
Rule 11.1	0
Rule 11.2	0
Rule 11.3	0
Rule 11.4	0
Rule 11.5	0
Rule 11.6	0
Rule 11.7	0
Rule 11.8	0
Rule 11.9	0
Rule 12.1	0
Rule 12.2	0
Rule 12.3	0
Rule 12.4	0
Rule 12.5	0
Rule 13.1	0
Rule 13.2	0
Rule 13.3	0
Rule 13.4	0
Rule 13.5	0
Rule 13.6	0
Rule 14.1	0
Rule 14.2	0
Rule 14.3	0
Rule 14.4	0
Rule 15.1	0
Rule 15.2	0
Rule 15.3	0
Rule 15.4	0
Rule 15.5	0
Rule 15.6	0
Rule 15.7	0
Rule 16.1	0

Rule 16.2	0
Rule 16.3	0
Rule 16.4	0
Rule 16.5	0
Rule 16.6	0
Rule 16.7	0
Rule 17.1	0
Rule 17.2	0
Rule 17.3	0
Rule 17.4	0
Rule 17.5	0
Rule 17.6	0
Rule 17.7	0
Rule 17.8	0
Rule 18.1	0
Rule 18.2	0
Rule 18.3	0
Rule 18.4	0
Rule 18.5	0
Rule 18.6	0
Rule 18.7	0
Rule 18.8	0
Rule 19.1	0
Rule 19.2	0
Rule 2.1	0
Rule 2.2	0
Rule 2.3	0
Rule 2.4	0
Rule 2.5	0
Rule 2.6	0
Rule 2.7	0
Rule 20.01	0
Rule 20.02	0
Rule 20.03	0
Rule 20.04	0
Rule 20.05	0

Rule 20.06	0
Rule 20.07	0
Rule 20.08	0
Rule 20.09	0
Rule 20.10	0
Rule 20.11	0
Rule 20.12	0
Rule 20.13	0
Rule 21.01	0
Rule 21.02	0
Rule 21.03	0
Rule 21.04	0
Rule 21.05	0
Rule 21.06	0
Rule 21.07	0
Rule 21.08	0
Rule 21.09	0
Rule 21.10	0
Rule 21.11	0
Rule 21.12	0
Rule 21.13	0
Rule 21.14	0
Rule 21.15	0
Rule 21.16	0
Rule 21.17	0
Rule 21.18	0
Rule 21.19	0
Rule 21.20	0
Rule 21.21	0
Rule 22.01	0
Rule 22.02	0
Rule 22.03	0
Rule 22.04	0
Rule 22.05	0
Rule 22.06	0
Rule 22.07	0

Rule 22.08	0
Rule 22.09	0
Rule 22.10	0
Rule 3.1	0
Rule 3.2	0
Rule 4.1	0
Rule 4.2	0
Rule 5.1	0
Rule 5.2	0
Rule 5.3	0
Rule 5.4	0
Rule 5.5	0
Rule 5.6	0
Rule 5.7	0
Rule 5.8	0
Rule 5.9	0
Rule 6.1	0
Rule 6.2	0
Rule 7.1	0
Rule 7.2	0
Rule 7.3	0
Rule 7.4	0
Rule 8.01	0
Rule 8.02	0
Rule 8.03	0
Rule 8.04	0
Rule 8.05	0
Rule 8.06	0
Rule 8.07	0
Rule 8.08	0
Rule 8.09	0
Rule 8.10	0
Rule 8.11	0
Rule 8.12	0
Rule 8.13	0
Rule 8.14	0

Rule 9.1	0
Rule 9.2	0
Rule 9.3	0
Rule 9.4	0
Rule 9.5	0

MISRA-C 2012 Amendment 3

레퍼런스 항목 이름	이슈 수
Directives 1.1	0
Directives 4.1	0
Directives 4.10	0
Directives 4.12	0
Directives 4.14	0
Directives 4.3	0
Directives 4.4	0
Directives 4.5	0
Directives 4.6	0
Directives 4.7	0
Directives 4.8	0
Directives 4.9	0
Rule 1.1	0
Rule 1.2	0
Rule 1.3	0
Rule 1.4	0
Rule 10.1	0
Rule 10.2	0
Rule 10.3	0
Rule 10.4	0
Rule 10.5	0
Rule 10.6	0
Rule 10.7	0
Rule 10.8	0
Rule 11.1	0
Rule 11.2	0
Rule 11.3	0

Rule 11.4	0
Rule 11.5	0
Rule 11.6	0
Rule 11.7	0
Rule 11.8	0
Rule 11.9	0
Rule 12.1	0
Rule 12.2	0
Rule 12.3	0
Rule 12.4	0
Rule 12.5	0
Rule 13.1	0
Rule 13.2	0
Rule 13.3	0
Rule 13.4	0
Rule 13.5	0
Rule 13.6	0
Rule 14.1	0
Rule 14.2	0
Rule 14.3	0
Rule 14.4	0
Rule 15.1	0
Rule 15.2	0
Rule 15.3	0
Rule 15.4	0
Rule 15.5	0
Rule 15.6	0
Rule 15.7	0
Rule 16.1	0
Rule 16.2	0
Rule 16.3	0
Rule 16.4	0
Rule 16.5	0
Rule 16.6	0
Rule 16.7	0
Rule 17.1	0

Rule 17.2	0
Rule 17.3	0
Rule 17.4	0
Rule 17.5	0
Rule 17.6	0
Rule 17.7	0
Rule 17.8	0
Rule 18.1	0
Rule 18.2	0
Rule 18.3	0
Rule 18.4	0
Rule 18.5	0
Rule 18.6	0
Rule 18.7	0
Rule 18.8	0
Rule 19.1	0
Rule 19.2	0
Rule 2.1	0
Rule 2.2	0
Rule 2.3	0
Rule 2.4	0
Rule 2.5	0
Rule 2.6	0
Rule 2.7	0
Rule 20.01	0
Rule 20.02	0
Rule 20.03	0
Rule 20.04	0
Rule 20.05	0
Rule 20.06	0
Rule 20.07	0
Rule 20.08	0
Rule 20.09	0
Rule 20.10	0
Rule 20.11	0
Rule 20.12	0

Rule 20.13	0
Rule 21.01	0
Rule 21.02	0
Rule 21.03	0
Rule 21.04	0
Rule 21.05	0
Rule 21.06	0
Rule 21.07	0
Rule 21.08	0
Rule 21.09	0
Rule 21.10	0
Rule 21.11	0
Rule 21.12	0
Rule 21.13	0
Rule 21.14	0
Rule 21.15	0
Rule 21.16	0
Rule 21.17	0
Rule 21.18	0
Rule 21.19	0
Rule 21.20	0
Rule 21.21	0
Rule 22.01	0
Rule 22.02	0
Rule 22.03	0
Rule 22.04	0
Rule 22.05	0
Rule 22.06	0
Rule 22.07	0
Rule 22.08	0
Rule 22.09	0
Rule 22.10	0
Rule 3.1	0
Rule 3.2	0
Rule 4.1	0
Rule 4.2	0

Rule 5.1	0
Rule 5.2	0
Rule 5.3	0
Rule 5.4	0
Rule 5.5	0
Rule 5.6	0
Rule 5.7	0
Rule 5.8	0
Rule 5.9	0
Rule 6.1	0
Rule 6.2	0
Rule 7.1	0
Rule 7.2	0
Rule 7.3	0
Rule 7.4	0
Rule 8.01	0
Rule 8.02	0
Rule 8.03	0
Rule 8.04	0
Rule 8.05	0
Rule 8.06	0
Rule 8.07	0
Rule 8.08	0
Rule 8.09	0
Rule 8.10	0
Rule 8.11	0
Rule 8.12	0
Rule 8.13	0
Rule 8.14	0
Rule 9.1	0
Rule 9.2	0
Rule 9.3	0
Rule 9.4	0
Rule 9.5	0

● MISRA-C++ 2008

레퍼런스 항목 이름	이슈 수
Rule 0-1-1	0
Rule 8-3-1	0
Rule0-1-10	0
Rule0-1-11	0
Rule0-1-12	0
Rule0-1-3	0
Rule0-1-4	0
Rule0-1-5	0
Rule0-1-6	0
Rule0-1-7	0
Rule0-1-8	0
Rule0-1-9	0
Rule0-2-1	0
Rule0-3-1	0
Rule10-1-1	0
Rule10-1-2	0
Rule10-1-3	0
Rule10-3-1	0
Rule10-3-2	0
Rule10-3-3	0
Rule11-0-1	0
Rule12-1-1	0
Rule12-1-2	0
Rule12-1-3	0
Rule12-8-1	0
Rule12-8-2	0
Rule14-5-1	0
Rule14-5-2	0
Rule14-5-3	0
Rule14-6-1	0
Rule14-6-2	0
Rule14-7-1	0
Rule14-7-3	0
Rule14-8-1	0

Rule14-8-2	0
Rule15-0-1	0
Rule15-0-2	0
Rule15-1-1	0
Rule15-1-2	0
Rule15-1-3	0
Rule15-3-1	0
Rule15-3-2	0
Rule15-3-3	0
Rule15-3-4	0
Rule15-3-5	0
Rule15-3-6	0
Rule15-3-7	0
Rule15-4-1	0
Rule15-5-1	0
Rule15-5-2	0
Rule15-5-3	0
Rule16-0-1	0
Rule16-0-2	0
Rule16-0-3	0
Rule16-0-4	0
Rule16-0-5	0
Rule16-0-6	0
Rule16-0-7	0
Rule16-0-8	0
Rule16-1-1	0
Rule16-2-1	0
Rule16-2-2	0
Rule16-2-3	0
Rule16-2-4	0
Rule16-2-5	0
Rule16-2-6	0
Rule16-3-1	0
Rule16-3-2	0
Rule17-0-1	0
Rule17-0-2	0

Rule17-0-3	0
Rule17-0-5	0
Rule18-0-1	0
Rule18-0-2	0
Rule18-0-3	0
Rule18-0-4	0
Rule18-0-5	0
Rule18-2-1	0
Rule18-4-1	0
Rule18-7-1	0
Rule19-3-1	0
Rule2-10-1	0
Rule2-10-2	0
Rule2-10-3	0
Rule2-10-4	0
Rule2-10-5	0
Rule2-10-6	0
Rule2-13-1	0
Rule2-13-2	0
Rule2-13-3	0
Rule2-13-4	0
Rule2-13-5	0
Rule2-3-1	0
Rule2-5-1	0
Rule2-7-1	0
Rule2-7-2	0
Rule2-7-3	0
Rule27-0-1	0
Rule3-1-1	0
Rule3-1-2	0
Rule3-1-3	0
Rule3-2-1	0
Rule3-2-2	0
Rule3-2-3	0
Rule3-2-4	0
Rule3-3-1	0

Rule3-3-2	0
Rule3-4-1	0
Rule3-9-1	0
Rule3-9-2	0
Rule3-9-3	0
Rule4-10-1	0
Rule4-10-2	0
Rule4-5-1	0
Rule4-5-2	0
Rule4-5-3	0
Rule5-0-1	0
Rule5-0-10	0
Rule5-0-11	0
Rule5-0-12	0
Rule5-0-13	0
Rule5-0-14	0
Rule5-0-15	0
Rule5-0-16	0
Rule5-0-17	0
Rule5-0-18	0
Rule5-0-19	0
Rule5-0-2	0
Rule5-0-20	0
Rule5-0-21	0
Rule5-0-3	0
Rule5-0-4	0
Rule5-0-5	0
Rule5-0-6	0
Rule5-0-7	0
Rule5-0-8	0
Rule5-0-9	0
Rule5-14-1	0
Rule5-18-1	0
Rule5-19-1	0
Rule5-2-1	0
Rule5-2-10	0

Rule5-2-11	0
Rule5-2-12	0
Rule5-2-2	0
Rule5-2-3	0
Rule5-2-4	0
Rule5-2-5	0
Rule5-2-6	0
Rule5-2-7	0
Rule5-2-8	0
Rule5-2-9	0
Rule5-3-1	0
Rule5-3-2	0
Rule5-3-3	0
Rule5-3-4	0
Rule5-8-1	0
Rule6-2-1	0
Rule6-2-2	0
Rule6-2-3	0
Rule6-3-1	0
Rule6-4-1	0
Rule6-4-2	0
Rule6-4-3	0
Rule6-4-4	0
Rule6-4-5	0
Rule6-4-6	0
Rule6-4-7	0
Rule6-4-8	0
Rule6-5-1	0
Rule6-5-2	0
Rule6-5-3	0
Rule6-5-4	0
Rule6-5-5	0
Rule6-5-6	0
Rule6-6-1	0
Rule6-6-2	0
Rule6-6-3	0

Rule6-6-4	0
Rule6-6-5	0
Rule7-1-1	0
Rule7-1-2	0
Rule7-2-1	0
Rule7-3-1	0
Rule7-3-2	0
Rule7-3-3	0
Rule7-3-4	0
Rule7-3-5	0
Rule7-3-6	0
Rule7-4-2	0
Rule7-4-3	0
Rule7-5-1	0
Rule7-5-2	0
Rule7-5-3	0
Rule7-5-4	0
Rule8-0-1	0
Rule8-4-1	0
Rule8-4-2	0
Rule8-4-3	0
Rule8-4-4	0
Rule8-5-1	0
Rule8-5-2	0
Rule8-5-3	0
Rule9-3-1	0
Rule9-3-2	0
Rule9-3-3	0
Rule9-5-1	0
Rule9-6-1	0
Rule9-6-2	0
Rule9-6-3	0
Rule9-6-4	0

OWASP 2017

레퍼런스 항목 이름	이슈 수
A1-Injection	7
A2-Broken Authentication	0
A3-Sensitive Data Exposure	0
A5-Broken Access Control	0
A6-Security Misconfiguration	0

OWASP 2021

레퍼런스 항목 이름	이슈 수
A03 Injection	29
A05 Security Misconfiguration	0
A07 Identification and Authentication Failures	0

Python 시큐어코딩 가이드 2022

레퍼런스 항목 이름	이슈 수
01.01. SQL 삽입	0
01.02. 코드 삽입	0
01.03. 경로 조작 및 자원 삽입	0
01.04. 크로스사이트 스크립트(XSS)	0
01.05. 운영체제 명령어 삽입	0
01.06. 위험한 형식 파일 업로드	0
01.07. 신뢰되지 않은 URL주소로 자동접속 연결	0
01.08. 부적절한 XML 외부 개체 참조	0
01.09. XML 삽입	0
01.10. LDAP 삽입	0
01.11. 크로스사이트 요청 위조(CSRF)	0
01.12. 서버사이드 요청 위조	0
01.13. HTTP 응답분할	0
01.14. 보안기능 결정에 사용되는 부적절한 입력값	0
01.15. 포맷 스트링 삽입	0
02.01. 적절한 인증 없는 중요 기능 허용	0
02.03. 중요한 자원에 대한 잘못된 권한 설정	0
02.04. 취약한 암호화 알고리즘 사용	0
02.06. 하드코드된 중요정보	0

02.07. 충분하지 않은 키 길이 사용	0
02.08. 적절하지 않은 난수 값 사용	0
02.09. 취약한 비밀번호 허용	0
02.10. 사용자 하드디스크에 저장되는 쿠키를 통한 정보 노출	0
02.11. 주석문 안에 포함된 시스템 주요정보	0
02.12. 솔트 없이 일방향 해쉬 함수 사용	0
02.13. 무결성 검사없는 코드 다운로드	0
03.01. 경쟁조건: 검사시점과 사용시점(TOCTOU)	0
03.02. 종료되지 않는 반복문 또는 재귀 함수	0
04.01. 오류 메시지 정보노출	0
04.02. 오류상황 대응 부재	0
04.03. 부적절한 예외 처리	0
05.01. Null Pointer 역참조	0
05.02. 부적절한 자원 해제	0
05.03. 신뢰할 수 없는 데이터의 역직렬화	0
06.02. 제거되지 않고 남은 디버그 코드	0
06.03. Public 메소드로부터 반환된 Private 배열	0
06.04. Private 배열에 Public 데이터 할당	0

● Rust ANSSI guide v1.0

레퍼런스 항목 이름	이슈 수
R10 RULE - Don't use unsafe blocks	0
R11 RULE - Use appropriate arithmetic operations regarding potential overflows	0
R13 RECO - Use the ? operator and do not use the try! macro	0
R14 RULE - Don't use functions that can cause panic!	0
R15 RULE - Test properly array indexing or use the get method	0
R16 RULE - Handle correctly panic! in FFI	0
R17 RULE - Do not use forget	0
R19 RULE - Do not leak memory	0
R2 RULE - Keep default values for critical variables in cargo profiles	0
R20 RULE - Do release value wrapped in ManuallyDrop	0
R21 RULE - Always call from_raw on into_rawed value	0
R22 RULE - Do not use uninitialized memory	0
R32 RULE - Use only C-compatible types in FFI	0

● 무기체계 소프트웨어 보안약점 점검 목록

레퍼런스 항목 이름	이슈 수
CWE-119	0
CWE-134	0
CWE-170	0
CWE-190	1
CWE-209	53
CWE-22	9
CWE-259	0
CWE-285	0
CWE-306	0
CWE-307	0
CWE-312	0
CWE-319	0
CWE-321	0
CWE-327	7
CWE-330	4
CWE-367	1
CWE-369	0
CWE-390	6
CWE-400	0
CWE-404	23
CWE-415	0
CWE-416	0
CWE-457	0
CWE-467	0
CWE-469	0
CWE-476	34
CWE-489	0
CWE-494	0
CWE-495	0
CWE-496	0
CWE-497	53
CWE-521	0
CWE-562	0

CWE-587	0
CWE-59	0
CWE-615	0
CWE-628	0
CWE-676	0
CWE-732	0
CWE-755	20
CWE-759	0
CWE-78	5
CWE-89	2
CWE-99	9

방위사업청 코딩규칙

레퍼런스 항목 이름	이슈 수
1-01. Switch 구문에서 첫 번째 Label 전에 코드 구문이 존재하면 안된다.	0
1-02. 함수/변수 선언 시 type을 명시해야 한다.	0
1-03. 의미 없는 구문은 사용하지 말아야 한다.(side effect)	0
1-04. 함수의 Return Type에 맞는 return을 사용해야 한다.	0
1-05. 선언 없이 함수를 사용하지 말아야 한다.(묵시적 선언이 사용됨)	0
1-06. 매크로의 정의 여부를 확인하지 않고 해당 매크로에 대하여 #if, #elseif 표현을 사용하지 말아야 한다.	0
1-07. goto 문 사용은 최대한 자제한다.	0
1-08. 하나의 함수는 하나의 Exit Point를 가져야 한다.	0
1-09. switch~case 문은 default 문이 포함되어야 한다.	0
1-10. 한 줄에 하나의 명령문을 사용한다.	0
1-11. if - else if 문은 else 문도 포함시킨다.	0
2-01. String 배열의 초기화에서 배열의 마지막 인자는 NULL로 종료되어야 한다.	0
2-02. 초기화 되지 않은 변수를 사용하지 말아야 한다.	0
2-03. 설정되지 않은 포인터를 함수의 Read-only(const)로 사용하면 안된다.	0
3-01. external과 internal linkage 의 특성을 동시에 가질 수 없다.	0
3-02. external linkage scope 에서 선언된 함수나 Object의 이름은 유일해야 한다.	0
3-03. external linkage scope 에서 정의된 함수나 Object의 데이터 형은 선언 시 정의와 동일해야 한다.	0
3-04. 바깥 scope 의 식별자를 가리는 정의를 해서는 안된다.	0
4-01. float 자료형에서 동등성 비교연산을 수행하지 말아야 한다.	0

4-02. 조건문의 결과가 항상 True거나 False면 안된다.	0
4-03. switch의 case 조건을 만족할 수 없는 Label을 사용하지 않는다.	0
4-04. switch 구문에서 Expression을 논리적 연산으로 사용하지 말아야 한다.	0
4-05. 수행되지 않는 소스코드를 작성하지 말아야 한다.	0
5-01. 선언된 데이터 형으로 표현할 수 있는 숫자의 영역을 초과하는 값을 할당하지 말아야 한다.	0
5-02. 가변인수를 받는 함수의 Conversion 지시자와 Argument의 type은 동일해야 한다.	0
5-03. 가변인수를 받는 함수의 Conversion 지시자와 Argument의 개수는 동일해야 한다.	0
5-04. Object 저장값을 표현할 수 없는 데이터로의 형 변환을 하지말아야 한다.	0
5-05. 음수값을 unsigned type으로 변환을 자제해야 한다.	0
5-06. Character 문자열과 Wide character 문자열을 혼용하지 말아야 한다.	0
5-07. 포인터 Cast의 결과로 이전 포인터의 Const 특성의 상실을 유의해야 한다.	0
5-08. 포인터 Cast의 결과로 이전 포인터의 Volatile 특성의 상실을 유의해야 한다.	0
6-01. Null pointer를 참조하지 않는다.	0
6-02. 지역 변수의 주소값을 더 넓은 scope를 가진 변수에 할당하지 말아야 한다.	0
6-03. 지역 변수의 주소값을 함수의 리턴값으로 사용하지 말아야 한다.	0
6-04. 선언된 배열의 크기를 초과하는 인덱스 값을 사용하지 말아야 한다.	0
6-05. Null Pointer를 산술연산 하지 않는다.	0
7-01. 하나의 Sequence Point 내에서 하나의 Object Value를 두 번 이상 변경하지 않아야 한다.	0
7-02. 0 으로 나눗셈 연산을 하지 않는다.	0
7-03. 하나의 Sequence Point 내에서 Object의 값을 변경하고 Access 하지 않아야 한다.	0
7-04. 음수 값 또는 데이터 사이즈를 초과하는 값을 사용하여 Shift operator를 하지 않는다.	0
7-05. Underlying type이 부호 없는 정수일 경우 단행 빼기 연산(-)을 사용하여 결과를 대입하지 말아야 한다.	0
7-06. sizeof의 인자는 side effect를 가지지 말아야 한다.	0
7-07. Boolean 표현 값에 &&, ||, ! 연산자를 제외하고 다른 연산자를 사용하지 말아야 한다.	0
7-08. 조건문에 직접적인 대입 연산자를 사용하지 말아야 한다.	0
7-09. Signed Value에서 Bitwise연산자(<<, ~, |, ^ 등)로 인한 Negative Value를 유의해야 한다.	0
8-01. Scanf의 Argument 는 Object Value의 저장된 주소에 값이 입력되어야 한다.	0
8-02. #include 구문에서 표준에 맞지 않는 Character set을 사용하지 않아야 한다.	0
8-03. Allocated되는 메모리 블록의 크기는 Pointer에 의해서 Address 되는 완전한 하나의 multiple size여야 한다.	0
8-04. 함수의 Argument type과 개수는 함수의 Prototype, 선언, 정의가 모두 같아야 한다.	0
8-05. 구조체/배열의 초기화 시 default 초기화 값(0)을 제외하고, 구조에 맞게 '{}'를 사용하	

여 선언된 Size에 맞게 초기화 해야 한다.	0
9-01. 동적 할당된 데이터를 해제할 때, 잘못된 메소드를 이용하여 해제하면 안된다.	0
9-02. 지역 변수의 주소 값을 처리하는 handle을 return하지 말아야 한다.	0
9-03. 함수 parameter의 주소 값을 처리하는 handle을 return하지 말아야 한다.	0
9-04. 소멸자내에서 처리할 수 없는 예외 상황을 발생시키지 말아야 한다.	0
9-05. 사용되지 않는 예외 처리 문을 작성하지 말아야 한다.	0
9-06. exception specification에 기술되지 않은 모든 throw에 대하여 예외처리를 해야만 한다.	0
9-07. main 함수에서 처리되지 않는 throw를 작성하지 말아야 한다.	0
9-08. 해제된 메모리 영역 사용하지 말아야 한다.	0
9-09.복사 연산자를 통해서, 복사되지 않는 멤버 변수가 존재하지 말아야 한다	0
9-10. C 코딩 방법으로 메모리를 할당 하면 안된다.	0
9-11. 순수 가상함수는 반드시 0으로 초기화 되어야 한다	0
9-12. 순수함수는 반드시 가상함수로 선언되어야 한다	0
9-13. virtual base 클래스의 포인터는 derived 클래스의 포인터로 cast 할 때에는 dynamic_cast만 사용해야 한다.	0
9-14. 생성자/소멸자 내에서 가상함수는 식별자 없이 호출하면 안된다.	0
9-15. 생성자/소멸자에 dynamic type을 사용하면 안된다.	0

● 소프트웨어 보안약점 진단가이드 2021

레퍼런스 항목 이름	이슈 수
DNS lookup에 의존한 보안 결정	0
HTTP 응답분할	7
LDAP 삽입	3
Null Pointer 역참조	34
Private 배열에 Public 데이터 할당	0
Public 메소드부터 반환된 Private 배열	0
SQL 삽입	2
XML 삽입	2
경로 조작 및 자원 삽입	18
경쟁조건: 검사시점과 사용시점(TOCTOU)	1
메모리 버퍼 오버플로우	0
무결성 검사없는 코드 다운로드	0
반복된 인증시도 제한 기능 부재	0
보안기능 결정에 사용되는 부적절한 입력값	26

부적절한 XML 외부개체 참조	0
부적절한 예외처리	20
부적절한 인가	0
부적절한 인증서 유효성 검증	0
부적절한 자원 해제	23
부적절한 전자서명 확인	0
사용자 하드디스크에 저장되는 쿠키를 통한 정보 노출	0
서버사이드 요청 위조	0
솔트 없이 일방향 해쉬 함수 사용	0
신뢰되지 않는 URL 주소로 자동 접속 연결	0
신뢰할 수 없는 데이터의 역직렬화	0
암호화되지 않은 중요정보	2
오류 상황 대응 부재	6
오류메시지 정보 노출	53
운영체제 명령어 삽입	5
위험한 형식 파일 업로드	0
잘못된 세션에 의한 데이터 정보 노출	0
적절하지 않은 난수 값 사용	4
적절한 인증없는 중요기능 허용	0
정수형 오버플로우	1
제거되지 않고 남은 디버그 코드	0
종료되지 않는 반복문 또는 재귀 함수	0
주석문 안에 포함된 시스템 주요정보	0
중요한 자원에 대한 잘못된 권한 설정	0
초기화되지 않은 변수 사용	0
충분하지 않은 키 길이 사용	0
취약한 API 사용	0
취약한 비밀번호 허용	0
취약한 암호화 알고리즘 사용	7
코드 삽입	0
크로스사이트 스크립트	22
크로스사이트 요청 위조	0
포맷스트링 삽입	0
하드코드된 중요정보	0
해제된 자원 사용	0

● 주요정보통신기반시설 취약점 분석·평가 기준

레퍼런스 항목 이름	이슈 수
SQL 인젝션	0
XPath 인젝션	0
경로 추적	0
디렉토리 인덱싱	0
세션 고정	0
세션 예측	0
약한 문자열 강도	0
운영체제 명령 실행	0
위치 공개	0
크로스사이트 스크립팅	0
파일 다운로드	0

■ 이슈 상세 결과

● [규칙 이름] catch 블록에서 반환 구문 사용 (보통, Java)

catch 블록에서 반환 구문 사용 체커는 catch 블록 안에 반환 구문이 있는 경우를 검출합니다.

값이 있는 반환이라면 반환할 값을 변수에 저장해두고 try 구문 전체가 끝난 후에 반환하도록 수정합니다.

- CWE 660 4.14
  - 460 - Improper Cleanup on Thrown Exception

- CWE 660 4.7
  - Improper Cleanup on Thrown Exception - (460)

위험한 예시

```
1. public static final boolean doStuff( ) {
2.
3.   boolean threadLock;
4.   boolean truthvalue=true;
5.   try {
6.
7.   while(
8. //check some condition
9.   ) {
10.
11.     threadLock=true; //do some stuff to truthvalue
12.     threadLock=false;
13.     }
14.   }
15.   catch (Exception e){
16.
17.   System.err.println("You did something bad");
18.   if (something) return truthvalue;
19.   }
20.   return truthvalue;
21. }
```

라인 18: catch 블록 안에서 값을 반환합니다.

안전한 예시

```
1. public static final boolean doStuff( ) {
2.
3.   boolean threadLock;
4.   boolean truthvalue=true;
5.   try {
6.
7.   while(
8. //check some condition
9.   ) {
10.
11.     threadLock=true; //do some stuff to truthvalue
12.     threadLock=false;
13.     }
14.   }
15.   catch (Exception e){
16.
17.   System.err.println("You did something bad");
18.   //if (something) return truthvalue;
19.   }
20.   return truthvalue;
21. }
```

라인 18: catch 블록 안에서는 반환 구문을 사용하지 않습니다.

이슈 ID	274457
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DataBaseServer.java
줄 번호	72

소스코드

```
67.          org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql, resp);
```

```
68.       } catch (java.sql.SQLException e) {
69.           if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.               e.printStackTrace();
71.               resp.add(new XMLMessage("Error processing request: " + e.
getMessage()));
72.               return new ResponseEntity<List<XMLMessage>>(resp, HttpStatus.OK);
73.           } else throw new ServletException(e);
74.       }
75.       return new ResponseEntity<List<XMLMessage>>(resp, HttpStatus.OK);
76.   }
77.
```

이슈 ID	274472
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /LDAPManager.java
줄 번호	131

소스코드

```
126.
127.           return true;
128.       } catch (Exception e) {
129.           System.out.println("LDAP error search: ");
130.           e.printStackTrace();
131.           return false;
132.       }
133.   }
134.
135.   public DirContext getDirContext() throws NamingException {
136.       if (ctx == null) {
```

이슈 ID	274498
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00007.java
줄 번호	67

소스코드

```
62.             org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
63.         } catch (IOException e) {
64.             System.out.println("Problem executing cmdi - TestCase");
65.             response.getWriter()
66.                     .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
67.             return;
68.         }
69.     }
70. }
```

이슈 ID	274500
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00017.java
줄 번호	69

소스코드

```
64.             org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
65.         } catch (IOException e) {
66.             System.out.println("Problem executing cmdi - TestCase");
67.             response.getWriter()
68.                     .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
69.             return;
70.         }
71.     }
72. }
```

이슈 ID	274501
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00018.java

줄 번호 64

소스코드

```
59.             int count = statement.executeUpdate(sql);
60.             org.owasp.benchmark.helpers.DatabaseHelper.outputUpdateComplete
(sql, response);
61.         } catch (java.sql.SQLException e) {
62.             if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
63.                 response.getWriter().println("Error processing request.");
64.                 return;
65.             } else throw new ServletException(e);
66.         }
67.     }
68. }
```

이슈 ID 274514

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00034.java

줄 번호 61

소스코드

```
56.             statement.execute(sql, java.sql.Statement.RETURN_GENERATED_KEYS);
57.             org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
response);
58.         } catch (java.sql.SQLException e) {
59.             if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
60.                 response.getWriter().println("Error processing request.");
61.                 return;
62.             } else throw new ServletException(e);
63.         }
64.     }
65. }
```

이슈 ID 274536

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00092.java
줄 번호	101

소스코드

```
96.             org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
97.         } catch (IOException e) {
98.             System.out.println("Problem executing cmdi - TestCase");
99.             response.getWriter()
100.                     .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
101.             return;
102.         }
103.     }
104. }
```

● [규칙 이름] 데이터베이스 연결 경쟁 조건 (보통, Java)

데이터베이스 연결 경쟁 조건 체커는 데이터베이스 연결 객체를 정적으로 선언한 경우를 검출합니다.

java.sql.Connection 타입의 객체가 데이터베이스 연결 객체로 간주됩니다.

트랜잭션 리소스 객체(예: JDBC Connection 객체)는 정적 필드에 저장되어서는 안됩니다. 이러한 객체는 오직 한 번에 한 트랜잭션과 연관될 수 있는데, 정적 필드에 이들을 저장하면 서로 다른 트랜잭션에서 스레드들간의 잘못된 공유가 일어날 수 있습니다. 따라서 Database Connection 객체를 다룰 때에는 정적을 사용하지 않도록 합니다. 다중 스레드 프로그램에서 여러 스레드가 정적 필드에 저장된 트랜잭션 리소스에 동시에 접근할 경우 경쟁 조건이 발생하여 원인을 찾기 어려운 오작동을 일으킬 수 있습니다.

해당 객체는 인스턴스 필드에 저장하는 것이 좋습니다.

- CWE 660 4.14
  - 362 - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
  - 366 - Race Condition within a Thread

- CWE 660 4.7

- Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') - (362)

위험한 예시

```
1. public class DBConnectionManager {
2.   private static Connection conn = connect(); //Avoid static
3. }
```

라인 2: 데이터베이스 연결 객체가 정적 필드에 저장되었습니다.

안전한 예시

```
1. public class DBConnectionManager {
2.   private Connection conn = connect();
3. }
```

라인 2: 데이터베이스 연결 객체를 정적 필드에 저장하지 않도록 합니다.

이슈 ID	274461
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	36

소스코드

```
31. import javax.sql.DataSource;
32. import org.owasp.benchmark.service.pojo.XMLMessage;
33. import org.owasp.esapi.ESAPI;
34.
35. public class DatabaseHelper {
36.     private static Connection conn;
37.     public static org.springframework.jdbc.core.JdbcTemplate JDBCtemplate;
38.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtil =
39.             new org.owasp.benchmark.helpers.HibernateUtil(false);
40.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtilClassic =
41.             new org.owasp.benchmark.helpers.HibernateUtil(true);
```

● [규칙 이름] 자원 누수 (보통, Java)

자원 누수 체커는 파일, 소켓 등 리소스를 할당한 후에 해제하지 않는 코드를 검출합니다.

유한한 프로그램 리소스를 할당받았다면 더 이상 사용하지 않게 되었을 때 적절히 반환하여야 합니다. 만일 프로그램 오류 또는 오류나 JVM의 가비지 콜렉터에 의해 리소스가 빠르게 회수되지 않은 경우 리소스가 고갈되는 문제가 발생할 수 있습니다. 이러한 경우 시스템 성능이 저하되거나, 기능이 중단되거나, DoS(Denial of Service)가 발생하거나, 또 다른 리소스를 획득하는 데 실패할 수 있습니다.

한 메소드 안에서 할당 받은 리소스는 그 메소드가 종료되기 전에 반드시 반환할 수 있도록 코드를 작성합니다. 특히 이런 리소스는 접근 도중 예외가 발생할 수 있는 경우가 많기 때문에 리소스가 사용되는 구간을 try 구문으로 감싸고 finally 블록에서 확실하게 반환하는 것이 좋습니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-404

- 소프트웨어 보안약점 진단가이드 2021
  - 부적절한 자원 해제

위험한 예시

```
1. public class ResourceLeakEx {
2.   public void testResourceLeak() {
3.     try {
4.       BufferedWriter out = new BufferedWriter(new FileWriter(
5.                                                   new File("test.txt")));
6.       out.write("This is Resource Leak sample code...");
7.       out.newLine();
8.     } catch (IOException e) {
9.       // ...
10.     }
11.   } /* BUG : RESOURCE_LEAK */
12. }
```

라인 4: 리소스를 할당받고 나서 예외가 발생할 경우 리소스가 해제되지 않습니다.

안전한 예시

```
1. public class ResourceLeakSafeEx {
2.   public void testResourceLeak() {
3.     try {
4.       BufferedWriter out = new BufferedWriter(new FileWriter(
5.                                                   new File("test.txt")));
6.       out.write("This is Resource Leak sample code...");
7.       out.newLine();
8.     } catch (IOException e) {
9.       // ...
10.     } finally {
11.       if (out != null) {
12.         out.close(); /* SAFE */
13.       }
14.     }
15.   }
16. }
```

라인 12: 예외 발생과 상관없이 항상 리소스를 해제해야 합니다.

이슈 ID	274707
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DataBaseServer.java
줄 번호	65

소스코드

```
60.       List<XMLMessage> resp = new ArrayList<XMLMessage>();
61.       String sql = "SELECT * from USERS";
62.       try {
63.          java.sql.Connection connection =
64.                org.owasp.benchmark.helpers.DatabaseHelper.getSqlConnection();
65.          java.sql.PreparedStatement statement = connection.prepareStatement(sql);
66.          statement.execute();
67.          org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
resp);
68.       } catch (java.sql.SQLException e) {
```

```
69.         if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.             e.printStackTrace();
```

이슈 ID	274538
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DataBaseServer.java
줄 번호	65

소스코드

```
60.     List<XMLMessage> resp = new ArrayList<XMLMessage>();
61.     String sql = "SELECT * from USERS";
62.     try {
63.         java.sql.Connection connection =
64.                 org.owasp.benchmark.helpers.DatabaseHelper.getSqlConnection();
65.         java.sql.PreparedStatement statement = connection.prepareStatement(sql);
66.         statement.execute();
67.         org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
resp);
68.     } catch (java.sql.SQLException e) {
69.         if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.             e.printStackTrace();
```

이슈 ID	274540
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	124

소스코드

```
119.     if (conn == null) {
120.         getSqlConnection();
121.     }
122.     Statement stmt = null;
123.     try {
```

```
124.         stmt = conn.createStatement();
125.     } catch (SQLException e) {
126.         System.out.println("Problem with database init.");
127.     }
128.
129.     return stmt;
```

이슈 ID	274594
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	124

소스코드

```
119.     if (conn == null) {
120.         getSqlConnection();
121.     }
122.     Statement stmt = null;
123.     try {
124.         stmt = conn.createStatement();
125.     } catch (SQLException e) {
126.         System.out.println("Problem with database init.");
127.     }
128.
129.     return stmt;
```

이슈 ID	274611
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	124

소스코드

```
119.     if (conn == null) {
120.         getSqlConnection();
```

```
121.        }
122.        Statement stmt = null;
123.        try {
124.            stmt = conn.createStatement();
125.        } catch (SQLException e) {
126.            System.out.println("Problem with database init.");
127.        }
128.
129.        return stmt;
```

이슈 ID	274541
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	161

소스코드

```
156.    public static java.sql.Connection getSqlConnection() {
157.        if (conn == null) {
158.            try {
159.                InitialContext ctx = new InitialContext();
160.                DataSource datasource = (DataSource) ctx.lookup("java:comp/env/jdbc/BenchmarkDB");
161.                conn = datasource.getConnection();
162.                conn.setAutoCommit(false);
163.            } catch (SQLException | NamingException e) {
164.                System.out.println("Problem with getSqlConnection.");
165.                e.printStackTrace();
166.            }
```

이슈 ID	274542
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	161

소스코드

```
156.    public static java.sql.Connection getSqlConnection() {
157.        if (conn == null) {
158.            try {
159.                InitialContext ctx = new InitialContext();
160.                DataSource datasource = (DataSource) ctx.lookup("java:comp/env/jdbc
/BenchmarkDB");
161.                conn = datasource.getConnection();
162.                conn.setAutoCommit(false);
163.            } catch (SQLException | NamingException e) {
164.                System.out.println("Problem with getSqlConnection.");
165.                e.printStackTrace();
166.            }
```

이슈 ID	274557
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	234

소스코드

```
229.                    + "<meta http-equiv=₩"Content-Type₩" content=₩"text/html;
charset=UTF-8₩">₩n"
230.                    + "</head>₩n"
231.                    + "<body>₩n"
232.                    + "<p>₩n");
233.
234.        BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
235.        BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
236.
237.        try {
238.            // read the output from the command
239.            // System.out.println("Here is the standard output of the
```

이슈 ID	274555
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	235

소스코드

```
230.                 + "</head>₩n"
231.                 + "<body>₩n"
232.                 + "<p>₩n");
233.
234.     BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
235.     BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
236.
237.     try {
238.       // read the output from the command
239.       // System.out.println("Here is the standard output of the
240.       // command:₩n");
```

이슈 ID	274556
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	235

소스코드

```
230.                 + "</head>₩n"
231.                 + "<body>₩n"
232.                 + "<p>₩n");
233.
234.     BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
235.     BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
236.
```

```
237.        try {
238.            // read the output from the command
239.            // System.out.println("Here is the standard output of the
240.            // command:₩n");
```

이슈 ID	274552
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	266

소스코드

```
261.
262.    // A method used by the Benchmark JAVA test cases to format OS Command
Output
263.    // This version is only used by the Web Services test cases.
264.    public static void printOSCommandResults(java.lang.Process proc,
List<XMLMessage> resp) {
265.
266.        BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
267.        BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
268.
269.        try {
270.            // read the output from the command
271.            resp.add(new XMLMessage("Here is the standard output of the
command:"));
```

이슈 ID	274553
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	267

소스코드

```
262.    // A method used by the Benchmark JAVA test cases to format OS Command
Output
263.    // This version is only used by the Web Services test cases.
264.    public static void printOSCommandResults(java.lang.Process proc,
List<XMLMessage> resp) {
265.
266.        BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
267.        BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
268.
269.        try {
270.            // read the output from the command
271.            resp.add(new XMLMessage("Here is the standard output of the
command:"));
272.            String s = null;
```

이슈 ID	274554
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	267

소스코드

```
262.    // A method used by the Benchmark JAVA test cases to format OS Command
Output
263.    // This version is only used by the Web Services test cases.
264.    public static void printOSCommandResults(java.lang.Process proc,
List<XMLMessage> resp) {
265.
266.        BufferedReader stdInput = new BufferedReader(new InputStreamReader
(proc.getInputStream()));
267.        BufferedReader stdError = new BufferedReader(new InputStreamReader
(proc.getErrorStream()));
268.
269.        try {
270.            // read the output from the command
```

```
271.            resp.add(new XMLMessage("Here is the standard output of the
command:"));
272.            String s = null;
```

이슈 ID	274558
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	383

소스코드

```
378.            Files.createDirectories(pathToFileDir);
379.            File f = new File(completeName);
380.            if (!f.exists()) {
381.                f.createNewFile();
382.            }
383.            FileOutputStream fos = new FileOutputStream(f, true);
384.            os = new PrintStream(fos);
385.            os.println(line);
386.        } catch (IOException e1) {
387.            result = false;
388.            e1.printStackTrace();
```

이슈 ID	274577
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00003.java
줄 번호	94

소스코드

```
89.             java.io.File fileTarget =
90.                     new java.io.File(
91.                             new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
92.                             "passwordFile.txt");
```

```
93.         java.io.FileWriter fw =
94.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
95.         fw.write(
96.             "hash_value="
97.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
98.                 + "₩n");
99.         fw.close();
```

이슈 ID	274578
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00005.java
줄 번호	93

소스코드

```
88.         java.io.File fileTarget =
89.             new java.io.File(
90.                 new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
91.                 "passwordFile.txt");
92.         java.io.FileWriter fw =
93.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
94.         fw.write(
95.             "secret_value="
96.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
97.                 + "₩n");
98.         fw.close();
```

이슈 ID	274584
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00009.java
줄 번호	94

소스코드

```
89.         java.io.File fileTarget =
90.             new java.io.File(
91.                 new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
92.                 "passwordFile.txt");
93.         java.io.FileWriter fw =
94.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
95.         fw.write(
96.             "hash_value="
97.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
98.                 + "₩n");
99.         fw.close();
```

이슈 ID	274596
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	88

소스코드

```
83.         java.io.File fileTarget =
84.             new java.io.File(
85.                 new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
86.                 "passwordFile.txt");
87.         java.io.FileWriter fw =
88.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
89.         fw.write(
90.             "secret_value="
91.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
92.                 + "₩n");
93.         fw.close();
```

이슈 ID	274624
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	89

소스코드

```
84.         java.io.File fileTarget =
85.             new java.io.File(
86.                 new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
87.                 "passwordFile.txt");
88.         java.io.FileWriter fw =
89.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
90.         fw.write(
91.             "secret_value="
92.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
93.                 + "₩n");
94.         fw.close();
```

이슈 ID	274659
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	96

소스코드

```
91.         java.io.File fileTarget =
92.             new java.io.File(
93.                 new java.io.File(org.owasp.benchmark.helpers.Utils.
TESTFILES_DIR),
94.                 "passwordFile.txt");
95.         java.io.FileWriter fw =
96.             new java.io.FileWriter(fileTarget, true); // the true will append the
new data
97.         fw.write(
```

```
98.                 "hash_value="
99.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
100.                         + "₩n");
101.         fw.close();
```

이슈 ID	274666
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00076.java
줄 번호	114

소스코드

```
109.         java.io.File fileTarget =
110.                 new java.io.File(
111.                         new java.io.File(org.owasp.benchmark.helpers.Utils.TESTFILES_DIR),
112.                         "passwordFile.txt");
113.         java.io.FileWriter fw =
114.                 new java.io.FileWriter(fileTarget, true); // the true will append the new data
115.         fw.write(
116.                 "hash_value="
117.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
118.                         + "₩n");
119.         fw.close();
```

이슈 ID	274700
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00207.java
줄 번호	62

소스코드

```
57.                                 param.getBytes())));
58.     }
59.
60.     try {
61.         java.io.FileInputStream file =
62.                 new java.io.FileInputStream(
63.                         org.owasp.benchmark.helpers.Utils.getFileFromClasspath(
64.                                 "employees.xml", this.getClass().getClassLoader()));
65.         javax.xml.parsers.DocumentBuilderFactory builderFactory =
66.                 javax.xml.parsers.DocumentBuilderFactory.newInstance();
67.         // Prevent XXE
```

이슈 ID	274705
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00442.java
줄 번호	55

소스코드

```
50.
51.     bar = (7 * 42) - num > 200 ? "This should never happen" : param;
52.
53.     try {
54.         java.io.FileInputStream file =
55.                 new java.io.FileInputStream(
56.                         org.owasp.benchmark.helpers.Utils.getFileFromClasspath(
57.                                 "employees.xml", this.getClass().getClassLoader()));
58.         javax.xml.parsers.DocumentBuilderFactory builderFactory =
59.                 javax.xml.parsers.DocumentBuilderFactory.newInstance();
60.         // Prevent XXE
```

● [규칙 이름] 널 역참조 (보통, Java)

널 역참조 체커는 널 상수나 널이 할당된 변수를 역참조하는 경우를 검출합니다.

널일 수도 있는 값을 검사 없이 역참조하면 프로그램 실행 중에 NullPointerException 예외가 발생합니다. 프로그램 실행이 비정상적으로 종료될 수 있습니다.

공격자는 NullPointerException 예외로 발생하는 취약점을 사용하여 후 공격을 계획할 수 있습니다.

변수가 널이 아니라는 보장이 없다면 해당 값을 역참조하기 전에 항상 널이 아닌지 확인해야 합니다.

- CWE 660 4.14
  - 476 - NULL Pointer Dereference

- CWE 660 4.7
  - NULL Pointer Dereference - (476)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-476

- 소프트웨어 보안약점 진단가이드 2021
  - Null Pointer 역참조

위험한 예시

```
1. public class ForwardNullEx {
2.   public void test() {
3.     String uppercased = toUpperCase(null); // FORWARD_NULL
4.   }
5.   public String toUpperCase(String arg) {
6.     arg.toUpperCase();
7.   }
8. }
```

라인 3: toUpperCase() 메소드에 널 값을 전달합니다.
라인 6: 인자를 확인 없이 역참조합니다.

안전한 예시

```
1. public class ForwardNullSafeEx {
2.   public void test() {
```

```
3.    String uppercased = toUpperCase(null);  // SAFE
4.  }
5.  public String toUpperCase(String arg) {
6.    if (arg != null) { // Do a null check
7.      arg.toUpperCase();
8.    } else {
9.      return null;
10.   }
11. }
12. }
```

라인 6: 널 역참조가 발생하지 않도록 검사를 수행한 후에 변수를 사용하도록 합니다.

이슈 ID	274559
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	390

소스코드

```
385.          os.println(line);
386.      } catch (IOException e1) {
387.          result = false;
388.          e1.printStackTrace();
389.      } finally {
390.          os.close();
391.      }
392.
393.      return result;
394.  }
395.
```

● [규칙 이름] 널 반환 값 역참조 (보통, Java)

널 반환 값 역참조 체커는 반환된 널 값을 확인 없이 역참조하는 경우를 검출합니다.

널을 반환할 수 있는 메소드를 호출한 후 그 반환 값을 확인 없이 역참조하면 예외가 발생할 수 있습니다.

공격자는 NullPointerException 예외로 발생하는 취약점을 사용하여 후 공격을 계획할 수 있습니다.

널을 반환할 수 있는 메소드의 반환 값은 항상 값 검사 후 사용합니다.

- CWE 660 4.14
  - 476 - NULL Pointer Dereference

- CWE 660 4.7
  - NULL Pointer Dereference - (476)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-476

- 소프트웨어 보안약점 진단가이드 2021
  - Null Pointer 역참조

위험한 예시

```
1. public class NullReturnEx {
2.   public Object returnNull() { return null; }
3.   public void testNull() {
4.     String str =returnNull().toString(); /* BUG : NULL_RETURN */
5.   }
6. }
```

라인 2: return널() 메소드는 널 값을 반환합니다.
라인 4: return널() 메소드의 반환값을 확인 없이 직접 역참조합니다.

안전한 예시

```
1. public class NullReturnSafeEx {
2.   public Object returnNull() { return null; }
```

```
3.  public void testNull() {
4.    Object x = returnNull();
5.    String str = x != null ? x.toString() : null;  // SAFE
6.  }
7. }
```

라인 5: 널을 반환하는 메소드를 호출한 경우 그 반환 값을 사용하기 전에 널 검사를 수행합니다.

이슈 ID	274545
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPServer.java
줄 번호	96

소스코드

```
91.      // BEGIN HACK
92.      try {
93.          String dir =
94.                  Utils.getFileFromClasspath(
95.                          "benchmark.properties", LDAPServer.class.getClassLoader())
96.                  .getParent();
97.          File workDir = new File(dir + "/../ldap");
98.          workDir.mkdirs();
99.          System.setProperty("workingDiretory", workDir.getPath());
100.
101.           init();
```

이슈 ID	274546
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	153

소스코드

```
148.            perms.add(PosixFilePermission.GROUP_EXECUTE);
149.            perms.add(PosixFilePermission.OTHERS_READ);
150.            perms.add(PosixFilePermission.OTHERS_EXECUTE);
151.
152.            try {
153.                Files.setPosixFilePermissions(script.toPath(), perms);
154.            } catch (IOException e) {
155.                System.out.println(
156.                        "Problem while changing executable permissions: " + e.
getMessage());
157.            }
158.        }
```

이슈 ID	274550
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	192

소스코드

```
187.
188.    public static String getInsecureOSCommandString(ClassLoader classLoader) {
189.        String command = null;
190.        String osName = System.getProperty("os.name");
191.        if (osName.indexOf("Windows") != -1) {
192.            command = Utils.getFileFromClasspath("insecureCmd.bat", classLoader).
getAbsolutePath();
193.        } else {
194.            command = Utils.getFileFromClasspath("insecureCmd.sh", classLoader).
getAbsolutePath();
195.        }
196.        return command;
197.    }
```

이슈 ID	274549

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	194

[소스코드]

```
189.        String command = null;
190.        String osName = System.getProperty("os.name");
191.        if (osName.indexOf("Windows") != -1) {
192.            command = Utils.getFileFromClasspath("insecureCmd.bat", classLoader).
getAbsolutePath();
193.        } else {
194.            command = Utils.getFileFromClasspath("insecureCmd.sh", classLoader).
getAbsolutePath();
195.        }
196.        return command;
197.    }
198.
199.    public static List<String> getOSCommandArray(String append) {
```

● [규칙 이름] 누락된 널 값 검사 (보통, Java)

누락된 널 값 검사 체커는 널 여부를 확인한 적이 있는 값을 후 확인 없이 역참조하는 경우를 검출합니다.

변수를 한 번이라도 널과 비교했다면 당시 개발자가 해당 지점에 도달하는 실행 경로에 따라 해당 변수에 널이 포함될 가능성이 있다고 판단했기 때문입니다. 그런데 같은 변수를 이후에는 널 확인 없이 역참조 했다면 작성자가 실수했을 가능성이 높기 때문에 검출합니다. 공격자는 NullPointerException 예외로 발생하는 취약점을 사용하여 후 공격을 계획할 수 있습니다.

변수가 널이 아니라는 보장이 없다면 해당 값을 역참조하기 전에 항상 널이 아닌지 확인해야 합니다. 반대로 변수가 널이 아니라는 보장이 있다면 혼동을 줄이기 위해 일관적으로 해당 변수를 널 확인 없이 직접 사용해야 합니다.

- CWE 660 4.14
  - 476 - NULL Pointer Dereference

- CWE 660 4.7

- NULL Pointer Dereference - (476)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-476

- 소프트웨어 보안약점 진단가이드 2021
  - Null Pointer 역참조

위험한 예시

```
1. public class UncheckedNullEx {
2.   public void test(String x) {
3.     String str = "";
4.     System.out.println(str);
5.     if(x != null) {
6.       str = x.toUpperCase();
7.     }
8.     x.toString(); /* BUG : UNCHECKED_NULL */
9.   }
10. }
```

라인 5: 변수 x를 널과 비교하고 있습니다.
라인 8: 한 번 널과 비교했던 변수 x를 이번에는 검사 없이 사용했습니다.

안전한 예시

```
1. public class UncheckedNullSafeEx {
2.   public void test(String x) {
3.     String str = "";
4.     System.out.println(str);
5.     if(x != null) {
6.       str = x.toUpperCase();
7.     }
8.     if(x != null) {
9.       x.toString(); /* SAFE */
10.    }
```

```
11.   }
12. }
```

라인 9: 널일 가능성이 있는 변수는 항상 검사 후에 역참조합니다.

이슈 ID	274706
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /DataBaseServer.java
줄 번호	65

소스코드

```
60.     List<XMLMessage> resp = new ArrayList<XMLMessage>();
61.     String sql = "SELECT * from USERS";
62.     try {
63.         java.sql.Connection connection =
64.                 org.owasp.benchmark.helpers.DatabaseHelper.getSqlConnection();
65.         java.sql.PreparedStatement statement = connection.prepareStatement(sql);
66.         statement.execute();
67.         org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
resp);
68.     } catch (java.sql.SQLException e) {
69.         if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.             e.printStackTrace();
```

이슈 ID	274537
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /DataBaseServer.java
줄 번호	65

소스코드

```
60.     List<XMLMessage> resp = new ArrayList<XMLMessage>();
61.     String sql = "SELECT * from USERS";
62.     try {
```

```
63.          java.sql.Connection connection =
64.                  org.owasp.benchmark.helpers.DatabaseHelper.getSqlConnection();
65.          java.sql.PreparedStatement statement = connection.prepareStatement(sql);
66.          statement.execute();
67.          org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
resp);
68.      } catch (java.sql.SQLException e) {
69.          if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.              e.printStackTrace();
```

이슈 ID	274539
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	124

소스코드

```
119.      if (conn == null) {
120.          getSqlConnection();
121.      }
122.      Statement stmt = null;
123.      try {
124.          stmt = conn.createStatement();
125.      } catch (SQLException e) {
126.          System.out.println("Problem with database init.");
127.      }
128.
129.      return stmt;
```

● [규칙 이름] 표준 라이브러리의 널 반환 값 역참조 (보통, Java)

표준 라이브러리의 널 반환 값 역참조 체커는 Java 표준 라이브러리 메소드중에서 널을 반환할 가능성이 있는 메소드의 반환 값을 확인 없이 역참조하는 경우를 검출합니다.

널을 반환할 수 있는 표준 라이브러리 메소드를 호출한 후 그 반환 값을 확인 없이 역참조하면 예외가 발생할 수 있습니다.

공격자는 NullPointerException 예외로 발생하는 취약점을 사용하여 후 공격을 계획할 수 있습니다.

널을 반환할 수 있는 표준 라이브러리 메소드의 반환 값은 항상 값 검사 후 사용합니다.

- CWE 660 4.14
  - 476 - NULL Pointer Dereference

- CWE 660 4.7
  - NULL Pointer Dereference - (476)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-476

- 소프트웨어 보안약점 진단가이드 2021
  - Null Pointer 역참조

위험한 예시

```
1. public class NullReturnStdEx {
2.   public void getInputFromFile() {
3.     try  {
4.       BufferedReader br =
5.         new BufferedReader(new FileReader("input.dat"));
6.       String str = br.readLine();  // BufferedReader.readLine() can return null
7.       str.toUpperCase();  // NULL_RETURN_STD
8.     } catch (IOException e) { e.printStackTrace(); }
9.   }
10. }
```

라인 6: java.io.BufferedReader 클래스의 readline() 메소드의 반환 값은 널일 가능성이 있습니다. 만약 널이 반환되었다면 이를 사용한 toUpperCase() 메소드가 호출될 널 역참조가 발생합니다.

안전한 예시

```
1. public class NullReturnStdSafeEx {
2.   public void getInputFromFile() {
3.     try  {
4.       BufferedReader br =
5.         new BufferedReader(new FileReader("input.dat"));
6.       String str = br.readLine();  // BufferedReader.readLine() can return null
7.       if (str != NULL) {
8.         str.toUpperCase();  // SAFE
9.       }
10.     } catch (IOException e) { e.printStackTrace(); }
11.   }
12. }
```

라인 7: readLine() 메소드의 반환 값이 널이 아닌지 확인 후에 사용해야 합니다.

이슈 ID	274543
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	84

소스코드

```
79.       InitialDirContext iniDirContext = (InitialDirContext) ctx;
80.
81.       try {
82.           iniDirContext.bind(name, ctx, matchAttrs);
83.       } catch (NamingException e) {
84.           if (!e.getMessage().contains("ENTRY_ALREADY_EXISTS")) {
85.               System.out.println("Record already exist or an error occurred: " + e.
getMessage());
86.           }
87.       }
88.
89.       return true;
```

이슈 ID	274544

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	117

소스코드

```
112.
113.        NamingEnumeration<SearchResult> results = ctx.search(base, filter, sc);
114.
115.        while (results.hasMore()) {
116.            SearchResult sr = (SearchResult) results.next();
117.            Attributes attrs = sr.getAttributes();
118.
119.            Attribute attr = attrs.get("uid");
120.            if (attr != null) {
121.                // logger.debug("record found " + attr.get());
122.                // System.out.println("record found " + attr.get());
```

이슈 ID	274547
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	165

소스코드

```
160.
161.    public static String getCookie(HttpServletRequest request, String paramName) {
162.        Cookie[] values = request.getCookies();
163.        String param = "none";
164.        if (paramName != null) {
165.            for (int i = 0; i < values.length; i++) {
166.                if (values[i].getName().equals(paramName)) {
167.                    param = values[i].getValue();
168.                    break; // break out of for loop when param found
169.                }
170.            }
```

이슈 ID	274560
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.         request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00001.html");
45.     rd.include(request, response);
46.   }
47.
48.   @Override
49.   public void doPost(HttpServletRequest request, HttpServletResponse response)
50.       throws ServletException, IOException {
```

이슈 ID	274568
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.         request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00002.html");
45.     rd.include(request, response);
```

```
46.  }
47.
48.  @Override
49.  public void doPost(HttpServletRequest request, HttpServletResponse response)
50.          throws ServletException, IOException {
```

이슈 ID	274573
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00003.java
줄 번호	45

소스코드

```
40.      userCookie.setPath(request.getRequestURI());
41.      userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.      response.addCookie(userCookie);
43.      javax.servlet.RequestDispatcher rd =
44.              request.getRequestDispatcher("/hash-00/BenchmarkTest00003.html");
45.      rd.include(request, response);
46.  }
47.
48.  @Override
49.  public void doPost(HttpServletRequest request, HttpServletResponse response)
50.          throws ServletException, IOException {
```

이슈 ID	274583
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00009.java
줄 번호	46

소스코드

```
41.      // some code
42.      response.setContentType("text/html;charset=UTF-8");
```

```
43.
44.     String param = "";
45.     java.util.Enumeration<String> names = request.getHeaderNames();
46.     while (names.hasMoreElements()) {
47.         String name = (String) names.nextElement();
48.
49.         if (org.owasp.benchmark.helpers.Utils.commonHeaders.contains(name)) {
50.             continue; // If standard header, move on to next one
51.         }
```

이슈 ID	274582
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00009.java
줄 번호	47

소스코드

```
42.     response.setContentType("text/html;charset=UTF-8");
43.
44.     String param = "";
45.     java.util.Enumeration<String> names = request.getHeaderNames();
46.     while (names.hasMoreElements()) {
47.         String name = (String) names.nextElement();
48.
49.         if (org.owasp.benchmark.helpers.Utils.commonHeaders.contains(name)) {
50.             continue; // If standard header, move on to next one
51.         }
52.
```

이슈 ID	274589
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00012.java
줄 번호	72

소스코드

```
67.         javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
68.             idc.search(base, filter, filters, sc);
69.         while (results.hasMore()) {
70.             javax.naming.directory.SearchResult sr =
71.                 (javax.naming.directory.SearchResult) results.next();
72.             javax.naming.directory.Attributes attrs = sr.getAttributes();
73.
74.             javax.naming.directory.Attribute attr = attrs.get("uid");
75.             javax.naming.directory.Attribute attr2 = attrs.get("street");
76.             if (attr != null) {
77.                 response.getWriter()
```

이슈 ID	274591
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00012.java
줄 번호	84

소스코드

```
79.                         "LDAP query results:<br>"
80.                             + "Record found with name "
81.                             + attr.get()
82.                             + "<br>"
83.                             + "Address: "
84.                             + attr2.get()
85.                             + "<br>");
86.             // System.out.println("record found " + attr.get());
87.             found = true;
88.         }
89.     }
```

이슈 ID	274598
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00021.java

줄 번호	63

소스코드

```
58.         javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
59.                 ctx.search(base, filter, filters, sc);
60.         while (results.hasMore()) {
61.             javax.naming.directory.SearchResult sr =
62.                     (javax.naming.directory.SearchResult) results.next();
63.             javax.naming.directory.Attributes attrs = sr.getAttributes();
64.
65.             javax.naming.directory.Attribute attr = attrs.get("uid");
66.             javax.naming.directory.Attribute attr2 = attrs.get("street");
67.             if (attr != null) {
68.                 response.getWriter()
```

이슈 ID	274599
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00021.java
줄 번호	75

소스코드

```
70.                             "LDAP query results:<br>"
71.                                     + "Record found with name "
72.                                     + attr.get()
73.                                     + "<br>"
74.                                     + "Address: "
75.                                     + attr2.get()
76.                                     + "<br>");
77.                 // System.out.println("record found " + attr.get());
78.                 found = true;
79.             }
80.         }
```

이슈 ID	274622
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00044.java
줄 번호	64

소스코드

```
59.         javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
60.             ctx.search(base, filter, sc);
61.         while (results.hasMore()) {
62.           javax.naming.directory.SearchResult sr =
63.               (javax.naming.directory.SearchResult) results.next();
64.           javax.naming.directory.Attributes attrs = sr.getAttributes();
65.
66.           javax.naming.directory.Attribute attr = attrs.get("uid");
67.           javax.naming.directory.Attribute attr2 = attrs.get("street");
68.           if (attr != null) {
69.             response.getWriter()
```

이슈 ID	274620
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00044.java
줄 번호	76

소스코드

```
71.                         "LDAP query results:<br>"
72.                             + "Record found with name "
73.                             + attr.get()
74.                             + "<br>"
75.                             + "Address: "
76.                             + attr2.get()
77.                             + "<br>");
78.             // System.out.println("record found " + attr.get());
```

```
79.                 found = true;
80.             }
81.         }
```

이슈 ID	274625
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00061.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.             request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00061.html");
45.     rd.include(request, response);
46. }
47.
48. @Override
49. public void doPost(HttpServletRequest request, HttpServletResponse response)
50.         throws ServletException, IOException {
```

이슈 ID	274630
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
```

```
42.         response.addCookie(userCookie);
43.         javax.servlet.RequestDispatcher rd =
44.                 request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00062.
html");
45.         rd.include(request, response);
46.     }
47.
48.     @Override
49.     public void doPost(HttpServletRequest request, HttpServletResponse response)
50.             throws ServletException, IOException {
```

이슈 ID	274638
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00064.java
줄 번호	45

소스코드

```
40.         userCookie.setPath(request.getRequestURI());
41.         userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.         response.addCookie(userCookie);
43.         javax.servlet.RequestDispatcher rd =
44.                 request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00064.
html");
45.         rd.include(request, response);
46.     }
47.
48.     @Override
49.     public void doPost(HttpServletRequest request, HttpServletResponse response)
50.             throws ServletException, IOException {
```

이슈 ID	274640
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java

줄 번호 45

[소스코드]

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.             request.getRequestDispatcher("/pathtraver-00/BenchmarkTest00065.
html");
45.     rd.include(request, response);
46.   }
47.
48.   @Override
49.   public void doPost(HttpServletRequest request, HttpServletResponse response)
50.           throws ServletException, IOException {
```

이슈 ID	274647
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00067.java
줄 번호	45

[소스코드]

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.             request.getRequestDispatcher("/weakrand-00/BenchmarkTest00067.
html");
45.     rd.include(request, response);
46.   }
47.
48.   @Override
49.   public void doPost(HttpServletRequest request, HttpServletResponse response)
50.           throws ServletException, IOException {
```

이슈 ID	274657
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	45

소스코드

```
40.       userCookie.setPath(request.getRequestURI());
41.       userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.       response.addCookie(userCookie);
43.       javax.servlet.RequestDispatcher rd =
44.               request.getRequestDispatcher("/hash-00/BenchmarkTest00070.html");
45.       rd.include(request, response);
46.   }
47.
48.   @Override
49.   public void doPost(HttpServletRequest request, HttpServletResponse response)
50.           throws ServletException, IOException {
```

이슈 ID	274662
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00076.java
줄 번호	45

소스코드

```
40.       userCookie.setPath(request.getRequestURI());
41.       userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.       response.addCookie(userCookie);
43.       javax.servlet.RequestDispatcher rd =
44.               request.getRequestDispatcher("/hash-00/BenchmarkTest00076.html");
45.       rd.include(request, response);
```

```
46.  }
47.
48.  @Override
49.  public void doPost(HttpServletRequest request, HttpServletResponse response)
50.          throws ServletException, IOException {
```

이슈 ID	274667
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00077.java
줄 번호	45

소스코드

```
40.      userCookie.setPath(request.getRequestURI());
41.      userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
42.      response.addCookie(userCookie);
43.      javax.servlet.RequestDispatcher rd =
44.              request.getRequestDispatcher("/cmdi-00/BenchmarkTest00077.html");
45.      rd.include(request, response);
46.  }
47.
48.  @Override
49.  public void doPost(HttpServletRequest request, HttpServletResponse response)
50.          throws ServletException, IOException {
```

이슈 ID	274670
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	45

소스코드

```
40.      userCookie.setPath(request.getRequestURI());
41.      userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
```

```
getHost());
42.         response.addCookie(userCookie);
43.         javax.servlet.RequestDispatcher rd =
44.                 request.getRequestDispatcher("/weakrand-00/BenchmarkTest00084.
html");
45.         rd.include(request, response);
46.     }
47.
48.     @Override
49.     public void doPost(HttpServletRequest request, HttpServletResponse response)
50.             throws ServletException, IOException {
```

이슈 ID	274680
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	45

소스코드

```
40.         userCookie.setPath(request.getRequestURI());
41.         userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.         response.addCookie(userCookie);
43.         javax.servlet.RequestDispatcher rd =
44.                 request.getRequestDispatcher("/weakrand-00/BenchmarkTest00086.
html");
45.         rd.include(request, response);
46.     }
47.
48.     @Override
49.     public void doPost(HttpServletRequest request, HttpServletResponse response)
50.             throws ServletException, IOException {
```

이슈 ID	274690
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode

파일	/BenchmarkTest00087.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.             request.getRequestDispatcher("/securecookie-00/BenchmarkTest00087.
html");
45.     rd.include(request, response);
46.   }
47.
48.   @Override
49.   public void doPost(HttpServletRequest request, HttpServletResponse response)
50.           throws ServletException, IOException {
```

이슈 ID	274696
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00092.java
줄 번호	45

소스코드

```
40.     userCookie.setPath(request.getRequestURI());
41.     userCookie.setDomain(new java.net.URL(request.getRequestURL().toString()).
getHost());
42.     response.addCookie(userCookie);
43.     javax.servlet.RequestDispatcher rd =
44.             request.getRequestDispatcher("/cmdi-00/BenchmarkTest00092.html");
45.     rd.include(request, response);
46.   }
47.
```

```
48.  @Override
49.  public void doPost(HttpServletRequest request, HttpServletResponse response)
50.          throws ServletException, IOException {
```

● [규칙 이름] 부적절한 예외 처리 (높음, Java)

부적절한 예외 처리 체커는 너무 다양한 예외를 포괄적으로 처리하는 코드를 검출합니다.

최상위 예외 클래스인 Throwable 혹은 Exception 등을 직접 잡는 것을 포괄적 예외 처리로 볼 수 있습니다.

지나치게 일반적인 예외 처리를 하게 되면 특별히 처리해야 하는 개별 예외를 적절히 처리하지 못할 뿐 아니라 이 지점에서 논리적으로 발생해서는 안되는 예외까지 처리하게 됩니다. 이러한 경우에 개발 및 검수 과정에서 발견되었어야 하는 설계 및 구현 상 결함이 발견되지 못한 채 배포 단계로 넘어가서 더 큰 문제의 원인이 될 수 있습니다.

해당 코드 구역에서 발생할 수 있는 세부적인 예외들을 명시적으로 나눠서 처리하는 것이 좋습니다. 일반적인 예외를 처리하더라도 예측 가능한 세부적인 예외들을 먼저 처리한 후 마지막에 처리하도록 합니다. 이렇게 작성했을 때 실행 흐름이 일반적인 예외 처리 부분에 도달했다면 개발 과정에서 예측하지 못한 예외가 발생한 것이므로 조용히 넘어가지 않고 상황에 대한 상세한 기록을 남겨서 추후 결함을 수정할 수 있도록 합니다.

- CWE 660 4.14
  - 397 - Declaration of Throws for Generic Exception

- CWE 660 4.7
  - Declaration of Throws for Generic Exception - (397)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-755

- 소프트웨어 보안약점 진단가이드 2021
  - 부적절한 예외처리

위험한 예시

```
1. public void readFromFile(String fileName){
2.   try{
3.     File myFile = new File(fileName);
4.     FileReader fr = new FileReader(myFile);
5.   } catch(Exception ex){ }
6. }
```

라인 5: 파일 관련 API를 사용하면서 발생 가능한 세부 입출력 예외 대신 포괄적인 Exception 예외를 처리하고 있습니다.

안전한 예시

```
1. public void readFromFile(String fileName) throws FileNotFoundException,
IOException, MyException {
2.   try {
3.     //Null check for fileName
4.     if(fileName == NULL) throw new MyException("error");
5.     File myFile = new File(fileName);
6.     FileReader fr = new FileReader(myFile);
7.   } catch(FileNotFoundException fe){
8.     ...
9.   } catch(IOException ie){
10.     ...
11.   }
12. }
```

라인 7: 실제 발생 가능한 입출력 예외들을 명시적으로 처리합니다.

이슈 ID	274466
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	112

소스코드

```
107.                     + "END;");
108.             conn.commit();
109.             initData();
110.
111.             System.out.println("DataBase tables/procedures created.");
112.         } catch (Exception e1) {
113.             System.out.println(
114.                     "Problem with database table/procedure creations: " + e1.
getMessage());
115.         }
116.     }
117.
```

이슈 ID	274464
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	151

소스코드

```
146.             executeSQLCommand("INSERT INTO SCORE (nick, score) VALUES('foo',
40)");
147.
148.             executeSQLCommand(
149.                     "INSERT INTO EMPLOYEE (first_name, last_name, salary) VALUES
('foo', 'bar', 34567)");
150.             conn.commit();
151.         } catch (Exception e1) {
152.             System.out.println("Problem with database init/reset: " + e1.
getMessage());
153.         }
154.     }
155.
156.     public static java.sql.Connection getSqlConnection() {
```

이슈 ID	274471

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	128

소스코드

```
123.             }
124.         }
125.         ctx.close();
126.
127.         return true;
128.     } catch (Exception e) {
129.         System.out.println("LDAP error search: ");
130.         e.printStackTrace();
131.         return false;
132.     }
133. }
```

이슈 ID	274475
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPServer.java
줄 번호	102

소스코드

```
97.         File workDir = new File(dir + "/../ldap");
98.         workDir.mkdirs();
99.         System.setProperty("workingDiretory", workDir.getPath());
100.
101.         init();
102.     } catch (Exception e) {
103.         System.out.println("Error initializing LDAP Server: " + e.getMessage());
104.         e.printStackTrace();
105.     }
106.
107.     LDAPManager emd = new LDAPManager();
```

이슈 ID	274485
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	326

소스코드

```
321.             BufferedReader br = new BufferedReader(fr); ) {
322.         String line;
323.         while ((line = br.readLine()) != null) {
324.             sourceLines.add(line);
325.         }
326.     } catch (Exception e) {
327.         try {
328.             System.out.println("Problem reading contents of file: " + file.
getCanonicalFile());
329.         } catch (IOException e2) {
330.             System.out.println("Problem reading file to get lines from.");
331.             e2.printStackTrace();
```

이슈 ID	274490
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	84

소스코드

```
79.                             + org.owasp
80.                                 .esapi
81.                                 .ESAPI
82.                                 .encoder()
83.                                 .encodeForHTML(new String(b, 0, size)));
84.     } catch (Exception e) {
85.         System.out.println("Couldn't open FileInputStream on file: '" + fileName +
"'");
86.         response.getWriter()
```

```
87.                 .println(
88.                         "Problem getting FileInputStream: "
89.                                 + org.owasp
```

이슈 ID	274491
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	99

소스코드

```
94.      } finally {
95.          if (fis != null) {
96.              try {
97.                  fis.close();
98.                  fis = null;
99.              } catch (Exception e) {
100.                 // we tried...
101.             }
102.         }
103.     }
104. }
```

이슈 ID	274493
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	78

소스코드

```
73.         response.getWriter()
74.                 .println(
75.                         "Now ready to write to file: "
76.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
```

```
77.
78.         } catch (Exception e) {
79.             System.out.println("Couldn't open FileOutputStream on file: '" + fileName
+ "'");
80.             //                  System.out.println("File exception caught and swallowed:
" + e.getMessage());
81.         } finally {
82.             if (fos != null) {
83.                 try {
```

이슈 ID	274494
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	86

소스코드

```
81.         } finally {
82.             if (fos != null) {
83.                 try {
84.                     fos.close();
85.                     fos = null;
86.                 } catch (Exception e) {
87.                     // we tried...
88.                 }
89.             }
90.         }
91.     }
```

이슈 ID	274499
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00012.java
줄 번호	101

소스코드

```
96.       } catch (javax.naming.NamingException e) {
97.           throw new ServletException(e);
98.       } finally {
99.           try {
100.              ads.closeDirContext();
101.          } catch (Exception e) {
102.              throw new ServletException(e);
103.          }
104.      }
105.  }
106. }
```

이슈 ID	274510
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00021.java
줄 번호	92

소스코드

```
87.       } catch (javax.naming.NamingException e) {
88.           throw new ServletException(e);
89.       } finally {
90.           try {
91.               ads.closeDirContext();
92.           } catch (Exception e) {
93.               throw new ServletException(e);
94.           }
95.       }
96.   }
97. }
```

이슈 ID	274512
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00028.java

줄 번호 63

소스코드

```
58.           response.getWriter()
59.                   .println(
60.                           "Now ready to write to file: "
61.                                   + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
62.
63.       } catch (Exception e) {
64.           System.out.println("Couldn't open FileOutputStream on file: '" + fileName
+ "'");
65.           //                  System.out.println("File exception caught and swallowed:
" + e.getMessage());
66.       } finally {
67.           if (fos != null) {
68.               try {
```

이슈 ID 274513

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00028.java

줄 번호 71

소스코드

```
66.       } finally {
67.           if (fos != null) {
68.               try {
69.                   fos.close();
70.                   fos = null;
71.               } catch (Exception e) {
72.                   // we tried...
73.               }
74.           }
75.       }
76.   }
```

이슈 ID	274515
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00044.java
줄 번호	93

소스코드

```
88.      } catch (javax.naming.NamingException e) {
89.          throw new ServletException(e);
90.      } finally {
91.          try {
92.              ads.closeDirContext();
93.          } catch (Exception e) {
94.              throw new ServletException(e);
95.          }
96.      }
97.  }
98. }
```

이슈 ID	274525
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	90

소스코드

```
85.                              + org.owasp
86.                                  .esapi
87.                                  .ESAPI
88.                                  .encoder()
89.                                  .encodeForHTML(new String(b, 0, size)));
90.      } catch (Exception e) {
91.          System.out.println("Couldn't open FileInputStream on file: '" + fileName +
"'");
92.          response.getWriter()
```

```
93.                 .println(
94.                         "Problem getting FileInputStream: "
95.                                 + org.owasp
```

이슈 ID	274526
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	105

소스코드

```
100.     } finally {
101.         if (fis != null) {
102.             try {
103.                 fis.close();
104.                 fis = null;
105.             } catch (Exception e) {
106.                 // we tried...
107.             }
108.         }
109.     }
110. }
```

이슈 ID	274528
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00064.java
줄 번호	84

소스코드

```
79.         response.getWriter()
80.                 .println(
81.                         "Now ready to write to file: "
82.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
```

```
83.
84.         } catch (Exception e) {
85.             System.out.println("Couldn't open FileOutputStream on file: '" + fileName
+ "'");
86.             //                  System.out.println("File exception caught and swallowed:
" + e.getMessage());
87.         } finally {
88.             if (fos != null) {
89.                 try {
```

이슈 ID	274529
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00064.java
줄 번호	92

소스코드

```
87.         } finally {
88.             if (fos != null) {
89.                 try {
90.                     fos.close();
91.                     fos = null;
92.                 } catch (Exception e) {
93.                     // we tried...
94.                 }
95.             }
96.         }
97.     }
```

이슈 ID	274531
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	90

소스코드

```
85.                             + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName)
86.                             + "' is:₩n₩n");
87.         response.getWriter()
88.                 .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(new String
(b, 0, size)));
89.         is.close();
90.       } catch (Exception e) {
91.         System.out.println("Couldn't open InputStream on file: '" + fileName + "'");
92.         response.getWriter()
93.                 .println(
94.                         "Problem getting InputStream: "
95.                                 + org.owasp
```

이슈 ID	274532
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	105

소스코드

```
100.      } finally {
101.        if (is != null) {
102.          try {
103.            is.close();
104.            is = null;
105.          } catch (Exception e) {
106.            // we tried...
107.          }
108.        }
109.      }
110.  }
```

● [규칙 이름] 빈 catch 블록 (보통, Java)

빈 catch 블록 체커는 예외를 처리하는 코드 내용이 없는 예외 처리 블록을 검출합니다.

catch 블록 내에 실제 실행 코드가 없으면 검출됩니다.

예외가 발생했을 때 예외를 잡아낸 후 아무 작업도 하지 않으면 프로그램 실행 시 오류가 발생했을 경우에 원인을 파악하기가 쉽지 않습니다.

빈 블록 안에 예외 처리 코드를 추가합니다. 특별히 처리할 것이 없다면 예외가 발생했었다는 오류 메시지를 남겨두는 것이 가장 무난합니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-390

- 소프트웨어 보안약점 진단가이드 2021
  - 오류 상황 대응 부재

위험한 예시

```
1. private Connection conn;
2. public Connection DBConnect(String url, String id, String password) {
3.   try {
4.     String CONNECT_STRING = url + ":" + id + ":" + password;
5.     InitialContext ctx = new InitialContext();
6.     DataSource datasource = (DataSource) ctx.lookup(CONNECT_STRING);
7.     conn = datasource.getConnection();
8.   } catch (SQLException e) {
9.     // Catch block is empty
10.   } catch (NamingException e) {
11.     // Catch block is empty
12.   }
13.   return conn;
14. }
```

라인 8: catch 블록에서 예외를 잡지만 아무런 조치를 취하지 않습니다.
라인 10: catch 블록에서 예외를 잡지만 아무런 조치를 취하지 않습니다.

안전한 예시

```
1. private Connection conn;
2. public Connection DBConnect(String url, String id, String password) {
```

```
3.  try {
4.    String CONNECT_STRING = url + ":" + id + ":" + password;
5.    InitialContext ctx = new InitialContext();
6.    DataSource datasource = (DataSource) ctx.lookup(CONNECT_STRING);
7.    conn = datasource.getConnection();
8.  } catch (SQLException e) {
9.    // Proper Exception handling
10.     if ( conn != null ) {
11.       try {
12.         conn.close();
13.       } catch (SQLException e1) {
14.         conn = null;
15.       }
16.     }
17.   } catch (NamingException e) {
18.     // Proper Exception handling.
19.     if ( conn != null ) {
20.       try {
21.         conn.close();
22.       } catch (SQLException e1) {
23.         conn = null;
24.       }
25.     }
26.   }
27.   return conn;
28. }
```

라인 10: 예외를 잡은 후 각각의 예외에 대해 적절한 처리 작업을 수행해야 합니다.
라인 17: 예외를 잡은 후 각각의 예외에 대해 적절한 처리 작업을 수행해야 합니다.

이슈 ID	274489
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	99

소스코드

```
94.      } finally {
95.          if (fis != null) {
96.              try {
97.                  fis.close();
98.                  fis = null;
99.              } catch (Exception e) {
100.                 // we tried...
101.             }
102.         }
103.     }
104. }
```

이슈 ID	274492
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	86

소스코드

```
81.      } finally {
82.          if (fos != null) {
83.              try {
84.                  fos.close();
85.                  fos = null;
86.              } catch (Exception e) {
87.                  // we tried...
88.              }
89.          }
90.      }
91. }
```

이슈 ID	274511
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00028.java
줄 번호	71

소스코드

```
66.       } finally {
67.           if (fos != null) {
68.               try {
69.                   fos.close();
70.                   fos = null;
71.               } catch (Exception e) {
72.                   // we tried...
73.               }
74.           }
75.       }
76.   }
```

이슈 ID	274524
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	105

소스코드

```
100.      } finally {
101.          if (fis != null) {
102.              try {
103.                  fis.close();
104.                  fis = null;
105.              } catch (Exception e) {
106.                  // we tried...
107.              }
108.          }
109.      }
110.  }
```

이슈 ID	274527
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode

파일	/BenchmarkTest00064.java
줄 번호	92

소스코드

```
87.      } finally {
88.          if (fos != null) {
89.              try {
90.                  fos.close();
91.                  fos = null;
92.              } catch (Exception e) {
93.                  // we tried...
94.              }
95.          }
96.      }
97.  }
```

이슈 ID	274530
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00065.java
줄 번호	105

소스코드

```
100.      } finally {
101.          if (is != null) {
102.              try {
103.                  is.close();
104.                  is = null;
105.              } catch (Exception e) {
106.                  // we tried...
107.              }
108.          }
109.      }
110.  }
```

● [규칙 이름] 신뢰 범위 위반 (매우 높음, Java)

신뢰 범위 위반 체커는 신뢰할 수 없는 데이터와 신뢰할 수 있는 데이터가 섞여있는 경우를 검출합니다.

세션 값이나 속성에 신뢰할 수 없는 외부 입력을 전달하는 것이 한 예입니다.

신뢰할 수 없는 데이터와 신뢰할 수 있는 데이터가 혼재하면, 프로그래머가 신뢰할 수 없는 데이터를 실수로 신뢰하여 사용할 수 있습니다. 신뢰할 수 있는 데이터의 경계는 하나의 선으로 생각해볼 수 있습니다. 선의 한쪽에서는 데이터를 신뢰하지 않습니다. 다른 한쪽은 신뢰할 수 있다고 생각합니다. 데이터를 검증하는 것은 이 경계를 넘어가기 위함입니다. 신뢰할 수 없는 데이터 쪽에서 신뢰할 수 있는 쪽으로 넘어가기 위해 데이터 검증을 사용하는 것입니다. Trust boundary violation은 신뢰할 수 있는 쪽과 신뢰할 수 없는 쪽의 경계가 흐려질때 발생합니다. 이런 경우는 신뢰할 수 있는 데이터와 신뢰할 수 없는 데이터가 같은 데이터 구조체에 혼재되는 경우 흔하게 발생합니다.

신뢰할 수 있는 데이터만 모여있는 저장 공간은 특별히 관리합니다. 그 공간에 외부 입력을 저장할 때는 검증을 거친 후에만 저장하도록 합니다.

위험한 예시

```
1. javax.servlet.http.Cookie[] theCookies = request.getCookies();
2. String param = "";
3. if (theCookies != null) {
4.   for (javax.servlet.http.Cookie theCookie : theCookies) {
5.     if (theCookie.getName().equals("danger")) {
6.       param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
7.       break;
8.     }
9.   }
10. }
11.
12. request.getSession().setAttribute( param, "danger param"); //Bad
```

라인 12: request의 쿠키를 통해 전달된 신뢰할 수 없는 값을 바로 세션에 저장합니다.

안전한 예시

```
1. javax.servlet.http.Cookie[] theCookies = request.getCookies();
2. String param = "";
3. if (theCookies != null) {
4.   for (javax.servlet.http.Cookie theCookie : theCookies) {
5.     if (theCookie.getName().equals("danger")) {
```

```
6.      param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
7.      break;
8.    }
9.  }
10. }
11.
12. if(isSafe(param)){
13.   request.getSession().setAttribute( param, "danger param"); //Good
14. else {
15.  //
16. }
```

라인 12: 외부 값이 원하는 데이터 형식에 맞는지 검사후 사용합니다.

이슈 ID	274606
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	82

소스코드

```
77.           rememberMe.setSecure(true);
78.           rememberMe.setHttpOnly(true);
79.           rememberMe.setDomain(new java.net.URL(request.getRequestURL().
toString()).getHost());
80.           rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
81.           // e.g., /benchmark/sql-01/BenchmarkTest01001
82.           request.getSession().setAttribute(cookieName, rememberMeKey);
83.           response.addCookie(rememberMe);
84.           response.getWriter()
85.                   .println(
86.                           user
87.                                   + " has been remembered with cookie: "
```

이슈 ID	274615

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	85

소스코드

```
80.                 new javax.servlet.http.Cookie(cookieName, rememberMeKey);
81.         rememberMe.setSecure(true);
82.         rememberMe.setHttpOnly(true);
83.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
84.         // e.g., /benchmark/sql-01/BenchmarkTest01001
85.         request.getSession().setAttribute(cookieName, rememberMeKey);
86.         response.addCookie(rememberMe);
87.         response.getWriter()
88.                 .println(
89.                         user
90.                                 + " has been remembered with cookie: "
```

이슈 ID	274650
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	124

소스코드

```
119.         rememberMe.setSecure(true);
120.         rememberMe.setHttpOnly(true);
121.         rememberMe.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
122.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
123.         // e.g., /benchmark/sql-01/BenchmarkTest01001
124.         request.getSession().setAttribute(cookieName, rememberMeKey);
125.         response.addCookie(rememberMe);
126.         response.getWriter()
127.                 .println(
```

```
128.                    user
129.                        + " has been remembered with cookie: "
```

이슈 ID	274671
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	103

소스코드

```
98.                 new javax.servlet.http.Cookie(cookieName, rememberMeKey);
99.             rememberMe.setSecure(true);
100.            rememberMe.setHttpOnly(true);
101.            rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
102.            // e.g., /benchmark/sql-01/BenchmarkTest01001
103.            request.getSession().setAttribute(cookieName, rememberMeKey);
104.            response.addCookie(rememberMe);
105.            response.getWriter()
106.                    .println(
107.                            user
108.                                    + " has been remembered with cookie: "
```

이슈 ID	274686
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	107

소스코드

```
102.                new javax.servlet.http.Cookie(cookieName, rememberMeKey);
103.            rememberMe.setSecure(true);
104.            rememberMe.setHttpOnly(true);
105.            rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
```

```
106.        // e.g., /benchmark/sql-01/BenchmarkTest01001
107.        request.getSession().setAttribute(cookieName, rememberMeKey);
108.        response.addCookie(rememberMe);
109.        response.getWriter()
110.                .println(
111.                        user
112.                                + " has been remembered with cookie: "
```

● [규칙 이름] 예측 가능한 난수 생성 (높음, Java)

예측 가능한 난수 생성 체커는 예측 가능한 난수를 사용하는 코드를 검출합니다.

예측 불가능한 숫자가 필요한 상황에서 예측 가능한 난수를 사용한다면, 공격자는 SW에서 생성되는 다음 숫자를 예상하여 시스템을 공격하는 것이 가능하게 됩니다. 일반적으로 언어마다 기본 제공되는 통계적 PRNG(유사 난수 생성기) 방식을 사용하면 생성될 난수를 공격자가 쉽게 예측 가능하며, 시드를 잘 설정하는 것만으로는 안전을 보장할 수 없습니다.

암호학적으로 충분히 안전한 방식을 사용하여 예측 불가능한 난수를 생성해야 합니다. SecureRandom 클래스로 안전한 난수를 생성할 수 있습니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-330

- 소프트웨어 보안약점 진단가이드 2021
  - 적절하지 않은 난수 값 사용

위험한 예시

```
1. public double roledice() {
2.   return Math.random();
3. }
```

라인 2: java.lang.Math 클래스의 random() 메소드는 시드를 재설정할 수 없기 때문에 위험합니다. 또한 시드 재설정이 가능한 java.util.Random 클래스도 여전히 위험합니다.

안전한 예시

```
1. import java.security.SecureRandom;
2. import java.util.Random;
3. import java.util.Date;
4. public int roledice() {
5.   Random numGen = SecureRandom.getInstance("SHA1PRNG");
6.   return (numGen.nextInt(6)) + 1;
7. }
```

라인 6: java.security.SecureRandom 클래스를 통해 직접 시드를 설정하면 java.util.Random을 사용하는 것보다 강력한 예측 불가 난수를 생성할 수 있습니다.

이슈 ID	274605
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	47

소스코드

```
42.       response.setContentType("text/html;charset=UTF-8");
43.
44.       String param = request.getParameter("BenchmarkTest00023");
45.       if (param == null) param = "";
46.
47.       float rand = new java.util.Random().nextFloat();
48.       String rememberMeKey = Float.toString(rand).substring(2); // Trim off the 0.
at the front.
49.
50.       String user = "Floyd";
51.       String fullClassName = this.getClass().getName();
52.       String testCaseNumber =
```

이슈 ID	274672
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	69

소스코드

```
64.
65.     org.owasp.benchmark.helpers.ThingInterface thing =
66.             org.owasp.benchmark.helpers.ThingFactory.createThing();
67.     String bar = thing.doSomething(param);
68.
69.     int r = new java.util.Random().nextInt();
70.     String rememberMeKey = Integer.toString(r);
71.
72.     String user = "Ingrid";
73.     String fullClassName = this.getClass().getName();
74.     String testCaseNumber =
```

이슈 ID	274687
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	72

소스코드

```
67.     // Simple if statement that assigns constant to bar on true condition
68.     int num = 86;
69.     if ((7 * 42) - num > 200) bar = "This_should_always_happen";
70.     else bar = param;
71.
72.     long l = new java.util.Random().nextLong();
73.     String rememberMeKey = Long.toString(l);
74.
75.     String user = "Logan";
76.     String fullClassName = this.getClass().getName();
77.     String testCaseNumber =
```

● [규칙 이름] 취약한 암호화 알고리즘 (높음, Java)

취약한 암호화 알고리즘 체커는 안전하다고 알려져 있지 않은 암호화 알고리즘 사용을 검출합니다.

표준화되지 않은 암호화 알고리즘을 사용하면 공격자가 알고리즘을 분석하여 무력화시킬 수 있는 가능성을 높일 수 있습니다. 오래된 몇몇 암호화 알고리즘의 경우 컴퓨터의 성능이 향상됨에 따라 취약해지기도 합니다. 따라서 예전에는 해독하는데 시간이 걸릴 것으로 예상된 알고리즘이 며칠이나 몇 시간 내에 해독되기도 합니다. 오래되거나 표준화되지 않은 암호화 알고리즘을 사용하면 공격자가 알고리즘을 분석하여 무력화시킬 수 있습니다.

자신만의 암호화 알고리즘을 개발하는 것은 위험하므로 학계 및 업계에서 이미 검증되고 표준화된 알고리즘을 사용해야 합니다. 이미 취약하다고 알려진 DES, RC5 등의 알고리즘 대신 3DES, AES, SEED 등의 안전한 알고리즘을 사용합니다. 또한 표준화된 안전한 알고리즘마다 적정선으로 제시되는 충분한 길이의 키를 사용해야 합니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-327

- 소프트웨어 보안약점 진단가이드 2021
  - 취약한 암호화 알고리즘 사용

위험한 예시

```
1. import java.security.*;
2. import javax.crypto.Cipher;
3. import javax.crypto.NoSuchPaddingException;
4. public class CryptoUtils {
5.   public byte[] encrypt(byte[] msg, Key k) {
6.     byte[] rslt = null;
7.     try {
8.       Cipher c = Cipher.getInstance("DES");
9.       c.init(Cipher.ENCRYPT_MODE, k);
10.        rslt = c.update(msg);
11.    } catch (InvalidKeyException e) {
12.      System.err.println("Exception occured!");
13.    } catch (NoSuchAlgorithmException e) {
14.      System.err.println("Exception occured!");
15.    } catch (NoSuchPaddingException e) {
16.      System.err.println("Exception occured!");
17.    }
```

```
18.     return rslt;
19.   }
20. }
```

라인 8: 취약한 DES 알고리즘으로 암호화를 수행합니다.

안전한 예시

```
1. import java.security.*;
2. import javax.crypto.Cipher;
3. import javax.crypto.NoSuchPaddingException;
4. public class CryptoUtils {
5.   public byte[] encrypt(byte[] msg, Key k) {
6.     byte[] rslt = null;
7.     try {
8.       Cipher c = Cipher.getInstance("AES/CBC/PKCS5Padding");
9.       c.init(Cipher.ENCRYPT_MODE, k);
10.       rslt = c.update(msg);
11.     } catch (InvalidKeyException e) {
12.       System.err.println("Exception occured!");
13.     } catch (NoSuchAlgorithmException e) {
14.       System.err.println("Exception occured!");
15.     } catch (NoSuchPaddingException e) {
16.       System.err.println("Exception occured!");
17.     }
18.     return rslt;
19.   }
20. }
```

라인 8: 안전하다고 알려진 AES 알고리즘을 사용합니다.

이슈 ID	274496
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00005.java
줄 번호	63

소스코드

```
58.       //              };
59.       java.security.SecureRandom random = new java.security.SecureRandom();
60.       byte[] iv = random.generateSeed(8); // DES requires 8 byte keys
61.
62.       try {
63.           javax.crypto.Cipher c = javax.crypto.Cipher.getInstance("DES/CBC
/PKCS5Padding");
64.
65.           // Prepare the cipher to encrypt
66.           javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
67.           java.security.spec.AlgorithmParameterSpec paramSpec =
68.                   new javax.crypto.spec.IvParameterSpec(iv);
```

이슈 ID	274497
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00005.java
줄 번호	66

소스코드

```
61.
62.       try {
63.           javax.crypto.Cipher c = javax.crypto.Cipher.getInstance("DES/CBC
/PKCS5Padding");
64.
65.           // Prepare the cipher to encrypt
66.           javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
67.           java.security.spec.AlgorithmParameterSpec paramSpec =
68.                   new javax.crypto.spec.IvParameterSpec(iv);
69.           c.init(javax.crypto.Cipher.ENCRYPT_MODE, key, paramSpec);
70.
71.           // encrypt and store the results
```

이슈 ID	274508

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	59

소스코드

```
54.      java.security.SecureRandom random = new java.security.SecureRandom();
55.      byte[] iv = random.generateSeed(8); // DES requires 8 byte keys
56.
57.      try {
58.        javax.crypto.Cipher c =
59.            javax.crypto.Cipher.getInstance("DES/CBC/PKCS5Padding", "SunJCE");
60.        // Prepare the cipher to encrypt
61.        javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
62.        java.security.spec.AlgorithmParameterSpec paramSpec =
63.            new javax.crypto.spec.IvParameterSpec(iv);
64.        c.init(javax.crypto.Cipher.ENCRYPT_MODE, key, paramSpec);
```

이슈 ID	274509
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	61

소스코드

```
56.
57.      try {
58.        javax.crypto.Cipher c =
59.            javax.crypto.Cipher.getInstance("DES/CBC/PKCS5Padding", "SunJCE");
60.        // Prepare the cipher to encrypt
61.        javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
62.        java.security.spec.AlgorithmParameterSpec paramSpec =
63.            new javax.crypto.spec.IvParameterSpec(iv);
```

```
64.         c.init(javax.crypto.Cipher.ENCRYPT_MODE, key, paramSpec);
65.
66.         // encrypt and store the results
```

이슈 ID	274522
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	60

소스코드

```
55.      java.security.SecureRandom random = new java.security.SecureRandom();
56.      byte[] iv = random.generateSeed(8); // DES requires 8 byte keys
57.
58.      try {
59.         javax.crypto.Cipher c =
60.                javax.crypto.Cipher.getInstance("DES/CBC/PKCS5Padding", "SunJCE");
61.         // Prepare the cipher to encrypt
62.         javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
63.         java.security.spec.AlgorithmParameterSpec paramSpec =
64.                new javax.crypto.spec.IvParameterSpec(iv);
65.         c.init(javax.crypto.Cipher.ENCRYPT_MODE, key, paramSpec);
```

이슈 ID	274523
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	62

소스코드

```
57.
58.      try {
59.         javax.crypto.Cipher c =
60.                javax.crypto.Cipher.getInstance("DES/CBC/PKCS5Padding", "SunJCE");
```

```
61.         // Prepare the cipher to encrypt
62.         javax.crypto.SecretKey key = javax.crypto.KeyGenerator.getInstance("DES").
generateKey();
63.         java.security.spec.AlgorithmParameterSpec paramSpec =
64.             new javax.crypto.spec.IvParameterSpec(iv);
65.         c.init(javax.crypto.Cipher.ENCRYPT_MODE, key, paramSpec);
66.
67.         // encrypt and store the results
```

이슈 ID	274534
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	73

소스코드

```
68.     int num = 106;
69.
70.     bar = (7 * 42) - num > 200 ? "This should never happen" : param;
71.
72.     try {
73.         java.security.MessageDigest md = java.security.MessageDigest.getInstance
("SHA1", "SUN");
74.         byte[] input = {(byte) '?'};
75.         Object inputParam = bar;
76.         if (inputParam instanceof String) input = ((String) inputParam).getBytes();
77.         if (inputParam instanceof java.io.InputStream) {
78.             byte[] strInput = new byte[1000];
```

● [규칙 이름] SQL 삽입 (높음, Java)

SQL 삽입 체커는 검증되지 않은 외부 입력값을 사용하는 SQL 쿼리를 검출합니다.

데이터베이스(DB)와 연동된 웹 애플리케이션에서 입력된 데이터에 대한 유효성 검증을 하지 않을 경우 공격자가 입력 폼 및 URL 입력란에 SQL 문을 삽입하여 DB로부터 정보를 열람하거나 조작할 수 있습니다.

PreparedStatement 객체 등을 사용하여 DB에 컴파일된 쿼리문(상수)을 전달하는 방법을 사용합니다. PreparedStatement를 사용하면 DB 쿼리에 사용되는 외부 입력값에 대한 특수문자 및 쿼리 예약어를 필터링할 수 있습니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-89

- 소프트웨어 보안약점 진단가이드 2021
  - SQL 삽입

위험한 예시

```
1. String query = "SELECT account_balance FROM"
2.                + "user_data WHERE user_name = "
3.                + request.getParameter("customerName");
4. try {
5.   Statement statement = connection.createStatement( ... );
6.   ResultSet results = statement.executeQuery(query);
7. }
```

라인 3: 외부에서 입력된 값 request.getParameter("customerName")이 적절한 검증 없이 query에 포함됩니다.
라인 6: query가 statement.executeQuery()의 인자로 전달됩니다. 이를 통해 공격자가 데이터베이스에 명령을 수행할 수 있습니다.

안전한 예시

```
1. String custname = request.getparameter("customerName"); // Verification required
2. // perform input validation to detect attacks
3. String query = "SELECT account_balance FROM user_data WHERE user_name = ?";
4. PreparedStatement pstmt = connection.prepareStatement(query);
5. pstmt.setString(1, custname);
6. ResultSet results = pstmt.executeQuery();
```

라인 5: 공격자가 SQL 명령을 삽입하더라도 preparedStatement()가 쿼리의 목적을 변경하는 것을 방지합니다.

이슈 ID	274593
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00018.java
줄 번호	59

소스코드

```
54.        String sql = "INSERT INTO users (username, password) VALUES ('foo','" +
param + "')";
55.
56.        try {
57.            java.sql.Statement statement =
58.                    org.owasp.benchmark.helpers.DatabaseHelper.getSqlStatement();
59.            int count = statement.executeUpdate(sql);
60.            org.owasp.benchmark.helpers.DatabaseHelper.outputUpdateComplete
(sql, response);
61.        } catch (java.sql.SQLException e) {
62.            if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
63.                response.getWriter().println("Error processing request.");
64.                return;
```

이슈 ID	274612
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00034.java
줄 번호	56

소스코드

```
51.        String sql = "SELECT * from USERS where USERNAME='foo' and
PASSWORD='" + param + "'";
52.
53.        try {
54.            java.sql.Statement statement =
55.                    org.owasp.benchmark.helpers.DatabaseHelper.getSqlStatement();
56.            statement.execute(sql, java.sql.Statement.RETURN_GENERATED_KEYS);
57.            org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
```

```
response);
58.         } catch (java.sql.SQLException e) {
59.             if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
60.                 response.getWriter().println("Error processing request.");
61.                 return;
```

● [규칙 이름] final 한정자 없는 중요 public 변수 (높음, Java)

final 한정자 없는 중요 public 변수 체커는 임의로 변경되면 안되는 중요한, final 없이 선언된 public static 멤버 변수를 검출합니다.

변수의 중요도 여부를 작성자의 의도 없이 기계적으로 판단하기 어려우므로, 일반적으로 전역 상수의 역할을 할 가능성이 높은 원시 타입 변수들을 중요한 변수로 간주합니다.

final로 선언하지 않으면 변수의 값을 외부에서 임의로 변경할 수 있습니다.

원시 타입 변수를 public static으로 선언할 경우 final 한정자를 함께 부여합니다. 값이 변할 수 있는 중요 변수는 public으로 공개하지 말고 정적 메소드를 통해 접근하도록 설계합니다.

- CWE 660 4.14
  - 493 - Critical Public Variable Without Final Modifier
  - 500 - Public Static Field Not Marked Final

- CWE 660 4.7
  - Critical Public Variable Without Final Modifier - (493)
  - Public Static Field Not Marked Final - (500)

위험한 예시

```
1. public final class myClass extends AppleIt{
2.   // var field is not final
3.   // value of var can be modified from external source.
4.   public static int var = 20;
5.   public int getTotal(int n){
6.   return var * n;
7. }
```

라인 4: public static으로 선언된 var 변수에 final 변경자가 지정되지 않았습니다.

안전한 예시

```
1. public final class myClass extends AppleIt{
2.   // declare variables using keyword final
3.   // if it should not be altered
4.   public static final int var = 20;
5.   public int getTotal(int n){
6.   return var * n;
7. }
```

라인 4: 중요한 public 변수에는 final 변경자를 함께 지정해야 합니다.

이슈 ID	274458
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /DatabaseHelper.java
줄 번호	37

소스코드

```
32. import org.owasp.benchmark.service.pojo.XMLMessage;
33. import org.owasp.esapi.ESAPI;
34.
35. public class DatabaseHelper {
36.     private static Connection conn;
37.     public static org.springframework.jdbc.core.JdbcTemplate JDBCtemplate;
38.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtil =
39.             new org.owasp.benchmark.helpers.HibernateUtil(false);
40.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtilClassic =
41.             new org.owasp.benchmark.helpers.HibernateUtil(true);
42.     public static final boolean hideSQLErrors =
```

이슈 ID	274459
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers

파일	/DatabaseHelper.java
줄 번호	38

소스코드

```
33. import org.owasp.esapi.ESAPI;
34.
35. public class DatabaseHelper {
36.     private static Connection conn;
37.     public static org.springframework.jdbc.core.JdbcTemplate JDBCtemplate;
38.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtil =
39.             new org.owasp.benchmark.helpers.HibernateUtil(false);
40.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtilClassic =
41.             new org.owasp.benchmark.helpers.HibernateUtil(true);
42.     public static final boolean hideSQLErrors =
43.             false; // If we want SQL Exceptions to be suppressed from being displayed
to the user of
```

이슈 ID	274460
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /DatabaseHelper.java
줄 번호	40

소스코드

```
35. public class DatabaseHelper {
36.     private static Connection conn;
37.     public static org.springframework.jdbc.core.JdbcTemplate JDBCtemplate;
38.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtil =
39.             new org.owasp.benchmark.helpers.HibernateUtil(false);
40.     public static org.owasp.benchmark.helpers.HibernateUtil hibernateUtilClassic =
41.             new org.owasp.benchmark.helpers.HibernateUtil(true);
42.     public static final boolean hideSQLErrors =
43.             false; // If we want SQL Exceptions to be suppressed from being displayed
to the user of
44.     // the web app.
45.
```

● [규칙 이름] 명령어 삽입 (높음, Java)

명령어 삽입 체커는 검증되지 않은 외부 입력값을 포함한 시스템 내부 명령어를 실행하는 코드를 검출합니다.

적절한 검증 절차를 거치지 않은 사용자 입력값을 운영 체제 명령어의 일부 또는 전부에 사용하는 경우 의도하지 않은 시스템 명령어가 실행되어 부적절하게 권한이 변경되거나 시스템 동작 및 운영에 장애가 발생할 수 있습니다.

외부 입력에 따라 명령어를 생성하거나 명령어를 생성하는 데 필요한 안전한 값을 미리 지정하고 외부 입력에 따라 선택하여 사용합니다.

- OWASP 2017
  - A1-Injection

- OWASP 2021
  - A03 Injection

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-78

- 소프트웨어 보안약점 진단가이드 2021
  - 운영체제 명령어 삽입

위험한 예시

```
1. public void foo() throws IOException{
2.   Properties props  = new Properties();
3.   String filename = "file_list";
4.   FileInputStream in = new FileInputStream(fileName);
5.   props.load(in);
6.   String version = props.getProperty("dir_type");
7.   // Unusual behavior if dir_type is an unintended string
```

```
8.  String cmd = new String("cmd.exe /K  ₩"rmanDB.bat ₩"");
9.  Runtime.getRuntime().exec(cmd + "c:₩₩prog_cmd₩₩" + version);
10. }
```

라인 9: 외부 입력값 props.getProperty("dir_type")을 적절한 검증 없이 명령 실행에 사용하고 있습니다.

안전한 예시

```
1. public void foo() throws IOException{
2.  Properties props  = new Properties();
3.  String filename = "file_list";
4.  FileInputStream in = new FileInputStream(fileName);
5.  props.load(in);
6.  String version[] = {"1.0", "1.0.1", "1.11", "1.4"};
7.  int versionSelection = Integer.parseInt(props.getProperty("version"));
8.  String cmd = new String("cmd.exe /K  ₩"rmanDB.bat ₩"");
9.  String vs = "";
10.  if(versionSelection == 0)
11.    vs = version[0];
12.  else if(versionSelection == 1)
13.    vs = version[1];
14.  else if(versionSelection == 2)
15.    vs = version[2];
16.  else if(versionSelection == 3)
17.    vs = version[3];
18.  else
19.    vs = vsersion[3];
20.  Runtime.getRuntime().exec(cmd + "c:₩₩prog_cmd₩₩" + vs);
21. }
```

라인 10: 외부 입력값 props.getProperty("dir_type")으로 명령을 생성하기 전에 의도된 형식인지 확인하고 입력에 따라 미리 생성해둔 version 배열에서 선택한 값을 명령에 사용합니다.

이슈 ID	274580
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00006.java

줄 번호 66

소스코드

```
61.     }
62.     argList.add("echo " + param);
63.
64.     ProcessBuilder pb = new ProcessBuilder();
65.
66.     pb.command(argList);
67.
68.     try {
69.         Process p = pb.start();
70.         org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
71.     } catch (IOException e) {
```

이슈 ID 274581

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00007.java

줄 번호 61

소스코드

```
56.     String[] argsEnv = {param};
57.
58.     Runtime r = Runtime.getRuntime();
59.
60.     try {
61.         Process p = r.exec(args, argsEnv);
62.         org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
63.     } catch (IOException e) {
64.         System.out.println("Problem executing cmdi - TestCase");
65.         response.getWriter()
66.                 .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
```

이슈 ID	274592
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00017.java
줄 번호	63

소스코드

```
58.      }
59.
60.      Runtime r = Runtime.getRuntime();
61.
62.      try {
63.          Process p = r.exec(cmd + param);
64.          org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
65.      } catch (IOException e) {
66.          System.out.println("Problem executing cmdi - TestCase");
67.          response.getWriter()
68.                  .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
```

이슈 ID	274669
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00077.java
줄 번호	98

소스코드

```
93.          argList.add("sh");
94.          argList.add("-c");
95.      }
96.      argList.add("echo " + bar);
97.
98.      ProcessBuilder pb = new ProcessBuilder(argList);
99.
100.     try {
101.         Process p = pb.start();
102.         org.owasp.benchmark.helpers.Utils.printOSCommandResults(p,
```

```
response);
103.        } catch (IOException e) {
```

이슈 ID	274697
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00092.java
줄 번호	95

소스코드

```
90.         String[] argsEnv = {bar};
91.
92.         Runtime r = Runtime.getRuntime();
93.
94.         try {
95.             Process p = r.exec(args, argsEnv);
96.             org.owasp.benchmark.helpers.Utils.printOSCommandResults(p, response);
97.         } catch (IOException e) {
98.             System.out.println("Problem executing cmdi - TestCase");
99.             response.getWriter()
100.                    .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(e.
getMessage()));
```

● [규칙 이름] LDAP 삽입 (높음, Java)

LDAP 삽입 체커는 검증되지 않은 외부 입력값으로 생성된 LDAP 쿼리문을 검출합니다.

공격자가 외부 입력을 통해서 의도하지 않은 LDAP(Lightweight Directory Access Protocol) 명령어를 수행할 수 있습니다. 즉, 웹 애플리케이션이 사용자가 제공한 입력을 올바르게 처리하지 못하면 공격자가 LDAP 명령문의 구성을 바꿀 수 있습니다. 이로 인해 프로세스가 명령을 실행한 컴포넌트와 동일한 권한을 가지고 동작하게 됩니다.

DN(Distinguished Name)과 필터에 사용되는 사용자 입력값에는 특수문자가 포함되지 않도록 특수문자를 제거해야 합니다. 만약 특수문자를 허용해야 하는 경우라면 특수문자(=, +, <, >, #, ;, 등)가 실행 명령이 아닌 일반문자로 인식되도록 처리합니다.

- 소프트웨어 보안약점 진단가이드 2021

■ LDAP 삽입

위험한 예시

```
1. private void searchRecord(String userSN, String userPassword) throws
NamingException {
2.   Hashtable<String, String> env = new Hashtable<String, String>();
3.   env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.
LdapCtxFactory");
4.   try {
5.     DirContext dctx = new InitialDirContext(env);
6.     SearchControls sc = new SearchControls();
7.     String[] attributeFilter = { "cn", "mail" };
8.     sc.setReturningAttributes(attributeFilter);
9.     sc.setSearchScope(SearchControls.SUBTREE_SCOPE);
10.      String base = "dc=example,dc=com";
11.      String filter = "(&(sn=" + userSN + ")(userPassword=" + userPassword + "))";
12.      NamingEnumeration<?> results = dctx.search(base, filter, sc);
13.      while (results.hasMore()) {
14.        SearchResult sr = (SearchResult) results.next();
15.        Attributes attrs = sr.getAttributes();
16.        Attribute attr = attrs.get("cn");
17.        ...
18.      }
19.      dctx.close();
20.   } catch (NamingException e) { ... }
21. }
```

라인 11: userSN과 userPassword의 변수 값으로 *를 전달할 경우 필터 문자열의 조건식은 항상 참이 되며, 이는 의도하지 않은 동작을 발생시킬 수 있습니다.

안전한 예시

```
1. private void searchRecord(String userSN, String userPassword) throws
NamingException {
2.   Hashtable<String, String> env = new Hashtable<String, String>();
3.   env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.
LdapCtxFactory");
4.   try {
5.     DirContext dctx = new InitialDirContext(env);
```

```
6.    SearchControls sc = new SearchControls();
7.    String[] attributeFilter = { "cn", "mail" };
8.    sc.setReturningAttributes(attributeFilter);
9.    sc.setSearchScope(SearchControls.SUBTREE_SCOPE);
10.    String base = "dc=example,dc=com";
11.    if (!userSN.matches("[₩₩w₩₩s]*") || !userPassword.matches("[₩₩w]*")) {
12.      throw new IllegalArgumentException("Invalid input");
13.    }
14.    String filter = "(&(sn=" + userSN + ")(userPassword=" + userPassword + "))";
15.    NamingEnumeration<?> results = dctx.search(base, filter, sc);
16.    while (results.hasMore()) {
17.      SearchResult sr = (SearchResult) results.next();
18.      Attributes attrs = sr.getAttributes();
19.      Attribute attr = attrs.get("cn");
20.      ...
21.    }
22.    dctx.close();
23.  } catch (NamingException e) { ... }
24. }
```

라인 11: 검색을 위한 필터 문자열로 사용되는 외부 입력에서 위험한 문자열을 제거하여 위험성을 부분적으로 감소 시킬 수 있습니다.

이슈 ID	274588
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00012.java
줄 번호	68

소스코드

```
63.          javax.naming.directory.DirContext ctx = ads.getDirContext();
64.          javax.naming.directory.InitialDirContext idc =
65.                (javax.naming.directory.InitialDirContext) ctx;
66.          boolean found = false;
67.          javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
68.                idc.search(base, filter, filters, sc);
69.          while (results.hasMore()) {
```

```
70.             javax.naming.directory.SearchResult sr =
71.                 (javax.naming.directory.SearchResult) results.next();
72.             javax.naming.directory.Attributes attrs = sr.getAttributes();
73.
```

이슈 ID	274597
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00021.java
줄 번호	59

소스코드

```
54.         String filter = "(&(objectclass=person))(|(uid=" + param + ")(street={0}))";
55.         Object[] filters = new Object[] {"The streetz 4 Ms bar"};
56.         // System.out.println("Filter " + filter);
57.         boolean found = false;
58.         javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
59.             ctx.search(base, filter, filters, sc);
60.         while (results.hasMore()) {
61.           javax.naming.directory.SearchResult sr =
62.                 (javax.naming.directory.SearchResult) results.next();
63.           javax.naming.directory.Attributes attrs = sr.getAttributes();
64.
```

이슈 ID	274623
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00044.java
줄 번호	60

소스코드

```
55.         sc.setSearchScope(javax.naming.directory.SearchControls.
SUBTREE_SCOPE);
56.         String filter = "(&(objectclass=person)(uid=" + param + "))";
```

```
57.         // System.out.println("Filter " + filter);
58.         boolean found = false;
59.         javax.naming.NamingEnumeration<javax.naming.directory.SearchResult>
results =
60.                 ctx.search(base, filter, sc);
61.         while (results.hasMore()) {
62.             javax.naming.directory.SearchResult sr =
63.                     (javax.naming.directory.SearchResult) results.next();
64.             javax.naming.directory.Attributes attrs = sr.getAttributes();
65.
```

● [규칙 이름] 자원 삽입 (높음, Java)

자원 삽입 체커는 검증되지 않은 외부 입력값으로 생성된 리소스(resource) 식별자를 검출합니다.

검증되지 않은 외부 입력값을 통해 파일 및 서버 등 시스템 리소스에 대한 접근 혹은 식별을 허용할 경우에 입력값 조작을 통해 시스템이 보호하는 리소스에 임의로 접근할 수 있습니다. 리소스 삽입 약점을 사용하여 공격자는 리소스의 수정/삭제, 시스템 정보 누출, 시스템 리소스 간 충돌로 인한 서비스 장애 등을 발생시킬 수 있습니다.

외부 입력을 리소스(소켓의 포트 등)의 식별자로 사용하는 경우에는 사전에 정의된 적합한 리스트에서 선택되도록 합니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-99

- 소프트웨어 보안약점 진단가이드 2021
  - 경로 조작 및 자원 삽입

위험한 예시

```
1. public void foo() {
2.   ServerSocket serverSocket;
3.   Properties props = new Properties();
4.   String filename = "file_list";
5.   FileInputStream in = new FileInputStream(fileName);
6.   props.load(in);
```

```
7.  String service = props.getProperty("Service No");
8.  int port = Integer.parseInt(service);
9.  // if wrong service value is used, crash with
10.  // port number
11.  if(port != 0)
12.    serverSocket = new ServerSocket(port);
13.  else
14.     serverSocket = new ServerSocket(4000);
15. }
```

라인 12: 외부에서 전달 받은 소켓 번호를 검증 없이 그대로 사용했습니다. 만약 공격자가 Service No의 값으로 80을 지정하면, 기존의 80포트에서 구동하는 서비스와 충돌하는 문제를 발생시킬 수 있습니다.

안전한 예시

```
1. public void foo() {
2.  ServerSocket serverSocket;
3.  Properties props = new Properties();
4.  String filename = "file_list";
5.  FileInputStream in = new FileInputStream(fileName);
6.  String service = "";
7.  if(in != null && in.available() > 0) {
8.     props.load(in);
9.     service = props.getProperty("Service No");
10.  }
11.  if("".equals(service)) service= "8080";
12.  int port = Integer.parseInt(service);
13.  switch(port) {
14.  case 1:
15.    port = 3001; break;
16.  case 2:
17.    port = 3002; break;
18.  case 3:
19.    port = 3003; break;
20.  default:
21.    port = 3003;
22.  }
23.  serverSocket = new ServerSocket(port);
24. }
```

라인 13: 외부 입력값에 따라 미리 정해진 포트를 선택하도록 함으로써 공격자가 임의의 포트를 사용하지 못하도록 합니다.

이슈 ID	274563
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	71

소스코드

```
66.  String fileName = null;
67.  java.io.FileInputStream fis = null;
68.
69.  try {
70.     fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.     fis = new java.io.FileInputStream(new java.io.File(fileName));
72.     byte[] b = new byte[1000];
73.     int size = fis.read(b);
74.     response.getWriter()
75.           .println(
76.                 "The beginning of file: '"
```

이슈 ID	274566
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	71

소스코드

```
66.  String fileName = null;
67.  java.io.FileInputStream fis = null;
68.
69.  try {
70.     fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.     fis = new java.io.FileInputStream(new java.io.File(fileName));
```

```
72.         byte[] b = new byte[1000];
73.         int size = fis.read(b);
74.         response.getWriter()
75.                 .println(
76.                         "The beginning of file: '"
```

이슈 ID	274571
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	72

소스코드

```
67.      java.io.FileOutputStream fos = null;
68.
69.      try {
70.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.
72.         fos = new java.io.FileOutputStream(fileName, false);
73.         response.getWriter()
74.                 .println(
75.                         "Now ready to write to file: "
76.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
77.
```

이슈 ID	274586
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00011.java
줄 번호	54

소스코드

```
49.      }
50.
```

```
51.       // URL Decode the header value since req.getHeaders() doesn't. Unlike req.
getParameters().
52.       param = java.net.URLDecoder.decode(param, "UTF-8");
53.
54.       java.io.File fileTarget = new java.io.File(param, "/Test.txt");
55.       response.getWriter()
56.               .println(
57.                       "Access to file: '"
58.                               + org.owasp
59.                                       .esapi
```

이슈 ID	274610
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00028.java
줄 번호	57

소스코드

```
52.       java.io.FileOutputStream fos = null;
53.
54.       try {
55.           fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
56.
57.           fos = new java.io.FileOutputStream(fileName, false);
58.           response.getWriter()
59.                   .println(
60.                           "Now ready to write to file: "
61.                                   + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
62.
```

이슈 ID	274628
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00061.java
줄 번호	74

소스코드

```
69.                     org.apache.commons.codec.binary.Base64.decodeBase64(
70.                         org.apache.commons.codec.binary.Base64.encodeBase64(
71.                             param.getBytes())));
72.     }
73.
74.     java.io.File fileTarget = new java.io.File(bar, "/Test.txt");
75.     response.getWriter()
76.             .println(
77.                     "Access to file: '"
78.                             + org.owasp
79.                                     .esapi
```

이슈 ID	274635
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	77

소스코드

```
72.     String fileName = null;
73.     java.io.FileInputStream fis = null;
74.
75.     try {
76.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
77.         fis = new java.io.FileInputStream(new java.io.File(fileName));
78.         byte[] b = new byte[1000];
79.         int size = fis.read(b);
80.         response.getWriter()
81.                 .println(
82.                         "The beginning of file: '"
```

이슈 ID	274636
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode

파일	/BenchmarkTest00062.java
줄 번호	77

소스코드

```
72.     String fileName = null;
73.     java.io.FileInputStream fis = null;
74.
75.     try {
76.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
77.         fis = new java.io.FileInputStream(new java.io.File(fileName));
78.         byte[] b = new byte[1000];
79.         int size = fis.read(b);
80.         response.getWriter()
81.                 .println(
82.                         "The beginning of file: '"
```

이슈 ID	274645
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00065.java
줄 번호	79

소스코드

```
74.     String fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
75.     java.io.InputStream is = null;
76.
77.     try {
78.         java.nio.file.Path path = java.nio.file.Paths.get(fileName);
79.         is = java.nio.file.Files.newInputStream(path, java.nio.file.
StandardOpenOption.READ);
80.         byte[] b = new byte[1000];
81.         int size = is.read(b);
82.         response.getWriter()
83.                 .println(
84.                         "The beginning of file: '"
```

● [규칙 이름] 경로 조작 (높음, Java)

경로 조작 체커는 검증되지 않은 외부 입력값으로 생성된 경로를 통해 파일 시스템에 접근하는 코드를 검출합니다.

검증되지 않은 외부 입력값을 통해 파일 및 서버 등 시스템 리소스에 접근하도록 허용하는 경우 공격자가 입력값을 조작하여 의도하지 않은 경로에 임의로 접근할 수 있습니다. 즉, 공격자가 경로 조작을 통해 허용되지 않은 권한을 획득하여 설정에 관련된 파일을 변경하거나 실행시킬 수 있습니다.

외부 입력을 리소스(파일 등)의 식별자로 사용하기 위해 적절한 검증을 거치도록 합니다. 특히, 외부 입력이 파일명인 경우에는 경로 순회(directory traversal) 공격의 위험이 있는 문자(", /, , .. 등)를 제거하는 필터를 사용합니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-22

- 소프트웨어 보안약점 진단가이드 2021
  - 경로 조작 및 자원 삽입

위험한 예시

```
1. public void foo(Properties request){
2.   String name = request.getProperty("filename");
3.   if(name!=null){
4.     File file = new File("/usr/local/tmp/" + name);
5.     // if other file names comes into name, harmful
6.     file.delete();
7.   }
8. }
```

라인 4: 외부 입력값 request.getProperty("filename")을 적절한 검증 없이 파일 접근에 사용하고 있습니다.

안전한 예시

```
1. public void foo(Properties request){
2.   String name = request.getProperty("filename");
3.   if(name!=null && !"".equeals(name)){
```

```
4.    name = name.replaceAll("/","");
5.    name = name.replaceAll("₩₩","");
6.    name = name.replaceAll(".","");
7.    name = name.replaceAll("&","");
8.    name = name + "-report";
9.    File file = new file("/usr/local/tmp/" + name);
10.     if(file != null)  file.delete();
11.  }
12. }
```

라인 4: 외부 입력값 request.getProperty("filename")으로 파일에 접근하기 전에, 공격 위험이 있는 특수 문자를 제거할 수 있는 필터를 사용합니다.

이슈 ID	274565
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	71

소스코드

```
66.     String fileName = null;
67.     java.io.FileInputStream fis = null;
68.
69.     try {
70.        fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.        fis = new java.io.FileInputStream(new java.io.File(fileName));
72.        byte[] b = new byte[1000];
73.        int size = fis.read(b);
74.        response.getWriter()
75.              .println(
76.                    "The beginning of file: '"
```

이슈 ID	274567
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java

줄 번호 71

소스코드

```
66.     String fileName = null;
67.     java.io.FileInputStream fis = null;
68.
69.     try {
70.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.         fis = new java.io.FileInputStream(new java.io.File(fileName));
72.         byte[] b = new byte[1000];
73.         int size = fis.read(b);
74.         response.getWriter()
75.                 .println(
76.                         "The beginning of file: '"
```

이슈 ID 274570

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java

줄 번호 72

소스코드

```
67.     java.io.FileOutputStream fos = null;
68.
69.     try {
70.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.
72.         fos = new java.io.FileOutputStream(fileName, false);
73.         response.getWriter()
74.                 .println(
75.                         "Now ready to write to file: "
76.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
77.
```

이슈 ID	274587
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00011.java
줄 번호	54

[소스코드]

```
49.     }
50.
51.     // URL Decode the header value since req.getHeaders() doesn't. Unlike req.
getParameters().
52.     param = java.net.URLDecoder.decode(param, "UTF-8");
53.
54.     java.io.File fileTarget = new java.io.File(param, "/Test.txt");
55.     response.getWriter()
56.             .println(
57.                     "Access to file: '"
58.                             + org.owasp
59.                                     .esapi
```

이슈 ID	274609
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00028.java
줄 번호	57

[소스코드]

```
52.     java.io.FileOutputStream fos = null;
53.
54.     try {
55.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
56.
57.         fos = new java.io.FileOutputStream(fileName, false);
58.         response.getWriter()
59.                 .println(
60.                         "Now ready to write to file: "
61.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
```

```
(fileName));
62.
```

이슈 ID	274627
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00061.java
줄 번호	74

소스코드

```
69.                     org.apache.commons.codec.binary.Base64.decodeBase64(
70.                         org.apache.commons.codec.binary.Base64.encodeBase64(
71.                             param.getBytes())));
72.     }
73.
74.     java.io.File fileTarget = new java.io.File(bar, "/Test.txt");
75.     response.getWriter()
76.             .println(
77.                     "Access to file: '"
78.                         + org.owasp
79.                             .esapi
```

이슈 ID	274633
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	77

소스코드

```
72.     String fileName = null;
73.     java.io.FileInputStream fis = null;
74.
75.     try {
76.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
77.         fis = new java.io.FileInputStream(new java.io.File(fileName));
```

```
78.         byte[] b = new byte[1000];
79.         int size = fis.read(b);
80.         response.getWriter()
81.                 .println(
82.                         "The beginning of file: '"
```

이슈 ID	274634
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	77

소스코드

```
72.      String fileName = null;
73.      java.io.FileInputStream fis = null;
74.
75.      try {
76.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
77.         fis = new java.io.FileInputStream(new java.io.File(fileName));
78.         byte[] b = new byte[1000];
79.         int size = fis.read(b);
80.         response.getWriter()
81.                 .println(
82.                         "The beginning of file: '"
```

이슈 ID	274644
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	79

소스코드

```
74.      String fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
75.      java.io.InputStream is = null;
76.
```

```
77.         try {
78.             java.nio.file.Path path = java.nio.file.Paths.get(fileName);
79.             is = java.nio.file.Files.newInputStream(path, java.nio.file.
StandardOpenOption.READ);
80.             byte[] b = new byte[1000];
81.             int size = is.read(b);
82.             response.getWriter()
83.                     .println(
84.                             "The beginning of file: '"
```

● [규칙 이름] HTTP 응답 분할 (보통, Java)

HTTP 응답 분할 체커는 검증되지 않은 외부 입력값으로 생성된 HTTP 응답을 검출합니다.

HTTP 요청에 들어 있는 인자 값이 HTTP 응답 헤더에 포함되어 사용자에게 다시 전달될 때, 입력값에 CR(Carriage Return) 이나 LF(Line Feed)와 같은 개행문자가 존재하면 HTTP 응답이 2개 이상으로 분리될 수 있습니다. 이 경우 공격자는 개행문자를 사용하여 첫 번째 응답을 종료시키고, 두 번째 응답에 악의적인 코드를 주입하여 XSS 및 캐시 훼손(Cache Poisoning) 공격 등을 수행할 수 있습니다.

요청 매개 변수의 값을 HTTP 응답 헤더(예, Set-Cookie 등)에 포함시킬 경우 CR, LF와 같은 개행문자를 제거해야 합니다.

- 소프트웨어 보안약점 진단가이드 2021
  - HTTP 응답분할

위험한 예시

```
1. ...
2. String lastLogin = request.getParameter("last_login");
3. if (lastLogin == null || "".equals(lastLogin)) {
4.   return;
5. }
6. Cookie c = new Cookie("LASTLOGIN", lastLogin);
7. c.setMaxAge(1000);
8. c.setSecure(true);
9. response.addCookie(c);
10. response.setContentType("text/html");
11. ...
```

라인 6: 외부에서 입력된 값 lastLogin으로 쿠키의 값을 설정하고 있습니다. 공격자가 "Wiley Hacker /r/nHTTP/1.1 200 OK/r/n" 으로 lastLogin 값을 설정하면, 응답이 분리되어 전달되며 분리된 응답 본문의 내용을 공격자가 마음대로 수정할 수 있습니다.

안전한 예시

```
1. ...
2. String lastLogin = request.getParameter("last_login");
3. if (lastLogin == null || "".equals(lastLogin)) {
4.   return;
5. }
6. lastLogin = lastLogin.replaceAll("[₩₩r₩₩n]", "");
7. Cookie c = new Cookie("LASTLOGIN", lastLogin);
8. c.setMaxAge(1000);
9. c.setSecure(true);
10. response.addCookie(c);
11. response.setContentType("text/html");
12. ...
```

라인 6: 응답이 여러 개로 나눠지는 것을 방지하기 위해 개행 문자를 제거한 후 응답 헤더의 값으로 사용합니다.

이슈 ID	274607
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	83

소스코드

```
78.         rememberMe.setHttpOnly(true);
79.         rememberMe.setDomain(new java.net.URL(request.getRequestURL().
toString()).getHost());
80.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
81.         // e.g., /benchmark/sql-01/BenchmarkTest01001
82.         request.getSession().setAttribute(cookieName, rememberMeKey);
83.         response.addCookie(rememberMe);
84.         response.getWriter()
```

```
85.                 .println(
86.                         user
87.                                 + " has been remembered with cookie: "
88.                                 + rememberMe.getName()
```

이슈 ID	274617
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	86

소스코드

```
81.             rememberMe.setSecure(true);
82.             rememberMe.setHttpOnly(true);
83.             rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
84.             // e.g., /benchmark/sql-01/BenchmarkTest01001
85.             request.getSession().setAttribute(cookieName, rememberMeKey);
86.             response.addCookie(rememberMe);
87.             response.getWriter()
88.                     .println(
89.                             user
90.                                     + " has been remembered with cookie: "
91.                                     + rememberMe.getName()
```

이슈 ID	274649
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	125

소스코드

```
120.             rememberMe.setHttpOnly(true);
121.             rememberMe.setDomain(new java.net.URL(request.getRequestURL().toString()).getHost());
```

```
122.            rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
123.            // e.g., /benchmark/sql-01/BenchmarkTest01001
124.            request.getSession().setAttribute(cookieName, rememberMeKey);
125.            response.addCookie(rememberMe);
126.            response.getWriter()
127.                    .println(
128.                            user
129.                                    + " has been remembered with cookie: "
130.                                    + rememberMe.getName()
```

이슈 ID	274674
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	104

소스코드

```
99.             rememberMe.setSecure(true);
100.            rememberMe.setHttpOnly(true);
101.            rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
102.            // e.g., /benchmark/sql-01/BenchmarkTest01001
103.            request.getSession().setAttribute(cookieName, rememberMeKey);
104.            response.addCookie(rememberMe);
105.            response.getWriter()
106.                    .println(
107.                            user
108.                                    + " has been remembered with cookie: "
109.                                    + rememberMe.getName()
```

이슈 ID	274682
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	108

소스코드

```
103.            rememberMe.setSecure(true);
104.            rememberMe.setHttpOnly(true);
105.            rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
106.            // e.g., /benchmark/sql-01/BenchmarkTest01001
107.            request.getSession().setAttribute(cookieName, rememberMeKey);
108.            response.addCookie(rememberMe);
109.            response.getWriter()
110.                    .println(
111.                            user
112.                                    + " has been remembered with cookie: "
113.                                    + rememberMe.getName()
```

이슈 ID	274694
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00087.java
줄 번호	89

소스코드

```
84.                 return;
85.             }
86.             str = new String(input, 0, i);
87.         }
88.         if ("".equals(str)) str = "No cookie value supplied";
89.         javax.servlet.http.Cookie cookie = new javax.servlet.http.Cookie
("SomeCookie", str);
90.
91.         cookie.setSecure(false);
92.         cookie.setHttpOnly(true);
93.         cookie.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
94.         // e.g., /benchmark/sql-01/BenchmarkTest01001
```

이슈 ID	274693
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00087.java
줄 번호	95

소스코드

```
90.
91.     cookie.setSecure(false);
92.     cookie.setHttpOnly(true);
93.     cookie.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
94.     // e.g., /benchmark/sql-01/BenchmarkTest01001
95.     response.addCookie(cookie);
96.
97.     response.getWriter()
98.             .println(
99.                     "Created cookie: 'SomeCookie': with value: '"
100.                            + org.owasp.esapi.ESAPI.encoder().encodeForHTML(str)
```

● [규칙 이름] XPath 삽입 (높음, Java)

XPATH 삽입 체커는 검증되지 않은 외부 입력값이 포함된 XPath 쿼리문을 검출합니다.

외부 입력값을 적절한 검사 과정 없이 XPath 쿼리문 생성을 위한 문자열로 사용하면, 공격자는 프로그래머가 의도하지 않았던 문자열을 전달하여 쿼리문의 의미를 왜곡시키거나 그 구조를 변경하고 임의의 쿼리를 실행하여 인가되지 않은 데이터를 열람할 수 있습니다.

XPath 쿼리에 사용되는 외부 입력값에 대하여 특수문자(", [,], /, =, @ 등) 및 쿼리 예약어 필터링을 수행하고 매개 변수화된 쿼리문을 지원하는 XQuery를 사용해야 합니다.

- OWASP 2017
  - A1-Injection

- OWASP 2021
  - A03 Injection

- 소프트웨어 보안약점 진단가이드 2021
  - XML 삽입

위험한 예시

```
1. String nm = props.getProperty("name");
2. String pw = props.getProperty("password");
3. ...
4. XPathFactory factory = XPathFactory.newInstance();
5. XPath xpath = factory.newXPath();
6. ...
7. XPathExpression expr = xpath.compile("//users/user[login/text()='"+nm+"' and
password/text()='"+pw+"']/home_dir/text()");
8. Object result = expr.evaluate(doc, XPathConstants.NODESET);
9. NodeList nodes = (NodeList) result;
10. for (int i=0; i<nodes.getLength(); i++) {
11.   String value = nodes.item(i).getNodeValue();
12.   if (value.indexOf(">") < 0) {
13.     ...
14.   }
15. }
16.
17. public static void main(String[] args) throws Exception {
18.   ...
19.   String name = args[0];
20.   DocumentBuilder docBuilder = DocumentBuilderFactory.newInstance().
newDocumentBuilder();
21.   Document doc = docBuilder.parse("http://www.w3schools.com/xml/simple.
xml");
22.   XPath xpath = XPathFactory.newInstance().newXPath();
23.   NodeList nodes = (NodeList) xpath.evaluate("//food[name='" + name + "']
/price", doc, XPathConstants.NODESET);
24.   for (int i = 0; i < nodes.getLength(); i++) {
25.     System.out.println(nodes.item(i).getTextContent());
26.   }
27. }
```

라인 7: name과 password에 대한 입력값 검증을 수행하지 않고 XPath 쿼리를 생성합니다. 인자로 넘겨 받은 외부 입력값을 XPath 구문 생성 및 실행에 사용하는 경우 공격자는 XPath 구문을 조작할 수 있습니다.

안전한 예시

```
1. declare variable $loginID as xs:string external;
2. declare variable $password as xs:string external;
3. //users/user[@loginID=$loginID and @password=$password]
4. String nm = props.getProperty("name");
5. String pw = props.getProperty("password");
6. Document doc = new Builder().build("users.xml");
7. XQuery xquery = new XQueryFactory().createXQuery(new File("login.xq"));
8. Map vars = new HashMap();
9. vars.put("loginID", nm);
10. vars.put("password", pw);
11. Nodes results = xquery.execute(doc, null, vars).toNodes();
12. for (int i=0; i<results.size(); i++) {
13.   System.out.println(results.get(i).toXML());
14. }
15.
16. public static void main(String[] args) throws Exception {
17.   ...
18.   String name = args[0];
19.   if (name != null) {
20.     name = name.replaceAll("[()₩₩-'₩₩[₩₩]:,*/]", "");
21.   }
22.   DocumentBuilder docBuilder = DocumentBuilderFactory.newInstance().
newDocumentBuilder();
23.   Document doc = docBuilder.parse("http://www.w3schools.com/xml/simple.
xml");
24.   XPath xpath = XPathFactory.newInstance().newXPath();
25.   NodeList nodes = (NodeList) xpath.evaluate("//food[name='" + name + "']
/price", doc, XPathConstants.NODESET);
26.   for (int i = 0; i < nodes.getLength(); i++) {
27.     System.out.println(nodes.item(i).getTextContent());
28.   }
29. }
```

라인 7: XQuery를 사용하여 미리 쿼리문 골격을 생성함으로써 외부 입력으로 인해 쿼리 구조가 변경되는 것을 막을 수 있습니다. XPath 구문을 조작할 수 있는 문자열을 제거하도록 합니다.

이슈 ID	274701
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00207.java
줄 번호	75

소스코드

```
70.             org.w3c.dom.Document xmlDocument = builder.parse(file);
71.             javax.xml.xpath.XPathFactory xpf = javax.xml.xpath.XPathFactory.
newInstance();
72.             javax.xml.xpath.XPath xp = xpf.newXPath();
73.
74.             String expression = "/Employees/Employee[@emplid='" + bar + "']";
75.             String result = xp.evaluate(expression, xmlDocument);
76.
77.             response.getWriter().println("Your query results are: " + result + "<br/>");
78.
79.         } catch (javax.xml.xpath.XPathExpressionException
80.                 | javax.xml.parsers.ParserConfigurationException
```

이슈 ID	274703
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00442.java
줄 번호	68

소스코드

```
63.             org.w3c.dom.Document xmlDocument = builder.parse(file);
64.             javax.xml.xpath.XPathFactory xpf = javax.xml.xpath.XPathFactory.
newInstance();
65.             javax.xml.xpath.XPath xp = xpf.newXPath();
66.
67.             String expression = "/Employees/Employee[@emplid='" + bar + "']";
68.             String result = xp.evaluate(expression, xmlDocument);
69.
```

```
70.             response.getWriter().println("Your query results are: " + result + "<br/>");
71.
72.         } catch (javax.xml.xpath.XPathExpressionException
73.                 | javax.xml.parsers.ParserConfigurationException
```

● [규칙 이름] 크로스 사이트 스크립팅 (보통, Java)

크로스 사이트 스크립팅 체커는 검증되지 않은 외부 입력값을 HTML에 포함시키는 코드를 검출합니다.

외부 입력을 검증하지 않은 채로 페이지 생성에 사용하는 경우 공격자가 해당 페이지에 악성 스크립트를 삽입할 가능성이 있습니다. 공격자는 해당 취약점을 이용하여 사용자의 쿠키나 세션 등 정보를 탈취하거나 비정상적인 기능을 수행하도록 할 수 있습니다.

외부 입력값에 스크립트가 삽입되지 않도록 문자 변환 함수 또는 메소드를 사용하여 <, >, &, " 등을 <, >, &, " 로 대체합니다. HTML 태그를 사용하도록 허용하는 게시판에서는 허용되는 HTML 태그를 화이트 리스트로 작성하여 리스트에 포함된 태그만 지원하도록 합니다.

- OWASP 2021
  - A03 Injection

- 소프트웨어 보안약점 진단가이드 2021
  - 크로스사이트 스크립트

위험한 예시

```
1. <%@page contentType="text/html" pageEncoding="UTF-8"%>
2. <html>
3.   <head>
4.     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5.   </head>
6.   <body>
7.     <h1>XSS Sample</h1>
8.     <%
9.     <!- Receive name from external source -->
10.       String name = request.getParameter("name");
11.     %>
```

```
12.    <!?Print name received from outer source -->
13.    <p>NAME:<%=name%></p>
14.  </body>
15. </html>
16.
17. <% String customerID = request.getParameter("id"); %>
18.
19.
```

라인 13: 외부 입력값이 들어있는 name을 특별한 처리 과정 없이 결과 페이지 생성에 사용하고 있습니다. 만약 악의적인 공격자가 name 값에 다음 아래의 스크립트를 넣으면, 희생자의 권한으로 attack.jsp 코드가 수행하며 희생자의 쿠키 정보 유출 등의 피해를 주게 됩니다.
(예 : <script>URL = "http://devil.com/attack.jsp";</script>)
라인 17: 매개 변수 id에 쿠키 정보를 출력하는 스크립트 코드가 입력되고 그대로 사용하는 경우, 공격자는 공격 코드를 사용하여 피해자의 쿠키 정보를 빼돌릴 수 있습니다.

안전한 예시

```
1. <%@page contentType="text/html" pageEncoding="UTF-8"%>
2. <html>
3.  <head>
4.    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5.  </head>
6.  <body>
7.    <h1>XSS Sample</h1>
8.    <%
9.    <!- Receive name from outer source -->
10.    String name = request.getParameter("name");
11.    <!? Print name received from outer source -->
12.    if ( name != null ) {
13.      name = name.replaceAll("<","&lt;");
14.      name = name.replaceAll(">","&gt;");
15.    } else {
16.      return;
17.    }
18.    %>
19.    <!-- Remove dangerous character from name received from
20.          outer source, then print it -->
21.    <p>NAME:<%=name%></p>
22.  </body>
23. </html>
```

```
24.
25. <textarea name="content">${ fn:escapeXml(model.content) }</textarea>
26. ...
27. <textarea name="content"><c:out value="${model.content}"/></textarea>
28. ...
29. XssFilter filter = XssFilter.getInstance("lucy-xss-superset.xml");
30. out.append(filter.doFilter(data));
31.
```

라인 13: replaceAll() 메소드를 사용하여 외부 입력 문자열에서 "<"와 ">"같은 HTML 스크립트 생성에 사용되는 모든 문자열을 "<"와 ">"로 변경함으로써 악의적인 스크립트 수행의 위험성을 줄일 수 있습니다. 그러나 이러한 방법이 위험성을 완전히 제거했음을 의미하지는 않습니다. 그 외에도 여러 가지 방법으로 이 공격을 방지할 수 있습니다.
라인 25: JSP에서 출력값에 JSTL HTML 인코딩을 합니다.
라인 27: JSP에서 출력값에 JSTL Core 출력 포맷을 사용하여 텍스트로 처리합니다.
라인 30: 잘 만들어진 외부 XSSFilter 라이브러리를 활용하여 출력값에 필터링 합니다.

이슈 ID	274576
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00003.java
줄 번호	101

소스코드

```
96.                 "hash_value="
97.                     + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
98.                     + "₩n");
99.         fw.close();
100.          response.getWriter()
101.                .println(
102.                    "Sensitive value '"
103.                        + org.owasp
104.                            .esapi
105.                            .ESAPI
106.                            .encoder()
```

이슈 ID	274579
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00005.java
줄 번호	100

소스코드

```
95.             "secret_value="
96.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
97.                 + "₩n");
98.     fw.close();
99.     response.getWriter()
100.            .println(
101.                "Sensitive value: '"
102.                    + org.owasp
103.                        .esapi
104.                        .ESAPI
105.                        .encoder()
```

이슈 ID	274585
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00009.java
줄 번호	101

소스코드

```
96.             "hash_value="
97.                 + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
98.                 + "₩n");
99.     fw.close();
100.      response.getWriter()
101.            .println(
102.                "Sensitive value '"
103.                    + org.owasp
104.                        .esapi
105.                        .ESAPI
106.                        .encoder()
```

이슈 ID	274590
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00012.java
줄 번호	78

소스코드

```
73.
74.             javax.naming.directory.Attribute attr = attrs.get("uid");
75.             javax.naming.directory.Attribute attr2 = attrs.get("street");
76.             if (attr != null) {
77.                 response.getWriter()
78.                         .println(
79.                                 "LDAP query results:<br>"
80.                                         + "Record found with name "
81.                                         + attr.get()
82.                                         + "<br>"
83.                                         + "Address: "
```

이슈 ID	274595
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	95

소스코드

```
90.                 "secret_value="
91.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
92.                         + "₩n");
93.         fw.close();
94.         response.getWriter()
95.                 .println(
96.                         "Sensitive value: '"
97.                                 + org.owasp
```

```
98.                         .esapi
99.                         .ESAPI
100.                         .encoder()
```

이슈 ID	274600
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00021.java
줄 번호	69

소스코드

```
64.
65.             javax.naming.directory.Attribute attr = attrs.get("uid");
66.             javax.naming.directory.Attribute attr2 = attrs.get("street");
67.             if (attr != null) {
68.                 response.getWriter()
69.                         .println(
70.                                 "LDAP query results:<br>"
71.                                         + "Record found with name "
72.                                         + attr.get()
73.                                         + "<br>"
74.                                         + "Address: "
```

이슈 ID	274602
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	73

소스코드

```
68.             }
69.         }
70.     }
71.
72.     if (foundUser) {
```

```
73.             response.getWriter().println("Welcome back: " + user + "<br/>");
74.         } else {
75.             javax.servlet.http.Cookie rememberMe =
76.                     new javax.servlet.http.Cookie(cookieName, rememberMeKey);
77.             rememberMe.setSecure(true);
78.             rememberMe.setHttpOnly(true);
```

이슈 ID	274601
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	85

소스코드

```
80.             rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
81.             // e.g., /benchmark/sql-01/BenchmarkTest01001
82.             request.getSession().setAttribute(cookieName, rememberMeKey);
83.             response.addCookie(rememberMe);
84.             response.getWriter()
85.                     .println(
86.                             user
87.                                     + " has been remembered with cookie: "
88.                                     + rememberMe.getName()
89.                                     + " whose value is: "
90.                                     + rememberMe.getValue()
```

이슈 ID	274618
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	77

소스코드

```
72.                 }
73.             }
74.         }
75.
76.         if (foundUser) {
77.             response.getWriter().println("Welcome back: " + user + "<br/>");
78.         } else {
79.             javax.servlet.http.Cookie rememberMe =
80.                     new javax.servlet.http.Cookie(cookieName, rememberMeKey);
81.             rememberMe.setSecure(true);
82.             rememberMe.setHttpOnly(true);
```

이슈 ID	274619
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	88

소스코드

```
83.             rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
84.             // e.g., /benchmark/sql-01/BenchmarkTest01001
85.             request.getSession().setAttribute(cookieName, rememberMeKey);
86.             response.addCookie(rememberMe);
87.             response.getWriter()
88.                     .println(
89.                             user
90.                                     + " has been remembered with cookie: "
91.                                     + rememberMe.getName()
92.                                     + " whose value is: "
93.                                     + rememberMe.getValue()
```

이슈 ID	274621
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00044.java

줄 번호 70

소스코드

```
65.
66.             javax.naming.directory.Attribute attr = attrs.get("uid");
67.             javax.naming.directory.Attribute attr2 = attrs.get("street");
68.             if (attr != null) {
69.                 response.getWriter()
70.                         .println(
71.                                 "LDAP query results:<br>"
72.                                         + "Record found with name "
73.                                         + attr.get()
74.                                         + "<br>"
75.                                         + "Address: "
```

이슈 ID 274651

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java

줄 번호 114

소스코드

```
109.             }
110.         }
111.     }
112.
113.     if (foundUser) {
114.         response.getWriter().println("Welcome back: " + user + "<br/>");
115.
116.     } else {
117.         javax.servlet.http.Cookie rememberMe =
118.                 new javax.servlet.http.Cookie(cookieName, rememberMeKey);
119.         rememberMe.setSecure(true);
```

이슈 ID 274654

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	127

소스코드

```
122.        rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
123.        // e.g., /benchmark/sql-01/BenchmarkTest01001
124.        request.getSession().setAttribute(cookieName, rememberMeKey);
125.        response.addCookie(rememberMe);
126.        response.getWriter()
127.                .println(
128.                        user
129.                                + " has been remembered with cookie: "
130.                                + rememberMe.getName()
131.                                + " whose value is: "
132.                                + rememberMe.getValue()
```

이슈 ID	274661
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	103

소스코드

```
98.                 "hash_value="
99.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
100.                        + "₩n");
101.        fw.close();
102.        response.getWriter()
103.                .println(
104.                        "Sensitive value '"
105.                                + org.owasp
106.                                        .esapi
107.                                        .ESAPI
108.                                        .encoder()
```

이슈 ID	274663
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00076.java
줄 번호	121

소스코드

```
116.                 "hash_value="
117.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result,
true)
118.                         + "₩n");
119.         fw.close();
120.         response.getWriter()
121.                 .println(
122.                         "Sensitive value '"
123.                                 + org.owasp
124.                                         .esapi
125.                                         .ESAPI
126.                                         .encoder()
```

이슈 ID	274679
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	95

소스코드

```
90.             }
91.         }
92.     }
93.
94.     if (foundUser) {
95.         response.getWriter().println("Welcome back: " + user + "<br/>");
96.     } else {
97.         javax.servlet.http.Cookie rememberMe =
```

```
98.                 new javax.servlet.http.Cookie(cookieName, rememberMeKey);
99.           rememberMe.setSecure(true);
100.           rememberMe.setHttpOnly(true);
```

이슈 ID	274673
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	106

소스코드

```
101.           rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
102.           // e.g., /benchmark/sql-01/BenchmarkTest01001
103.           request.getSession().setAttribute(cookieName, rememberMeKey);
104.           response.addCookie(rememberMe);
105.           response.getWriter()
106.                   .println(
107.                           user
108.                                   + " has been remembered with cookie: "
109.                                   + rememberMe.getName()
110.                                   + " whose value is: "
111.                                   + rememberMe.getValue()
```

이슈 ID	274681
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	98

소스코드

```
93.             }
94.         }
95.     }
96.
```

```
97.      if (foundUser) {
98.          response.getWriter().println("Welcome back: " + user + "<br/>");
99.
100.     } else {
101.         javax.servlet.http.Cookie rememberMe =
102.                 new javax.servlet.http.Cookie(cookieName, rememberMeKey);
103.         rememberMe.setSecure(true);
```

이슈 ID	274685
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	110

소스코드

```
105.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
106.         // e.g., /benchmark/sql-01/BenchmarkTest01001
107.         request.getSession().setAttribute(cookieName, rememberMeKey);
108.         response.addCookie(rememberMe);
109.         response.getWriter()
110.                 .println(
111.                         user
112.                                 + " has been remembered with cookie: "
113.                                 + rememberMe.getName()
114.                                 + " whose value is: "
115.                                 + rememberMe.getValue()
```

이슈 ID	274699
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00207.java
줄 번호	77

소스코드

```
72.         javax.xml.xpath.XPath xp = xpf.newXPath();
73.
74.         String expression = "/Employees/Employee[@emplid='" + bar + "']";
75.         String result = xp.evaluate(expression, xmlDocument);
76.
77.         response.getWriter().println("Your query results are: " + result + "<br/>");
78.
79.      } catch (javax.xml.xpath.XPathExpressionException
80.             | javax.xml.parsers.ParserConfigurationException
81.             | org.xml.sax.SAXException e) {
82.         response.getWriter()
```

이슈 ID	274702
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00207.java
줄 번호	83

소스코드

```
78.
79.      } catch (javax.xml.xpath.XPathExpressionException
80.             | javax.xml.parsers.ParserConfigurationException
81.             | org.xml.sax.SAXException e) {
82.         response.getWriter()
83.                 .println(
84.                         "Error parsing XPath input: '"
85.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML(bar)
86.                                 + "'");
87.         throw new ServletException(e);
88.      }
```

이슈 ID	274704
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00442.java
줄 번호	70

소스코드

```
65.         javax.xml.xpath.XPath xp = xpf.newXPath();
66.
67.         String expression = "/Employees/Employee[@emplid='" + bar + "']";
68.         String result = xp.evaluate(expression, xmlDocument);
69.
70.         response.getWriter().println("Your query results are: " + result + "<br/>");
71.
72.       } catch (javax.xml.xpath.XPathExpressionException
73.             | javax.xml.parsers.ParserConfigurationException
74.             | org.xml.sax.SAXException e) {
75.         response.getWriter()
```

● [규칙 이름] 부적절한 난수 생성 (높음, Java)

부적절한 난수 생성 체커는 예측 가능한 난수를 사용하는 코드를 검출합니다.

사용을 금지할 난수 생성 메소드를 옵션으로 지정할 수 있습니다.

예측 불가능한 숫자가 필요한 상황에서 예측 가능한 난수를 사용한다면, 공격자는 SW에서 생성되는 다음 숫자를 예상하여 시스템을 공격하는 것이 가능하게 됩니다.

옵션으로 지정된 금지 메소드보다 안전한 방식을 사용하여 난수를 생성합니다.

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-330

- 소프트웨어 보안약점 진단가이드 2021
  - 적절하지 않은 난수 값 사용

위험한 예시

```
1. import java.Math;
2. ...
3. public static int[] insertRandom(int[] Cnt, inti, int scope) {
4.   int ran = (int) (Math.random() * scope) - 1;
```

```
5.  if (checkDigit(ran, Cnt)) {
6.    Cnt[i] = ran;
7.  } else {
8.    insertRandom(Cnt, i, scope);
9.  }
10.  return Cnt;
11. }
```

라인 4: java.lang.Math 클래스의 random() 메소드는 시드를 재설정할 수 없기 때문에 위험합니다.

안전한 예시

```
1. import java.util.Random;
2. ...
3. public static int[] insertRandom(int[] Cnt, inti, int scope) {
4.  Random jur = new Random();
5.  jur.setSeed(new Date().getTime());
6.  int ran = (int) (jur.nextInt() * scope) - 1;
7.  if (checkDigit(ran, Cnt)) {
8.    Cnt[i] = ran;
9.  } else {
10.    insertRandom(Cnt, i, scope);
11.  }
12.  return Cnt;
13. }
```

라인 5: java.util.Random 클래스를 사용하면 시드를 재설정할 수 있으나, 여전히 위험합니다. 따라서 Random 클래스를 사용하는 것이 더 안전합니다.

이슈 ID	274533
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	88

소스코드

```
83.      org.owasp.benchmark.helpers.ThingInterface thing =
84.          org.owasp.benchmark.helpers.ThingFactory.createThing();
```

```
85.        String g71153 = "barbarians_at_the_gate"; // This is static so this whole flow
is 'safe'
86.        String bar = thing.doSomething(g71153); // reflection
87.
88.        double value = java.lang.Math.random();
89.        String rememberMeKey = Double.toString(value).substring(2); // Trim off the
0. at the front.
90.
91.        String user = "Doug";
92.        String fullClassName = this.getClass().getName();
93.        String testCaseNumber =
```

● [규칙 이름] 안전하지 않은 쿠키 (높음, Java)

안전하지 않은 쿠키 체커는 쿠키를 암호화하지 않은 채로 전송하는 경우를 검출합니다.

브라우저 쿠키에 데이터를 저장할 때 Cookie 객체의 setSecure() 메소드를 호출하여 해당 속성을 true로 설정해야 합니다.

일반적으로 HTTPS로만 서비스를 운영하면 모든 정보가 암호화 되어 안전하게 전송될 것으로 생각하기 쉽습니다 하지만 보안에 민감한 데이터를 브라우저 쿠키에 저장할 때 보안 속성을 세팅하지 않으면 SSL 등의 보안 프로토콜을 통해 전송이 되지 않습니다. 그로 인해 중요 정보가 공격자에게 단순한 텍스트의 형태로 노출될 수 있습니다. 이것은 쿠키가 개인 정보나 세션 ID를 포함하는 경우에 특히 중요합니다. 단, 한 사이트(도메인)에서 HTTP나 HTTP와 HTTPS를 함께 사용하는 경우 setSecure 메소드를 호출하면 브라우저 쿠키의 데이터가 서버에 전송되지 않아 장애가 발생할 수 있다는 점을 주의해야 합니다.

모든 Cookie 객체에 대해 항상 setSecure() 메소드에 true를 전달하며 호출합니다.

- 소프트웨어 보안약점 진단가이드 2021
  - 암호화되지 않은 중요정보

위험한 예시

```
1. private final String ACCOUNT_ID = "account";
2. public void setupCookies(ServletRequest r, HttpServletResponse response) {
3.   String acctID = r.getParameter("accountID");
4.   // Cookie without security attributes
5.   Cookie c = new Cookie(ACCOUNT_ID, acctID);
```

```
6.  response.addCookie(c);
7. }
```

라인 6: HTTPS로만 서비스하는 경우 민감한 정보를 가진 쿠키를 전송하는 과정에서, 보안속성을 설정하지 않으면 공격자에게 정보가 노출될 수 있습니다.

안전한 예시

```
1. private final String ACCOUNT_ID = "account";
2. public void setupCookies(ServletRequest r, HttpServletResponse response) {
3.  String acctID = r.getParameter("accountID");
4.  // Check validity of account
5.  if (acctID == null || "".equals(acctID)) return;
6.  String filtered_ID = acctID.replaceAll("₩r", "");
7.  Cookie c = new Cookie(ACCOUNT_ID, filtered_ID);
8.  // Cookie with sensitive information need to have secure attributes.
9.  c.setSecure(true);
10.  response.addCookie(c);
11. }
```

라인 9: HTTPS로만 서비스하는 환경에서 민감한 정보를 가진 쿠키를 사용할 경우에는 반드시 Cookie 객체의 setSecure(true) 메소드를 호출해야 합니다.

이슈 ID	274691
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00087.java
줄 번호	95

소스코드

```
90.
91.      cookie.setSecure(false);
92.      cookie.setHttpOnly(true);
93.      cookie.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
94.      // e.g., /benchmark/sql-01/BenchmarkTest01001
95.      response.addCookie(cookie);
96.
97.      response.getWriter()
```

```
98.             .println(
99.                     "Created cookie: 'SomeCookie': with value: '"
100.                            + org.owasp.esapi.ESAPI.encoder().encodeForHTML(str)
```

이슈 ID	274535
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00087.java
줄 번호	95

소스코드

```
90.
91.     cookie.setSecure(false);
92.     cookie.setHttpOnly(true);
93.     cookie.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
94.     // e.g., /benchmark/sql-01/BenchmarkTest01001
95.     response.addCookie(cookie);
96.
97.     response.getWriter()
98.             .println(
99.                     "Created cookie: 'SomeCookie': with value: '"
100.                            + org.owasp.esapi.ESAPI.encoder().encodeForHTML(str)
```

● [규칙 이름] 정수 오버플로우 (높음, Java)

정수 오버플로우 체커는 정수 연산의 결과가 해당 타입의 정수에 허용되는 범위를 넘어서는 코드를 검출합니다.

정수형 변수가 표현할 수 있는 정수의 범위를 초과하게 될 경우 오버플로우가 발생하여 음수 또는 예상하지 못한 결과값을 얻을 수 있습니다. 예상하지 못한 결과값을 사용하여 메모리를 할당하거나 반복문의 조건을 작성하는 경우 보안 취약점이 발생합니다.

언어 및 플랫폼에 따라 정수형의 범위를 확인하고 사용하도록 합니다. 정수형 변수를 연산에 사용하는 경우 먼저 결과값의 범위를 확인하도록 합니다. 외부 입력값을 동적 메모리 할당에 사용하는 경우 변수가 적절한 범위 내에 존재하는지 확인해야 합니다.

- CWE 660 4.14

- 191 - Integer Underflow (Wrap or Wraparound)

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-190

- 소프트웨어 보안약점 진단가이드 2021
  - 정수형 오버플로우

위험한 예시

```
1. public static void main(String[] args) {
2.   int size = new Integer(args[0]).intValue();
3.   size += new Integer(args[1]).intValue();
4.   MyClass[] data = new MyClass[size];
5. }
```

라인 2: 외부 입력(args[0], args[1])을 사용하여 동적으로 계산한 값을 배열의 크기(size)를 결정하는데 사용하고 있습니다.
라인 4: 만일 외부 입력으로부터 계산된 값(size)이 오버플로우에 의해 음수값이 되면 배열의 크기가 음수가 되어 시스템에 문제가 발생할 수 있습니다.

안전한 예시

```
1. public static void main(String[] args) {
2.   int size = new Integer(args[0]).intValue();
3.   size += new Integer(args[1]).intValue();
4.   // Check if size of the array is negative.
5.   if (size < 0) return ;
6.   MyClass[ ] data = new MyClass[size];
7. }
```

라인 5: 동적 메모리 할당을 위해 크기를 사용하는 경우 그 값이 음수가 아닌지 검사하는 문장이 필요합니다.

이슈 ID 274653

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	69

소스코드

```
64.
65.     // Chain a bunch of propagators in sequence
66.     String a71153 = param; // assign
67.     StringBuilder b71153 = new StringBuilder(a71153); // stick in stringbuilder
68.     b71153.append(" SafeStuff"); // append some safe content
69.     b71153.replace(
70.         b71153.length() - "Chars".length(),
71.         b71153.length(),
72.         "Chars"); // replace some of the end content
73.     java.util.HashMap<String, Object> map71153 = new java.util.
HashMap<String, Object>();
74.     map71153.put("key71153", b71153.toString()); // put in a collection
```

● [규칙 이름] TOCTOU 경쟁 조건 (보통, Java)

TOCTOU 경쟁 조건 체커는 리소스의 상태를 확인하고 실제 사용하는 시간 간의 차이로 인해 발생하는 경쟁 조건을 검출합니다.

리소스의 상태를 확인하고 그 결과에 따라 리소스에 접근하는 코드를 작성할 때는 보통 리소스에 접근하는 순간에도 조금 전 확인한 그 상태가 유지되고 있을 것을 가정하게 됩니다. 하지만 그 사이에 리소스의 상태는 변할 수 있으며, 특히 파일과 같은 리소스는 프로그램이 다중 스레드로 작성되지 않더라도 다른 프로그램이나 운영 체제에 의해 상태가 변할 수 있습니다. 이러한 경쟁 조건으로 인해 프로그램이 의도하지 않게 동작하거나 예외가 발생하거나 공격의 진입로로 활용될 수 있습니다.

리소스의 상태에 의존하여 리소스 접근 여부를 결정해야 한다면 동기화 등의 방법을 활용하여 리소스에 독점적으로 접근할 수 있도록 합니다. 혹은 몇몇 파일 관련 동작의 경우에 일단 주어진 동작을 시도하고 리소스의 상태에 따라 다른 결과를 반환하는 방식으로 작동하는 API가 존재하므로 이를 적극 활용하도록 합니다.

- CWE 660 4.14
  - 1341 - Multiple Releases of Same Resource or Handle
  - 366 - Race Condition within a Thread

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-367

- 소프트웨어 보안약점 진단가이드 2021
  - 경쟁조건: 검사시점과 사용시점(TOCTOU)

위험한 예시

```
1. class FileMgmtThread extends Thread {
2.   private String manageType = "";
3.   public FileMgmtThread (String type) {
4.     manageType = type;
5.   }
6.   public void run() {
7.     try {
8.       if ( manageType.equals("READ") ) {
9.         File f = new File("Test_367.txt");
10.          if (f.exists()) { // Read contents if a file exists
11.            BufferedReader br = new BufferedReader(new FileReader(f));
12.            br.close();
13.          }
14.       } else if ( manageType.equals("DELETE") ) {
15.          File f = new File("Test_367.txt");
16.          if (f.exists()) { // delete a file if it exists
17.            f.delete();
18.          } else {... }
19.       }
20.    } catch (IOException e) {...}
21.  }
22. }
23. public class CWE367 {
24.   public static void main(String[] args) {
25.     // Read and delete a file simultaneously
26.     FileMgmtThread fileAccessThread = new FileMgmtThread("READ");
27.     FileMgmtThread fileDeleteThread = new FileMgmtThread("DELETE");
28.     fileAccessThread.start();
```

```
29.     fileDeleteThread.start();
30.   }
31. }
```

라인 10: 파일의 존재를 확인하고 그 결과에 따라 분기가 일어납니다.
라인 16: 파일 존재를 확인하는 시점과 파일을 사용하는 시점이 다르므로, 그 사이에 파일에 대한 삭제가 발생하면 프로그램이 예상하지 못하는 형태로 수행될 수 있습니다.

안전한 예시

```
1. class FileMgmtThread extends Thread {
2.       private String manageType = "";
3.       public FileMgmtThread (String type) {
4.          manageType = type;
5.       }
6.
7.       // Add synchronized for TOCTOU problem
8.       public synchronized void run() {
9.          try {
10.             if ( manageType.equals("READ") ) {
11.                File f = new File("Test_367.txt");
12.                // Read contents of a file if it exists
13.                if (f.exists()) {
14.                   try {
15.                      BufferedReader br = new BufferedReader(new FileReader(f));
16.                      br.close();
17.                   } catch (FileNotFoundException e) {
18.                      // handle race condition
19.                   }
20.                }
21.             } else if ( manageType.equals("DELETE") ) {
22.                File f = new File("Test_367.txt");
23.                if (f.exists()) { // Delete a file if it exists
24.                   if (f.delete()) {
25.                      // successful
26.                   } else {
27.                      // handle race condition
28.                   }
29.                } else {...}
30.             }
31.          } catch (IOException e) {...}
```

```
32.       }
33. }
34.
35. public class CWE367 {
36.     public static void main(String[] args) {
37.         // Read and delete a file simultaneously
38.         FileMgmtThread fileAccessThread = new FileMgmtThread("READ");
39.         FileMgmtThread fileDeleteThread = new FileMgmtThread("DELETE");
40.         fileAccessThread.start();
41.         fileDeleteThread.start();
42.     }
43. }
```

라인 8: 공유 리소스를 여러 스레드가 접근하여 사용할 경우, 동기화 구문을 사용하여 한 번에 하나의 스레드만 접근 가능하도록 변경합니다.
라인 17: 접근 시점에 파일이 존재하지 않을 경우 FileNotFound 예외가 발생하므로 이 예외를 처리해 줍니다.

이슈 ID	274551
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	323

소스코드

```
318.       List<String> sourceLines = new ArrayList<String>();
319.
320.       try (FileReader fr = new FileReader(file);
321.               BufferedReader br = new BufferedReader(fr); ) {
322.           String line;
323.           while ((line = br.readLine()) != null) {
324.               sourceLines.add(line);
325.           }
326.       } catch (Exception e) {
327.           try {
328.               System.out.println("Problem reading contents of file: " + file.
getCanonicalFile());
```

● [규칙 이름] 신뢰할 수 없는 입력에 의존한 보안 결정 (높음, Java)

신뢰할 수 없는 입력에 의존한 보안 결정 체커는 사용자의 인증 정보를 쿠키에 저장하는 코드를 검출합니다.

개발자들은 흔히 쿠키, 환경 변수 또는 히든 필드와 같은 입력값이 조작될 수 없다고 가정합니다. 하지만 공격자는 다양한 방법을 통해 이러한 입력값들을 변경할 수 있고 조작된 내용은 탐지되지 않을 수 있습니다. 이러한 입력값(쿠키, 환경 변수, 히든 필드 등)에 기반하여 인증이나 인가와 같은 보안 결정을 수행하는 경우 공격자는 해당 값을 조작하여 애플리케이션의 보안을 우회할 수 있으므로 외부 사용자에 의한 입력값을 신뢰해서는 안됩니다.

상태 정보나 민감한 데이터, 사용자 세션 정보와 같은 중요한 정보는 서버에 저장하고 보안 확인 절차도 서버에서 실행합니다. 또한 보안 설계 관점에서 신뢰할 수 없는 입력값이 애플리케이션 내부로 들어올 수 있는 지점을 파악해야 하며, 보안 결정에 사용되는 입력값을 식별하고 제공되는 입력값에 의존할 필요가 없는 구조로 변경할 수 있는지 검토해야 합니다.

- 소프트웨어 보안약점 진단가이드 2021
  - 보안기능 결정에 사용되는 부적절한 입력값

위험한 예시

```
1. <%
2. String username = request.getParameter("username");
3. String password = request.getParameter("password");
4. if (username==nill || password==null || !isAuthenticatedUser(usename,
5. password)) {
6. throw new MyException("Authentification error");
7. }
8. Cookie userCookie = new Cookie("user",username);
9. Cookie authCookie = new Cookie("authenticated","1");
10. response.addCookie(userCookie);
11. response.addCookie(authCookie);
12. %>
```

라인 9: 평문으로 사용자의 인증정보 및 "authenticated"를 쿠키에 저장하고 있습니다. 공격자는 쿠키 정보를 변경 가능하기 때문에 위험합니다.

안전한 예시

```
1. <%
2. String username = request.getParameter("username");
3. String password = request.getParameter("password");
4. if (username==nill || password==null || !isAuthenticatedUser(usename,
5. password)) {
6.   throw new MyException("Authentication Error");
7. }
8. // Save user information in session
9. HttpSession ses = request.getSession();
10. ses.setAttribute("user",username);
11. ses.setAttribute("authenticated","1");
12. %>
```

라인 11: 사용자의 인증 정보를 세션에 저장하면 인증 정보가 외부에 노출될 위험을 제거할 수 있습니다.

이슈 ID	274548
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	167

소스코드

```
162.     Cookie[] values = request.getCookies();
163.     String param = "none";
164.     if (paramName != null) {
165.         for (int i = 0; i < values.length; i++) {
166.             if (values[i].getName().equals(paramName)) {
167.                 param = values[i].getValue();
168.                 break; // break out of for loop when param found
169.             }
170.         }
171.     }
172.     return param;
```

이슈 ID	274564

파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	60

소스코드

```
55.
56.     String param = "noCookieValueSupplied";
57.     if (theCookies != null) {
58.         for (javax.servlet.http.Cookie theCookie : theCookies) {
59.             if (theCookie.getName().equals("BenchmarkTest00001")) {
60.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
61.                 break;
62.             }
63.         }
64.     }
65.
```

이슈 ID	274569
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	60

소스코드

```
55.
56.     String param = "noCookieValueSupplied";
57.     if (theCookies != null) {
58.         for (javax.servlet.http.Cookie theCookie : theCookies) {
59.             if (theCookie.getName().equals("BenchmarkTest00002")) {
60.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
61.                 break;
62.             }
63.         }
64.     }
65.
```

이슈 ID	274575
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00003.java
줄 번호	60

소스코드

```
55.
56.     String param = "noCookieValueSupplied";
57.     if (theCookies != null) {
58.         for (javax.servlet.http.Cookie theCookie : theCookies) {
59.             if (theCookie.getName().equals("BenchmarkTest00003")) {
60.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
61.                 break;
62.             }
63.         }
64.     }
65.
```

이슈 ID	274608
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	65

소스코드

```
60.     javax.servlet.http.Cookie[] cookies = request.getCookies();
61.     if (cookies != null) {
62.         for (int i = 0; !foundUser && i < cookies.length; i++) {
63.             javax.servlet.http.Cookie cookie = cookies[i];
64.             if (cookieName.equals(cookie.getName())) {
65.                 if (cookie.getValue().equals(request.getSession().getAttribute
(cookieName))) {
66.                     foundUser = true;
67.                 }
68.             }
```

```
69.         }
70.     }
```

이슈 ID	274604
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	90

소스코드

```
85.                 .println(
86.                         user
87.                                 + " has been remembered with cookie: "
88.                                 + rememberMe.getName()
89.                                 + " whose value is: "
90.                                 + rememberMe.getValue()
91.                                 + "<br/>");
92.     }
93.
94.     response.getWriter().println("Weak Randomness Test java.util.Random.nextFloat() executed");
95.   }
```

이슈 ID	274613
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	68

소스코드

```
63.         javax.servlet.http.Cookie[] cookies = request.getCookies();
64.         if (cookies != null) {
65.             for (int i = 0; !foundUser && i < cookies.length; i++) {
66.                 javax.servlet.http.Cookie cookie = cookies[i];
67.                 if (cookieName.equals(cookie.getName())) {
```

```
68.                     if (cookie.getValue()
69.                             .equals(request.getSession().getAttribute(cookieName))) {
70.                         foundUser = true;
71.                     }
72.                 }
73.             }
```

이슈 ID	274616
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	93

소스코드

```
88.                     .println(
89.                             user
90.                                     + " has been remembered with cookie: "
91.                                     + rememberMe.getName()
92.                                     + " whose value is: "
93.                                     + rememberMe.getValue()
94.                                     + "<br/>");
95.             }
96.         } catch (java.security.NoSuchAlgorithmException e) {
97.             System.out.println("Problem executing SecureRandom.nextInt() - TestCase");
98.             throw new ServletException(e);
```

이슈 ID	274626
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00061.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00061")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274631
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00062")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274639
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00064.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00064")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274641
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00065")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274655
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode

파일	/BenchmarkTest00067.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00067")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274652
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00067.java
줄 번호	106

소스코드

```
101.        javax.servlet.http.Cookie[] cookies = request.getCookies();
102.        if (cookies != null) {
103.            for (int i = 0; !foundUser && i < cookies.length; i++) {
104.                javax.servlet.http.Cookie cookie = cookies[i];
105.                if (cookieName.equals(cookie.getName())) {
106.                    if (cookie.getValue().equals(request.getSession().getAttribute
(cookieName))) {
107.                        foundUser = true;
108.                    }
109.                }
110.            }
111.        }
```

이슈 ID	274656
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	132

소스코드

```
127.                 .println(
128.                         user
129.                                 + " has been remembered with cookie: "
130.                                 + rememberMe.getName()
131.                                 + " whose value is: "
132.                                 + rememberMe.getValue()
133.                                 + "<br/>");
134.         }
135.         response.getWriter().println("Weak Randomness Test java.lang.Math.random() executed");
136.     }
137. }
```

이슈 ID	274658
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	59

소스코드

```
54.
55.      String param = "noCookieValueSupplied";
56.      if (theCookies != null) {
57.          for (javax.servlet.http.Cookie theCookie : theCookies) {
58.              if (theCookie.getName().equals("BenchmarkTest00070")) {
59.                  param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                  break;
61.              }
62.          }
```

```
63.     }
64.
```

이슈 ID	274664
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00076.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00076")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274668
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00077.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00077")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
```

```
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274678
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00084")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274675
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	87

소스코드

```
82.     javax.servlet.http.Cookie[] cookies = request.getCookies();
83.     if (cookies != null) {
84.         for (int i = 0; !foundUser && i < cookies.length; i++) {
```

```
85.             javax.servlet.http.Cookie cookie = cookies[i];
86.             if (cookieName.equals(cookie.getName())) {
87.                 if (cookie.getValue().equals(request.getSession().getAttribute
(cookieName))) {
88.                     foundUser = true;
89.                 }
90.             }
91.         }
92.     }
```

이슈 ID	274676
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java
줄 번호	111

소스코드

```
106.                 .println(
107.                         user
108.                                 + " has been remembered with cookie: "
109.                                 + rememberMe.getName()
110.                                 + " whose value is: "
111.                                 + rememberMe.getValue()
112.                                 + "<br/>");
113.     }
114.
115.     response.getWriter().println("Weak Randomness Test java.util.Random.
nextInt() executed");
116. }
```

이슈 ID	274684
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	59

소스코드

```
54.
55.     String param = "noCookieValueSupplied";
56.     if (theCookies != null) {
57.         for (javax.servlet.http.Cookie theCookie : theCookies) {
58.             if (theCookie.getName().equals("BenchmarkTest00086")) {
59.                 param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                 break;
61.             }
62.         }
63.     }
64.
```

이슈 ID	274688
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java
줄 번호	90

소스코드

```
85.     javax.servlet.http.Cookie[] cookies = request.getCookies();
86.     if (cookies != null) {
87.         for (int i = 0; !foundUser && i < cookies.length; i++) {
88.             javax.servlet.http.Cookie cookie = cookies[i];
89.             if (cookieName.equals(cookie.getName())) {
90.                 if (cookie.getValue().equals(request.getSession().getAttribute
(cookieName))) {
91.                     foundUser = true;
92.                 }
93.             }
94.         }
95.     }
```

이슈 ID	274689
	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode

파일	/BenchmarkTest00086.java
줄 번호	115

소스코드

```
110.                 .println(
111.                         user
112.                                 + " has been remembered with cookie: "
113.                                 + rememberMe.getName()
114.                                 + " whose value is: "
115.                                 + rememberMe.getValue()
116.                                 + "<br/>");
117.         }
118.
119.         response.getWriter().println("Weak Randomness Test java.util.Random.
nextLong() executed");
120.     }
```

이슈 ID	274695
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode /BenchmarkTest00087.java
줄 번호	59

소스코드

```
54.
55.      String param = "noCookieValueSupplied";
56.      if (theCookies != null) {
57.          for (javax.servlet.http.Cookie theCookie : theCookies) {
58.              if (theCookie.getName().equals("BenchmarkTest00087")) {
59.                  param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                  break;
61.              }
62.          }
63.      }
64.
```

이슈 ID	274698
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00092.java
줄 번호	59

소스코드

```
54.
55.      String param = "noCookieValueSupplied";
56.      if (theCookies != null) {
57.          for (javax.servlet.http.Cookie theCookie : theCookies) {
58.              if (theCookie.getName().equals("BenchmarkTest00092")) {
59.                  param = java.net.URLDecoder.decode(theCookie.getValue(), "UTF-8");
60.                  break;
61.              }
62.          }
63.      }
64.
```

● [규칙 이름] 시스템 정보 노출 (보통, Java)

시스템 정보 노출 체커는 오류 메시지 등을 통해 시스템 정보를 출력하는 코드를 검출합니다.

애플리케이션이 실행 환경, 사용자 등 관련 데이터에 대한 민감한 정보를 포함하는 오류 메시지를 생성하여 외부에 제공하는 경우, 공격자의 악성 행위를 도울 수 있습니다. 예외 발생 시 예외 이름이나 스택 트레이스를 출력하면, 프로그램 내부 구조를 쉽게 파악할 수 있는 위험이 있습니다.

오류 메시지는 정해진 사용자에게 유용한 최소한의 정보만 포함하도록 합니다. 소스 코드에서 예외 상황은 내부적으로 처리하며, 민감한 정보를 포함하는 오류를 출력하지 않도록 합니다.

- CWE 660 4.14
  - 209 - Generation of Error Message Containing Sensitive Information

- 무기체계 소프트웨어 보안약점 점검 목록
  - CWE-209

- CWE-497

- 소프트웨어 보안약점 진단가이드 2021
  - 오류메시지 정보 노출

위험한 예시

```
1. try {
2.   ...
3. } catch (Exception e) {
4.   e.printStackTrace();
5. }
```

라인 4: 예외 이름이나 스택 트레이스를 출력하면 프로그램 내부 정보가 유출됩니다.

안전한 예시

```
1. try {
2.   ...
3. } catch (Exception e) {
4.   logger.error("Connection Exception occurred");
5. }
```

라인 4: 예외 이름이나 스택 트레이스를 출력하지 않습니다.

이슈 ID	274456
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DataBaseServer.java
줄 번호	70

소스코드

```
65.         java.sql.PreparedStatement statement = connection.prepareStatement(sql);
66.         statement.execute();
67.         org.owasp.benchmark.helpers.DatabaseHelper.printResults(statement, sql,
```

```
resp);
68.         } catch (java.sql.SQLException e) {
69.             if (org.owasp.benchmark.helpers.DatabaseHelper.hideSQLErrors) {
70.                 e.printStackTrace();
71.                 resp.add(new XMLMessage("Error processing request: " + e.
getMessage()));
72.                 return new ResponseEntity<List<XMLMessage>>(resp, HttpStatus.OK);
73.             } else throw new ServletException(e);
74.         }
75.         return new ResponseEntity<List<XMLMessage>>(resp, HttpStatus.OK);
```

이슈 ID	274465
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	113

소스코드

```
108.             conn.commit();
109.             initData();
110.
111.             System.out.println("DataBase tables/procedures created.");
112.         } catch (Exception e1) {
113.             System.out.println(
114.                     "Problem with database table/procedure creations: " + e1.
getMessage());
115.         }
116.     }
117.
118.     public static java.sql.Statement getSqlStatement() {
```

이슈 ID	274463
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/DatabaseHelper.java
줄 번호	152

소스코드

```
147.
148.            executeSQLCommand(
149.                    "INSERT INTO EMPLOYEE (first_name, last_name, salary) VALUES
('foo', 'bar', 34567)");
150.            conn.commit();
151.        } catch (Exception e1) {
152.            System.out.println("Problem with database init/reset: " + e1.
getMessage());
153.        }
154.    }
155.
156.    public static java.sql.Connection getSqlConnection() {
157.        if (conn == null) {
```

이슈 ID	274462
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers /DatabaseHelper.java
줄 번호	165

소스코드

```
160.                DataSource datasource = (DataSource) ctx.lookup("java:comp/env/jdbc
/BenchmarkDB");
161.                conn = datasource.getConnection();
162.                conn.setAutoCommit(false);
163.            } catch (SQLException | NamingException e) {
164.                System.out.println("Problem with getSqlConnection.");
165.                e.printStackTrace();
166.            }
167.        }
168.        return conn;
169.    }
170.
```

이슈 ID	274467
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	52

소스코드

```
47.  public LDAPManager() {
48.      try {
49.          ctx = getDirContext();
50.      } catch (NamingException e) {
51.          // FIXME: Don't eat exceptions!
52.          System.out.println("Failed to get Directory Context: " + e.getMessage());
53.          e.printStackTrace();
54.      }
55.  }
56.
57.  protected Hashtable<Object, Object> createEnv() {
```

이슈 ID	274468
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	53

소스코드

```
48.      try {
49.          ctx = getDirContext();
50.      } catch (NamingException e) {
51.          // FIXME: Don't eat exceptions!
52.          System.out.println("Failed to get Directory Context: " + e.getMessage());
53.          e.printStackTrace();
54.      }
55.  }
56.
57.  protected Hashtable<Object, Object> createEnv() {
58.      Hashtable<Object, Object> env = new Hashtable<Object, Object>();
```

이슈 ID	274469
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	85

소스코드

```
80.
81.         try {
82.             iniDirContext.bind(name, ctx, matchAttrs);
83.         } catch (NamingException e) {
84.             if (!e.getMessage().contains("ENTRY_ALREADY_EXISTS")) {
85.                 System.out.println("Record already exist or an error occurred: " + e.
getMessage());
86.             }
87.         }
88.
89.         return true;
90.     }
```

이슈 ID	274470
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPManager.java
줄 번호	130

소스코드

```
125.             ctx.close();
126.
127.             return true;
128.         } catch (Exception e) {
129.             System.out.println("LDAP error search: ");
130.             e.printStackTrace();
131.             return false;
```

```
132.     }
133.   }
134.
135.   public DirContext getDirContext() throws NamingException {
```

이슈 ID	274473
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPServer.java
줄 번호	103

소스코드

```
98.       workDir.mkdirs();
99.       System.setProperty("workingDiretory", workDir.getPath());
100.
101.        init();
102.      } catch (Exception e) {
103.        System.out.println("Error initializing LDAP Server: " + e.getMessage());
104.        e.printStackTrace();
105.      }
106.
107.      LDAPManager emd = new LDAPManager();
108.      LDAPPerson ldapP = new LDAPPerson();
```

이슈 ID	274474
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/LDAPServer.java
줄 번호	104

소스코드

```
99.       System.setProperty("workingDiretory", workDir.getPath());
100.
101.        init();
102.      } catch (Exception e) {
```

```
103.            System.out.println("Error initializing LDAP Server: " + e.getMessage());
104.            e.printStackTrace();
105.        }
106.
107.        LDAPManager emd = new LDAPManager();
108.        LDAPPerson ldapP = new LDAPPerson();
109.        ldapP.setName("foo");
```

이슈 ID	274476
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	107

소스코드

```
102.            safeDocBuilderFactory.setFeature(
103.                    "http://apache.org/xml/features/disallow-doctype-decl", true);
104.        } catch (ParserConfigurationException e) {
105.            System.out.println(
106.                    "ERROR: couldn't set http://apache.org/xml/features/disallow-
doctype-decl");
107.            e.printStackTrace();
108.        }
109.
110.        File tempDir = new File(TESTFILES_DIR);
111.        if (!tempDir.exists()) {
112.            tempDir.mkdir();
```

이슈 ID	274477
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	119

소스코드

```
114.        try {
115.            PrintWriter out = new PrintWriter(testFile);
116.            out.write("Test is a test file.₩n");
117.            out.close();
118.        } catch (FileNotFoundException e) {
119.            e.printStackTrace();
120.        }
121.        File testFile2 = new File(TESTFILES_DIR + "SafeText");
122.        try {
123.            PrintWriter out = new PrintWriter(testFile2);
124.            out.write("Test is a 'safe' test file.₩n");
```

이슈 ID	274478
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	127

소스코드

```
122.        try {
123.            PrintWriter out = new PrintWriter(testFile2);
124.            out.write("Test is a 'safe' test file.₩n");
125.            out.close();
126.        } catch (FileNotFoundException e) {
127.            e.printStackTrace();
128.        }
129.        File secreTestFile = new File(TESTFILES_DIR + "SecretFile");
130.        try {
131.            PrintWriter out = new PrintWriter(secreTestFile);
132.            out.write("Test is a 'secret' file that no one should find.₩n");
```

이슈 ID	274479
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	135

소스코드

```
130.          try {
131.             PrintWriter out = new PrintWriter(secreTestFile);
132.             out.write("Test is a 'secret' file that no one should find.₩n");
133.             out.close();
134.          } catch (FileNotFoundException e) {
135.             e.printStackTrace();
136.          }
137.       }
138.
139.       // The target script is exploded out of the WAR file. When this occurs, the
file
140.       // loses its execute permissions. So this hack adds the required execute
permissions back.
```

이슈 ID	274486
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	258

소스코드

```
253.             out.write(ESAPI.encoder().encodeForHTML(s));
254.             out.write("<br>");
255.          }
256.       } catch (IOException e) {
257.          System.out.println("An error occurred while reading
OSCommandResults");
258.          e.printStackTrace();
259.       }
260.    }
261.
262.    // A method used by the Benchmark JAVA test cases to format OS Command
Output
263.    // This version is only used by the Web Services test cases.
```

이슈 ID	274487
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	289

소스코드

```
284.            }
285.
286.            resp.add(new XMLMessage(outError.toString()));
287.        } catch (IOException e) {
288.            System.out.println("An error occurred while reading
OSCommandResults");
289.            e.printStackTrace();
290.        }
291.    }
292.
293.    public static File getFileFromClasspath(String fileName, ClassLoader
classLoader) {
294.        URL url = classLoader.getResource(fileName);
```

이슈 ID	274481
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	301

소스코드

```
296.            try {
297.                return new File(url.toURI().getPath());
298.            } catch (URISyntaxException e) {
299.                System.out.println(
300.                        "The file '" + fileName + "' cannot be loaded from the classpath.");
301.                e.printStackTrace();
302.            }
303.        } else System.out.println("The file '" + fileName + "' cannot be found on the
```

```
classpath.");
304.        return null;
305.    }
306.
```

이슈 ID	274482
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	313

소스코드

```
308.        if (!file.exists()) {
309.            try {
310.                System.out.println("Can't find file to get lines from: " + file.getCanonicalFile());
311.            } catch (IOException e) {
312.                System.out.println("Can't find file to get lines from.");
313.                e.printStackTrace();
314.            }
315.            return null;
316.        }
317.
318.        List<String> sourceLines = new ArrayList<String>();
```

이슈 ID	274483
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	331

소스코드

```
326.        } catch (Exception e) {
327.            try {
328.                System.out.println("Problem reading contents of file: " + file.
```

```
getCanonicalFile());
329.            } catch (IOException e2) {
330.                System.out.println("Problem reading file to get lines from.");
331.                e2.printStackTrace();
332.            }
333.            e.printStackTrace();
334.        }
335.
336.        return sourceLines;
```

이슈 ID	274484
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	333

소스코드

```
328.                System.out.println("Problem reading contents of file: " + file.
getCanonicalFile());
329.            } catch (IOException e2) {
330.                System.out.println("Problem reading file to get lines from.");
331.                e2.printStackTrace();
332.            }
333.            e.printStackTrace();
334.        }
335.
336.        return sourceLines;
337.    }
338.
```

이슈 ID	274488
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	388

소스코드

```
383.            FileOutputStream fos = new FileOutputStream(f, true);
384.            os = new PrintStream(fos);
385.            os.println(line);
386.        } catch (IOException e1) {
387.            result = false;
388.            e1.printStackTrace();
389.        } finally {
390.            os.close();
391.        }
392.
393.        return result;
```

이슈 ID	274480
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/helpers/Utils.java
줄 번호	417

소스코드

```
412.                cipher.init(javax.crypto.Cipher.ENCRYPT_MODE, publicKey);
413.            } catch (NoSuchAlgorithmException
414.                    | NoSuchProviderException
415.                    | NoSuchPaddingException
416.                    | InvalidKeyException e) {
417.                e.printStackTrace();
418.            }
419.        }
420.        return cipher;
421.    }
422.
```

이슈 ID	274562
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java

줄 번호	75

[소스코드]

```
70.         fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.         fis = new java.io.FileInputStream(new java.io.File(fileName));
72.         byte[] b = new byte[1000];
73.         int size = fis.read(b);
74.         response.getWriter()
75.                 .println(
76.                         "The beginning of file: '"
77.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName)
78.                                 + "' is:₩n₩n"
79.                                 + org.owasp
80.                                         .esapi
```

이슈 ID	274561
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00001.java
줄 번호	85

[소스코드]

```
80.                                         .esapi
81.                                         .ESAPI
82.                                         .encoder()
83.                                         .encodeForHTML(new String(b, 0, size)));
84.         } catch (Exception e) {
85.             System.out.println("Couldn't open FileInputStream on file: '" + fileName +
"'");
86.             response.getWriter()
87.                     .println(
88.                             "Problem getting FileInputStream: "
89.                                     + org.owasp
90.                                             .esapi
```

이슈 ID	274572
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00002.java
줄 번호	74

소스코드

```
69.      try {
70.          fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + param;
71.
72.          fos = new java.io.FileOutputStream(fileName, false);
73.          response.getWriter()
74.                  .println(
75.                          "Now ready to write to file: "
76.                                  + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName));
77.
78.      } catch (Exception e) {
79.          System.out.println("Couldn't open FileOutputStream on file: '" + fileName
+ "'");
```

이슈 ID	274574
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00003.java
줄 번호	101

소스코드

```
96.                  "hash_value="
97.                          + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
98.                          + "₩n");
99.          fw.close();
100.         response.getWriter()
101.                 .println(
102.                         "Sensitive value '"
103.                                 + org.owasp
```

```
104.                        .esapi
105.                        .ESAPI
106.                        .encoder()
```

이슈 ID	274495
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00005.java
줄 번호	118

소스코드

```
113.              | java.security.InvalidKeyException
114.              | java.security.InvalidAlgorithmParameterException e) {
115.          response.getWriter()
116.                  .println(
117.                          "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
118.          e.printStackTrace(response.getWriter());
119.          throw new ServletException(e);
120.      }
121.  }
122. }
```

이슈 ID	274502
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	108

소스코드

```
103.
104.      } catch (java.security.NoSuchAlgorithmException e) {
105.          response.getWriter()
106.                  .println(
107.                          "Problem executing crypto - javax.crypto.Cipher.getInstance
```

```
(java.lang.String,java.security.Provider) Test Case");
108.            e.printStackTrace(response.getWriter());
109.            throw new ServletException(e);
110.        } catch (java.security.NoSuchProviderException e) {
111.            response.getWriter()
112.                    .println(
113.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274503
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	114

소스코드

```
109.            throw new ServletException(e);
110.        } catch (java.security.NoSuchProviderException e) {
111.            response.getWriter()
112.                    .println(
113.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
114.            e.printStackTrace(response.getWriter());
115.            throw new ServletException(e);
116.        } catch (javax.crypto.NoSuchPaddingException e) {
117.            response.getWriter()
118.                    .println(
119.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274504
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	120

소스코드

```
115.            throw new ServletException(e);
116.        } catch (javax.crypto.NoSuchPaddingException e) {
117.            response.getWriter()
118.                    .println(
119.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
120.            e.printStackTrace(response.getWriter());
121.            throw new ServletException(e);
122.        } catch (javax.crypto.IllegalBlockSizeException e) {
123.            response.getWriter()
124.                    .println(
125.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274505
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	126

소스코드

```
121.            throw new ServletException(e);
122.        } catch (javax.crypto.IllegalBlockSizeException e) {
123.            response.getWriter()
124.                    .println(
125.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
126.            e.printStackTrace(response.getWriter());
127.            throw new ServletException(e);
128.        } catch (javax.crypto.BadPaddingException e) {
129.            response.getWriter()
130.                    .println(
131.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274506
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	132

소스코드

```
127.            throw new ServletException(e);
128.        } catch (javax.crypto.BadPaddingException e) {
129.            response.getWriter()
130.                    .println(
131.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
132.            e.printStackTrace(response.getWriter());
133.            throw new ServletException(e);
134.        } catch (java.security.InvalidKeyException e) {
135.            response.getWriter()
136.                    .println(
137.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274507
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00020.java
줄 번호	144

소스코드

```
139.            throw new ServletException(e);
140.        } catch (java.security.InvalidAlgorithmParameterException e) {
141.            response.getWriter()
142.                    .println(
143.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
144.            e.printStackTrace(response.getWriter());
145.            throw new ServletException(e);
146.        }
```

```
147.        response.getWriter()
148.                .println(
149.                        "Crypto Test javax.crypto.Cipher.getInstance(java.lang.String,java.
lang.String) executed");
```

이슈 ID	274603
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00023.java
줄 번호	85

소스코드

```
80.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
81.         // e.g., /benchmark/sql-01/BenchmarkTest01001
82.         request.getSession().setAttribute(cookieName, rememberMeKey);
83.         response.addCookie(rememberMe);
84.         response.getWriter()
85.                 .println(
86.                         user
87.                                 + " has been remembered with cookie: "
88.                                 + rememberMe.getName()
89.                                 + " whose value is: "
90.                                 + rememberMe.getValue()
```

이슈 ID	274614
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00042.java
줄 번호	88

소스코드

```
83.             rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
84.             // e.g., /benchmark/sql-01/BenchmarkTest01001
```

```
85.             request.getSession().setAttribute(cookieName, rememberMeKey);
86.             response.addCookie(rememberMe);
87.             response.getWriter()
88.                     .println(
89.                             user
90.                                     + " has been remembered with cookie: "
91.                                     + rememberMe.getName()
92.                                     + " whose value is: "
93.                                     + rememberMe.getValue()
```

이슈 ID	274516
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	109

소스코드

```
104.
105.         } catch (java.security.NoSuchAlgorithmException e) {
106.             response.getWriter()
107.                     .println(
108.                             "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
109.             e.printStackTrace(response.getWriter());
110.             throw new ServletException(e);
111.         } catch (java.security.NoSuchProviderException e) {
112.             response.getWriter()
113.                     .println(
114.                             "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274517
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	115

소스코드

```
110.            throw new ServletException(e);
111.        } catch (java.security.NoSuchProviderException e) {
112.            response.getWriter()
113.                    .println(
114.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
115.            e.printStackTrace(response.getWriter());
116.            throw new ServletException(e);
117.        } catch (javax.crypto.NoSuchPaddingException e) {
118.            response.getWriter()
119.                    .println(
120.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274518
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	121

소스코드

```
116.            throw new ServletException(e);
117.        } catch (javax.crypto.NoSuchPaddingException e) {
118.            response.getWriter()
119.                    .println(
120.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
121.            e.printStackTrace(response.getWriter());
122.            throw new ServletException(e);
123.        } catch (javax.crypto.IllegalBlockSizeException e) {
124.            response.getWriter()
125.                    .println(
126.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274519
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	127

소스코드

```
122.            throw new ServletException(e);
123.        } catch (javax.crypto.IllegalBlockSizeException e) {
124.            response.getWriter()
125.                    .println(
126.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
127.            e.printStackTrace(response.getWriter());
128.            throw new ServletException(e);
129.        } catch (javax.crypto.BadPaddingException e) {
130.            response.getWriter()
131.                    .println(
132.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274520
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	133

소스코드

```
128.            throw new ServletException(e);
129.        } catch (javax.crypto.BadPaddingException e) {
130.            response.getWriter()
131.                    .println(
132.                            "Problem executing crypto - javax.crypto.Cipher.getInstance(java.lang.String,java.security.Provider) Test Case");
133.            e.printStackTrace(response.getWriter());
134.            throw new ServletException(e);
```

```
135.        } catch (java.security.InvalidKeyException e) {
136.            response.getWriter()
137.                    .println(
138.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
```

이슈 ID	274521
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java
줄 번호	145

소스코드

```
140.            throw new ServletException(e);
141.        } catch (java.security.InvalidAlgorithmParameterException e) {
142.            response.getWriter()
143.                    .println(
144.                            "Problem executing crypto - javax.crypto.Cipher.getInstance
(java.lang.String,java.security.Provider) Test Case");
145.            e.printStackTrace(response.getWriter());
146.            throw new ServletException(e);
147.        }
148.        response.getWriter()
149.                .println(
150.                        "Crypto Test javax.crypto.Cipher.getInstance(java.lang.String,java.
lang.String) executed");
```

이슈 ID	274629
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00061.java
줄 번호	76

소스코드

```
71.                                 param.getBytes())));
72.         }
73.
74.         java.io.File fileTarget = new java.io.File(bar, "/Test.txt");
75.         response.getWriter()
76.                 .println(
77.                         "Access to file: '"
78.                                 + org.owasp
79.                                         .esapi
80.                                         .ESAPI
81.                                         .encoder()
```

이슈 ID	274632
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java
줄 번호	81

소스코드

```
76.             fileName = org.owasp.benchmark.helpers.Utils.TESTFILES_DIR + bar;
77.             fis = new java.io.FileInputStream(new java.io.File(fileName));
78.             byte[] b = new byte[1000];
79.             int size = fis.read(b);
80.             response.getWriter()
81.                     .println(
82.                             "The beginning of file: '"
83.                                     + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName)
84.                                     + "' is:₩n₩n"
85.                                     + org.owasp
86.                                             .esapi
```

이슈 ID	274637
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00062.java

줄 번호 91

소스코드

```
86.                                 .esapi
87.                                 .ESAPI
88.                                 .encoder()
89.                                 .encodeForHTML(new String(b, 0, size)));
90.       } catch (Exception e) {
91.           System.out.println("Couldn't open FileInputStream on file: '" + fileName +
"'");
92.           response.getWriter()
93.                   .println(
94.                           "Problem getting FileInputStream: "
95.                                   + org.owasp
96.                                           .esapi
```

이슈 ID 274646

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java

줄 번호 83

소스코드

```
78.           java.nio.file.Path path = java.nio.file.Paths.get(fileName);
79.           is = java.nio.file.Files.newInputStream(path, java.nio.file.
StandardOpenOption.READ);
80.           byte[] b = new byte[1000];
81.           int size = is.read(b);
82.           response.getWriter()
83.                   .println(
84.                           "The beginning of file: '"
85.                                   + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName)
86.                                   + "' is:₩n₩n");
87.           response.getWriter()
88.                   .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(new String
(b, 0, size)));
```

이슈 ID	274643
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	88

소스코드

```
83.                 .println(
84.                         "The beginning of file: '"
85.                                 + org.owasp.esapi.ESAPI.encoder().encodeForHTML
(fileName)
86.                                 + "' is:₩n₩n");
87.             response.getWriter()
88.                     .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(new String
(b, 0, size)));
89.             is.close();
90.         } catch (Exception e) {
91.             System.out.println("Couldn't open InputStream on file: '" + fileName + "'");
92.             response.getWriter()
93.                     .println(
```

이슈 ID	274642
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00065.java
줄 번호	91

소스코드

```
86.                                 + "' is:₩n₩n");
87.             response.getWriter()
88.                     .println(org.owasp.esapi.ESAPI.encoder().encodeForHTML(new String
(b, 0, size)));
89.             is.close();
90.         } catch (Exception e) {
```

```
91.         System.out.println("Couldn't open InputStream on file: '" + fileName + "'");
92.         response.getWriter()
93.                 .println(
94.                         "Problem getting InputStream: "
95.                                 + org.owasp
96.                                         .esapi
```

이슈 ID	274648
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00067.java
줄 번호	127

소스코드

```
122.         rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
123.         // e.g., /benchmark/sql-01/BenchmarkTest01001
124.         request.getSession().setAttribute(cookieName, rememberMeKey);
125.         response.addCookie(rememberMe);
126.         response.getWriter()
127.                 .println(
128.                         user
129.                                 + " has been remembered with cookie: "
130.                                 + rememberMe.getName()
131.                                 + " whose value is: "
132.                                 + rememberMe.getValue()
```

이슈 ID	274660
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00070.java
줄 번호	103

소스코드

```
98.                 "hash_value="
99.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
100.                         + "₩n");
101.         fw.close();
102.         response.getWriter()
103.                 .println(
104.                         "Sensitive value '"
105.                                 + org.owasp
106.                                         .esapi
107.                                         .ESAPI
108.                                         .encoder()
```

이슈 ID	274665
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00076.java
줄 번호	121

소스코드

```
116.                 "hash_value="
117.                         + org.owasp.esapi.ESAPI.encoder().encodeForBase64(result, true)
118.                         + "₩n");
119.         fw.close();
120.         response.getWriter()
121.                 .println(
122.                         "Sensitive value '"
123.                                 + org.owasp
124.                                         .esapi
125.                                         .ESAPI
126.                                         .encoder()
```

이슈 ID	274677
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00084.java

줄 번호 106

소스코드

```
101.    rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
102.    // e.g., /benchmark/sql-01/BenchmarkTest01001
103.    request.getSession().setAttribute(cookieName, rememberMeKey);
104.    response.addCookie(rememberMe);
105.    response.getWriter()
106.        .println(
107.            user
108.                + " has been remembered with cookie: "
109.                + rememberMe.getName()
110.                + " whose value is: "
111.                + rememberMe.getValue()
```

이슈 ID 274683

파일 BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00086.java

줄 번호 110

소스코드

```
105.    rememberMe.setPath(request.getRequestURI()); // i.e., set path to JUST
this servlet
106.    // e.g., /benchmark/sql-01/BenchmarkTest01001
107.    request.getSession().setAttribute(cookieName, rememberMeKey);
108.    response.addCookie(rememberMe);
109.    response.getWriter()
110.        .println(
111.            user
112.                + " has been remembered with cookie: "
113.                + rememberMe.getName()
114.                + " whose value is: "
115.                + rememberMe.getValue()
```

이슈 ID	274692
파일	BenchmarkJava-master/src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00087.java
줄 번호	98

소스코드

```
93.     cookie.setPath(request.getRequestURI()); // i.e., set path to JUST this servlet
94.     // e.g., /benchmark/sql-01/BenchmarkTest01001
95.     response.addCookie(cookie);
96.
97.     response.getWriter()
98.             .println(
99.                     "Created cookie: 'SomeCookie': with value: '"
100.                            + org.owasp.esapi.ESAPI.encoder().encodeForHTML(str)
101.                            + "' and secure flag set to: false");
102.    }
103. }
```

● [규칙 이름] Apache-2.0 라이선스 컴포넌트 사용 (낮음, 공통)

Apache License 2.0(SPDX ID: Apache-2.0) 라이선스에는 고지 의무가 있습니다. 라이선스 전문을 확인하여 사용에 유의해야 합니다.

오픈소스 라이선스 정보

라이선스 이름	Apache License 2.0

오픈소스 컴포넌트

이슈 ID	274447
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	commons-lang:commons-lang
컴포넌트 버전	2.6

오픈소스 라이선스 정보

라이선스 이름	Apache License 2.0

오픈소스 컴포넌트

이슈 ID	274452
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	commons-dbcp:commons-dbcp
컴포넌트 버전	1.4

● [규칙 이름] CDDL-1.0 라이선스 컴포넌트 사용 (보통, 공통)

Common Development and Distribution License 1.0(SPDX ID: CDDL-1.0) 라이선스에는 고지 의무, 파일 단위의 소스 코드 공개 의무가 있습니다. 라이선스 전문을 확인하여 사용에 유의해야 합니다.

오픈소스 라이선스 정보

라이선스 이름	Common Development and Distribution License 1.0

오픈소스 컴포넌트

이슈 ID	274435
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	javax:javaee-api
컴포넌트 버전	8.0.1

● [규칙 이름] CDDL-1.1 라이선스 컴포넌트 사용 (보통, 공통)

Common Development and Distribution License 1.1(SPDX ID: CDDL-1.1) 라이선스에는 고지 의무, 파일 단위의 소스 코드 공개 의무가 있습니다. 라이선스 전문을 확인하여 사용에 유의해야 합니다.

오픈소스 라이선스 정보

라이선스 이름	Common Development and Distribution License 1.1

오픈소스 컴포넌트

이슈 ID	274436
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	javax:javaee-api
컴포넌트 버전	8.0.1

오픈소스 라이선스 정보

라이선스 이름	Common Development and Distribution License 1.1

오픈소스 컴포넌트

이슈 ID	274455
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	com.sun.jersey:jersey-servlet
컴포넌트 버전	1.19.4

● [규칙 이름] GPL-2.0-with-classpath-exception 라이선스 컴포넌트 사용 (높음, 공통)

GNU General Public License v2.0 w/Classpath exception(SPDX ID: GPL-2.0-with-classpath-exception) 라이선스에는 고지 의무, 제약 사항, 코드 공개 범위 정보가 없습니다. 라이선스 전문을 확인해야 합니다.

오픈소스 라이선스 정보

라이선스 이름	GNU General Public License v2.0 w/Classpath exception

오픈소스 컴포넌트

이슈 ID	274454

파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	com.sun.jersey:jersey-servlet
컴포넌트 버전	1.19.4

● [규칙 이름] W3C 라이선스 컴포넌트 사용 (낮음, 공통)

W3C Software Notice and License (2002-12-31)(SPDX ID: W3C) 라이선스에는 고지 의무가 있습니다. 라이선스 전문을 확인하여 사용에 유의해야 합니다.

오픈소스 라이선스 정보

라이선스 이름	W3C Software Notice and License (2002-12-31)

오픈소스 컴포넌트

이슈 ID	274453
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	xml-apis:xml-apis
컴포넌트 버전	1.4.01

● [규칙 이름] 취약한 컴포넌트 사용 (매우 높음, 공통)

애플리케이션의 보안을 위해 중요한 조치 중 하나는 알려진 보안 취약점을 가진 컴포넌트를 사용하지 않는 것입니다.

취약한 컴포넌트는 꼭 최신 안전 버전으로 업데이트하거나 다른 안전한 컴포넌트로 교체해야 합니다.

알려진 컴포넌트 취약점

취약점 이름	CVE-2024-30171
취약점 설명	An issue was discovered in Bouncy Castle Java TLS API and JSSE Provider before 1.78. Timing-based leakage may occur in RSA based handshakes because of exception processing.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A

-	-	-	-	-	-	-	-	-

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274437
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	GHSA-wjxj-5m7g-mg7q
취약점 설명	Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle org.bouncycastle.openssl. PEMParser class. This class parses OpenSSL PEM encoded streams containing X.509 certificates, PKCS8 encoded keys, and PKCS7 objects. Parsing a file that has crafted ASN.1 data through the PEMParser causes an OutOfMemoryError, which can enable a denial of service attack.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	LOCAL	LOW	NONE	REQUIRED	UNCHANGED	NONE	NONE	HIGH

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274438
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	CVE-2023-33202
취약점 설명	Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle org.bouncycastle.openssl. PEMParser class. This class parses OpenSSL PEM encoded streams containing X.509 certificates, PKCS8 encoded keys, and PKCS7 objects. Parsing a file that has crafted ASN.1 data through the PEMParser causes an OutOfMemoryError, which can enable a denial of service attack. (For users of the FIPS Java API: BC-FJA 1.0.2.3 and earlier are affected; BC-FJA 1.0.2.4 is fixed.)

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	LOCAL	LOW	NONE	REQUIRED	UNCHANGED	NONE	NONE	HIGH

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274439
파일	BenchmarkJava-master/pom.xml

컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	GHSA-8xfc-gm6g-vgpv
취약점 설명	An issue was discovered in ECCurve.java and ECCurve.cs in Bouncy Castle Java (BC Java) before 1.78, BC Java LTS before 2.73.6, BC-FJA before 1.0.2.5, and BC C# .Net before 2.3.1. Importing an EC certificate with crafted F2m parameters can lead to excessive CPU consumption during the evaluation of the curve parameters.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	NONE	NONE	UNCHANGED	NONE	NONE	LOW

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274440
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	CVE-2023-33201
취약점 설명	Bouncy Castle For Java before 1.74 is affected by an LDAP injection vulnerability. The vulnerability only affects applications that use an LDAP CertStore from Bouncy Castle to validate X.509 certificates. During the certificate validation process, Bouncy Castle inserts the certificate's Subject Name into an LDAP search filter without any escaping, which leads to an LDAP injection vulnerability.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	NONE	NONE	UNCHANGED	LOW	NONE	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274441
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

1.46 이상 1.48 이하 버전으로 업데이트하세요.

알려진 컴포넌트 취약점

취약점 이름	CVE-2024-29857
취약점 설명	An issue was discovered in ECCurve.java and ECCurve.cs in Bouncy Castle Java (BC Java) before 1.78, BC Java LTS before 2.73.6, BC-FJA

before 1.0.2.5, and BC C# .Net before 2.3.1. Importing an EC certificate with crafted F2m parameters can lead to excessive CPU consumption during the evaluation of the curve parameters.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	-	-	-	-	-	-	-	-

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274442
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	GHSA-m44j-cfrm-g8qc
취약점 설명	An issue was discovered in Bouncy Castle Java Cryptography APIs starting in 1.73 and before 1.78. An Ed25519 verification code infinite loop can occur via a crafted signature and public key.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	NONE	NONE	UNCHANGED	NONE	NONE	LOW

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274443
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	CVE-2024-30172
취약점 설명	An issue was discovered in Bouncy Castle Java Cryptography APIs before 1.78. An Ed25519 verification code infinite loop can occur via a crafted signature and public key.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	-	-	-	-	-	-	-	-

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274444
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on

컴포넌트 버전 1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름 GHSA-hr8g-6v94-x4m9

취약점 설명 Bouncy Castle provides the `X509LDAPCertStoreSpi.java` class which can be used in conjunction with the CertPath API for validating certificate paths. Pre-1.73 the implementation did not check the X.500 name of any certificate, subject, or issuer being passed in for LDAP wild cards, meaning the presence of a wild car may lead to Information Disclosure. A potential attack would be to generate a self-signed certificate with a subject name that contains special characters, e.g: `CN=Subject*)(objectclass=`. This will be included into the filter and provides the attacker ability to specify additional attributes in the search query. This can be exploited as a blind LDAP injection: an attacker can enumerate valid attribute values using the boolean blind injection technique. The exploitation depends on the structure of the target LDAP directory, as well as what kind of errors are exposed to the user. Changes to the `X509LDAPCertStoreSpi.java` class add the additional checking of any X.500 name used to correctly escape wild card characters.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	NONE	NONE	UNCHANGED	LOW	NONE	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID 274445

파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

1.46 이상 1.48 이하 버전으로 업데이트하세요.

알려진 컴포넌트 취약점

취약점 이름	GHSA-v435-xc8x-wvr9
취약점 설명	An issue was discovered in Bouncy Castle Java TLS API and JSSE Provider before 1.78. Timing-based leakage may occur in RSA based handshakes because of exception processing.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	HIGH	NONE	NONE	UNCHANGED	HIGH	NONE	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274446
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.bouncycastle:bcprov-jdk15on
컴포넌트 버전	1.70

조치 방안

검색된 조치 방안이 없습니다.

알려진 컴포넌트 취약점

취약점 이름	CVE-2020-25638
취약점 설명	A flaw was found in hibernate-core in versions prior to and including 5.4.23.Final. A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals when a literal is used in the SQL comments of the query. This flaw could allow an attacker to access unauthorized information or possibly conduct further attacks. The highest threat from this vulnerability is to data confidentiality and integrity.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	HIGH	NONE	NONE	UNCHANGED	HIGH	HIGH	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	NETWORK	MEDIUM	NONE	PARTIAL	PARTIAL	NONE

오픈소스 컴포넌트

이슈 ID	274448
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.hibernate:hibernate-core
컴포넌트 버전	3.6.10.Final

조치 방안

5.4.24.Final 이상 5.6.15.Final 이하 버전으로 업데이트하세요.

알려진 컴포넌트 취약점

취약점 이름	GHSA-j8jw-g6fq-mp7h
	A flaw was found in hibernate-core in versions prior to 5.3.20.Final and in 5.4.0.Final up to and including 5.4.23.Final. A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals

취약점 설명	when a literal is used in the SQL comments of the query. This flaw could allow an attacker to access unauthorized information or possibly conduct further attacks. The highest threat from this vulnerability is to data confidentiality and integrity.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	HIGH	NONE	NONE	UNCHANGED	HIGH	HIGH	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274449
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.hibernate:hibernate-core
컴포넌트 버전	3.6.10.Final

조치 방안

5.4.24.Final 이상 5.6.15.Final 이하 버전으로 업데이트하세요.

알려진 컴포넌트 취약점

취약점 이름	GHSA-8grg-q944-cch5
취약점 설명	A flaw was found in Hibernate ORM in versions before 5.3.18, 5.4.18 and 5.5.0.Beta1. A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals when a literal is used in the SELECT or GROUP BY parts of the query. This flaw could allow an attacker to access unauthorized information or possibly conduct further attacks.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	LOW	NONE	UNCHANGED	HIGH	NONE	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	-	-	-	-	-	-

오픈소스 컴포넌트

이슈 ID	274450
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.hibernate:hibernate-core
컴포넌트 버전	3.6.10.Final

조치 방안

5.5.0.Beta1 이상 5.6.15.Final 이하 버전으로 업데이트하세요.

알려진 컴포넌트 취약점

취약점 이름	CVE-2019-14900
취약점 설명	A flaw was found in Hibernate ORM in versions before 5.3.18, 5.4.18 and 5.5.0.Beta1. A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals when a literal is used in the SELECT or GROUP BY parts of the query. This flaw could allow an attacker to access unauthorized information or possibly conduct further attacks.

CVSS 3

점수	AV	AC	PR	UI	S	C	I	A
-	NETWORK	LOW	LOW	NONE	UNCHANGED	HIGH	NONE	NONE

CVSS 2

점수	AV	AC	Au	C	I	A
-	NETWORK	LOW	SINGLE	PARTIAL	NONE	NONE

오픈소스 컴포넌트

이슈 ID	274451
파일	BenchmarkJava-master/pom.xml
컴포넌트 이름	org.hibernate:hibernate-core
컴포넌트 버전	3.6.10.Final

조치 방안

5.5.0.Beta1 이상 5.6.15.Final 이하 버전으로 업데이트하세요.

■ 제외된 이슈 정보

제외된 이슈가 없습니다.

■ 이슈 검출 규칙 정보

유형	위험도	언어	이름
소스코드	낮음	ABAP	너무 긴 한 줄
소스코드	낮음	ABAP	부적절한 WHEN OTHERS 절 위치
소스코드	낮음	ABAP	DATA 변수 명명 규칭 위반
소스코드	낮음	ABAP	클래스 명명 규칙 위반
소스코드	낮음	ABAP	폼 명명 규칙 위반
소스코드	낮음	ABAP	함수 명명 규칙 위반
소스코드	낮음	ABAP	인터페이스 명명 규칙 위반
소스코드	낮음	ABAP	메소드 명명 규칙 위반
소스코드	낮음	ABAP	REPORT 명명 규칙 위반
소스코드	낮음	ABAP	프로그램 명명 규칙 위반
소스코드	낮음	C	포인터 변수에 정수형 널 대입
소스코드	매우 낮음	C	구조체 및 공용체 내부 구현 노출
소스코드	낮음	C++	비 표준 이스케이프 시퀀스
소스코드	낮음	C++	다른 범위의 중복된 이름
소스코드	낮음	C++	유일하지 않은 사용자 정의 타입 이름
소스코드	낮음	C++	중복된 태그 이름
소스코드	낮음	C++	재사용된 정적 식별자
소스코드	매우 낮음	C++	재사용된 식별자
소스코드	매우 낮음	C++	타입 정의 시 누락된 크기 명시
소스코드	낮음	C++	부적절한 논리 연산의 피연산자
소스코드	낮음	C++	continue 구문 사용
소스코드	매우 낮음	C++	누락된 switch 구문 내 default 케이스
소스코드	낮음	C++	너무 높은 포인터 단계
소스코드	낮음	C++	너무 복잡한 함수
소스코드	낮음	C++	논리 연산의 피연산자로 사용된 비 불린 값
소스코드	낮음	C++	여러 개의 break 구문
소스코드	매우 낮음	C++	중첩된 블록 내 케이스 구문
소스코드	낮음	C++	누락된 파라미터 식별자
소스코드	매우 낮음	C++	일치하지 않는 인자 이름
소스코드	낮음	C++	후행하는 포함 구문
소스코드	낮음	C++	일관성 없는 파라미터 선언
소스코드	매우 낮음	C++	비 const 멤버 함수

소스코드	매우 낮음	C++	헤더 파일 내 외부 전역 변수 선언
소스코드	매우 낮음	C++	헤더 파일 내 누락된 인라인 함수 정의
소스코드	매우 낮음	C++	소스 파일 내 템플릿 정의
소스코드	낮음	C++	불분명한 식별자
소스코드	매우 낮음	C++	모호한 문법 사용
소스코드	낮음	C++	식별자 내 혼란스러운 문자
소스코드	낮음	C++	하나의 타입 정의에 작성된 여러 개의 타입 이름
소스코드	낮음	C++	사용자 정의 배열 타입
소스코드	낮음	C++	C 스타일 배열 사용
소스코드	매우 낮음	C++	* 및 & 토큰 규칙 위반
소스코드	매우 낮음	C++	부호가 없는 타입 사용
소스코드	낮음	C++	부적절한 상수 값과 비교
소스코드	매우 낮음	C++	삼항 연산자 사용
소스코드	낮음	C++	비 정적으로 오버로딩된 new 연산자
소스코드	낮음	C++	전처리 구문 공백 규칙 위반
소스코드	낮음	C++	중첩된 전처리 구문 공백 규칙 위반
소스코드	낮음	C++	#if 지시자 사용
소스코드	낮음	C++	포함 구문 내 경로 지정자
소스코드	낮음	C++	널 매크로 사용
소스코드	낮음	C++	클래스 구성 규칙 위반
소스코드	낮음	C++	읽기 어려운 식별자
소스코드	낮음	C++	소스 파일 내 타입 선언
소스코드	낮음	C++	익명 네임스페이스 밖의 비 멤버 함수
소스코드	낮음	C++	구조체 내 멤버 함수
소스코드	낮음	C++	열거형 타입과 정수 값 비교
소스코드	매우 낮음	C++	상위 범위 변수 이름을 포함하는 하위 범위 변수 이름
소스코드	매우 낮음	C++	변수 명명 규칙 위반(BSSC)
소스코드	매우 낮음	C++	함수 명명 규칙 위반
소스코드	매우 낮음	C++	사용자 정의 타입 명명 규칙 위반
소스코드	매우 낮음	C++	클래스 명명 규칙 위반
소스코드	낮음	C++	부적절한 접근 지정자
소스코드	낮음	C++	이른 변수 정의
소스코드	낮음	C++	쓸모없는 형변환
소스코드	낮음	C++	부적절한 매크로 이름
소스코드	보통	C++	지정되지 않은 예외 생성
소스코드	보통	C++	누락된 sizeof 함수 호출

소스코드	보통	C++	잘못된 필드 배열의 길이
소스코드	보통	C++	중복된 파라미터
소스코드	매우 낮음	C++	주석 내 코드
소스코드	매우 낮음	C++	가상 함수의 중복 정의
소스코드	매우 낮음	C++	순수 가상 함수의 오버라이딩
소스코드	보통	C++	잘못된 루프 카운터 비교
소스코드	매우 낮음	C++	1개가 아닌 루프 카운터
소스코드	낮음	C++	가까운 범위에 있지 않은 case 구문
소스코드	매우 낮음	C++	헤더 파일에 있는 이름 없는 네임스페이스
소스코드	매우 낮음	C++	메인 함수가 아닌 함수의 이름을 main으로 사용
소스코드	낮음	C++	C++에서 함수 반환 결과 미사용
소스코드	낮음	C++	같은 네임스페이스에서의 함수 오버로딩
소스코드	낮음	C++	cvalue 식을 캐스팅
소스코드	낮음	C++	포인터로 붕괴하는 배열
소스코드	매우 낮음	C++	잘못된 전처리 구문 토큰
소스코드	매우 낮음	C++	전역 범위에서 변수 및 함수 선언
소스코드	매우 낮음	C++	부모 클래스의 생성자 호출 누락
소스코드	매우 낮음	C++	예외 명세에 포함되지 않은 예외 던짐
소스코드	매우 낮음	C++	다른 파일에서 선언된 템플릿 특수화
소스코드	매우 낮음	C++	하나 이상의 정의 금지
소스코드	매우 낮음	C++	너무 많은 함수 호출
소스코드	보통	C++	goto 구문 사용
소스코드	낮음	C++	너무 많은 반환 구문
소스코드	매우 낮음	C++	비 매크로 상수
소스코드	매우 낮음	C++	반복문 조건 내 상수
소스코드	매우 낮음	C++	함수 중간에 존재하는 변수 선언
소스코드	매우 낮음	C++	너무 많은 연속 if 구문
소스코드	매우 낮음	C++	enum 내 상수에 지정된 접두사 미사용
소스코드	매우 낮음	C++	헝가리안 표기법 위반
소스코드	높음	C++	누락된 변수의 초기화
소스코드	매우 낮음	C++	매크로 명명 규칙 위반
소스코드	매우 낮음	C++	상수 명명 규칙 위반
소스코드	보통	C++	개별 라인을 가지지 않은 중괄호 형식
소스코드	낮음	C++	중복된 이름
소스코드	매우 낮음	C++	전역 변수 명명 규칙 위반
소스코드	매우 낮음	C++	너무 긴 이름

소스코드	보통	C++	누락된 명시적 배열 크기
소스코드	매우 낮음	C++	블록 주석 형식 위반
소스코드	매우 낮음	C++	너무 긴 주석 전 소스 코드
소스코드	매우 낮음	C++	누락된 주석 앞 빈 줄
소스코드	매우 낮음	C++	주석 들여쓰기 규칙 위반
소스코드	낮음	C++	잘못된 열거자 연산
소스코드	매우 낮음	C++	typedef로 부호 없는 숫자 타입 선언
소스코드	매우 낮음	C++	잘못된 매크로의 이름 길이
소스코드	매우 낮음	C++	매크로와 중복된 이름
소스코드	매우 낮음	C++	default 절로 시작하거나 끝나지 않은 switch 구문
소스코드	보통	C++	문자 배열의 잘못된 초기화
소스코드	매우 낮음	C++	case 문에서 break 구문 누락(Misra2008)
소스코드	낮음	C++	switch 구문에 case 절 누락(Misra2012)
소스코드	매우 낮음	C++	extern으로 선언된 빈 배열의 크기 미지정
소스코드	매우 낮음	C++	if-else 문에서 else 누락
소스코드	매우 낮음	C++	잘못된 전역 변수의 이름 길이(Misra2012)
소스코드	낮음	C++	부적절한 sizeof 인자의 연산자
소스코드	매우 낮음	C++	포인터 변수 명명 규칙 위반
소스코드	매우 낮음	C++	너무 긴 한 줄
소스코드	매우 낮음	C++	잘못된 길이의 들여쓰기
소스코드	매우 낮음	C++	한 줄에 작성된 여러 개의 구문
소스코드	매우 낮음	C++	키워드 주위 공백 규칙 위반
소스코드	매우 낮음	C++	누락된 이항 연산자 주위 공백
소스코드	매우 낮음	C++	메소드 호출 시 공백 규칙 위반
소스코드	매우 낮음	C++	for 구문 공백 규칙 위반
소스코드	매우 낮음	C++	단항 연산자 주위 공백
소스코드	매우 낮음	C++	형변환 시 공백 규칙 위반
소스코드	매우 낮음	C++	접근 연산자 주위 공백 규칙 위반
소스코드	매우 낮음	C++	잘못된 포함 순서
소스코드	매우 낮음	C++	포인터 연산자 주위 공백 규칙 위반
소스코드	매우 낮음	C++	함수 주위 공백 규칙 위반
소스코드	매우 낮음	C++	괄호 내 공백 규칙 위반
소스코드	매우 낮음	C++	세미콜론 주위 공백 규칙 위반
소스코드	매우 낮음	C++	블록 들여쓰기 규칙 위반
소스코드	매우 낮음	C++	누락된 복합 구문 중괄호
소스코드	매우 낮음	C++	누락된 복합 구문 시작 중괄호

소스코드	매우 낮음	C++	while 구문 공백 규칙 위반
소스코드	매우 낮음	C++	if 구문 공백 규칙 위반
소스코드	매우 낮음	C++	switch 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	C++	switch 구문 공백 규칙 위반
소스코드	매우 낮음	C++	case 및 default 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	C++	쉼표 주위 공백 규칙 위반
소스코드	매우 낮음	C++	너무 많은 논리 연산
소스코드	매우 낮음	C++	타입 명명 규칙 위반
소스코드	매우 낮음	C++	함수 사이 전역 변수 선언
소스코드	매우 낮음	C++	블록 상단에 선언되지 않은 지역 변수
소스코드	매우 낮음	C++	반복문 내 변수 선언 규칙 위반
소스코드	매우 낮음	C++	부적절한 크기 검사
소스코드	매우 낮음	C++	하나의 소스 파일에서만 사용되는 비 표준 전역 변수
소스코드	매우 낮음	C++	헤더 파일 내 데이터 정의 시 누락된 extern
소스코드	매우 낮음	C++	타입 주석 형식 위반
소스코드	매우 낮음	C++	한 줄에 작성된 여러 개의 선언
소스코드	낮음	C++	불필요한 변수 범위
소스코드	보통	C++	클래스 주석 형식 위반
소스코드	매우 낮음	C++	함수 주석 형식 위반
소스코드	보통	C++	필드 주석 형식 위반
소스코드	보통	C++	파일 주석 형식 위반
소스코드	매우 낮음	C++	변수 명명 규칙 위반
소스코드	보통	C#	비어 있는 인터페이스
소스코드	보통	C#	구조체 안에 변경 가능한 형식 포함
소스코드	낮음	C#	out 매개 변수 사용
소스코드	매우 낮음	C#	잘못된 네임스페이스 이름
소스코드	매우 낮음	C#	파라미터 명명 규칙 위반
소스코드	매우 낮음	C#	잘못된 지역 변수 이름
소스코드	낮음	C#	단일 필드 클래스 정의
소스코드	낮음	Java	정수 리터럴 직접 사용
소스코드	보통	Java	허용되지 않은 타입의 예외 던짐
소스코드	매우 낮음	Java	최대 줄길이 제한
소스코드	매우 낮음	Java	일반적 중괄호 줄 바꿈 규칙 위반
소스코드	매우 낮음	Java	조건으로 단일 변수 사용
소스코드	매우 낮음	Java	누락된 else 절
소스코드	매우 낮음	Java	한정자 나열 순서 위반

소스코드	매우 낮음	Java	복합 구문에서 중괄호 누락
소스코드	매우 낮음	Java	너무 넓은 변수 활성 범위
소스코드	보통	Java	Switch 문에서 Default 미사용
소스코드	매우 낮음	Java	if-else 반환문의 잘못된 사용
소스코드	매우 낮음	Java	빈 while 구문
소스코드	매우 낮음	Java	문서화 주석 형식 위반
소스코드	매우 낮음	Java	If 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	while 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	for 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	do-while 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	switch 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	try-catch 키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	증분 혹은 감소 연산자에서 잘못된 공백 사용
소스코드	매우 낮음	Java	assert 키워드 주위 공백 규칙 위반
소스코드	보통	Java	클래스 주석 형식 위반
소스코드	보통	Java	필드 주석 형식 위반
소스코드	보통	Java	일반 주석 형식 위반
소스코드	보통	Java	메소드 주석 형식 위반
소스코드	보통	Java	패키지 주석 형식 위반
소스코드	보통	Java	파일 주석 형식 위반
소스코드	매우 낮음	Java	이름에 사용된 특수 문자
소스코드	매우 낮음	Java	너무 긴 이름
소스코드	매우 낮음	Java	하드코드된 숫자
소스코드	매우 낮음	Java	모호한 수식 우선순위
소스코드	매우 낮음	Java	너무 복잡한 for 구문
소스코드	매우 낮음	Java	선언 시 누락된 탭 문자
소스코드	매우 낮음	Java	파라미터 명명 규칙 위반
소스코드	낮음	Java	중복된 이름
소스코드	매우 낮음	Java	블록 주석 형식 위반
소스코드	매우 낮음	Java	한 줄 주석 형식 위반
소스코드	매우 낮음	Java	소스 코드 뒤 주석 형식 위반
소스코드	매우 낮음	Java	누락된 블록 마지막 주석
소스코드	매우 낮음	Java	단항 연산자 주위 공백
소스코드	매우 낮음	Java	괄호 없이 조건 연산 사용
소스코드	매우 낮음	Java	증감연산자의 부적절한 사용
소스코드	매우 낮음	Java	누락된 이항 연산자 주위 공백

소스코드	매우 낮음	Java	한 줄에 작성된 여러 개의 구문
소스코드	매우 낮음	Java	블록 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	누락된 패키지 선언
소스코드	매우 낮음	Java	반환 구문에서 누락된 괄호
소스코드	매우 낮음	Java	누락된 if 구문 중괄호
소스코드	매우 낮음	Java	누락된 finally 블록
소스코드	매우 낮음	Java	for 구문 공백 규칙 위반
소스코드	매우 낮음	Java	형변환 시 공백 규칙 위반
소스코드	매우 낮음	Java	if 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	한 줄에 작성된 여러 개의 선언
소스코드	매우 낮음	Java	블록 중간에 존재하는 선언
소스코드	매우 낮음	Java	중괄호 형식 위반
소스코드	매우 낮음	Java	부적절한 배열 선언
소스코드	낮음	Java	동기화되지 않은 public 메소드
소스코드	매우 낮음	Java	동기화 구문 사용
소스코드	매우 낮음	Java	누락된 복합 구문 시작 중괄호
소스코드	매우 낮음	Java	상수 명명 규칙 위반
소스코드	매우 낮음	Java	변수 명명 규칙 위반
소스코드	매우 낮음	Java	메소드 명명 규칙 위반
소스코드	매우 낮음	Java	클래스 명명 규칙 위반
소스코드	매우 낮음	Java	반복문 내에서의 잠금 혹은 잠금 해제
소스코드	매우 낮음	Java	반복문 내에서의 스트림 읽기
소스코드	매우 낮음	Java	누락된 빈 줄
소스코드	매우 낮음	Java	누락된 public 클래스 혹은 인터페이스
소스코드	매우 낮음	Java	처리되지 않은 추가 예외
소스코드	매우 낮음	Java	메소드 호출 시 공백 규칙 위반
소스코드	매우 낮음	Java	누락된 지역 변수의 초기화
소스코드	매우 낮음	Java	필드의 즉시 초기화
소스코드	매우 낮음	Java	괄호 내 공백 규칙 위반
소스코드	매우 낮음	Java	키워드 주위 공백 규칙 위반
소스코드	매우 낮음	Java	클래스 구성 규칙 위반
소스코드	매우 낮음	Java	소스 코드 시작점에 존재하는 들여쓰기
소스코드	매우 낮음	Java	너무 긴 한 줄
소스코드	매우 낮음	Java	잘못된 길이의 들여쓰기
소스코드	매우 낮음	Java	for 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	메소드 선언 규칙 위반

소스코드	매우 낮음	Java	접근 연산자 주위 공백 규칙 위반
소스코드	매우 낮음	Java	줄 바꿈 규칙 위반
소스코드	매우 낮음	Java	import 구문 사이 누락된 빈 줄
소스코드	매우 낮음	Java	메소드 구역 사이 누락된 빈 줄
소스코드	매우 낮음	Java	온디맨드 방식으로 import
소스코드	매우 낮음	Java	클래스 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	메소드 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	필드 선언 규칙 위반
소스코드	매우 낮음	Java	필드 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	세미콜론 주위 공백 규칙 위반
소스코드	매우 낮음	Java	변수 선언 규칙 위반
소스코드	매우 낮음	Java	반복문 내 변수 선언 규칙 위반
소스코드	매우 낮음	Java	정수 타입 사용
소스코드	매우 낮음	Java	실수 타입 사용
소스코드	매우 낮음	Java	while 구문 공백 규칙 위반
소스코드	매우 낮음	Java	do 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	if 구문 공백 규칙 위반
소스코드	매우 낮음	Java	switch 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	switch 구문 공백 규칙 위반
소스코드	매우 낮음	Java	case 및 default 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	한 줄에 작성된 여러 개의 case 구문
소스코드	매우 낮음	Java	try 구문 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	catch 구문 공백 규칙 위반
소스코드	매우 낮음	Java	쉼표 주위 공백 규칙 위반
소스코드	매우 낮음	Java	내부 클래스의 상수
소스코드	매우 낮음	Java	너무 많은 논리 연산
소스코드	매우 낮음	Java	+ 및 += 연산자로 더해진 문자열
소스코드	매우 낮음	Java	반복문 내에서의 인스턴스 생성
소스코드	매우 낮음	Java	패키지 명명 규칙 위반
소스코드	매우 낮음	Java	패키지 들여쓰기 규칙 위반
소스코드	매우 낮음	Java	빈 분기문
소스코드	매우 낮음	Java	equals 메소드를 통한 문자열 비교
소스코드	매우 낮음	Java	인스턴스에 접근
소스코드	매우 낮음	Java	쓸모없는 이름
소스코드	매우 낮음	Java	주석 들여쓰기 규칙 위반
소스코드	낮음	JS/TS	불필요한 블록

소스코드	낮음	JS/TS	불필요한 괄호
소스코드	매우 낮음	JS/TS	누락된 중괄호
소스코드	매우 낮음	JS/TS	변수 명명 규칙 위반
소스코드	보통	SQL	SELECT 구문에서 누락된 컬럼 주석
소스코드	매우 낮음	SQL	테이블 명명 규칙 위반
소스코드	매우 높음	SQL	금지된 테이블 사용
소스코드	매우 높음	SQL	금지된 테이블 컬럼 사용
소스코드	보통	ABAP	BREAK-POINT 구문 사용
소스코드	보통	ABAP	SYSTEM-CALL 구문 사용
소스코드	보통	ABAP	시스템 C 함수 사용
소스코드	낮음	ABAP	CX_ROOT 예외 처리
소스코드	보통	ABAP	누락된 WHEN OTHERS 절
소스코드	낮음	ABAP	너무 적은 WHEN 절
소스코드	낮음	ABAP	빈 catch 블록
소스코드	낮음	ABAP	누락된 ELSE 절
소스코드	보통	ABAP	DATA BEGIN OF OCCURS 구문 사용
소스코드	보통	ABAP	NOT IN 사용
소스코드	높음	ABAP	DELETE 구문에서 누락된 WHERE 절
소스코드	보통	ABAP	반복문 내에서의 EXIT 및 CHECK 구문 사용
소스코드	보통	ABAP	SELECT 구문 내 * 사용
소스코드	보통	ABAP	네이티브 SQL 사용
소스코드	낮음	ABAP	FORM 구문 사용
소스코드	높음	ABAP	UPDATE 구문에서 누락된 WHERE 절
소스코드	낮음	ABAP	REFRESH FROM TABLE 구문 사용
소스코드	낮음	ABAP	반복문 내에서의 SELECT 구문 사용
소스코드	보통	ABAP	너무 깊이 중첩된 제어 흐름
소스코드	보통	ABAP	너무 큰 파일
소스코드	보통	ABAP	중첩된 SELECT 구문 사용
소스코드	보통	ABAP	반복문 내 너무 많은 분기
소스코드	보통	ABAP	누락된 SORT 필드
소스코드	보통	ABAP	SELECT 구문에서 누락된 ORDER BY 절
소스코드	보통	ABAP	SELECT 구문에서 누락된 WHERE 절
소스코드	보통	ABAP	내부 소스 코드 처리 구문 사용
소스코드	보통	ABAP	BYPASSING BUFFER 절 사용
소스코드	보통	ABAP	DISTINCT 연산자 사용
소스코드	낮음	ABAP	중복된 문자열

소스코드	보통	C	잘못된 TRY 및 CATCH 매크로 사용
소스코드	보통	C	트랜잭션에서 누락된 commit 혹은 rollback 함수 호출
소스코드	낮음	C	금지된 한정자 restrict 사용
소스코드	낮음	C	금지된 헤더 tgmath.h 사용
소스코드	보통	C	부적절한 scanf 인자의 타입
소스코드	보통	C	부적절한 할당 블록 크기
소스코드	보통	C	부적절한 타입으로 매개변수 할당
소스코드	보통	C	문자 및 정수 타입을 혼용한 할당
소스코드	높음	C	누락된 필수 함수 호출
소스코드	낮음	C	누락된 함수 선언
소스코드	보통	C	FILE 객체의 잘못된 사용
소스코드	보통	C	배열 타입 매개 변수에 static 키워드 사용
소스코드	낮음	C	값 전달 호출 함수에서 매개 변수 수정
소스코드	매우 낮음	C	inline 함수 선언에서 static 누락
소스코드	낮음	C	키워드를 매크로 이름으로 사용
소스코드	매우 낮음	C	한 줄 주석문을 여러 줄로 작성
소스코드	낮음	C	불필요한 레이블
소스코드	낮음	C	불필요한 태그 선언
소스코드	낮음	C	불필요한 매크로 선언
소스코드	낮음	C	외부 연결 식별자 중복 선언
소스코드	보통	C	초기화 리스트에서 부수 효과 발생
소스코드	보통	C	지정 초기화 리스트에서 부수 효과 발생
소스코드	보통	C	지정 초기화 리스트에서 일부 항목 누락
소스코드	낮음	C	특정 컴파일러 전용 확장 코드 사용
소스코드	매우 낮음	C	전처리 구문 조건식의 결과값 제약 위반
소스코드	매우 낮음	C	지정 초기화 리스트에서 중복 초기화
소스코드	낮음	C	금지된 fenv.h 헤더의 예외 처리 사용
소스코드	낮음	C	금지된 stdarg.h 헤더의 기능 사용
소스코드	낮음	C	너무 긴 함수 코드
소스코드	낮음	C	너무 깊은 중첩 블록
소스코드	낮음	C	함수의 상수 인자 제한 범위 위반
소스코드	낮음	C	너무 많은 실행 경로
소스코드	낮음	C	불확실한 값으로 나누기
소스코드	매우 낮음	C	암시적 함수 선언
소스코드	보통	C	읽기 전용 파일에 쓰기 시도
소스코드	보통	C	반복문 없이 cnd_wait 사용

소스코드	보통	C	스레드 이중 해제
소스코드	낮음	C	추가된 언어 기능 사용 금지
소스코드	낮음	C	<ctype.h> 함수에 전달되는 값의 유효성 검사
소스코드	낮음	C	포인터 인자 타입 호환성 확인
소스코드	낮음	C	반환된 포인터 const 처리
소스코드	낮음	C	반환된 포인터 재사용 금지
소스코드	낮음	C	반환된 EOF 데이터 비교
소스코드	낮음	C	errno 설정 후 오류 확인
소스코드	낮음	C	잘못된 errno 검사
소스코드	낮음	C	불필요한 할당
소스코드	낮음	C++	금지된 인자 사용
소스코드	보통	C++	부적절한 문자열 인자의 값
소스코드	보통	C++	명시되지 않은 인자 사용
소스코드	보통	C++	부적절한 가변 인자
소스코드	보통	C++	암시적 실수 업캐스팅
소스코드	보통	C++	캡슐화되지 않은 어셈블리 언어
소스코드	낮음	C++	너무 긴 식별자
소스코드	매우 낮음	C++	네임스페이스 안에서 중복된 이름
소스코드	보통	C++	부적절한 타입으로 할당
소스코드	높음	C++	대체 된 문자 타입 변수
소스코드	높음	C++	연산의 피연산자로 사용된 문자 값
소스코드	보통	C++	누락된 명시적 형변환
소스코드	보통	C++	사용되지 않은 래퍼 함수
소스코드	낮음	C++	대체 된 부호가 있는 문자 타입 변수
소스코드	매우 낮음	C++	기본 타입의 직접 사용
소스코드	높음	C++	부적절한 비트 필드 타입
소스코드	보통	C++	너무 작은 부호가 있는 정수 타입
소스코드	낮음	C++	8진수 사용
소스코드	낮음	C++	의심스러운 8진수 이스케이프 시퀀스
소스코드	낮음	C++	함수 내 함수 선언
소스코드	보통	C++	불필요한 전역 변수
소스코드	보통	C++	중복된 외부 객체 혹은 함수
소스코드	보통	C++	함수 호출 후 인자 사용 규칙 위반
소스코드	보통	C++	외부 참조 없이 비 정적 객체 혹은 함수 사용
소스코드	보통	C++	누락된 배열 초기화 중괄호
소스코드	보통	C++	불충분한 초기화 값

소스코드	보통	C++	부분적으로 초기화 된 열거형 리스트
소스코드	보통	C++	암시적 정수 형변환
소스코드	보통	C++	암시적 정수 업캐스팅
소스코드	보통	C++	암시적 정수 다운캐스팅
소스코드	보통	C++	일치하지 않는 정의 및 선언의 반환 타입
소스코드	보통	C++	금지된 매크로 사용
소스코드	보통	C++	암시적 실수 형변환
소스코드	보통	C++	암시적 실수 다운캐스팅
소스코드	높음	C++	더 큰 타입으로 정수 형변환
소스코드	높음	C++	부호가 있는 데이터와 없는 데이터 간 변환
소스코드	낮음	C++	더 큰 타입으로 실수 형변환
소스코드	보통	C++	누락된 비트 연산 결과에 대한 형변환
소스코드	높음	C++	포인터 타입에서 정수 타입으로 형변환
소스코드	높음	C++	정수 타입에서 포인터 타입으로 형변환
소스코드	보통	C++	포인터 타입으로 형변환
소스코드	보통	C++	인자로 금지된 문자열 사용
소스코드	보통	C++	모호한 수식 우선순위
소스코드	보통	C++	수식의 부수 효과
소스코드	보통	C++	함수 호출의 부수 효과
소스코드	보통	C++	파라미터의 부수 효과
소스코드	보통	C++	피연산자에 사용된 할당
소스코드	낮음	C++	복합 수식 내에서 volatile 타입 변수 사용
소스코드	보통	C++	크기 검사의 부수 효과
소스코드	보통	C++	잘못된 호출 순서로 인한 함수 호출의 부수 효과
소스코드	높음	C++	논리 연산의 부수 효과
소스코드	보통	C++	명시된 함수 인자 미사용
소스코드	보통	C++	이미 열려진 파일 열기
소스코드	낮음	C++	부적절한 배열 해제
소스코드	낮음	C++	void 파라미터 타입 사용
소스코드	높음	C++	논리 연산 내에서의 volatile 변수 접근
소스코드	높음	C++	부호가 있는 값에 대해 비트 연산
소스코드	높음	C++	부호가 있는 값에 대해 시프트
소스코드	보통	C++	부호가 없는 값에 대해 부정 연산
소스코드	낮음	C++	, 연산자 사용
소스코드	높음	C++	부동 소수점 값에 대해 비트 표현 사용
소스코드	보통	C++	암시적 비교 수식

소스코드	보통	C++	반복문 인덱스로 실수 타입 사용
소스코드	낮음	C++	누락된 반복문 인덱스 변경
소스코드	낮음	C++	누락된 반복문 조건
소스코드	낮음	C++	반복문 내에서 반복문 인덱스를 제외한 변수 변경
소스코드	낮음	C++	반복문 내에서의 여러 개의 초기화
소스코드	보통	C++	명시된 인자 미사용
소스코드	낮음	C++	반복문 내에서 변경된 반복문 인덱스
소스코드	낮음	C++	switch 구문 조건 내 불린 수식
소스코드	낮음	C++	빈 switch 구문
소스코드	낮음	C++	함수의 가변 인자
소스코드	매우 낮음	C++	부적절한 읽기 전용 파라미터 선언
소스코드	보통	C++	함수 식별자의 직접 사용
소스코드	보통	C++	일반 포인터에 대해 산술 연산
소스코드	보통	C++	다른 객체 포인터의 감소 연산
소스코드	보통	C++	인자로 리터럴 사용
소스코드	보통	C++	다른 객체 포인터 간 비교
소스코드	보통	C++	포인터 타입 변수에 대한 배열 접근
소스코드	보통	C++	겹치는 메모리 영역의 객체 간 대입 연산
소스코드	높음	C++	중복 저장소에 할당
소스코드	낮음	C++	금지된 표준 라이브러리 함수 사용
소스코드	낮음	C++	금지된 시간 함수 사용
소스코드	매우 낮음	C++	switch 구문 밖에서의 break 구문 사용
소스코드	보통	C++	비 const 함수 포인터
소스코드	보통	C++	인자로 변수 사용
소스코드	낮음	C++	구조체 및 공용체 내 익명 필드
소스코드	낮음	C++	포인터 타입 재정의
소스코드	낮음	C++	소문자 long 접미사
소스코드	보통	C++	부적절한 가변형 배열 멤버 선언
소스코드	높음	C++	memcmp의 인자로 구조체 사용
소스코드	매우 낮음	C++	가정된 상수 값
소스코드	보통	C++	부적절한 연산자 결합
소스코드	보통	C++	구문과 같은 라인에 위치한 세미콜론
소스코드	보통	C++	비 volatile 참조를 통한 volatile 객체 접근
소스코드	보통	C++	변수가 아닌 인자 사용
소스코드	낮음	C++	임시 객체 수정
소스코드	낮음	C++	누락된 입력 함수에 대한 오류 처리

소스코드	보통	C++	단일 식 내 시프트 및 산술 연산
소스코드	낮음	C++	가정된 부호가 있는 정수 표현법
소스코드	낮음	C++	누락된 부동 소수점 반환 값 형변환
소스코드	매우 낮음	C++	비 const 문자열 포인터
소스코드	보통	C++	다른 종류의 문자열 연결
소스코드	보통	C++	문자열 리터럴의 수정
소스코드	낮음	C++	누락된 메모리 할당 후 형변환
소스코드	높음	C++	하드코드된 데이터
소스코드	보통	C++	가변 길이 멤버 배열과 정적 할당 및 복사
소스코드	낮음	C++	안전하지 않은 시스템 간 바이너리 데이터 전송
소스코드	낮음	C++	잘못된 파일 열기 모드
소스코드	낮음	C++	여러 번 ungetc 함수 사용
소스코드	낮음	C++	부적절한 fsetpot 인자의 값
소스코드	낮음	C++	파일 스트림 오류 확인을 위한 errno 사용
소스코드	보통	C++	포인터 파라미터에 누락된 필요 한정자
소스코드	보통	C++	부적절한 종료 방법
소스코드	보통	C++	초기화되지 않은 errno
소스코드	낮음	C++	time_t 타입 변수의 직접 사용
소스코드	높음	C++	부적절한 assert 구문
소스코드	낮음	C++	너무 복잡한 switch 구문
소스코드	보통	C++	포인터 파라미터에 금지된 한정자 사용
소스코드	낮음	C++	POSIX 잠금 중 차단
소스코드	낮음	C++	누락된 switch 구문 중괄호
소스코드	낮음	C++	누락된 if-else 구문 중괄호
소스코드	보통	C++	유효하지 않은 오류 코드 사용
소스코드	낮음	C++	불완전한 구조체 및 공용체 타입
소스코드	높음	C++	공용체 사용
소스코드	낮음	C++	포함 구문 내 비 표준 문자
소스코드	낮음	C++	포함 파일 이름 형식 위반
소스코드	낮음	C++	블록 내에서 정의 혹은 해제된 매크로
소스코드	낮음	C++	#undef 지시자 사용
소스코드	보통	C++	오류 발생 시 누락된 초기화 함수 호출
소스코드	보통	C++	함수 같은 매크로
소스코드	보통	C++	일치하지 않는 매크로 인자 수
소스코드	낮음	C++	매크로 내 전처리 구문
소스코드	보통	C++	누락된 매크로 인자 주위 괄호

소스코드	낮음	C++	정의되지 않은 식별자
소스코드	낮음	C++	여러 개의 # 혹은 ## 연산
소스코드	매우 낮음	C++	# 혹은 ## 연산자 사용
소스코드	낮음	C++	비 표준 전처리 구문 형식
소스코드	낮음	C++	전처리 구문 뒤 추가 문자
소스코드	낮음	C++	전처리 구문에 후행하는 세미콜론
소스코드	보통	C++	오류 발생 시 부적절한 반환 값
소스코드	매우 낮음	C++	동적 메모리 할당
소스코드	매우 낮음	C++	errno 사용
소스코드	매우 낮음	C++	offsetof 매크로 사용
소스코드	매우 낮음	C++	setjmp 함수 사용
소스코드	매우 낮음	C++	longjmp 함수 사용
소스코드	낮음	C++	시그널 핸들링 사용
소스코드	매우 낮음	C++	stdio.h 사용
소스코드	낮음	C++	와이드 문자열 사용
소스코드	낮음	C++	void 반환 타입 함수 내에서 반환
소스코드	낮음	C++	포인터 연산 사용
소스코드	낮음	C++	setlocale 함수 사용
소스코드	보통	C++	시그널 핸들러의 공유 객체 접근
소스코드	낮음	C++	시그널 핸들러 반환
소스코드	낮음	C++	사용되지 않은 런타임 지정 핸들러
소스코드	보통	C++	멀티스레드 프로그램 내에서 signal 함수 사용
소스코드	보통	C++	오버로딩된 논리 연산자
소스코드	보통	C++	오류 발생 시 누락된 로깅
소스코드	보통	C++	C 스타일 형변환 사용
소스코드	보통	C++	복사 초기화 사용
소스코드	보통	C++	복잡한 클래스에서 누락된 특별 함수
소스코드	보통	C++	동적 할당을 사용한 클래스에서 누락된 특별 함수
소스코드	보통	C++	복잡한 클래스에서 누락된 명시적 소멸자
소스코드	보통	C++	동적 할당을 사용한 클래스에서 누락된 명시적 소멸자
소스코드	매우 낮음	C++	너무 복잡한 인라인 함수
소스코드	매우 낮음	C++	비 가상 인라인 함수
소스코드	매우 낮음	C++	너무 긴 인라인 함수
소스코드	매우 낮음	C++	인라인 멤버 함수 사용
소스코드	높음	C++	하드코드된 인자
소스코드	보통	C++	기본 타입에 대한 변환 연산자

소스코드	보통	C++	클래스 타입에 대한 변환 연산자
소스코드	매우 낮음	C++	누락된 출력 연산자
소스코드	낮음	C++	비 표준 인터페이스
소스코드	보통	C++	누락된 explicit 키워드
소스코드	낮음	C++	비 public 파생
소스코드	보통	C++	베이스 클래스의 비 가상 소멸자
소스코드	보통	C++	위험한 다운캐스팅
소스코드	보통	C++	가상 클래스로의 형변환
소스코드	높음	C++	하드코드된 초기화 값
소스코드	보통	C++	누락된 오버로딩된 함수의 오버라이딩
소스코드	낮음	C++	너무 많은 파라미터
소스코드	낮음	C++	여러 번 변경된 반복문 인덱스
소스코드	매우 낮음	C++	변할 수 있는 반복문 조건
소스코드	낮음	C++	여러 개의 진입 및 탈출 지점
소스코드	낮음	C++	부적절한 switch 구문
소스코드	낮음	C++	for 구문 내에서의 부적절한 변수 선언
소스코드	매우 낮음	C++	누락된 정수 상수 접미사
소스코드	매우 낮음	C++	누락된 부동 소수점 상수 접미사
소스코드	낮음	C++	비 const 전역 및 정적 변수
소스코드	보통	C++	인자로 금지된 숫자 사용
소스코드	높음	C++	부동 소수점 타입에서 정수 타입으로 형변환
소스코드	낮음	C++	암시적 인자 형변환
소스코드	낮음	C++	전역 변수 사용
소스코드	매우 낮음	C++	using 지시자 사용
소스코드	매우 낮음	C++	포함 구문 전 using 구문
소스코드	낮음	C++	네임스페이스 안에서 정적 객체 선언
소스코드	매우 낮음	C++	extern 사용
소스코드	보통	C++	헤더 파일 내 정적 지정자 사용
소스코드	매우 낮음	C++	auto 키워드 사용
소스코드	매우 낮음	C++	register 키워드 사용
소스코드	낮음	C++	누락된 const 한정자
소스코드	보통	C++	대입 연산 결과 사용
소스코드	보통	C++	혼합된 부호가 있는 데이터 및 부호가 없는 데이터
소스코드	보통	C++	혼합된 산술 정밀도
소스코드	낮음	C++	값으로 전달
소스코드	낮음	C++	오버로딩된 숫자 및 포인터 타입

소스코드	낮음	C++	오버로딩된 함수에 대해 기본 인자 사용
소스코드	낮음	C++	배열에 대해 사용자 정의 타입의 미사용
소스코드	낮음	C++	해제 후 초기화 되지 않은 포인터
소스코드	낮음	C++	매크로 내 주석
소스코드	낮음	C++	소스 코드 마지막의 비어있지 않은 줄
소스코드	낮음	C++	포함 파일 이름에 사용된 대문자
소스코드	낮음	C++	부적절한 상수 정의
소스코드	낮음	C++	배열 타입 사용
소스코드	낮음	C++	이중자 사용
소스코드	낮음	C++	비 표준 헤더 파일 사용
소스코드	낮음	C++	비효율적인 컨테이너 내 객체 복사
소스코드	낮음	C++	복사 생성자 사용
소스코드	낮음	C++	동적 배열 할당
소스코드	보통	C++	불필요한 재할당
소스코드	낮음	C++	부적절한 C 스타일 함수로의 벡터 전달
소스코드	높음	C++	키 수정
소스코드	낮음	C++	혼합된 반복자 타입
소스코드	보통	C++	비 순수 조건자 함수
소스코드	보통	C++	금지된 타입 사용
소스코드	보통	C++	STL 알고리즘 사용
소스코드	낮음	C++	auto_ptr 사용
소스코드	낮음	C++	클래스 선언에서 인라인 멤버 함수 정의
소스코드	낮음	C++	누락된 메모리 할당에 대한 예외 처리
소스코드	낮음	C++	너무 높은 역참조 수준
소스코드	낮음	C++	매크로 내에서 포인터 역참조 연산 사용
소스코드	낮음	C++	파일 간 중복된 함수 혹은 변수 이름
소스코드	낮음	C++	누락된 extern 변수 선언
소스코드	낮음	C++	포인터에 대해 사용자 정의 타입의 미사용
소스코드	보통	C++	매크로 연결 시 국제 문자 사용
소스코드	낮음	C++	누락된 상수 간 관계
소스코드	낮음	C++	반복문 내에서 복합 리터럴 주소 사용
소스코드	낮음	C++	유일하지 않은 서로에게 보이는 식별자
소스코드	낮음	C++	시그널 핸들러 내에서의 비 volatile 변수 사용
소스코드	보통	C++	선점된 식별자 사용
소스코드	보통	C++	매크로 인자의 부수 효과
소스코드	보통	C++	사용자 정의 타입의 미사용

소스코드	높음	C++	호환되지 않는 포인터를 통한 변수 접근
소스코드	높음	C++	부적절한 정수 형변환
소스코드	보통	C++	FILE 객체 복사
소스코드	보통	C++	재정의된 errno
소스코드	낮음	C++	함수 프로토타입 내 파라미터 이름
소스코드	보통	C++	생성자 및 소멸자 내에서 가상 함수 호출
소스코드	낮음	C++	하나만 오버로딩된 new 및 delete
소스코드	낮음	C++	오버로딩된 후위 연산자에서의 비 const 값 반환
소스코드	낮음	C++	정적 객체 초기화 중 재귀 호출
소스코드	낮음	C++	멤버 변수에 대해 사용자 정의 타입의 미사용
소스코드	낮음	C++	불완전 클래스 포인터 삭제 혹은 형변환
소스코드	보통	C++	부동 소수점이 아닌 값에 대한 부동 소수점 연산
소스코드	낮음	C++	비 정적 필드를 포함하는 조건자 함수
소스코드	보통	C++	지정된 문자 배열의 길이
소스코드	보통	C++	복사 할당 연산 시 누락된 동일성 검사
소스코드	낮음	C++	정렬되지 않은 포인터 사용
소스코드	보통	C++	소멸자 내에서의 예외 생성
소스코드	보통	C++	생성자의 핸들러 내에서의 필드 접근
소스코드	매우 높음	C++	잘못된 예외 처리 순서
소스코드	낮음	C++	해제 중 예외 생성
소스코드	낮음	C++	금지된 배열 타입 사용
소스코드	보통	C++	비 가상 소멸자
소스코드	보통	C++	복사 할당 연산 시 예외 생성
소스코드	낮음	C++	누락된 생성자 내 초기화
소스코드	보통	C++	잘못된 생성자 내에서의 초기화 순서
소스코드	낮음	C++	추상 클래스의 public 생성자
소스코드	보통	C++	오버로딩된 비 가상 함수
소스코드	낮음	C++	일치하지 않는 기본값
소스코드	낮음	C++	비 protected 복사 생성자
소스코드	보통	C++	허용되지 않은 타입 사용
소스코드	낮음	C++	암시적 가상 함수
소스코드	낮음	C++	const 멤버 함수에서의 비 const 핸들 반환
소스코드	매우 낮음	C++	비 추상 베이스 클래스
소스코드	낮음	C++	너무 많은 베이스 클래스
소스코드	낮음	C++	멤버 이진 연산자
소스코드	낮음	C++	부적절한 [] 연산자 오버로딩

소스코드	낮음	C++	너무 많은 실행 경로
소스코드	낮음	C++	비 클래스 타입 객체 예외 생성
소스코드	보통	C++	참조에 의한 예외 처리
소스코드	낮음	C++	금지된 복합 타입 사용
소스코드	낮음	C++	불린 값에 대해 증가 연산자 사용
소스코드	낮음	C++	정수 타입에서 열거형 타입으로 형변환
소스코드	낮음	C++	암시적 클래스 템플릿 변환
소스코드	보통	C++	상충되는 클래스 템플릿 내 메소드
소스코드	보통	C++	베이스 클래스 객체 컨테이너 내 파생 클래스 객체
소스코드	낮음	C++	컨테이너 크기를 0과 비교
소스코드	낮음	C++	public 베이스 클래스로 STL 컨테이너 사용
소스코드	낮음	C++	금지된 포인터 타입 사용
소스코드	높음	C++	auto_ptr 컨테이너 생성
소스코드	보통	C++	불린 타입 벡터 사용
소스코드	낮음	C++	식별자 명명 규칙 위반
소스코드	매우 낮음	C++	소스 파일 내 extern 변수 및 함수
소스코드	낮음	C++	매직 넘버 사용
소스코드	낮음	C++	금지된 사용자 정의 타입 사용
소스코드	낮음	C++	누락된 void 키워드
소스코드	낮음	C++	public 클래스 멤버 변수
소스코드	낮음	C++	정의되지 않은 생성자
소스코드	낮음	C++	정의되지 않은 소멸자
소스코드	낮음	C++	정의되지 않은 복사 생성자
소스코드	낮음	C++	정의되지 않은 할당 연산자
소스코드	보통	C++	부적절하게 오버로딩된 할당 연산자
소스코드	낮음	C++	비 멤버 비대칭 연산자
소스코드	높음	C++	누락된 배열 인덱스 검사
소스코드	낮음	C++	비 friend 대칭 연산자
소스코드	보통	C++	부적절한 메모리 재할당
소스코드	보통	C++	상수 값의 수정
소스코드	보통	C++	일치하지 않는 삼항 연산 내 타입
소스코드	높음	C++	불충분한 할당 메모리
소스코드	낮음	C++	#pragma 지시자 사용
소스코드	낮음	C++	주석 없는 빈 구문
소스코드	낮음	C++	표준 라이브러리 내 이름
소스코드	낮음	C++	friend 함수 및 클래스 사용

소스코드	보통	C++	부적절한 조건 비교
소스코드	보통	C++	누락된 더 큰 정수 타입으로의 형변환
소스코드	높음	C++	void 포인터 타입에서 비 void 포인터 타입으로 형변환
소스코드	낮음	C++	누락된 실패 시 문자열 초기화
소스코드	보통	C++	누락된 배열 길이 검사
소스코드	보통	C++	void 반환 타입 사용
소스코드	낮음	C++	효과가 없는 구문
소스코드	보통	C++	타입 정의 내에서 포인터 역참조 연산 사용
소스코드	낮음	C++	실수 변환에서 정밀성 손실
소스코드	보통	C++	누락된 디버그 모드 검사
소스코드	낮음	C++	인스턴스로 매크로 사용
소스코드	보통	C++	누락된 클래스 멤버 할당
소스코드	보통	C++	누락된 베이스 클래스 멤버 할당
소스코드	보통	C++	rename의 잘못된 사용
소스코드	낮음	C++	장치에 대해 연산 수행
소스코드	낮음	C++	잘못된 오류 타입
소스코드	낮음	C++	중복된 헤더 파일 이름
소스코드	높음	C++	부적절한 문자열 토큰 변환
소스코드	보통	C++	누락된 부동 소수점 오류에 대한 예외 처리
소스코드	보통	C++	필드에 포인터 연산 수행
소스코드	낮음	C++	부적절한 파일 열기 및 생성
소스코드	낮음	C++	반한된 포인터를 비 const 변수에 저장
소스코드	보통	C++	반복문 내에서의 변수 선언
소스코드	낮음	C++	컴파일러 최적화 중 코드 삭제
소스코드	낮음	C++	불확정 값에 대해 va_arg 함수 사용
소스코드	낮음	C++	extern 인라인 함수의 제약 조건 위반
소스코드	보통	C++	비 표준 문자
소스코드	보통	C++	사용하지 않는 매개 변수
소스코드	매우 낮음	C++	do-while 구문 사용
소스코드	낮음	C++	되돌아가는 goto 구문
소스코드	낮음	C++	블록 간 점프
소스코드	낮음	C++	구조체 내 가변 길이 배열 사용
소스코드	보통	C++	부동 소수점 값에 대한 == 연산
소스코드	매우 낮음	C++	읽기 어려운 주석
소스코드	매우 낮음	C++	다른 주석 길이
소스코드	매우 낮음	C++	주석 형식 위반

소스코드	매우 낮음	C++	코드 주위 주석
소스코드	낮음	C++	파라미터에 대해 사용자 정의 타입의 미사용
소스코드	낮음	C++	비 상수 값 반환
소스코드	보통	C++	반환 타입에 대해 사용자 정의 타입의 미사용
소스코드	보통	C++	문자 타입 사용
소스코드	보통	C++	중복된 헤더 파일 포함
소스코드	보통	C++	사용되지 않은 정적 변수
소스코드	낮음	C++	비트 필드 타입 사용
소스코드	보통	C++	실수 타입 사용
소스코드	보통	C++	extern 혹은 정적 지정자 사용
소스코드	보통	C++	금지된 한정자 사용
소스코드	매우 낮음	C++	너무 많은 함수
소스코드	보통	C++	부적절한 파일 입력 및 출력
소스코드	낮음	C++	선언된 함수의 정의 누락
소스코드	보통	C++	불린 타입에 대해 비트 연산 수행
소스코드	낮음	C++	한 줄에 작성된 여러 개의 다른 타입의 선언
소스코드	낮음	C++	금지된 헤더 파일 포함
소스코드	낮음	C++	타입 구조 노출
소스코드	보통	C++	널 배열 반환
소스코드	보통	C++	상위 범위에의 지역 주소 할당
소스코드	낮음	C++	잘못된 파일 포함 방법
소스코드	보통	C++	첫 번째 케이스 구문 전 실행 가능한 구문
소스코드	낮음	C++	포함 시 절대 경로 사용
소스코드	낮음	C++	누락된 헤더 파일
소스코드	매우 낮음	C++	부적절한 구조체 정렬
소스코드	매우 낮음	C++	패딩된 구조체
소스코드	낮음	C++	너무 큰 구조체
소스코드	보통	C++	도달할 수 없는 케이스 구문
소스코드	보통	C++	일치하지 않는 가변 인자 수
소스코드	보통	C++	한정자 없이 가상 함수 호출
소스코드	보통	C++	생성자 및 소멸자 내에서 동적 타입 사용
소스코드	보통	C++	소멸자 내에서 처리되지 않은 예외
소스코드	낮음	C++	사용되지 않은 예외 처리
소스코드	낮음	C++	나머지 연산 사용
소스코드	낮음	C++	파라미터 주소 반환
소스코드	보통	C++	가상 베이스 클래스 포인터에 대해 정적 형변환

소스코드	보통	C++	main 함수 내에서 처리되지 않은 예외
소스코드	보통	C++	부호가 있는 값에 대해 나눗셈
소스코드	보통	C++	누락된 반환 값 비교
소스코드	높음	C++	일치하지 않는 피연산자 타입
소스코드	낮음	C++	상충되는 저장소 클래스
소스코드	낮음	C++	읽기 전용 파라미터로 사용된 초기화 되지 않은 포인터
소스코드	보통	C++	일관성 없는 전역 변수 선언
소스코드	높음	C++	일치하지 않는 문자 타입
소스코드	높음	C++	일치하지 않는 재할당 타입
소스코드	보통	C++	케이스 구문에서 누락된 break 구문
소스코드	낮음	C++	다차원 배열 사용
소스코드	낮음	C++	블록 중간에 존재하는 선언
소스코드	낮음	C++	if 구문 조건 내에서의 함수 사용
소스코드	낮음	C++	== 및 != 연산자를 사용한 문자열 비교
소스코드	보통	C++	문자 타입을 정수 타입으로 할당
소스코드	보통	C++	10진수가 아닌 정수 상수
소스코드	높음	C++	무효화된 반복자
소스코드	보통	C++	지역 변수 주소 접근
소스코드	보통	C++	일치하지 않는 인자 수
소스코드	높음	C++	재사용된 토큰화 된 문자열
소스코드	보통	C++	심볼릭 링크 경쟁 조건
소스코드	낮음	C++	함수 포인터에 물리 주소 직접 할당
소스코드	높음	C++	비 동기 스레드 종료
소스코드	매우 높음	C++	비교 대신 할당 수행
소스코드	매우 높음	C++	할당 대신 비교 수행
소스코드	매우 낮음	C++	비 가상 순수 함수
소스코드	매우 낮음	C++	순수 가상 함수의 잘못된 초기화
소스코드	보통	C++	일치하지 않는 함수 정의 및 선언 타입
소스코드	보통	C++	부적절한 memcpy 인자의 값
소스코드	매우 낮음	C++	쓸모없는 이름
소스코드	보통	C++	누락된 반환 타입
소스코드	낮음	C++	빈 while 구문
소스코드	낮음	C++	16진수 이스케이프 시퀀스 사용
소스코드	보통	C++	매크로 정의 시 후행하는 세미콜론
소스코드	높음	C++	부적절한 매크로 사용
소스코드	높음	C++	안전한 함수 교체

소스코드	높음	C++	괄호로 묶이지 않은 여러 개의 구문을 포함하는 매크로
소스코드	보통	C++	중복된 매크로
소스코드	보통	C++	일치하지 않는 바이트 순서
소스코드	매우 높음	C++	이미 잠금된 자원 잠금
소스코드	매우 높음	C++	이미 잠금 해제된 자원 잠금 해제
소스코드	높음	C++	이미 잠금된 자원 trylock
소스코드	보통	C++	부적절한 파일 지시자의 값
소스코드	높음	C++	너무 큰 스택 크기
소스코드	매우 낮음	C++	효과가 없는 함수 호출
소스코드	낮음	C++	복사 대입 연산자를 정의하지 않은 템플릿을 포함한 클래스
소스코드	낮음	C++	외부 부수 효과를 가지지 않은 함수 선언
소스코드	매우 낮음	C++	타입이 서로 다른 비트 연산자의 피연산자
소스코드	낮음	C++	다이아몬드 문제가 발생하는 비가상 상속 관계
소스코드	높음	C++	지역 변수 주소를 범위 밖 변수에 대입
소스코드	낮음	C++	메인 함수에서 모든 예외 핸들러 누락
소스코드	매우 낮음	C++	cstdio 헤더 사용
소스코드	매우 낮음	C++	cstring 헤더 사용
소스코드	매우 낮음	C++	ctime 헤더 사용
소스코드	보통	C++	널 예외 던짐
소스코드	낮음	C++	포인터 타입 예외 던짐
소스코드	매우 낮음	C++	빈 throw 구문
소스코드	낮음	C++	정수값으로 사용한 널 포인터
소스코드	낮음	C++	열거형 비트 필드 선언
소스코드	보통	C++	참조로 받은 인자 값의 반환
소스코드	낮음	C++	주소 연산자 오버로딩
소스코드	매우 낮음	C++	가상 상속 사용
소스코드	낮음	C++	같은 계층 간 다른 상속 사용
소스코드	매우 낮음	C++	한 번도 호출되지 않은 함수
소스코드	매우 낮음	C++	잘못 사용한 예외 던짐
소스코드	낮음	C++	허용되지 않은 타입으로 비트 필드 선언
소스코드	매우 낮음	C++	asm 선언 없이 어셈블리어 명령어 사용
소스코드	낮음	C++	가상 함수에서 사용하지 않은 매개 변수
소스코드	매우 낮음	C++	코드 영역에서 사용한 주석
소스코드	낮음	C++	생성자 및 소멸자에서 예외 던짐
소스코드	낮음	C++	복사 생성자에서 정적 필드값 변경
소스코드	낮음	C++	복사 생성자를 정의하지 않은 템플릿을 포함한 클래스

소스코드	매우 낮음	C++	인스턴스화되지 않은 템플릿
소스코드	낮음	C++	명시적 특수화된 함수의 모호한 호출
소스코드	매우 낮음	C++	템플릿 특수화된 함수의 모호한 호출
소스코드	낮음	C++	throw 구문의 수식에서 예외 발생
소스코드	매우 낮음	C++	서로 다른 예외 명세
소스코드	매우 낮음	C++	네임스페이스에서 비멤버 템플릿 함수 선언
소스코드	낮음	C++	모호한 템플릿 부모 클래스 메소드 호출
소스코드	매우 낮음	C++	const char 포인터 외의 자료형에 문자열 할당
소스코드	보통	C++	조건 혹은 증감절에 의해 루프 제어 변수값 변동
소스코드	보통	C++	Boolean 타입이 아닌 루프 제어 변수
소스코드	낮음	C++	비정형적인 for 구문에서 continue 구문 사용
소스코드	낮음	C++	정적 스토리지 클래스를 가지지 않은 내부 링크 함수
소스코드	보통	C++	명시적인 예외 처리에 catch 문 미처리
소스코드	낮음	C++	non-const 핸들러를 클래스 데이터로 반환 금지
소스코드	매우 낮음	C++	재선언 시 토큰 불일치
소스코드	매우 낮음	C++	unsigned int 자료형에 접미사 U 사용
소스코드	보통	C++	case문 종료 누락
소스코드	보통	C++	C 스타일 메모리 관련 함수 사용
소스코드	보통	C++	무한 반복
소스코드	보통	C++	소켓의 잘못된 사용
소스코드	매우 낮음	C++	금지된 함수 사용
소스코드	보통	C++	반환 값 대신 함수 포인터 비교
소스코드	보통	C++	포인터 연산을 위한 포인터 크기 조정 금지
소스코드	매우 낮음	C++	헤더 파일 내 함수 선언
소스코드	보통	C++	모호한 템플릿 함수 호출
소스코드	보통	C++	const 한정자를 제거하는 캐스팅 사용
소스코드	보통	C++	va_start 매크로 제약사항 위반
소스코드	높음	C++	tss_create() 함수 사용 후 메모리 해제 확인
소스코드	보통	C++	표현식에서 원자 변수 두 번 참조
소스코드	낮음	C++	atomic_compare_exchange_weak() 함수 실패
소스코드	보통	C++	스토리지 클래스 충돌
소스코드	보통	C++	사용되지 않은 표준라이브러리 반환 값
소스코드	낮음	C++	정확하지 않은 가변인자 타입
소스코드	보통	C++	float 타입에 대해 비교 연산 수행
소스코드	보통	C++	부적절한 정수 정밀도
소스코드	보통	C++	다른 스레드의 뮤텍스 조작

소스코드	보통	C++	지역 변수를 공유하는 스레드
소스코드	낮음	C++	열거형 타입 명명 규칙 위반
소스코드	낮음	C++	클래스 정의 헤더에 인클루드 가드 누락
소스코드	매우 낮음	C++	연산자 사이 공백 누락
소스코드	낮음	C++	switch 구문에서 constexpr 사용
소스코드	보통	C++	수학 함수 범위 오류
소스코드	보통	C++	잠겨 있는 뮤텍스 파괴
소스코드	보통	C++	저장 기간이 끝난 공유 객체
소스코드	보통	C++	데드락 발생
소스코드	보통	C++	잘못된 조건 변수 사용
소스코드	높음	C++	이미 소유한 포인터
소스코드	보통	C++	참조 타입에 volatile 한정자 사용
소스코드	낮음	C++	noreturn 으로 선언한 함수에서 반환
소스코드	보통	C++	배열을 다형성으로 취급 금지
소스코드	높음	C++	반복자에서 1보다 큰 값으로 증감 금지
소스코드	높음	C++	뮤텍스 unlock() 함수 예외 처리 없이 호출 금지
소스코드	보통	C++	최대 크기 버퍼 사용
소스코드	낮음	C++	bool 형식에 감소 연산자 사용
소스코드	매우 낮음	C++	누락된 지역 변수의 초기화
소스코드	보통	C++	헤더 파일에서 누락된 #define
소스코드	보통	C++	문자열 비교로 인한 도달할 수 없는 코드
소스코드	높음	C++	음수에 대해 나머지 연산
소스코드	높음	C++	부호가 없는 타입에 음수 할당
소스코드	보통	C++	0으로 나누기
소스코드	보통	C++	누락된 0으로 나누기 검사
소스코드	높음	C++	지역 변수 주소 반환
소스코드	보통	C++	스택 메모리 해제
소스코드	높음	C++	누락된 반환 구문
소스코드	낮음	C++	겹치는 메모리 영역
소스코드	높음	C++	비트 폭을 초과하는 시프트
소스코드	높음	C++	음수 값에 대해 시프트
소스코드	보통	C++	부적절한 길이 인자의 값
소스코드	낮음	C++	쓸모없는 조건
소스코드	낮음	C++	도달할 수 없는 코드
소스코드	낮음	C++	사용되지 않은 값
소스코드	높음	C++	위험한 함수 포인터 형변환

소스코드	보통	C++	일반 주석 형식 위반
소스코드	매우 낮음	C++	레이블 구문 사용
소스코드	보통	C++	반복문 내에서 금지된 함수 사용
소스코드	보통	C#	누락된 while 구문 중괄호
소스코드	보통	C#	누락된 for 구문 중괄호
소스코드	낮음	C#	문자열의 인스턴스화
소스코드	보통	C#	누락된 switch 구문 내 default 케이스
소스코드	보통	C#	케이스 구문에서 누락된 break 구문
소스코드	보통	C#	반복문 마지막에 점프문 사용
소스코드	보통	C#	변수에 대해 Equals 사용
소스코드	보통	C#	+ 연산에 사용된 빈 문자열
소스코드	높음	C#	하드코드된 IP
소스코드	보통	C#	여러 변수에 할당
소스코드	매우 낮음	C#	반복문 내에서의 인스턴스 생성
소스코드	보통	C#	너무 깊이 중첩된 if 구문
소스코드	보통	C#	NullReferenceException 예외 생성
소스코드	보통	C#	부동 소수점 값에 대한 == 연산
소스코드	보통	C#	+= 연산자로 더해진 문자열
소스코드	보통	C#	컬렉션 크기를 0과 비교
소스코드	보통	C#	사용되지 않은 지역 변수
소스코드	보통	C#	사용되지 않은 파라미터
소스코드	보통	C#	사용되지 않은 private 필드
소스코드	보통	C#	새로운 예외 생성
소스코드	보통	C#	pulse 되지 않은 객체
소스코드	낮음	C#	비교 대신 할당 수행
소스코드	보통	C#	빈 while 구문
소스코드	보통	C#	sealed 클래스에 잘못된 접근 한정자 사용
소스코드	보통	C#	배열에 대해 toString 사용
소스코드	매우 낮음	Java	안드로이드 Activity에서 부모 클래스 메소드 호출 누락
소스코드	매우 낮음	Java	onMeasure 메소드 오버라이딩 중 필수 호출 누락
소스코드	낮음	Java	특정 변수에 리터럴 직접 대입
소스코드	높음	Java	컬렉션 내 객체에 대한 위험한 다운캐스팅
소스코드	보통	Java	의심스러운 메소드 이름
소스코드	높음	Java	equals 및 == 연산자의 잘못된 사용
소스코드	높음	Java	부정확한 오버라이딩
소스코드	높음	Java	비정적 메소드 내에서의 정적 필드 변경

소스코드	보통	Java	notify 사용
소스코드	낮음	Java	제어 없이 동기화 블록 내에서의 wait 사용
소스코드	낮음	Java	동기화 블록 내에서의 notify 사용
소스코드	매우 낮음	Java	지정된 예외 처리 방식 위반
소스코드	낮음	Java	제어 블록 밖에서의 sleep 혹은 yield 사용
소스코드	보통	Java	부동 소수점 값에 대한 == 연산
소스코드	낮음	Java	잘못된 컬렉션 내 객체 제거
소스코드	낮음	Java	외부 클래스의 private 멤버에 접근
소스코드	보통	Java	해제되지 않은 잠금
소스코드	보통	Java	notify 되지 않은 객체
소스코드	낮음	Java	불필요한 반복자 변수 형변환
소스코드	낮음	Java	함수 호출 시 발생하는 암시적 형변환
소스코드	낮음	Java	대입 시 발생하는 암시적 형변환
소스코드	낮음	Java	반환 시 발생하는 암시적 형변환
소스코드	낮음	Java	중첩된 try 구문 사용
소스코드	낮음	Java	외부 클래스의 private 필드 반환
소스코드	낮음	Java	생성자를 사용한 문자열 변환
소스코드	매우 낮음	Java	Runtime.exec 사용
소스코드	매우 낮음	Java	종료 시 finalizer 사용
소스코드	보통	Java	원시 타입 사용
소스코드	매우 낮음	Java	더 이상 사용되지 않는 API 사용
소스코드	매우 낮음	Java	반복문 내에서의 동기화 메소드 사용
소스코드	매우 낮음	Java	반복문 내에서 일정한 색인으로 배열 접근
소스코드	매우 낮음	Java	레이블을 사용한 break 구문
소스코드	매우 낮음	Java	volatile 사용
소스코드	매우 낮음	Java	short 타입 사용
소스코드	매우 낮음	Java	public 필드 사용
소스코드	낮음	Java	사용되지 않은 import 구문
소스코드	매우 낮음	Java	메소드 내의 public 필드 사용
소스코드	보통	Java	여러 변수에 할당
소스코드	매우 높음	Java	비교 대신 할당 수행
소스코드	낮음	JS/TS	잘못된 피연산자 타입
소스코드	낮음	JS/TS	with 구문 사용
소스코드	매우 낮음	JS/TS	부적절한 동등 비교 연산자
소스코드	낮음	JS/TS	피연산자에 사용된 할당
소스코드	낮음	JS/TS	비트 연산 수행

소스코드	보통	JS/TS	수정된 네이티브 객체
소스코드	낮음	JS/TS	for 구문 내에서의 부적절한 변수 선언
소스코드	매우 낮음	JS/TS	도달할 수 없는 코드
소스코드	높음	Obj-C	심볼릭 링크 경쟁 조건
소스코드	높음	Obj-C	Apple API의 잘못된 사용
소스코드	높음	Obj-C	@synchronized 지시자에 nil 전달
소스코드	매우 높음	Obj-C	비 인스턴스 객체 해제
소스코드	보통	Obj-C	CFNumberCreate의 잘못된 사용
소스코드	높음	Obj-C	반복문 인덱스로 실수 타입 사용
소스코드	보통	Obj-C	이항 연산에서 초기화되지 않은 값 사용
소스코드	보통	Obj-C	배열에서 초기화되지 않은 값 사용
소스코드	보통	Obj-C	초기화되지 않은 값 대입
소스코드	보통	Obj-C	조건식에서 초기화되지 않은 값 사용
소스코드	보통	Obj-C	초기화되지 않은 값 반환
소스코드	보통	Obj-C	CFErrorRef의 잘못된 사용
소스코드	높음	Obj-C	malloc의 잘못된 사용
소스코드	높음	Obj-C	부적절한 malloc 인자의 값
소스코드	높음	Obj-C	스택 변수 주소 할당
소스코드	보통	Obj-C	잘못된 문자열 함수 인자
소스코드	보통	Obj-C	CFRetain, CFRelease 및 CFMakeCollectable의 잘못된 사용
소스코드	낮음	Obj-C	초기화되지 않은 self
소스코드	낮음	Obj-C	Secure Keychain API의 잘못된 사용
소스코드	낮음	Obj-C	사용되지 않은 변수
소스코드	낮음	Obj-C	함수 호출 및 메시징의 논리 오류
소스코드	보통	Obj-C	부적절한 길이 인자의 값
소스코드	낮음	Obj-C	IPv4 전용 API 사용
소스코드	낮음	Obj-C	하드코드된 IP
소스코드	보통	Obj-C	잠금 해제되지 않은 뮤텍스
소스코드	높음	Obj-C	버퍼 오버플로우
소스코드	낮음	Obj-C	할당 시 정밀성 손실
소스코드	낮음	Obj-C	누락된 pthread_mutex_lock 반환 값 검사
소스코드	낮음	Obj-C	부동 소수점 값에 대한 == 연산
소스코드	매우 낮음	Obj-C	금지된 함수 사용
소스코드	보통	Python	ConnectionError 예외 처리 누락
소스코드	보통	Python	BlockingIOError 예외 처리 누락

소스코드	보통	Python	FileExistError 예외 처리 누락
소스코드	매우 높음	SQL	SELECT 구문에서 누락된 WHERE 절
소스코드	매우 높음	SQL	존재하지 않는 테이블
소스코드	매우 높음	SQL	존재하지 않는 컬럼
소스코드	낮음	SQL	사용하지 않는 테이블
소스코드	낮음	SQL	사용하지 않는 컬럼
소스코드	높음	SQL	유효하지 않은 숫자 형식
소스코드	높음	SQL	컬럼 크기를 초과하는 입력
소스코드	매우 높음	SQL	잘못된 GROUP BY 절
소스코드	높음	SQL	널 값을 허용하지 않는 컬럼에 널 값 입력
소스코드	매우 높음	SQL	날짜 형식 오류
소스코드	매우 높음	SQL	구문 오류
소스코드	매우 높음	SQL	유효하지 않은 테이블 이름
소스코드	높음	SQL	PRIMARY KEY 컬럼에 널 값 입력
소스코드	매우 높음	SQL	SELECT 구문에 포함되지 않은 ORDER BY 절 컬럼
소스코드	매우 높음	SQL	INSERT 쿼리에서 일치하지 않는 컬럼 및 값의 개수
소스코드	매우 높음	SQL	DELETE 및 UPDATE 구문에서 누락된 WHERE 절
소스코드	매우 낮음	Swift	대입 연산의 좌측 피연산자 갱신
소스코드	매우 낮음	Swift	동일한 값을 반환하는 삼항 연산자
소스코드	매우 낮음	Swift	중복된 조건 검사
소스코드	매우 낮음	Swift	시스템 함수 호출
소스코드	매우 낮음	Swift	같은 피연산자를 가진 이항 연산
소스코드	매우 낮음	Swift	제어문 후 작성된 코드
소스코드	매우 낮음	Swift	조건문 속에 포함되지 않은 제어문
소스코드	매우 낮음	Swift	부동 소수점 값에 대한 관계 연산
소스코드	매우 낮음	Swift	강제로 해제한 옵셔널
소스코드	매우 낮음	Swift	연산자 우선순위 변경
소스코드	매우 낮음	Swift	암시적으로 해제한 옵셔널 타입
소스코드	낮음	Swift	지문 요청 사유 메시지 누락
소스코드	매우 낮음	기타	지정한 문자열 사용
소스코드	매우 낮음	VBS	금지된 함수 사용
소스코드	보통	VBS	누락된 빈 값 검사
소스코드	낮음	VBS	with 구문 사용
소스코드	보통	VBS	누락된 반환 구문
소스코드	보통	Java	catch 블록에서 반환 구문 사용
소스코드	높음	JS/TS	이진 타입을 사용하지 않는 트랜잭션

소스코드	낮음	Swift	암시적인 클래스 접근 제어
소스코드	높음	VB.Net	제한되지 않은 동작
소스코드	보통	VB.Net	헤더 검사 비활성화
소스코드	높음	VB.Net	뷰 상태 MAC 비활성화
소스코드	낮음	VB.Net	가장된 권한에 의한 실행
소스코드	보통	VB.Net	남은 디버그 코드
소스코드	낮음	VB.Net	HttpSessionState 속성으로 비직렬화 객체 저장
소스코드	높음	VB.Net	Required 속성이 없는 상위 모델
소스코드	보통	VB.Net	지속적인 권한 설정
소스코드	높음	VB.Net	신뢰할 수 없는 모델
소스코드	보통	ABAP	동적 절과 함께 SELECT 사용
소스코드	낮음	C	가변 길이 배열 사용
소스코드	낮음	C	예약어를 매크로 이름으로 사용
소스코드	보통	C	금지된 재귀 호출
소스코드	낮음	C	불필요한 타입 선언
소스코드	낮음	C	문자 함수에 signed 문자형 전달
소스코드	보통	C	잘못된 타입의 getchar 함수 반환값 사용
소스코드	보통	C	할당 실패로 인한 널 역참조
소스코드	보통	C	널 값 검사 전 역참조
소스코드	낮음	C++	삼중자 사용
소스코드	보통	C++	너무 많은 초기화 값
소스코드	높음	C++	형변환 시 const 자격 손실
소스코드	높음	C++	형변환 시 volatile 자격 손실
소스코드	낮음	C++	누락된 오류 감지
소스코드	보통	C++	assert 구문의 잘못된 사용
소스코드	낮음	C++	PTHREAD_MUTEX_NORMAL 타입 뮤텍스 잠금 사용
소스코드	높음	C++	vfork 함수 사용
소스코드	낮음	C++	pthread_kill 함수 사용
소스코드	보통	C++	시그널 함수 사용
소스코드	높음	C++	배열 타입 파라미터에 대해 sizeof 함수 사용
소스코드	높음	C++	부적절한 숫자 타입 간 변환
소스코드	높음	C++	부적절한 할당 메모리 크기
소스코드	보통	C++	임시 혹은 공개 디렉토리에 파일 출력
소스코드	높음	C++	system 함수 사용
소스코드	높음	C++	atexit 핸들러 상에서 종료
소스코드	높음	C++	시그널 핸들러 내에서의 위험한 함수 사용

소스코드	보통	C++	더 이상 사용되지 않는 API 사용
소스코드	보통	C++	스레드 조기 종료
소스코드	높음	C++	포인터 타입에 대해 sizeof 함수 사용
소스코드	보통	C++	지나치게 일반적인 예외 생성
소스코드	높음	C++	잘못된 권한 포기 순서
소스코드	높음	C++	안전하지 않은 권한 포기
소스코드	높음	C++	정수 언더플로우
소스코드	높음	C++	부적절한 정수 다운캐스팅
소스코드	높음	C++	해제된 자원 사용
소스코드	높음	C++	일치하지 않는 버퍼 사이즈
소스코드	보통	C++	너무 작은 경로 버퍼
소스코드	낮음	C++	const 한정자 없는 중요 public 변수
소스코드	보통	C++	잘못된 순서로 매크로 사용
소스코드	높음	C++	asctime의 잘못된 사용
소스코드	높음	C++	정수 오버플로우
소스코드	높음	C++	위험한 함수 사용
소스코드	보통	C++	안전하지 않은 멀티 바이트 문자열 함수 사용
소스코드	보통	C++	무한 재귀 호출
소스코드	보통	C++	포멧 스트링과 대상 인자 개수 불일치
소스코드	매우 낮음	C++	빈 분기문
소스코드	보통	C++	다중 스레드 환경에서 비트 필드 접근
소스코드	보통	C++	표준 네임스페이스 변경
소스코드	낮음	C++	atexit() 함수로 등록한 핸들러에서 예외 발생
소스코드	높음	C++	컨테이너 오버플로우
소스코드	높음	C++	동적 할당 버퍼 오버플로우
소스코드	낮음	C++	복사 연산자 내에서 원본 객체 수정
소스코드	보통	C++	적절하지 않은 클래스 객체를 대상으로 하는 C 표준 라이브러리 사용
소스코드	낮음	C++	const로 선언되지 않은 public 정적 멤버 변수
소스코드	보통	C++	메모리 누수
소스코드	높음	C++	자원 누수
소스코드	높음	C++	이미 해제된 메모리 해제
소스코드	높음	C++	해제된 메모리 사용
소스코드	높음	C++	초기화되지 않은 값
소스코드	보통	C++	널 역참조
소스코드	보통	C++	누락된 널 검사

소스코드	높음	C++	해제된 메모리 반환
소스코드	보통	C++	동적 할당된 메모리에 대한 delete 시도
소스코드	보통	C++	동적 할당된 메모리에 대한 delete[] 시도
소스코드	보통	C++	new[]로 할당된 메모리에 대한 delete 시도
소스코드	보통	C++	new[]로 할당된 메모리에 대한 free 시도
소스코드	보통	C++	new로 할당된 메모리에 대한 delete[] 시도
소스코드	보통	C++	new로 할당된 메모리에 대한 free 시도
소스코드	높음	C++	이미 해제된 자원 해제
소스코드	낮음	C++	지정된 라이브러리 함수 호출 누락
소스코드	보통	C++	지정된 매개변수를 사용하는 라이브러리 함수 호출 누락
소스코드	보통	C++	누락된 필수 인자를 사용한 필수 라이브러리 함수 호출
소스코드	보통	C++	누락된 필수 함수 호출
소스코드	보통	C++	malloc의 잘못된 사용
소스코드	보통	C++	프로세스 API의 잘못된 사용
소스코드	보통	C++	자원 API의 잘못된 사용
소스코드	보통	C++	신호 API의 잘못된 사용
소스코드	보통	C++	TP API의 잘못된 사용
소스코드	보통	C++	임시 파일 관련 함수 사용
소스코드	보통	C++	타이머 API의 잘못된 사용
소스코드	보통	C++	printf 함수 사용
소스코드	매우 낮음	C++	[] 연산자를 통한 배열 접근
소스코드	보통	C++	fopen 함수 호출 전 umask 함수 사용
소스코드	보통	C++	구조체 멤버에 여러번 접근
소스코드	보통	C#	0으로 나누기
소스코드	보통	C#	널 역참조
소스코드	보통	C#	널 반환 값 역참조
소스코드	보통	C#	누락된 널 값 검사
소스코드	보통	C#	Application.exit 메소드 사용
소스코드	보통	C#	IndexOf의 인자로 문자열 사용
소스코드	보통	C#	Monitor.Pulse 사용
소스코드	높음	C#	자원 누수
소스코드	보통	C#	표준 라이브러리의 널 반환 값 역참조
소스코드	낮음	C#	빈 finally 블록
소스코드	매우 낮음	C#	금지된 인터페이스 사용
소스코드	보통	C#	누락된 Serializable 속성
소스코드	보통	C#	Equals 메소드를 통한 널 검사

소스코드	보통	C#	하나만 정의된 GetHashCode 및 Equals
소스코드	낮음	C#	SqlClientPermission의 잘못된 사용
소스코드	보통	C#	하드코드된 파일 구분 문자
소스코드	보통	Java	안드로이드에서의 자바스크립트 허용
소스코드	보통	Java	안드로이드에서의 URI 권한 부여
소스코드	보통	Java	안드로이드에서의 인텐트 브로드캐스팅
소스코드	보통	Java	Android에서 JavaScript 사용
소스코드	높음	Java	안드로이드에서의 클래스 로딩 하이재킹
소스코드	보통	Java	절대 경로에 의한 안드로이드에서의 클래스 로딩 하이재킹
소스코드	보통	Java	안드로이드에서의 권한 검사 무시
소스코드	낮음	Java	AWT 및 Swing 사용 제한
소스코드	낮음	Java	Java IO 사용 제한
소스코드	보통	Java	안드로이드 보안 경고 알림
소스코드	보통	Java	escapeXml 설정으로 인한 크로스 사이트 스크립팅
소스코드	보통	Java	Android에서 브로드캐스터 권한 미설정
소스코드	보통	Java	Android 크로스 사이트 스크립트
소스코드	높음	Java	public static final 배열 사용
소스코드	높음	Java	오버로딩된 equals 메소드
소스코드	높음	Java	배열들 간의 equals 사용
소스코드	보통	Java	누락된 문자열 메소드 반환 값 검사
소스코드	높음	Java	연결의 직접 관리
소스코드	높음	Java	소켓의 직접 사용
소스코드	보통	Java	데이터베이스 연결 경쟁 조건
소스코드	보통	Java	자원 누수
소스코드	보통	Java	0으로 나누기
소스코드	보통	Java	널 역참조
소스코드	보통	Java	널 반환 값 역참조
소스코드	보통	Java	누락된 널 값 검사
소스코드	높음	Java	누락된 부모 클래스 메소드 호출
소스코드	높음	Java	compareTo 내에서 변할 수 있는 필드 사용
소스코드	높음	Java	hashCode 내에서 변할 수 있는 필드 사용
소스코드	보통	Java	표준 라이브러리의 널 반환 값 역참조
소스코드	높음	Java	부적절한 예외 처리
소스코드	높음	Java	잘못된 serialPersistentFields 필드 한정자
소스코드	보통	Java	오버라이딩 중 누락된 동기화
소스코드	높음	Java	중복 검사된 잠금

소스코드	높음	Java	EJB에서의 서버 소켓 사용
소스코드	보통	Java	무한 재귀 호출
소스코드	보통	Java	빈 catch 블록
소스코드	높음	Java	부정확한 hashCode 및 equals 오버라이딩
소스코드	높음	Java	스레드의 직접 사용
소스코드	높음	Java	System.exit 메소드 사용
소스코드	매우 낮음	Java	금지된 함수 사용
소스코드	낮음	Java	strictfp 한정자 없는 나눗셈
소스코드	보통	Java	케이스 구문에서 누락된 break 구문
소스코드	매우 높음	Java	전역적으로 접근 가능한 파일
소스코드	매우 낮음	JS/TS	금지된 함수 사용
소스코드	보통	JS/TS	동기화된 트랜잭션
소스코드	보통	JS/TS	빈 catch 블록
소스코드	보통	Kotlin	빈 catch 블록
소스코드	보통	Kotlin	널 반환 값 역참조
소스코드	매우 낮음	Kotlin	금지된 함수 사용
소스코드	높음	Obj-C	system 함수 사용
소스코드	높음	Obj-C	안전하지 않은 권한 포기
소스코드	매우 높음	Obj-C	널 역참조
소스코드	매우 높음	Obj-C	중복 형식화된 문자열
소스코드	보통	Obj-C	CFArrayGetValueAtIndex에서의 버퍼 오버플로우
소스코드	매우 높음	Obj-C	0으로 나누기
소스코드	높음	Obj-C	부적절한 가변 길이 배열 크기
소스코드	보통	Obj-C	mktemp의 잘못된 사용
소스코드	보통	Obj-C	vfork의 잘못된 사용
소스코드	보통	Obj-C	CFArrayCreate의 잘못된 사용
소스코드	보통	Obj-C	getpw의 잘못된 사용
소스코드	보통	Obj-C	gets의 잘못된 사용
소스코드	보통	Obj-C	random의 잘못된 사용
소스코드	보통	Obj-C	strcpy의 잘못된 사용
소스코드	보통	Obj-C	Unix API의 잘못된 사용
소스코드	보통	Obj-C	무한 재귀 호출
소스코드	높음	Obj-C	안전하지 않은 권한 제어
소스코드	높음	Obj-C	안전하지 않은 권한 초기화
소스코드	높음	Obj-C	익명 LDAP 바인딩 사용
소스코드	매우 높음	Obj-C	해제된 메모리 사용

소스코드	높음	Obj-C	위험한 함수 사용
소스코드	매우 높음	Obj-C	메모리 누수
소스코드	보통	PHP	무한 재귀 호출
소스코드	보통	PHP	빈 catch 블록
소스코드	보통	PHP	지나치게 일반적인 예외 처리
소스코드	보통	PHP	널 역참조
소스코드	높음	PHP	자원 누수
소스코드	높음	PHP	해제된 자원 사용
소스코드	매우 낮음	PHP	금지된 함수 사용
소스코드	보통	Python	하드코딩된 경로 구분자
소스코드	매우 낮음	Python	금지된 함수 사용
소스코드	매우 높음	Rust	부적절한 cargo 설정
소스코드	매우 높음	Rust	unsafe 사용 금지
소스코드	높음	Rust	잠재적 오버플로우 래핑
소스코드	매우 낮음	Rust	try! 사용 금지
소스코드	매우 높음	Rust	panic 유발 함수 금지
소스코드	보통	Rust	부적절한 배열 접근
소스코드	매우 높음	Rust	0으로 나누기
소스코드	높음	Rust	forget 사용 금지
소스코드	높음	Rust	메모리 누수 방지
소스코드	보통	Rust	부적절한 ManuallyDrop 해제
소스코드	매우 높음	Rust	부적절한 포인터 반환
소스코드	낮음	Rust	초기화되지 않은 메모리 사용
소스코드	보통	Rust	호환되지 않은 타입 사용 금지
소스코드	보통	Swift	HTTP 프로토콜을 통한 전송
소스코드	매우 낮음	Swift	금지된 함수 사용
소스코드	보통	Swift	무한 재귀 호출
소스코드	높음	기타	활성화된 Android 매니페스트 내의 debuggable 옵션
소스코드	보통	기타	활성화 된 안드로이드 매니페스트 내의 sharedUserId 옵션
소스코드	보통	기타	활성화 된 안드로이드 매니페스트 내의 exported 옵션
소스코드	높음	기타	소스 코드 내 계좌 정보
소스코드	높음	기타	소스 코드 내 신용카드 정보
소스코드	높음	기타	소스 코드 내 이메일 주소
소스코드	높음	기타	소스 코드 내 외국인 등록번호
소스코드	높음	기타	소스 코드 내 IP 정보
소스코드	높음	기타	소스 코드 내 주민등록번호

소스코드	높음	기타	소스 코드 내 여권 번호
소스코드	높음	기타	소스 코드 내 전화 번호
소스코드	높음	기타	소스 코드 내 운전면허 정보
소스코드	매우 낮음	VB.Net	금지된 함수 사용
소스코드	매우 낮음	VB.Net	빈 분기문
소스코드	보통	ASP	크로스 사이트 스크립팅
소스코드	높음	ASP	SQL 삽입
소스코드	높음	ASP	명령어 삽입
소스코드	높음	ASP	LDAP 삽입
소스코드	높음	ASP	경로 조작
소스코드	매우 높음	ASP	XQuery 삽입
소스코드	보통	ASP	HTTP 응답 분할
소스코드	높음	ASP	XPath 삽입
소스코드	높음	ASP	자원 삽입
소스코드	보통	ASP	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	보통	ASP	민감한 정보의 일반 텍스트 전송
소스코드	보통	ASP	에러 메시지를 통한 크로스 사이트 스크립팅
소스코드	높음	ASP	DNS lookup에 의존한 보안 결정
소스코드	높음	ASP	취약한 비밀번호 요구 조건
소스코드	보통	ASP	영속적인 쿠키
소스코드	보통	ASP	시스템 정보 노출
소스코드	보통	ASP	신뢰할 수 없는 정규식
소스코드	높음	ASP	제한되지 않은 파일 업로드
소스코드	보통	ASP	암호화 없이 비밀번호 저장
소스코드	보통	C	누락된 반환 값 검사
소스코드	높음	C++	안전하지 않은 파일 식별
소스코드	매우 높음	C++	파일 생성 시 잘못된 권한 할당
소스코드	높음	C++	중복된 환경 변수 이름
소스코드	낮음	C++	환경 변수 포인터에 의존
소스코드	낮음	C++	In-band 오류 표시자 사용
소스코드	보통	C++	누락된 심볼릭 링크 검사
소스코드	낮음	C++	putenv의 인자로 비 정적 변수 사용
소스코드	낮음	C++	파일 지시자 경쟁 조건
소스코드	낮음	C++	시그널 핸들러의 잘못된 사용
소스코드	낮음	C++	중단 가능한 시그널 핸들러 내에서 signal 함수 사용
소스코드	낮음	C++	라이브러리 경쟁 조건

소스코드	낮음	C++	구조체 패딩으로 인한 정보 누출
소스코드	높음	C++	누락된 문자열 길이 제한
소스코드	낮음	C++	디스크에 민감한 정보 저장
소스코드	보통	C++	누락된 입력 길이 검증
소스코드	보통	C++	윈도우 함수 사용 권한 위반
소스코드	높음	C++	취약한 해쉬
소스코드	보통	C++	부적절한 암호화
소스코드	높음	C++	부적절한 난수 생성
소스코드	보통	C++	남은 디버그 코드
소스코드	보통	C++	동일 핸들러에 여러 개의 시그널 할당
소스코드	보통	C++	OAEP 없는 RSA 사용
소스코드	보통	C++	힙 검사
소스코드	보통	C++	지나치게 일반적인 예외 처리
소스코드	보통	C++	switch 구문 경쟁 조건
소스코드	보통	C++	부적절한 포인터 확장
소스코드	보통	C++	불충분한 암호화 키 지수
소스코드	높음	C++	잘못된 조건으로 인한 동적 할당 버퍼 오버플로우
소스코드	높음	C++	동적 할당 버퍼 언더플로우
소스코드	높음	C++	잘못된 조건으로 인한 동적 할당 버퍼 언더플로우
소스코드	보통	C++	파일 이름 TOCTOU 경쟁 조건
소스코드	낮음	C++	public 메소드에 의해 반환된 private 컬렉션
소스코드	낮음	C++	외부 데이터를 private 컬렉션에 저장
소스코드	보통	C++	잘못된 umask 인자
소스코드	낮음	C++	외부 데이터를 private 필드에 저장
소스코드	낮음	C++	중요 public 변수
소스코드	보통	C++	정렬된 메모리의 재할당
소스코드	높음	C++	시스템 함수 호출 명령어 삽입
소스코드	높음	C++	누락된 널 종료 문자
소스코드	높음	C++	제한 없는 입력의 잘못된 사용
소스코드	높음	C++	함수의 버퍼 오버플로우
소스코드	높음	C++	잘못된 조건으로 인한 함수의 버퍼 오버플로우
소스코드	높음	C++	함수의 필드 버퍼 오버플로우
소스코드	높음	C++	잘못된 조건으로 인한 함수의 필드 버퍼 오버플로우
소스코드	높음	C++	함수의 동적 할당 버퍼 오버플로우
소스코드	높음	C++	잘못된 조건으로 인한 함수의 동적 할당 버퍼 오버플로우
소스코드	높음	C++	함수의 필드 버퍼 언더플로우

소스코드	높음	C++	잘못된 조건으로 인한 함수의 필드 버퍼 언더플로우
소스코드	보통	C++	누락된 작업 디렉토리 변경
소스코드	높음	C++	함수의 버퍼 언더플로우
소스코드	높음	C++	잘못된 조건으로 인한 함수의 버퍼 언더플로우
소스코드	높음	C++	함수의 동적 할당 버퍼 언더플로우
소스코드	높음	C++	잘못된 조건으로 인한 함수의 동적 할당 버퍼 언더플로우
소스코드	보통	C++	DNS lookup에 의존한 보안 결정
소스코드	보통	C++	멀티스레드 프로그램 내에서 getlogin 함수 사용
소스코드	높음	C++	하드코드된 비밀번호
소스코드	높음	C++	하드코드된 사용자 이름
소스코드	보통	C++	취약한 암호화 알고리즘
소스코드	보통	C++	안전하지 않은 권한 제어
소스코드	보통	C++	안전하지 않은 권한 초기화
소스코드	높음	C++	불충분한 암호화 키 길이
소스코드	보통	C++	부적절한 RSA 패딩
소스코드	높음	C++	하드코드된 솔트
소스코드	높음	C++	취약한 비밀번호 암호화 알고리즘
소스코드	보통	C++	동일 포트에 여러 개의 바인딩
소스코드	매우 높음	C++	중요한 리소스에 대한 잘못된 권한 할당
소스코드	높음	C++	SQL 삽입
소스코드	높음	C++	경로 조작
소스코드	높음	C++	명령어 삽입
소스코드	높음	C++	LDAP 삽입
소스코드	높음	C++	자원 삽입
소스코드	매우 높음	C++	시스템 혹은 구성 설정의 외부 제어
소스코드	높음	C++	안전하지 않은 직접 객체 참조
소스코드	높음	C++	주석문 안의 비밀번호
소스코드	높음	C++	부적절한 인증
소스코드	보통	C++	비밀번호의 일반 텍스트 저장
소스코드	높음	C++	하드코드된 암호화 키
소스코드	보통	C++	오염된 값의 위험한 사용
소스코드	보통	C++	빈 catch 블록
소스코드	높음	C++	제한되지 않은 파일 업로드
소스코드	보통	C++	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	매우 높음	C++	XQuery 삽입
소스코드	높음	C++	XPath 삽입

소스코드	높음	C++	일치하지 않는 자원 해제 방법
소스코드	보통	C++	영속적인 쿠키
소스코드	높음	C++	부적절한 XML 외부 엔티티 참조
소스코드	보통	C++	취약한 서버 인증서 검증
소스코드	높음	C++	반환 주소 수정에 의한 코드 삽입
소스코드	높음	C++	누락된 로그인 제어
소스코드	낮음	C++	누락된 인증
소스코드	높음	C++	누락된 비밀번호 복구 제어
소스코드	높음	C++	주요 보안 정보 및 차량 정보의 평문 전송
소스코드	높음	C++	낮은 보안 강도의 암호화 알고리즘 사용
소스코드	높음	C++	MAC 생성 시 누락된 메시지 식별 값
소스코드	낮음	C++	문법적으로 애매한 선언
소스코드	낮음	C++	예외 안정성을 보장하지 않는 코드
소스코드	보통	C++	누락된 예외 처리
소스코드	매우 높음	C++	이동한 객체의 재사용
소스코드	보통	C++	적절하지 않은 객체를 대상으로 하는 메모리 지정 new 연산
소스코드	보통	C++	정적 범위의 객체 선언에서 발생하는 처리되지 않는 예외
소스코드	매우 높음	C++	존재하지 않는 멤버에 접근하는 멤버-포인터 연산자
소스코드	보통	C++	범위 밖에 있는 열거 값 캐스팅
소스코드	높음	C++	버퍼 오버플로우
소스코드	높음	C++	잘못된 조건으로 인한 버퍼 오버플로우
소스코드	높음	C++	잘못된 조건으로 인한 필드 버퍼 오버플로우
소스코드	높음	C++	필드 버퍼 오버플로우
소스코드	높음	C++	범위 지정 복사로 인한 버퍼 오버플로우
소스코드	보통	C++	크로스 사이트 스크립팅
소스코드	높음	C++	취약한 비밀번호 요구 조건
소스코드	높음	C++	솔트 없는 해쉬 사용
소스코드	높음	C++	누락된 로그인 횟수 제한
소스코드	보통	C++	TOCTOU 경쟁 조건
소스코드	보통	C++	누락된 반환 값 검사
소스코드	높음	C++	무결성 검사 없는 코드 다운로드
소스코드	높음	C++	신뢰할 수 없는 입력에 의존한 보안 결정
소스코드	높음	C++	잘못된 조건으로 인한 필드 버퍼 언더플로우
소스코드	높음	C++	필드 버퍼 언더플로우
소스코드	높음	C++	버퍼 언더플로우
소스코드	매우 낮음	C++	부적절한 순차 메모리 할당

소스코드	매우 낮음	C++	싱글톤 패턴에서 동기화 없는 멀티 스레드 사용
소스코드	높음	C++	잘못된 조건으로 인한 버퍼 언더플로우
소스코드	높음	C++	포맷 스트링 삽입
소스코드	높음	C++	시스템 정보 노출
소스코드	높음	C#	하드코드된 사용자 이름과 비밀번호
소스코드	보통	C#	파일 이름 및 경로에 사용된 비 아스키 문자
소스코드	높음	C#	정수 오버플로우
소스코드	높음	C#	누락된 XML 검증
소스코드	높음	C#	솔트 없는 해쉬 사용
소스코드	높음	C#	private 배열에 저장된 public 데이터
소스코드	높음	C#	public 메소드에 의해 반환된 private 컬렉션
소스코드	보통	C#	이름에 기반한 타입 검사
소스코드	높음	C#	부적절한 인증
소스코드	높음	C#	누락된 로그인 제어
소스코드	높음	C#	중요 기능에 대해 누락된 인증
소스코드	높음	C#	주석문 안의 비밀번호
소스코드	높음	C#	XSLT 삽입
소스코드	높음	C#	세션 간 데이터 누출
소스코드	높음	C#	하드코드된 암호화 키
소스코드	높음	C#	무결성 검사 없는 코드 다운로드
소스코드	보통	C#	TOCTOU 경쟁 조건
소스코드	높음	C#	중요한 리소스에 대한 잘못된 권한 할당
소스코드	보통	C#	취약한 직렬화 생성 허용
소스코드	높음	C#	비어 있는 비밀번호
소스코드	높음	C#	하드코딩된 HMAC 비밀 키
소스코드	보통	C#	하드코딩된 초기화 벡터
소스코드	높음	C#	비어 있는 HMAC 비밀 키
소스코드	높음	C#	빈 비밀번호로 키 파생 함수 사용
소스코드	높음	C#	하드코딩된 비밀번호로 키 파생 함수 사용
소스코드	매우 높음	C#	안전하지 않은 DLL 사용
소스코드	보통	C#	누락된 헤더 검사
소스코드	높음	C#	EnableViewStateMac 속성 비활성화
소스코드	보통	C#	쉬운 솔트 값 사용
소스코드	높음	C#	반복 횟수가 적은 키 파생 함수 사용
소스코드	보통	C#	부적절한 RSA 패딩
소스코드	높음	C#	익명 LDAP 바인딩 사용

소스코드	높음	C#	HttpOnly가 아닌 쿠키
소스코드	높음	C#	광범위 도메인을 허용하는 쿠키 사용
소스코드	높음	C#	광범위 경로를 허용하는 쿠키 사용
소스코드	높음	C#	ECB 모드로 암호화 알고리즘 사용
소스코드	높음	C#	불충분한 전자 서명 키 길이
소스코드	높음	C#	UI가 그려지는 동안 민감한 정보 노출
소스코드	높음	C#	하드코딩된 대칭 키 알고리즘 비밀 키
소스코드	높음	C#	하드코딩된 비밀번호
소스코드	높음	C#	하드코딩된 비밀번호 비교
소스코드	높음	C#	예외 정보 노출
소스코드	높음	C#	동적 코드 조작 취약점
소스코드	높음	C#	취약한 XML 변환 보안
소스코드	높음	C#	서버측 요청 변조
소스코드	높음	C#	부적절한 서명
소스코드	보통	C#	취약한 서버 인증서 검증
소스코드	높음	C#	신뢰할 수 없는 데이터의 역직렬화
소스코드	높음	C#	크로스 사이트 요청 위조
소스코드	보통	C#	NullPointerException 예외 처리
소스코드	보통	C#	빈 catch 블록
소스코드	매우 높음	C#	HtmlInputHidden 사용
소스코드	보통	C#	시스템 정보 노출
소스코드	보통	C#	지나치게 일반적인 예외 처리
소스코드	낮음	C#	안전하지 않은 로깅
소스코드	높음	C#	취약한 해쉬
소스코드	높음	C#	취약한 암호화 알고리즘
소스코드	높음	C#	불충분한 암호화 키 길이
소스코드	높음	C#	부적절한 난수 생성
소스코드	높음	Dart	하드코드된 API 키
소스코드	높음	Dart	하드코드된 인증 정보
소스코드	높음	Dart	하드코드된 이메일 주소
소스코드	높음	Dart	하드코드된 IP 주소
소스코드	높음	Dart	취약한 해시
소스코드	높음	Dart	취약한 난수
소스코드	높음	Dart	취약한 암호화 알고리즘
소스코드	높음	Dart	충분하지 않은 RSA 암호화 키 길이
소스코드	높음	Dart	주석문 안의 API 키

소스코드	높음	Dart	주석문 안의 인증 정보
소스코드	높음	Dart	주석문 안의 이메일 주소
소스코드	높음	Dart	주석문 안의 IP 주소
소스코드	높음	Go	크로스 사이트 스크립트(HTML)
소스코드	높음	Go	크로스 사이트 스크립트(JS)
소스코드	높음	Go	크로스 사이트 스크립트(template)
소스코드	높음	Go	크로스 사이트 스크립트(etc.)
소스코드	높음	Go	크로스 사이트 스크립트(URL)
소스코드	높음	Go	SQL 삽입
소스코드	높음	Go	취약한 해쉬
소스코드	높음	Go	취약한 난수
소스코드	보통	Go	오류 메시지 정보 노출
소스코드	보통	Go	부적절한 쿠키 플래그 설정
소스코드	낮음	Go	취약한 패키지
소스코드	높음	Go	하드코드된 비밀번호
소스코드	높음	Go	부적절한 네트워크 설정
소스코드	높음	Go	부적절한 SSH 키 설정
소스코드	높음	Go	서버사이드 요청 위조
소스코드	높음	Go	정수형 변환 과정의 오버플로우
소스코드	보통	Go	Slowloris 공격
소스코드	높음	Go	부적절한 권한 설정
소스코드	높음	Go	경로 조작
소스코드	보통	Go	부적절한 TLS 설정
소스코드	높음	Go	충분하지 않은 RSA 암호화 키 길이
소스코드	높음	HTML	비밀번호 노출
소스코드	높음	HTML	다른 사이트의 스크립트 포함
소스코드	보통	HTML	비밀번호 필드에 적용된 자동완성
소스코드	보통	Java	제한 없이 컬렉션에 추가
소스코드	보통	Java	반복문 내에서의 컬렉션 항목 삭제
소스코드	매우 높음	Java	비 정적 직렬화 가능 내부 클래스
소스코드	높음	Java	부적절한 Externalizable 인터페이스의 구현
소스코드	매우 높음	Java	AccessController.doPrivileged 내 신뢰할 수 없는 입력
소스코드	보통	Java	클래스 초기화 시 순환
소스코드	보통	Java	클래스 간 초기화 시 순환
소스코드	보통	Java	비 final 반복문 인덱스
소스코드	보통	Java	assert 구문 내 수식

소스코드	보통	Java	형변환 시 정밀성 손실
소스코드	보통	Java	원시 타입 컬렉션 사용
소스코드	높음	Java	안전하지 않은 변할 수 있는 클래스
소스코드	높음	Java	민감한 클래스 상속
소스코드	높음	Java	로그 삽입
소스코드	보통	Java	안전하지 않은 ZipInputStream에서의 파일 추출
소스코드	보통	Java	파일 이름 및 경로에 사용된 비 아스키 문자
소스코드	보통	Java	신뢰할 수 없는 정규식
소스코드	매우 높음	Java	입력 검증에 사용된 비 문자
소스코드	보통	Java	누락된 실수 입력에 대한 NaN 검사
소스코드	높음	Java	부동 소수점 값의 문자열 표현과 비교
소스코드	보통	Java	위험한 다운캐스팅
소스코드	보통	Java	변할 수 있는 입력 및 내부 구성 요소의 직접 사용
소스코드	보통	Java	입출력 버퍼에 대해 차단된 외부 프로세스
소스코드	보통	Java	부적절한 write 인자의 정수 값
소스코드	보통	Java	신뢰할 수 없는 파일 링크 사용
소스코드	보통	Java	잘못된 직렬화 순서
소스코드	보통	Java	직렬화 도중 발생하는 메모리 및 자원 누수
소스코드	매우 높음	Java	기본 자동 서명 인증 사용
소스코드	보통	Java	readInt 메소드 반환 값의 부적절한 사용
소스코드	매우 높음	Java	생성자 내에서의 예외 생성
소스코드	보통	Java	부적절한 키 객체 간 비교
소스코드	보통	Java	부적절한 URL 비교
소스코드	보통	Java	예외에 대한 부적절한 복원
소스코드	보통	Java	같은 스레드 풀 사용
소스코드	보통	Java	재 초기화 되지 않은 ThreadLocal 타입 필드
소스코드	높음	Java	부분적으로 초기화 된 객체
소스코드	매우 높음	Java	읽기 시 반환 값의 부적절한 사용
소스코드	매우 높음	Java	불완전한 정적 초기화 블록
소스코드	높음	Java	시스템 환경 변수 사용
소스코드	매우 높음	Java	ReflectPermission 사용
소스코드	높음	Java	클래스 이름 비교
소스코드	높음	Java	assert 구문 내에서 파라미터 사용
소스코드	높음	Java	final 한정자 없는 중요 public 메소드
소스코드	높음	Java	메소드 접근성 증가
소스코드	매우 높음	Java	clone 메소드 내 오버라이딩 가능한 메소드 호출

소스코드	보통	Java	public 정적 메소드 오버라이딩
소스코드	보통	Java	finally 블록 내에서 처리되지 않은 예외
소스코드	높음	Java	비 volatile 스레드 간 공유 필드
소스코드	보통	Java	스레드 안전하지 않은 메소드 체인
소스코드	보통	Java	비 원자적 데이터 타입 사용
소스코드	높음	Java	잠금 인스턴스로 재사용 가능한 객체 사용
소스코드	높음	Java	동기화 시 클래스 객체 사용
소스코드	높음	Java	잠금 인스턴스로 상위 수준 동시성 객체 사용
소스코드	보통	Java	다른 컬렉션으로부터 초기화
소스코드	보통	Java	비 동기화 정적 필드
소스코드	매우 높음	Java	교착상태
소스코드	보통	Java	잠금 중 대기
소스코드	보통	Java	무한 대기 스레드
소스코드	보통	Java	갑작스러운 스레드 종료
소스코드	보통	Java	스레드 풀의 미사용
소스코드	보통	Java	동기화 블록 내 중단할 수 없는 스레드
소스코드	보통	Java	부적절한 스레드 풀 생성
소스코드	보통	Java	생성자 내에서의 스레드 실행
소스코드	보통	Java	초기화 전 게시
소스코드	보통	Java	정적 초기화 블록 내에서 스레드 생성
소스코드	높음	Java	File 인스턴스에 대한 delete의 잘못된 사용
소스코드	높음	Java	deleteOnExit의 잘못된 사용
소스코드	매우 높음	Java	버퍼 노출
소스코드	보통	Java	누락된 읽기 길이 인자
소스코드	보통	Java	빅 엔디안 전용 메소드 사용
소스코드	높음	Java	직렬화 된 클래스의 필드 노출
소스코드	높음	Java	부적절한 Serializable 인터페이스의 구현
소스코드	보통	Java	readObject 메소드 내 오버라이딩 가능한 메소드 호출
소스코드	매우 높음	Java	누락된 권한 검사
소스코드	매우 높음	Java	리플렉션 사용
소스코드	매우 높음	Java	getPermissions 메소드에 누락된 부모 클래스 메소드 호출
소스코드	매우 높음	Java	신뢰할 수 없는 클래스 import
소스코드	보통	Java	누락된 실수 입력에 대한 무한대 검사
소스코드	보통	Java	요청의 속성을 통한 크로스 사이트 스크립팅
소스코드	매우 높음	Java	신뢰 범위 위반
소스코드	높음	Java	예측 가능한 난수 생성

소스코드	높음	Java	박스형 프리미티브의 값 비교 금지
소스코드	보통	Java	공유되는 immutable 객체의 가시성 보장
소스코드	높음	Java	Socket 클래스 사용 금지
소스코드	높음	Java	동적 코드 조작 취약점
소스코드	높음	Java	확인되지 않은 SpEL 식
소스코드	높음	Java	확인되지 않은 OGNL 식
소스코드	높음	Java	취약한 XML 변환 보안
소스코드	보통	Java	취약한 서버 인증서 검증
소스코드	높음	Java	서버측 요청 변조
소스코드	높음	Java	안전하지 않은 Jackson 역직렬화
소스코드	높음	Java	안전하지 않은 XStream 역직렬화
소스코드	보통	Java	안드로이드에서 안전하지 않은 호스트 이름 확인
소스코드	보통	Java	유효성 검사 전 문자열 정규화 필요
소스코드	보통	Java	XML 삽입 방지
소스코드	보통	Java	동일 데이터에 비트 연산과 산술 연산의 동시 수행 금지
소스코드	낮음	Java	시프트(Shift) 연산자의 올바른 사용
소스코드	높음	Java	메서드 인수의 유효성 검사 필요
소스코드	보통	Java	불완전한 validate 메서드 정의
소스코드	낮음	Java	Validation 클래스 상속 금지
소스코드	보통	Java	동기화 기본 요소 사용 주의
소스코드	낮음	Java	단일 바이트 혹은 문자 스트림에 여러 개의 버퍼 래퍼 생성
소스코드	낮음	Java	Java 표준 라이브러리의 공용 식별자 재사용
소스코드	높음	Java	Java 런타임 오류 메시지를 통한 정보 노출
소스코드	보통	Java	로그에 민감한 정보 기록
소스코드	보통	Java	응답이 커밋된 서블릿의 작업 수행
소스코드	보통	Java	안드로이드에서 민감한 정보 로그 기록
소스코드	매우 높음	Java	컨텐트 프로바이더를 통한 파일 교환 전 정규화 누락
소스코드	보통	Java	유효성 검사 전 정규화 누락
소스코드	보통	Java	Geolocation API의 사용 전 사용자 권한 확인 누락
소스코드	보통	Java	XML 외부 엔티티 공격
소스코드	보통	Java	equals 내에서 변할 수 있는 필드 사용
소스코드	보통	Java	직렬화할 수 없는 객체의 세션 저장
소스코드	높음	Java	래핑되지 않은 네이티브 메소드
소스코드	매우 높음	Java	신뢰할 수 없는 출처에 대한 보안 검사
소스코드	높음	Java	신뢰할 수 없는 데이터의 역직렬화
소스코드	매우 낮음	Java	주석문 안에 포함된 주요정보

소스코드	높음	Java	안전하지 않은 JNI의 직접 사용
소스코드	높음	Java	하드코드된 사용자 이름과 비밀번호
소스코드	높음	Java	불충분한 세션 만료
소스코드	높음	Java	안전하지 않은 직접 객체 참조
소스코드	높음	Java	동적 클래스 로딩 사용
소스코드	높음	Java	취약한 암호화 알고리즘
소스코드	높음	Java	SQL 삽입
소스코드	낮음	Java	public 메소드에 의해 반환된 private 컬렉션
소스코드	보통	Java	누락된 J2EE 오류페이지 설정
소스코드	매우 낮음	Java	비활성화된 Struts 유효성 검사
소스코드	매우 낮음	Java	유효성 검사와 폼 필드 불일치
소스코드	높음	Java	불안전한 리플렉션 입력 검증
소스코드	보통	Java	공유 데이터에 대한 비동기화된 접근
소스코드	매우 낮음	Java	EJB 환경에서 클래스 로더 사용
소스코드	매우 낮음	Java	public으로 선언된 finalize()
소스코드	매우 낮음	Java	직렬화할 수 없는 객체 저장
소스코드	매우 낮음	Java	ActionSupport에서 public 필드 사용
소스코드	매우 낮음	Java	반복문 내 잘못된 오토박싱 및 언박싱
소스코드	매우 낮음	Java	잘못된 정수 비트 시프트 연산
소스코드	보통	Java	스레드 안전하지 않은 싱글톤의 사용
소스코드	높음	Java	취약한 비밀번호 요구 조건
소스코드	높음	Java	세션 간 데이터 누출
소스코드	보통	Java	남은 디버그 코드
소스코드	높음	Java	민감한 데이터를 포함하는 내부 클래스
소스코드	높음	Java	final 한정자 없는 중요 public 변수
소스코드	높음	Java	JDO API에서의 SQL 삽입
소스코드	높음	Java	J2EE Persistence API에서의 SQL 삽입
소스코드	높음	Java	Hibernate에서의 SQL 삽입
소스코드	높음	Java	명령어 삽입
소스코드	높음	Java	LDAP 삽입
소스코드	높음	Java	자원 삽입
소스코드	높음	Java	경로 조작
소스코드	보통	Java	HTTP 응답 분할
소스코드	매우 높음	Java	시스템 혹은 구성 설정의 외부 제어
소스코드	보통	Java	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	매우 높음	Java	XQuery 삽입

소스코드	높음	Java	XPath 삽입
소스코드	보통	Java	영속적인 쿠키
소스코드	보통	Java	크로스 사이트 스크립팅
소스코드	보통	Java	DOM 기반 크로스 사이트 스크립팅
소스코드	보통	Java	직접 동적 코드 평가
소스코드	높음	Java	DNS lookup에 의존한 보안 결정
소스코드	높음	Java	사이트 간 요청 위조
소스코드	높음	Java	메모리 덤프를 통한 비밀번호 노출
소스코드	높음	Java	불충분한 암호화 키 길이
소스코드	높음	Java	하드코드된 암호화 키
소스코드	보통	Java	부적절한 RSA 패딩
소스코드	높음	Java	하드코드된 솔트
소스코드	높음	Java	부적절한 난수 생성
소스코드	높음	Java	주소창을 통한 비밀번호 노출
소스코드	높음	Java	동일 포트에 여러 개의 바인딩
소스코드	높음	Java	안전하지 않은 쿠키
소스코드	높음	Java	중요한 리소스에 대한 잘못된 권한 할당
소스코드	높음	Java	정수 오버플로우
소스코드	높음	Java	중요 기능에 대해 누락된 인증
소스코드	높음	Java	부적절한 인증
소스코드	보통	Java	암호화 없이 비밀번호 저장
소스코드	높음	Java	주석문 안의 비밀번호
소스코드	높음	Java	솔트 없는 해쉬 사용
소스코드	높음	Java	무결성 검사 없는 코드 다운로드
소스코드	보통	Java	TOCTOU 경쟁 조건
소스코드	높음	Java	private 배열에 저장된 public 데이터
소스코드	높음	Java	신뢰할 수 없는 입력에 의존한 보안 결정
소스코드	보통	Java	시스템 정보 노출
소스코드	높음	Java	포맷 스트링 삽입
소스코드	보통	Java	에러 메시지를 통한 크로스 사이트 스크립팅
소스코드	보통	Java	민감한 정보의 일반 텍스트 전송
소스코드	높음	Java	서블릿 주석문 안의 비밀번호
소스코드	높음	Java	제한되지 않은 파일 업로드
소스코드	높음	Java	누락된 비밀번호 복구 제어
소스코드	높음	Java	누락된 로그인 제어
소스코드	낮음	Java	관리자 페이지 노출

소스코드	높음	Java	특권 블록으로 인한 정보 누출
소스코드	낮음	Java	위험한 메소드 노출
소스코드	낮음	Java	누락된 입력 검증
소스코드	낮음	Java	이메일 주소 교체
소스코드	낮음	Java	누락된 인증
소스코드	낮음	Java	안전하지 않은 비밀번호 복구
소스코드	보통	Java	안드로이드에서의 정보 누출
소스코드	매우 높음	Java	특권 블록 내의 신뢰할 수 없는 데이터
소스코드	보통	Java	SSI 삽입
소스코드	보통	Java	무한 반복
소스코드	보통	JS/TS	크로스 사이트 스크립트(ExpressJS)
소스코드	높음	JS/TS	운영체제 명령어 삽입
소스코드	보통	JS/TS	DOM 기반 크로스 사이트 스크립팅
소스코드	낮음	JS/TS	부적절한 프로퍼티 값
소스코드	보통	JS/TS	원격 코드 실행
소스코드	보통	JS/TS	로컬 저장소에서 세션 저장소로 정보 누출
소스코드	보통	JS/TS	세션 저장소에서 로컬 저장소로 정보 누출
소스코드	보통	JS/TS	크로스 도큐먼트 메시징
소스코드	높음	JS/TS	SQL 삽입
소스코드	높음	JS/TS	파일 생성 시 외부 데이터 사용
소스코드	높음	JS/TS	예측 가능한 데이터베이스 이름
소스코드	높음	JS/TS	하드코드된 비밀번호
소스코드	높음	JS/TS	빈 비밀번호
소스코드	높음	JS/TS	데이터베이스의 권한 제어 취약점
소스코드	높음	JS/TS	명령어 삽입
소스코드	보통	JS/TS	직접 동적 코드 평가
소스코드	보통	JS/TS	innerHTML에 할당
소스코드	보통	JS/TS	신뢰할 수 없는 사이트로 리다이렉션
소스코드	매우 높음	JS/TS	XHR 삽입
소스코드	보통	JS/TS	localStorage 사용
소스코드	보통	JS/TS	로그 삽입
소스코드	보통	JS/TS	크로스 사이트 스크립팅
소스코드	높음	JS/TS	안전하지 않은 MiUpdater
소스코드	보통	JS/TS	남은 디버그 코드
소스코드	높음	JS/TS	GET 방식으로 트랜잭션 수행
소스코드	매우 높음	JS/TS	트랜잭션 삽입

소스코드	보통	JS/TS	일반 텍스트를 사용하는 트랜잭션
소스코드	높음	JS/TS	민감한 정보의 Dataset
소스코드	보통	JS/TS	민감한 정보의 일반 텍스트 저장
소스코드	높음	JS/TS	레지스트리 누수
소스코드	보통	JS/TS	다이얼로그를 통한 신뢰할 수 없는 사이트로 리다이렉션
소스코드	높음	JS/TS	다이얼로그를 통한 명령어 삽입
소스코드	낮음	JS/TS	너무 긴 quicktabstestfont 속성 문자열
소스코드	보통	JS/TS	Azure에서의 권한 제어 취약점
소스코드	보통	JS/TS	HTTP 응답 분할
소스코드	보통	JS/TS	민감한 정보 노출
소스코드	높음	JS/TS	JSON 삽입
소스코드	보통	JS/TS	잘못된 로거 사용
소스코드	보통	JS/TS	투비소프트 플랫폼 로그 조작
소스코드	높음	JS/TS	투비소프트 플랫폼 경로 조작
소스코드	보통	JS/TS	시스템 정보 노출
소스코드	높음	JS/TS	동적 코드 조작 취약점
소스코드	높음	JS/TS	서버측 요청 변조
소스코드	높음	JS/TS	부적절한 서명
소스코드	보통	JS/TS	취약한 서버 인증서 검증
소스코드	높음	JS/TS	신뢰할 수 없는 데이터의 역직렬화
소스코드	높음	JS/TS	크로스 사이트 요청 위조
소스코드	보통	JS/TS	uncaughtException 사용
소스코드	높음	JS/TS	정규식 서비스 거부
소스코드	높음	JS/TS	Strict 모드
소스코드	높음	JS/TS	SQL 삽입(mysql)
소스코드	높음	JS/TS	SQL 삽입(postgreSQL)
소스코드	높음	JS/TS	SQL 삽입(noSQL)
소스코드	낮음	JS/TS	부적절한 쿠키 플래그 설정
소스코드	보통	JS/TS	HTTP 보안 헤더 설정
소스코드	보통	JS/TS	부적절한 HTTP 헤더
소스코드	높음	JS/TS	취약한 암호화 알고리즘 사용
소스코드	보통	JS/TS	크로스 사이트 스크립트(dangerouslySetInnerHTML)
소스코드	높음	JS/TS	취약한 기능 사용(findDomNode)
소스코드	보통	JS/TS	URL 변조
소스코드	보통	JS/TS	스크립트 삽입
소스코드	높음	JS/TS	SQL 삽입(ORM)

소스코드	보통	JS/TS	크로스 사이트 스크립트(VanillaJS)
소스코드	높음	JS/TS	부적절한 XML 외부 개체 참조
소스코드	보통	JS/TS	솔트 없이 일방향 해쉬 함수 사용
소스코드	높음	JS/TS	충분하지 않은 키 길이 사용
소스코드	높음	JS/TS	경로 조작
소스코드	높음	JS/TS	적절하지 않은 난수 값 사용
소스코드	보통	JS/TS	종료되지 않는 반복문 또는 재귀 함수
소스코드	보통	JS/TS	오류 메시지 정보노출
소스코드	보통	JS/TS	제거되지 않고 남은 디버그 코드
소스코드	높음	JS/TS	자원 삽입
소스코드	높음	JS/TS	코드 삽입
소스코드	높음	Kotlin	SQL 삽입
소스코드	높음	Kotlin	취약한 암호화 알고리즘
소스코드	높음	Kotlin	경로 조작
소스코드	높음	Kotlin	명령어 삽입
소스코드	높음	Kotlin	제한되지 않은 파일 업로드
소스코드	보통	Kotlin	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	높음	Kotlin	XPath 삽입
소스코드	높음	Kotlin	LDAP 삽입
소스코드	높음	Kotlin	포맷 스트링 삽입
소스코드	높음	Kotlin	중요 기능에 대한 인증 제어 누락
소스코드	높음	Kotlin	부적절한 인증
소스코드	높음	Kotlin	중요한 리소스에 대한 잘못된 권한 할당
소스코드	보통	Kotlin	민감한 정보를 평문으로 저장
소스코드	보통	Kotlin	민감한 정보의 일반 텍스트 전송
소스코드	높음	Kotlin	하드코딩된 비밀번호
소스코드	높음	Kotlin	불충분한 암호화 키 길이
소스코드	높음	Kotlin	부적절한 난수 생성
소스코드	높음	Kotlin	하드코딩된 암호화 키
소스코드	높음	Kotlin	취약한 비밀번호 요구 조건
소스코드	보통	Kotlin	영속적인 쿠키
소스코드	높음	Kotlin	주석문 안의 비밀번호
소스코드	높음	Kotlin	솔트 없는 해쉬 사용
소스코드	높음	Kotlin	누락된 로그인 제어
소스코드	보통	Kotlin	TOCTOU 경쟁 조건
소스코드	보통	Kotlin	무한 반복

소스코드	보통	Kotlin	지나치게 일반적인 예외 처리
소스코드	높음	Kotlin	세션 간 데이터 누출
소스코드	보통	Kotlin	남은 디버그 코드
소스코드	보통	Kotlin	시스템 정보 노출
소스코드	높음	Kotlin	public 메소드에 의해 반환된 private 컬렉션
소스코드	높음	Kotlin	private 배열에 저장된 public 데이터
소스코드	높음	Kotlin	DNS lookup에 의존한 보안 결정
소스코드	높음	Kotlin	동적 코드 조작 취약점
소스코드	높음	Kotlin	취약한 XML 변환 보안
소스코드	높음	Kotlin	서버측 요청 변조
소스코드	높음	Kotlin	부적절한 서명
소스코드	보통	Kotlin	취약한 서버 인증서 검증
소스코드	높음	Kotlin	신뢰할 수 없는 데이터의 역직렬화
소스코드	보통	Kotlin	안드로이드에서 안전하지 않은 호스트 이름 확인
소스코드	높음	Lua	하드코드된 인증 정보
소스코드	높음	Lua	하드코드된 이메일 주소
소스코드	높음	Lua	하드코드된 IP 주소
소스코드	높음	Lua	하드코드된 API 키
소스코드	높음	Lua	취약한 해시
소스코드	높음	Lua	취약한 난수
소스코드	높음	Lua	충분하지 않은 RSA 암호화 키 길이
소스코드	높음	Lua	주석문 안의 인증 정보
소스코드	높음	Lua	주석문 안의 이메일 주소
소스코드	높음	Lua	주석문 안의 IP 주소
소스코드	높음	Lua	주석문 안의 API 키
소스코드	보통	Lua	제거되지 않고 남은 디버그 코드
소스코드	높음	Lua	취약한 암호화 알고리즘
소스코드	보통	Obj-C	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	낮음	Obj-C	시그널 핸들러의 잘못된 사용
소스코드	보통	Obj-C	누락된 반환 값 검사
소스코드	보통	Obj-C	TOCTOU 경쟁 조건
소스코드	낮음	Obj-C	주석문 안의 비밀번호
소스코드	높음	Obj-C	문자열 복사로 인한 버퍼 오버플로우
소스코드	보통	Obj-C	크로스 사이트 스크립팅
소스코드	보통	Obj-C	위험한 임시 파일 생성
소스코드	높음	Obj-C	명령어 삽입

소스코드	보통	Obj-C	빈 catch 블록
소스코드	보통	Obj-C	지나치게 일반적인 예외 처리
소스코드	보통	Obj-C	DNS lookup에 의존한 보안 결정
소스코드	높음	Obj-C	민감한 정보를 평문으로 저장
소스코드	매우 높음	Obj-C	XQuery 삽입
소스코드	높음	Obj-C	XPath 삽입
소스코드	높음	Obj-C	LDAP 삽입
소스코드	높음	Obj-C	신뢰할 수 없는 쿠키 값에 의존한 보안 결정
소스코드	보통	Obj-C	민감한 정보의 일반 텍스트 전송
소스코드	높음	Obj-C	무결성 검사 없는 코드 다운로드
소스코드	높음	Obj-C	광범위 경로를 허용하는 쿠키 사용
소스코드	높음	Obj-C	광범위 도메인을 허용하는 쿠키 사용
소스코드	높음	Obj-C	민감한 데이터를 지속적인 쿠키에 저장
소스코드	높음	Obj-C	비어 있는 솔트
소스코드	보통	Obj-C	널 솔트 사용
소스코드	높음	Obj-C	부적절한 난수 생성
소스코드	낮음	Obj-C	시스템 정보 노출
소스코드	보통	Obj-C	남은 디버그 코드
소스코드	높음	Obj-C	누락된 로그인 제어
소스코드	높음	Obj-C	취약한 비밀번호 요구 조건
소스코드	높음	Obj-C	누락된 인증
소스코드	보통	Obj-C	HTTP 응답 분할
소스코드	높음	Obj-C	정수 오버플로우
소스코드	높음	Obj-C	부적절한 XML 외부 엔티티 참조
소스코드	보통	Obj-C	취약한 서버 인증서 검증
소스코드	높음	Obj-C	신뢰할 수 없는 데이터의 역직렬화
소스코드	보통	Obj-C	임시 혹은 공개 디렉토리에 파일 출력
소스코드	보통	Obj-C	취약한 SSL 인증서
소스코드	보통	Obj-C	GET 방식으로 전송
소스코드	보통	Obj-C	HTTP 프로토콜을 통한 전송
소스코드	높음	Obj-C	빈 비밀번호
소스코드	높음	Obj-C	하드코드된 비밀번호
소스코드	보통	Obj-C	SMS 전송 API 사용
소스코드	높음	Obj-C	하드코드된 암호화 키
소스코드	높음	Obj-C	취약한 해쉬
소스코드	높음	Obj-C	불충분한 암호화 키 길이

소스코드	높음	Obj-C	취약한 암호화 알고리즘
소스코드	높음	Obj-C	포맷 스트링 삽입
소스코드	매우 높음	Obj-C	로그 삽입
소스코드	높음	Obj-C	경로 조작
소스코드	높음	Obj-C	자원 삽입
소스코드	높음	Obj-C	SQL 삽입
소스코드	높음	Obj-C	안전하지 않은 리플렉션
소스코드	보통	PHP	헤더 조작
소스코드	높음	PHP	명령어 삽입
소스코드	높음	PHP	원격 코드 실행
소스코드	보통	PHP	기본 경로로 설정된 쿠키
소스코드	높음	PHP	취약한 해쉬
소스코드	높음	PHP	제한되지 않은 파일 업로드
소스코드	보통	PHP	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	매우 높음	PHP	XQuery 삽입
소스코드	높음	PHP	XPath 삽입
소스코드	높음	PHP	LDAP 삽입
소스코드	높음	PHP	사이트 간 요청 위조
소스코드	높음	PHP	신뢰할 수 없는 입력에 의존한 보안 결정
소스코드	높음	PHP	포맷 스트링 삽입
소스코드	높음	PHP	중요 기능에 대해 누락된 인증
소스코드	높음	PHP	부적절한 인증
소스코드	높음	PHP	취약한 암호화 알고리즘
소스코드	보통	PHP	민감한 정보의 일반 텍스트 저장
소스코드	보통	PHP	민감한 정보의 일반 텍스트 전송
소스코드	높음	PHP	하드코드된 비밀번호
소스코드	높음	PHP	불충분한 암호화 키 길이
소스코드	높음	PHP	부적절한 난수 생성
소스코드	높음	PHP	하드코드된 암호화 키
소스코드	높음	PHP	취약한 비밀번호 요구 조건
소스코드	높음	PHP	누락된 로그인 제어
소스코드	보통	PHP	오류 정보 누출
소스코드	보통	PHP	남은 디버그 코드
소스코드	높음	PHP	DNS lookup에 의존한 보안 결정
소스코드	높음	PHP	주석문 안의 비밀번호
소스코드	보통	PHP	HTTP 응답 분할

소스코드	보통	PHP	민감한 정보의 쿠키
소스코드	높음	PHP	솔트 없는 해쉬 사용
소스코드	높음	PHP	취약한 XML 변환 보안
소스코드	높음	PHP	서버측 요청 변조
소스코드	보통	PHP	취약한 서버 인증서 검증
소스코드	높음	PHP	신뢰할 수 없는 데이터의 역직렬화
소스코드	높음	PHP	과도하게 허용된 Cross-Origin Resource Sharing 정책
소스코드	높음	PHP	GET 방식으로 전송
소스코드	높음	PHP	활성화 된 allowed_url_fopen 옵션
소스코드	높음	PHP	활성화 된 allowed_url_include 옵션
소스코드	보통	PHP	비활성화된 session.cookie_secure 옵션
소스코드	높음	PHP	비활성화 된 cgi.force_redirect 옵션
소스코드	보통	PHP	비활성화 된 safe_mode 옵션
소스코드	높음	PHP	활성화 된 file_uploads 옵션
소스코드	높음	PHP	활성화 된 magic_quotes_gpc 옵션
소스코드	높음	PHP	활성화 된 magic_quotes_runtime 옵션
소스코드	높음	PHP	활성화 된 magic_quotes_sybase 옵션
소스코드	보통	PHP	활성화 된 register_globals 옵션
소스코드	보통	PHP	활성화된 display_errors 옵션
소스코드	보통	PHP	시스템 정보 노출
소스코드	높음	PHP	누락된 open_basedir 옵션
소스코드	보통	PHP	누락된 safe_mode_exec_dir 옵션
소스코드	보통	PHP	기본 도메인을 사용하는 쿠키 설정
소스코드	보통	PHP	기본 경로를 사용하는 쿠키 설정
소스코드	보통	PHP	영속적인 쿠키
소스코드	보통	PHP	비활성화 된 session.cookie_httponly 옵션
소스코드	높음	PHP	활성화 된 session.use_trans_sid 옵션
소스코드	높음	PHP	CakePHP에서의 과도한 세션 타임아웃
소스코드	보통	PHP	CakePHP에서의 정보 누출
소스코드	보통	PHP	쿠키 평문 전송
소스코드	보통	PHP	HTTP 프로토콜을 통한 전송
소스코드	보통	PHP	크로스 사이트 스크립팅
소스코드	높음	PHP	자원 권한 조작
소스코드	높음	PHP	SQL 삽입
소스코드	높음	PHP	경로 조작
소스코드	매우 높음	PHP	외부 변수 수정

소스코드	보통	PHP	로그 삽입
소스코드	높음	Prop	설정 파일에 하드코딩된 비밀번호
소스코드	높음	Prop	비어 있는 비밀번호
소스코드	높음	Python	SQL 삽입
소스코드	높음	Python	코드 삽입
소스코드	높음	Python	경로 조작 및 자원 삽입
소스코드	보통	Python	크로스 사이트 스크립트(HTML)
소스코드	높음	Python	운영체제 명령어 삽입
소스코드	높음	Python	위험한 형식 파일 업로드
소스코드	보통	Python	신뢰되지 않은 URL 주소로 자동접속 연결
소스코드	높음	Python	부적절한 XML 외부 개체 참조
소스코드	높음	Python	XML 삽입
소스코드	높음	Python	LDAP 삽입
소스코드	높음	Python	크로스 사이트 요청 위조
소스코드	높음	Python	서버사이드 요청 위조
소스코드	보통	Python	위험한 임시 파일 생성
소스코드	높음	Python	잘못된 umask 인자
소스코드	보통	Python	직접 동적 코드 평가
소스코드	보통	Python	로그 조작
소스코드	높음	Python	SMTP 명령어 삽입
소스코드	보통	Python	Memcached 삽입
소스코드	높음	Python	경로 조작
소스코드	매우 높음	Python	시스템 혹은 구성 설정의 외부 제어
소스코드	높음	Python	django 파일 응답 조작
소스코드	보통	Python	HTTP 응답 분할
소스코드	높음	Python	보안 기능 결정에 사용되는 부적절한 입력값
소스코드	높음	Python	포맷 스트링 삽입
소스코드	낮음	Python	django 디버그 설정에 의한 정보 노출
소스코드	높음	Python	django에서 sleep() 함수 사용
소스코드	높음	Python	중요한 자원에 대한 잘못된 권한 설정
소스코드	높음	Python	부적절한 인증
소스코드	보통	Python	민감한 정보를 평문으로 저장
소스코드	보통	Python	민감한 정보를 평문으로 전송
소스코드	높음	Python	하드코딩된 암호화 키
소스코드	높음	Python	누락된 로그인 제어
소스코드	보통	Python	시스템 정보 노출

소스코드	보통	Python	오류 정보 노출
소스코드	매우 높음	Python	XQuery 삽입
소스코드	높음	Python	DNS lookup에 의존한 보안 결정
소스코드	높음	Python	취약한 암호화 알고리즘 사용
소스코드	보통	Python	취약한 서버 인증서 검증
소스코드	높음	Python	하드코드된 중요정보
소스코드	높음	Python	충분하지 않은 키 길이 사용
소스코드	높음	Python	적절하지 않은 난수 값 사용
소스코드	보통	Python	사용자 하드디스크에 저장되는 쿠키를 통한 정보 노출
소스코드	높음	Python	주석문 안에 포함된 시스템 주요정보
소스코드	보통	Python	솔트 없이 일방향 해쉬 함수 사용
소스코드	높음	Python	무결성 검사없는 코드 다운로드
소스코드	보통	Python	경쟁조건: 검사시점과 사용시점(TOCTOU)
소스코드	보통	Python	종료되지 않는 반복문 또는 재귀 함수
소스코드	보통	Python	오류 메시지 정보 노출
소스코드	보통	Python	오류상황 대응 부재
소스코드	보통	Python	부적절한 예외 처리
소스코드	보통	Python	Null Pointer 역참조
소스코드	보통	Python	부적절한 자원 해제
소스코드	높음	Python	신뢰할 수 없는 데이터의 역직렬화
소스코드	보통	Python	제거되지 않고 남은 디버그 코드
소스코드	높음	Python	Public 메소드로부터 반환된 Private 배열
소스코드	높음	Python	Private 배열에 Public 데이터 할당
소스코드	높음	Python	적절한 인증 없는 중요 기능 허용
소스코드	높음	Python	취약한 비밀번호 허용
소스코드	보통	Python	크로스 사이트 스크립트
소스코드	높음	SQL	iBatis에서의 SQL 삽입
소스코드	높음	SQL	myBatis에서의 SQL 삽입
소스코드	높음	Swift	하드코딩된 비밀번호
소스코드	높음	Swift	비어 있는 비밀번호
소스코드	보통	Swift	취약한 해쉬 알고리즘
소스코드	보통	Swift	ECB 모드로 암호화 알고리즘 사용
소스코드	높음	Swift	하드코딩된 IP
소스코드	보통	Swift	GET 방식으로 전송
소스코드	높음	Swift	광범위 도메인을 허용하는 쿠키 사용
소스코드	높음	Swift	광범위 경로를 허용하는 쿠키 사용

소스코드	높음	Swift	민감한 데이터를 지속적인 쿠키에 저장
소스코드	높음	Swift	비어 있는 HMAC 키
소스코드	높음	Swift	비어 있는 솔트
소스코드	높음	Swift	하드코딩된 솔트
소스코드	보통	Swift	안전하지 않은 쿠키
소스코드	보통	Swift	취약한 프레임워크에 의존한 보안 결정
소스코드	낮음	Swift	시스템 정보 노출
소스코드	높음	Swift	취약한 SSL 프로토콜
소스코드	낮음	Swift	SMS 기능 사용
소스코드	매우 낮음	Swift	자동 다이얼링 공격 방지
소스코드	낮음	Swift	HTTP 헤더 조작
소스코드	보통	Swift	반복 횟수가 적은 키 파생 함수 사용
소스코드	보통	Swift	널 솔트 사용
소스코드	높음	Swift	SQL 삽입
소스코드	높음	Swift	자원 삽입
소스코드	보통	Swift	크로스 사이트 스크립팅
소스코드	높음	Swift	명령어 삽입
소스코드	높음	Swift	민감한 정보를 평문으로 저장
소스코드	높음	Swift	하드코딩된 암호화 키
소스코드	높음	Swift	XPath 삽입
소스코드	높음	Swift	LDAP 삽입
소스코드	보통	Swift	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	높음	Swift	불충분한 암호화 키 길이
소스코드	보통	Swift	민감한 정보의 일반 텍스트 전송
소스코드	높음	Swift	무결성 검사 없는 코드 다운로드
소스코드	보통	Swift	빈 catch 블록
소스코드	보통	Swift	지나치게 일반적인 예외 처리
소스코드	높음	Swift	누락된 로그인 제어
소스코드	높음	Swift	취약한 비밀번호 요구 조건
소스코드	보통	Swift	남은 디버그 코드
소스코드	보통	Swift	TOCTOU 경쟁 조건
소스코드	보통	Swift	HTTP 응답 분할
소스코드	높음	Swift	주석문 안의 비밀번호
소스코드	높음	Swift	포맷 스트링 삽입
소스코드	높음	Swift	신뢰할 수 없는 쿠키 값에 의존한 보안 결정
소스코드	높음	Swift	누락된 인증

소스코드	높음	Swift	부적절한 인증
소스코드	보통	Swift	DNS lookup에 의존한 보안 결정
소스코드	높음	Swift	부적절한 난수 생성
소스코드	높음	Swift	부적절한 XML 외부 엔티티 참조
소스코드	보통	Swift	취약한 서버 인증서 검증
소스코드	높음	Swift	신뢰할 수 없는 데이터의 역직렬화
소스코드	높음	TS	Strict 모드 그룹 사용
소스코드	높음	TS	암묵적 Any 유형 사용 확인
소스코드	보통	TS	명확한 Null 유형 확인
소스코드	높음	TS	항상 strict 모드 추가
소스코드	보통	기타	활성화된 원격 모니터링
소스코드	보통	기타	비활성화된 바이트코드 검증
소스코드	낮음	기타	JavaScript 내에 존재하는 사용자 계정 정보
소스코드	보통	기타	주석문에 비밀번호 노출
소스코드	보통	기타	하드코딩된 비밀번호
소스코드	높음	VB.Net	명령어 삽입
소스코드	보통	VB.Net	빈 catch 블록
소스코드	높음	VB.Net	하드코드된 사용자 이름과 비밀번호
소스코드	보통	VB.Net	HTTP 응답 분할
소스코드	높음	VB.Net	LDAP 삽입
소스코드	보통	VB.Net	시스템 정보 노출
소스코드	보통	VB.Net	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	낮음	VB.Net	지나치게 일반적인 예외 처리
소스코드	높음	VB.Net	경로 조작
소스코드	높음	VB.Net	DNS lookup에 의존한 보안 결정
소스코드	높음	VB.Net	SQL 삽입
소스코드	높음	VB.Net	제한되지 않은 파일 업로드
소스코드	높음	VB.Net	취약한 암호화 알고리즘
소스코드	높음	VB.Net	불충분한 암호화 키 길이
소스코드	높음	VB.Net	XPath 삽입
소스코드	매우 높음	VB.Net	XQuery 삽입
소스코드	높음	VB.Net	취약한 비밀번호 요구 조건
소스코드	보통	VB.Net	크로스 사이트 스크립팅
소스코드	높음	VB.Net	매개 변수화되지 않은 쿼리 사용
소스코드	높음	VB.Net	부적절한 서명
소스코드	보통	VB.Net	취약한 서버 인증서 검증

소스코드	높음	VBS	과도하게 허용된 Cross-Origin Resource Sharing 정책
소스코드	높음	VBS	명령어 삽입
소스코드	높음	VBS	직접 동적 코드 평가
소스코드	높음	VBS	부적절한 난수 생성
소스코드	높음	VBS	시스템 정보 노출
소스코드	보통	VBS	로그 삽입
소스코드	보통	VBS	신뢰할 수 없는 사이트로의 리다이렉션
소스코드	높음	VBS	경로 조작
소스코드	매우 높음	VBS	설정 조작
소스코드	높음	VBS	SQL 삽입
소스코드	보통	VBS	안전하지 않은 리플렉션
소스코드	높음	VBS	취약한 암호화 알고리즘
소스코드	높음	VBS	취약한 해쉬
소스코드	높음	VBS	불충분한 암호화 키 길이
소스코드	보통	VBS	크로스 사이트 스크립팅
소스코드	보통	VBS	헤더 조작
소스코드	높음	VBS	자원 삽입
소스코드	보통	XML	부적절한 로그 레벨 설정
소스코드	보통	XML	중복된 서블릿 매핑
소스코드	보통	XML	중복된 보안 역할
소스코드	보통	XML	과도한 서블릿 매핑
소스코드	높음	XML	과도한 세션 유지 시간
소스코드	보통	XML	JSP 직접 접근
소스코드	보통	XML	불충분한 세션 ID 길이
소스코드	보통	XML	잘못 설정한 서블릿 사용
소스코드	보통	XML	존재하지 않는 필터 사용
소스코드	높음	XML	하드코딩된 비밀번호
소스코드	높음	XML	디버그 정보 노출
소스코드	높음	XML	기본 오류 페이지 사용
소스코드	높음	XML	Trace 로그 노출
소스코드	높음	XML	HttpOnly가 아닌 쿠키
소스코드	보통	XML	스트럿츠 : 중복 밸리데이션 양식
소스코드	보통	XML	Struts 입력 유효성 검사 프레임워크 미사용
오픈소스	매우 낮음	공통	0BSD 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AAL 라이선스 컴포넌트 사용
오픈소스	높음	공통	Abstyles 라이선스 컴포넌트 사용

오픈소스	높음	공통	AdaCore-doc 라이선스 컴포넌트 사용
오픈소스	높음	공통	Adobe-2006 라이선스 컴포넌트 사용
오픈소스	높음	공통	Adobe-Glyph 라이선스 컴포넌트 사용
오픈소스	높음	공통	ADSL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AFL-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AFL-1.2 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AFL-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AFL-2.1 라이선스 컴포넌트 사용
오픈소스	보통	공통	AFL-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Afmparse 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	AGPL-1.0-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	AGPL-1.0-or-later 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	AGPL-3.0-only 라이선스 컴포넌트 사용
오픈소스	보통	공통	AGPL-3.0-or-later 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Aladdin 라이선스 컴포넌트 사용
오픈소스	높음	공통	AMDPLPA 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AML 라이선스 컴포넌트 사용
오픈소스	낮음	공통	AMPAS 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ANTLR-PD 라이선스 컴포넌트 사용
오픈소스	높음	공통	ANTLR-PD-fallback 라이선스 컴포넌트 사용
오픈소스	높음	공통	Apache-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Apache-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Apache-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	APAFML 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	APL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	App-s2p 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	APSL-1.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	APSL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	APSL-1.2 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	APSL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Arphic-1999 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Artistic-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Artistic-1.0-cl8 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Artistic-1.0-Perl 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Artistic-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Baekmuk 라이선스 컴포넌트 사용

오픈소스	높음	공통	Bahyph 라이선스 컴포넌트 사용
오픈소스	높음	공통	Barr 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Beerware 라이선스 컴포넌트 사용
오픈소스	높음	공통	Bitstream-Charter 라이선스 컴포넌트 사용
오픈소스	높음	공통	Bitstream-Vera 라이선스 컴포넌트 사용
오픈소스	높음	공통	BitTorrent-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	BitTorrent-1.1 라이선스 컴포넌트 사용
오픈소스	매우 낮음	공통	blessing 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BlueOak-1.0.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Borceux 라이선스 컴포넌트 사용
오픈소스	높음	공통	Brian-Gladman-3-Clause 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-1-Clause 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-2-Clause 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-2-Clause-FreeBSD 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-2-Clause-NetBSD 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-2-Clause-Patent 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-2-Clause-Views 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause-Attribution 라이선스 컴포넌트 사용
오픈소스	보통	공통	BSD-3-Clause-Clear 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause-LBNL 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-Modification 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-No-Military-License 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause-No-Nuclear-License 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause-No-Nuclear-License-2014 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-No-Nuclear-Warranty 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-3-Clause-Open-MPI 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-4-Clause 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-4-Clause-Shortened 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-4-Clause-UC 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-4.3RENO 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-4.3TAHOE 라이선스 컴포넌트 사용

오픈소스	높음	공통	BSD-Advertising-Acknowledgement 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-Attribution-HPND-disclaimer 라이선스 컴포넌트 사용
오픈소스	보통	공통	BSD-Protection 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSD-Source-Code 라이선스 컴포넌트 사용
오픈소스	낮음	공통	BSL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	BUSL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	bzip2-1.0.5 라이선스 컴포넌트 사용
오픈소스	낮음	공통	bzip2-1.0.6 라이선스 컴포넌트 사용
오픈소스	높음	공통	C-UDA-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CAL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CAL-1.0-Combined-Work-Exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	Caldera 라이선스 컴포넌트 사용
오픈소스	높음	공통	CATOSL-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-2.5 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-2.5-AU 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0-AT 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0-DE 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0-IGO 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0-NL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-3.0-US 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CC-BY-4.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-2.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-2.5 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-3.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-3.0-DE 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-4.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-2.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-2.5 라이선스 컴포넌트 사용

오픈소스	보통	공통	CC-BY-NC-ND-3.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-3.0-DE 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-3.0-IGO 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-NC-ND-4.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-1.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-2.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-2.0-DE 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-2.0-FR 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-2.0-UK 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-2.5 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-3.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-3.0-DE 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-3.0-IGO 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CC-BY-NC-SA-4.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-2.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-2.5 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-3.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-3.0-DE 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-ND-4.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-2.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-2.0-UK 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-2.1-JP 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-2.5 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-3.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-3.0-AT 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-3.0-DE 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-4.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CC-PDDC 라이선스 컴포넌트 사용
오픈소스	매우 낮음	공통	CC0-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CDDL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CDDL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	CDL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CDLA-Permissive-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CDLA-Permissive-2.0 라이선스 컴포넌트 사용

오픈소스	보통	공통	CDLA-Sharing-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CECILL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CECILL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CECILL-2.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CECILL-2.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	CECILL-B 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CECILL-C 라이선스 컴포넌트 사용
오픈소스	높음	공통	CERN-OHL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	CERN-OHL-1.2 라이선스 컴포넌트 사용
오픈소스	높음	공통	CERN-OHL-P-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CERN-OHL-S-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CERN-OHL-W-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	CFITSIO 라이선스 컴포넌트 사용
오픈소스	높음	공통	checkmk 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	ClArtistic 라이선스 컴포넌트 사용
오픈소스	높음	공통	Clips 라이선스 컴포넌트 사용
오픈소스	높음	공통	CMU-Mach 라이선스 컴포넌트 사용
오픈소스	높음	공통	CNRI-Jython 라이선스 컴포넌트 사용
오픈소스	낮음	공통	CNRI-Python 라이선스 컴포넌트 사용
오픈소스	높음	공통	CNRI-Python-GPL-Compatible 라이선스 컴포넌트 사용
오픈소스	높음	공통	COIL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Community-Spec-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Condor-1.1 라이선스 컴포넌트 사용
오픈소스	보통	공통	copyleft-next-0.3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	copyleft-next-0.3.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	Cornell-Lossless-JPEG 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	CPAL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CPL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	CPOL-1.02 라이선스 컴포넌트 사용
오픈소스	높음	공통	Crossword 라이선스 컴포넌트 사용
오픈소스	높음	공통	CrystalStacker 라이선스 컴포넌트 사용
오픈소스	높음	공통	CUA-OPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Cube 라이선스 컴포넌트 사용
오픈소스	낮음	공통	curl 라이선스 컴포넌트 사용
오픈소스	높음	공통	D-FSL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	diffmark 라이선스 컴포넌트 사용

오픈소스	높음	공통	DL-DE-BY-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	DOC 라이선스 컴포넌트 사용
오픈소스	높음	공통	Dotseqn 라이선스 컴포넌트 사용
오픈소스	높음	공통	DRL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	DSDP 라이선스 컴포넌트 사용
오픈소스	높음	공통	dvipdfm 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ECL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ECL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	eCos-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	EFL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	EFL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	eGenix 라이선스 컴포넌트 사용
오픈소스	보통	공통	Elastic-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Entessa 라이선스 컴포넌트 사용
오픈소스	높음	공통	EPICS 라이선스 컴포넌트 사용
오픈소스	보통	공통	EPL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	EPL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	ErlPL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	etalab-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	EUDatagrid 라이선스 컴포넌트 사용
오픈소스	높음	공통	EUPL-1.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	EUPL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	EUPL-1.2 라이선스 컴포넌트 사용
오픈소스	높음	공통	Eurosym 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Fair 라이선스 컴포넌트 사용
오픈소스	높음	공통	FDK-AAC 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Frameworx-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	FreeBSD-DOC 라이선스 컴포넌트 사용
오픈소스	높음	공통	FreeImage 라이선스 컴포넌트 사용
오픈소스	낮음	공통	FSFAP 라이선스 컴포넌트 사용
오픈소스	높음	공통	FSFUL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	FSFULLR 라이선스 컴포넌트 사용
오픈소스	높음	공통	FSFULLRWD 라이선스 컴포넌트 사용
오픈소스	낮음	공통	FTL 라이선스 컴포넌트 사용
오픈소스	높음	공통	GD 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.1-invariants-only 라이선스 컴포넌트 사용

오픈소스	높음	공통	GFDL-1.1-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.1-no-invariants-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.1-no-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.1-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.1-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-invariants-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-no-invariants-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-no-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.2-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-invariants-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-no-invariants-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-no-invariants-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-only 라이선스 컴포넌트 사용
오픈소스	높음	공통	GFDL-1.3-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	Giftware 라이선스 컴포넌트 사용
오픈소스	높음	공통	GL2PS 라이선스 컴포넌트 사용
오픈소스	높음	공통	Glide 라이선스 컴포넌트 사용
오픈소스	높음	공통	Glulxe 라이선스 컴포넌트 사용
오픈소스	높음	공통	GLWTPL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	gnuplot 라이선스 컴포넌트 사용
오픈소스	보통	공통	GPL-1.0-only 라이선스 컴포넌트 사용
오픈소스	보통	공통	GPL-1.0-or-later 라이선스 컴포넌트 사용
오픈소스	보통	공통	GPL-2.0-only 라이선스 컴포넌트 사용
오픈소스	보통	공통	GPL-2.0-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-2.0-with-autoconf-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-2.0-with-bison-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-2.0-with-classpath-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-2.0-with-font-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-2.0-with-GCC-exception 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	GPL-3.0-only 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	GPL-3.0-or-later 라이선스 컴포넌트 사용

오픈소스	높음	공통	GPL-3.0-with-autoconf-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	GPL-3.0-with-GCC-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	Graphics-Gems 라이선스 컴포넌트 사용
오픈소스	높음	공통	gSOAP-1.3b 라이선스 컴포넌트 사용
오픈소스	높음	공통	HaskellReport 라이선스 컴포넌트 사용
오픈소스	높음	공통	Hippocratic-2.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	HP-1986 라이선스 컴포넌트 사용
오픈소스	낮음	공통	HPND 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-export-US 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Markus-Kuhn 라이선스 컴포넌트 사용
오픈소스	낮음	공통	HPND-sell-variant 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-sell-variant-MIT-disclaimer 라이선스 컴포넌트 사용
오픈소스	높음	공통	HTMLTIDY 라이선스 컴포넌트 사용
오픈소스	높음	공통	IBM-pibs 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ICU 라이선스 컴포넌트 사용
오픈소스	높음	공통	IEC-Code-Components-EULA 라이선스 컴포넌트 사용
오픈소스	낮음	공통	IJG 라이선스 컴포넌트 사용
오픈소스	높음	공통	IJG-short 라이선스 컴포넌트 사용
오픈소스	높음	공통	ImageMagick 라이선스 컴포넌트 사용
오픈소스	높음	공통	iMatix 라이선스 컴포넌트 사용
오픈소스	높음	공통	Imlib2 라이선스 컴포넌트 사용
오픈소스	높음	공통	Info-ZIP 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Intel 라이선스 컴포넌트 사용
오픈소스	높음	공통	Intel-ACPI 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Interbase-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	IPA 라이선스 컴포넌트 사용
오픈소스	보통	공통	IPL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ISC 라이선스 컴포넌트 사용
오픈소스	높음	공통	Jam 라이선스 컴포넌트 사용
오픈소스	높음	공통	JasPer-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	JPL-image 라이선스 컴포넌트 사용
오픈소스	높음	공통	JPNIC 라이선스 컴포넌트 사용
오픈소스	낮음	공통	JSON 라이선스 컴포넌트 사용
오픈소스	높음	공통	Kazlib 라이선스 컴포넌트 사용

오픈소스	높음	공통	KiCad-libraries-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	Knuth-CTAN 라이선스 컴포넌트 사용
오픈소스	높음	공통	LAL-1.2 라이선스 컴포넌트 사용
오픈소스	높음	공통	LAL-1.3 라이선스 컴포넌트 사용
오픈소스	높음	공통	Latex2e 라이선스 컴포넌트 사용
오픈소스	높음	공통	Leptonica 라이선스 컴포넌트 사용
오픈소스	보통	공통	LGPL-2.0-only 라이선스 컴포넌트 사용
오픈소스	보통	공통	LGPL-2.0-or-later 라이선스 컴포넌트 사용
오픈소스	보통	공통	LGPL-2.1-only 라이선스 컴포넌트 사용
오픈소스	보통	공통	LGPL-2.1-or-later 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	LGPL-3.0-only 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	LGPL-3.0-or-later 라이선스 컴포넌트 사용
오픈소스	높음	공통	LGPLLR 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Libpng 라이선스 컴포넌트 사용
오픈소스	낮음	공통	libpng-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	libselinux-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	libtiff 라이선스 컴포넌트 사용
오픈소스	높음	공통	libutil-David-Nugent 라이선스 컴포넌트 사용
오픈소스	낮음	공통	LiLiQ-P-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	LiLiQ-R-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	LiLiQ-Rplus-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	Linux-man-pages-copyleft 라이선스 컴포넌트 사용
오픈소스	높음	공통	Linux-OpenIB 라이선스 컴포넌트 사용
오픈소스	높음	공통	LOOP 라이선스 컴포넌트 사용
오픈소스	낮음	공통	LPL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	LPL-1.02 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPPL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPPL-1.2 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPPL-1.3a 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPPL-1.3c 라이선스 컴포넌트 사용
오픈소스	높음	공통	LZMA-SDK-9.11-to-9.20 라이선스 컴포넌트 사용
오픈소스	높음	공통	LZMA-SDK-9.22 라이선스 컴포넌트 사용
오픈소스	높음	공통	MakeIndex 라이선스 컴포넌트 사용
오픈소스	높음	공통	Martin-Birgmeier 라이선스 컴포넌트 사용
오픈소스	높음	공통	Minpack 라이선스 컴포넌트 사용

오픈소스	낮음	공통	MirOS 라이선스 컴포넌트 사용
오픈소스	낮음	공통	MIT 라이선스 컴포넌트 사용
오픈소스	낮음	공통	MIT-0 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-advertising 라이선스 컴포넌트 사용
오픈소스	낮음	공통	MIT-CMU 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-enna 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-feh 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-Modern-Variant 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-open-group 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-Wu 라이선스 컴포넌트 사용
오픈소스	높음	공통	MITNFA 라이선스 컴포넌트 사용
오픈소스	높음	공통	Motosoto 라이선스 컴포넌트 사용
오픈소스	높음	공통	mpi-permissive 라이선스 컴포넌트 사용
오픈소스	높음	공통	mpich2 라이선스 컴포넌트 사용
오픈소스	보통	공통	MPL-1.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	MPL-1.1 라이선스 컴포넌트 사용
오픈소스	보통	공통	MPL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	MPL-2.0-no-copyleft-exception 라이선스 컴포넌트 사용
오픈소스	높음	공통	mplus 라이선스 컴포넌트 사용
오픈소스	보통	공통	MS-LPL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	MS-PL 라이선스 컴포넌트 사용
오픈소스	보통	공통	MS-RL 라이선스 컴포넌트 사용
오픈소스	높음	공통	MTLL 라이선스 컴포넌트 사용
오픈소스	높음	공통	MulanPSL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	MulanPSL-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Multics 라이선스 컴포넌트 사용
오픈소스	높음	공통	Mup 라이선스 컴포넌트 사용
오픈소스	높음	공통	NAIST-2003 라이선스 컴포넌트 사용
오픈소스	높음	공통	NASA-1.3 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Naumen 라이선스 컴포넌트 사용
오픈소스	높음	공통	NBPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NCGL-UK-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	NCSA 라이선스 컴포넌트 사용
오픈소스	높음	공통	Net-SNMP 라이선스 컴포넌트 사용
오픈소스	높음	공통	NetCDF 라이선스 컴포넌트 사용
오픈소스	높음	공통	Newsletr 라이선스 컴포넌트 사용

오픈소스	높음	공통	NGPL 라이선스 컴포넌트 사용
오픈소스	높음	공통	NICTA-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NIST-PD 라이선스 컴포넌트 사용
오픈소스	높음	공통	NIST-PD-fallback 라이선스 컴포넌트 사용
오픈소스	높음	공통	NLOD-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NLOD-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NLPL 라이선스 컴포넌트 사용
오픈소스	보통	공통	Nokia 라이선스 컴포넌트 사용
오픈소스	높음	공통	NOSL 라이선스 컴포넌트 사용
오픈소스	높음	공통	Noweb 라이선스 컴포넌트 사용
오픈소스	높음	공통	NPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NPL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	NPOSL-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	NRL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	NTP 라이선스 컴포넌트 사용
오픈소스	높음	공통	NTP-0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Nunit 라이선스 컴포넌트 사용
오픈소스	높음	공통	O-UDA-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OCCT-PL 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OCLC-2.0 라이선스 컴포넌트 사용
오픈소스	보통	공통	ODbL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ODC-By-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFFIS 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFL-1.0-no-RFN 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFL-1.0-RFN 라이선스 컴포넌트 사용
오픈소스	보통	공통	OFL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFL-1.1-no-RFN 라이선스 컴포넌트 사용
오픈소스	높음	공통	OFL-1.1-RFN 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGC-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGDL-Taiwan-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGL-Canada-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGL-UK-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGL-UK-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGL-UK-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OGTSL 라이선스 컴포넌트 사용

오픈소스	보통	공통	OLDAP-1.1 라이선스 컴포넌트 사용
오픈소스	보통	공통	OLDAP-1.2 라이선스 컴포넌트 사용
오픈소스	보통	공통	OLDAP-1.3 라이선스 컴포넌트 사용
오픈소스	보통	공통	OLDAP-1.4 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.0.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.2 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.2.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.2.2 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.3 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.4 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.5 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.6 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.7 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OLDAP-2.8 라이선스 컴포넌트 사용
오픈소스	높음	공통	OML 라이선스 컴포넌트 사용
오픈소스	높음	공통	OpenPBS-2.3 라이선스 컴포넌트 사용
오픈소스	낮음	공통	OpenSSL 라이선스 컴포넌트 사용
오픈소스	높음	공통	OPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OPUBL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	OSET-PL-2.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	OSL-1.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	OSL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	OSL-2.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	OSL-2.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	OSL-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Parity-6.0.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Parity-7.0.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	PDDL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	PHP-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	PHP-3.01 라이선스 컴포넌트 사용
오픈소스	높음	공통	Plexus 라이선스 컴포넌트 사용
오픈소스	높음	공통	PolyForm-Noncommercial-1.0.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	PolyForm-Small-Business-1.0.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	PostgreSQL 라이선스 컴포넌트 사용

오픈소스	낮음	공통	PSF-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	psfrag 라이선스 컴포넌트 사용
오픈소스	높음	공통	psutils 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Python-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Python-2.0.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	Qhull 라이선스 컴포넌트 사용
오픈소스	보통	공통	QPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	QPL-1.0-INRIA-2004 라이선스 컴포넌트 사용
오픈소스	높음	공통	Rdisc 라이선스 컴포넌트 사용
오픈소스	높음	공통	RHeCos-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	RPL-1.1 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	RPL-1.5 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	RPSL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	RSA-MD 라이선스 컴포넌트 사용
오픈소스	높음	공통	RSCPL 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Ruby 라이선스 컴포넌트 사용
오픈소스	높음	공통	SAX-PD 라이선스 컴포넌트 사용
오픈소스	높음	공통	Saxpath 라이선스 컴포넌트 사용
오픈소스	높음	공통	SCEA 라이선스 컴포넌트 사용
오픈소스	높음	공통	Sendmail 라이선스 컴포넌트 사용
오픈소스	높음	공통	Sendmail-8.23 라이선스 컴포넌트 사용
오픈소스	낮음	공통	SGI-B-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	SGI-B-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	SGI-B-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SHL-0.5 라이선스 컴포넌트 사용
오픈소스	높음	공통	SHL-0.51 라이선스 컴포넌트 사용
오픈소스	높음	공통	SimPL-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SISSL 라이선스 컴포넌트 사용
오픈소스	높음	공통	SISSL-1.2 라이선스 컴포넌트 사용
오픈소스	높음	공통	Sleepycat 라이선스 컴포넌트 사용
오픈소스	높음	공통	SMLNJ 라이선스 컴포넌트 사용
오픈소스	높음	공통	SMPPL 라이선스 컴포넌트 사용
오픈소스	높음	공통	SNIA 라이선스 컴포넌트 사용
오픈소스	높음	공통	snprintf 라이선스 컴포넌트 사용
오픈소스	높음	공통	Spencer-86 라이선스 컴포넌트 사용
오픈소스	높음	공통	Spencer-94 라이선스 컴포넌트 사용

오픈소스	높음	공통	Spencer-99 라이선스 컴포넌트 사용
오픈소스	높음	공통	SPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SSH-OpenSSH 라이선스 컴포넌트 사용
오픈소스	높음	공통	SSH-short 라이선스 컴포넌트 사용
오픈소스	보통	공통	SSPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SugarCRM-1.1.3 라이선스 컴포넌트 사용
오픈소스	높음	공통	SunPro 라이선스 컴포넌트 사용
오픈소스	높음	공통	SWL 라이선스 컴포넌트 사용
오픈소스	높음	공통	Symlinks 라이선스 컴포넌트 사용
오픈소스	높음	공통	TAPR-OHL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	TCL 라이선스 컴포넌트 사용
오픈소스	높음	공통	TCP-wrappers 라이선스 컴포넌트 사용
오픈소스	높음	공통	TMate 라이선스 컴포넌트 사용
오픈소스	높음	공통	TORQUE-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	TOSL 라이선스 컴포넌트 사용
오픈소스	높음	공통	TPDL 라이선스 컴포넌트 사용
오픈소스	높음	공통	TPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	TTWL 라이선스 컴포넌트 사용
오픈소스	높음	공통	TU-Berlin-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	TU-Berlin-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	UCAR 라이선스 컴포넌트 사용
오픈소스	높음	공통	UCL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Unicode-DFS-2015 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Unicode-DFS-2016 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Unicode-TOU 라이선스 컴포넌트 사용
오픈소스	매우 낮음	공통	Unlicense 라이선스 컴포넌트 사용
오픈소스	낮음	공통	UPL-1.0 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	Vim 라이선스 컴포넌트 사용
오픈소스	높음	공통	VOSTROM 라이선스 컴포넌트 사용
오픈소스	높음	공통	VSL-1.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	W3C 라이선스 컴포넌트 사용
오픈소스	낮음	공통	W3C-19980720 라이선스 컴포넌트 사용
오픈소스	낮음	공통	W3C-20150513 라이선스 컴포넌트 사용
오픈소스	높음	공통	w3m 라이선스 컴포넌트 사용
오픈소스	높음	공통	Watcom-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Wsuipa 라이선스 컴포넌트 사용

오픈소스	매우 낮음	공통	WTFPL 라이선스 컴포넌트 사용
오픈소스	높음	공통	wxWindows 라이선스 컴포넌트 사용
오픈소스	낮음	공통	X11 라이선스 컴포넌트 사용
오픈소스	높음	공통	X11-distribute-modifications-variant 라이선스 컴포넌트 사용
오픈소스	높음	공통	Xerox 라이선스 컴포넌트 사용
오픈소스	낮음	공통	XFree86-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	xinetd 라이선스 컴포넌트 사용
오픈소스	높음	공통	xlock 라이선스 컴포넌트 사용
오픈소스	높음	공통	Xnet 라이선스 컴포넌트 사용
오픈소스	낮음	공통	xpp 라이선스 컴포넌트 사용
오픈소스	높음	공통	XSkat 라이선스 컴포넌트 사용
오픈소스	높음	공통	YPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	YPL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	ZCL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	Zed 라이선스 컴포넌트 사용
오픈소스	높음	공통	Zend-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	Zimbra-1.3 라이선스 컴포넌트 사용
오픈소스	높음	공통	Zimbra-1.4 라이선스 컴포넌트 사용
오픈소스	낮음	공통	Zlib 라이선스 컴포넌트 사용
오픈소스	낮음	공통	zlib-acknowledgement 라이선스 컴포넌트 사용
오픈소스	높음	공통	ZPL-1.1 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ZPL-2.0 라이선스 컴포넌트 사용
오픈소스	낮음	공통	ZPL-2.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	3D-Slicer-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	AMD-newlib 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-2-Clause-first-lines 라이선스 컴포넌트 사용
오픈소스	높음	공통	Catharon 라이선스 컴포넌트 사용
오픈소스	높음	공통	Gutmann 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Intel 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-UC-export-US 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-export-US-acknowledgement 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-export2-US 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-merchantability-variant 라이선스 컴포넌트 사용
			HPND-sell-variant-MIT-disclaimer-rev 라이선스 컴포넌트

오픈소스	높음	공통	사용
오픈소스	높음	공통	MIT-Khronos-old 라이선스 컴포넌트 사용
오픈소스	높음	공통	NCBI-PD 라이선스 컴포넌트 사용
오픈소스	높음	공통	NCL 라이선스 컴포넌트 사용
오픈소스	높음	공통	OAR 라이선스 컴포넌트 사용
오픈소스	높음	공통	PPL 라이선스 컴포넌트 사용
오픈소스	높음	공통	Sun-PPP-2000 라이선스 컴포넌트 사용
오픈소스	높음	공통	any-OSI 라이선스 컴포넌트 사용
오픈소스	높음	공통	cve-tou 라이선스 컴포넌트 사용
오픈소스	높음	공통	pkgconf 라이선스 컴포넌트 사용
오픈소스	높음	공통	threeparttable 라이선스 컴포넌트 사용
오픈소스	높음	공통	xzoom 라이선스 컴포넌트 사용
오픈소스	높음	공통	Adobe-Display-PostScript 라이선스 컴포넌트 사용
오픈소스	높음	공통	Adobe-Utopia 라이선스 컴포넌트 사용
오픈소스	높음	공통	AML-glslang 라이선스 컴포넌트 사용
오픈소스	높음	공통	ASWF-Digital-Assets-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	ASWF-Digital-Assets-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	bcrypt-Solar-Designer 라이선스 컴포넌트 사용
오픈소스	높음	공통	Boehm-GC 라이선스 컴포넌트 사용
오픈소스	높음	공통	Brian-Gladman-2-Clause 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-2-Clause-Darwin 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-acpica 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-flex 라이선스 컴포넌트 사용
오픈소스	보통	공통	CC-BY-SA-3.0-IGO 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-HP 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-3-Clause-Sun 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-Inferno-Nettverk 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-Source-beginning-file 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-Systemics 라이선스 컴포넌트 사용
오픈소스	높음	공통	BSD-Systemics-W3Works 라이선스 컴포넌트 사용
오픈소스	높음	공통	Caldera-no-preamble 라이선스 컴포넌트 사용
오픈소스	높음	공통	CC-BY-3.0-AU 라이선스 컴포넌트 사용
오픈소스	높음	공통	check-cvs 라이선스 컴포넌트 사용
오픈소스	높음	공통	CMU-Mach-nodoc 라이선스 컴포넌트 사용
오픈소스	높음	공통	Cronyx 라이선스 컴포넌트 사용
오픈소스	높음	공통	DEC-3-Clause 라이선스 컴포넌트 사용

오픈소스	높음	공통	DL-DE-ZERO-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	DocBook-Schema 라이선스 컴포넌트 사용
오픈소스	높음	공통	DocBook-XML 라이선스 컴포넌트 사용
오픈소스	높음	공통	DRL-1.1 라이선스 컴포넌트 사용
오픈소스	높음	공통	dtoa 라이선스 컴포넌트 사용
오픈소스	높음	공통	FBM 라이선스 컴포넌트 사용
오픈소스	높음	공통	Ferguson-Twofish 라이선스 컴포넌트 사용
오픈소스	높음	공통	FSFAP-no-warranty-disclaimer 라이선스 컴포넌트 사용
오픈소스	높음	공통	Furuseth 라이선스 컴포넌트 사용
오픈소스	높음	공통	fwlw 라이선스 컴포넌트 사용
오픈소스	높음	공통	GCR-docs 라이선스 컴포넌트 사용
오픈소스	높음	공통	LPD-document 라이선스 컴포넌트 사용
오픈소스	높음	공통	gtkbook 라이선스 컴포넌트 사용
오픈소스	높음	공통	hdparm 라이선스 컴포넌트 사용
오픈소스	높음	공통	HIDAPI 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-doc-sell 라이선스 컴포넌트 사용
오픈소스	높음	공통	HP-1989 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-DEC 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-doc 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-export-US-modify 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Fenneberg-Livingston 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-INRIA-IMAG 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Kevlin-Henney 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-MIT-disclaimer 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Netrek 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-Pbmplus 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-sell-MIT-disclaimer-xserver 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-sell-regexpr 라이선스 컴포넌트 사용
오픈소스	높음	공통	HPND-UC 라이선스 컴포넌트 사용
오픈소스	높음	공통	Kastrup 라이선스 컴포넌트 사용
오픈소스	높음	공통	Inner-Net-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	ISC-Veillard 라이선스 컴포넌트 사용
오픈소스	높음	공통	Latex2e-translated-notice 라이선스 컴포넌트 사용
오픈소스	높음	공통	Linux-man-pages-1-para 라이선스 컴포넌트 사용
			Linux-man-pages-copyleft-2-para 라이선스 컴포넌트 사

오픈소스	높음	공통	용
오픈소스	높음	공통	Linux-man-pages-copyleft-var 라이선스 컴포넌트 사용
오픈소스	높음	공통	lsof 라이선스 컴포넌트 사용
오픈소스	높음	공통	Lucida-Bitmap-Fonts 라이선스 컴포넌트 사용
오픈소스	높음	공통	Mackerras-3-Clause 라이선스 컴포넌트 사용
오픈소스	높음	공통	Mackerras-3-Clause-acknowledgment 라이선스 컴포넌트 사용
오픈소스	높음	공통	magaz 라이선스 컴포넌트 사용
오픈소스	높음	공통	mailprio 라이선스 컴포넌트 사용
오픈소스	높음	공통	McPhee-slideshow 라이선스 컴포넌트 사용
오픈소스	높음	공통	metamail 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-Festival 라이선스 컴포넌트 사용
오픈소스	높음	공통	MIT-testregex 라이선스 컴포넌트 사용
오픈소스	높음	공통	MMIXware 라이선스 컴포넌트 사용
오픈소스	높음	공통	MPEG-SSG 라이선스 컴포넌트 사용
오픈소스	높음	공통	NIST-Software 라이선스 컴포넌트 사용
오픈소스	높음	공통	OLFL-1.3 라이선스 컴포넌트 사용
오픈소스	높음	공통	OpenSSL-standalone 라이선스 컴포넌트 사용
오픈소스	높음	공통	OpenVision 라이선스 컴포넌트 사용
오픈소스	높음	공통	OPL-UK-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	PADL 라이선스 컴포넌트 사용
오픈소스	높음	공통	Pixar 라이선스 컴포넌트 사용
오픈소스	높음	공통	pnmstitch 라이선스 컴포넌트 사용
오픈소스	높음	공통	python-ldap 라이선스 컴포넌트 사용
오픈소스	높음	공통	radvd 라이선스 컴포넌트 사용
오픈소스	높음	공통	Ruby-pty 라이선스 컴포넌트 사용
오픈소스	높음	공통	SAX-PD-2.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SchemeReport 라이선스 컴포넌트 사용
오픈소스	높음	공통	Sun-PPP 라이선스 컴포넌트 사용
오픈소스	높음	공통	SGI-OpenGL 라이선스 컴포넌트 사용
오픈소스	높음	공통	SGP4 라이선스 컴포넌트 사용
오픈소스	높음	공통	Ubuntu-font-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	SL 라이선스 컴포넌트 사용
오픈소스	높음	공통	softSurfer 라이선스 컴포넌트 사용
오픈소스	높음	공통	Soundex 라이선스 컴포넌트 사용
오픈소스	높음	공통	ssh-keyscan 라이선스 컴포넌트 사용

오픈소스	높음	공통	SSLeay-standalone 라이선스 컴포넌트 사용
오픈소스	높음	공통	swrule 라이선스 컴포넌트 사용
오픈소스	높음	공통	TermReadKey 라이선스 컴포넌트 사용
오픈소스	높음	공통	TGPPL-1.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	TTYP0 라이선스 컴포넌트 사용
오픈소스	높음	공통	ulem 라이선스 컴포넌트 사용
오픈소스	높음	공통	UMich-Merit 라이선스 컴포넌트 사용
오픈소스	높음	공통	Unicode-3.0 라이선스 컴포넌트 사용
오픈소스	높음	공통	UnixCrypt 라이선스 컴포넌트 사용
오픈소스	높음	공통	URT-RLE 라이선스 컴포넌트 사용
오픈소스	높음	공통	Widget-Workshop 라이선스 컴포넌트 사용
오픈소스	높음	공통	X11-swapped 라이선스 컴포넌트 사용
오픈소스	높음	공통	Xdebug-1.03 라이선스 컴포넌트 사용
오픈소스	높음	공통	Xfig 라이선스 컴포넌트 사용
오픈소스	높음	공통	xkeyboard-config-Zinoviev 라이선스 컴포넌트 사용
오픈소스	높음	공통	Zeeff 라이선스 컴포넌트 사용
오픈소스	매우 높음	공통	취약한 컴포넌트 사용