Sparrow Analysis Report

Sparrow Cloud

Summary

Project Name	Demo Project
Analysis ID	432
Analysis Type	URL
Started at	2025-02-06 13:06:01
Ended at	2025-02-06 13:13:57
Analyzing Time	7 m 55 s
Total Issues	28

Risk Level and Issues

Critical	High	Medium	Low	Trivial
1	4	11	4	8

Reference and Issues

Reference Name	Total Issues
.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	0
CWE 660 4.7	0
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	11
OWASP 2021	8
Python 시큐어코딩 가이드 2022	0
Rust ANSSI guide v1.0	0
무기체계 소프트웨어 보안약점 점검 목록	0
방위사업청 코딩규칙	0
소프트웨어 보안약점 진단가이드 2021	1
주요정보통신기반시설 취약점 분석·평가 기준	1

● .NET framework design guideline

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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	0
191 - Integer Underflow (Wrap or Wraparound)	0
192 - Integer Coercion Error	0
197 - Numeric Truncation Error	0
209 - Generation of Error Message Containing Sensitive Information	0
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')	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
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	0
460 - Improper Cleanup on Thrown Exception	0
470 - Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection')	0
476 - NULL Pointer Dereference	0
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	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
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

Reference Chapter	Issues

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)	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
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)	0
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)	0
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)	0
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)

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues

A1-Injection	0
A2-Broken Authentication	0
A3-Sensitive Data Exposure	2
A5-Broken Access Control	1
A6-Security Misconfiguration	8

OWASP 2021

Reference Chapter	Issues
A03 Injection	0
A05 Security Misconfiguration	8
A07 Identification and Authentication Failures	0

Python 시큐어코딩 가이드 2022

Reference Chapter	Issues
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

Reference Chapter	Issues
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

Reference Chapter	Issues
CWE-119	0
CWE-134	0
CWE-170	0
CWE-190	0
CWE-209	0
CWE-22	0
CWE-259	0
CWE-285	0
CWE-306	0
CWE-307	0
CWE-312	0
CWE-319	0
CWE-321	0
CWE-327	0
CWE-330	0
CWE-367	0
CWE-369	0
CWE-390	0
CWE-400	0
CWE-404	0
CWE-415	0
CWE-416	0
CWE-457	0
CWE-467	0
CWE-469	0
CWE-476	0
CWE-489	0
CWE-494	0
CWE-495	0
CWE-496	0
CWE-497	0
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	0
CWE-759	0
CWE-78	0
CWE-89	0
CWE-99	0

● 방위사업청 코딩규칙

Reference Chapter	Issues
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 단항 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 사용하여 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

Reference Chapter	Issues
DNS lookup에 의존한 보안 결정	0
HTTP 응답분할	0
LDAP 삽입	0
Null Pointer 역참조	0
Private 배열에 Public 데이터 할당	0
Public 메소드부터 반환된 Private 배열	0
SQL 삽입	0
XML 삽입	0
경로 조작 및 자원 삽입	0
경쟁조건: 검사시점과 사용시점(TOCTOU)	0
메모리 버퍼 오버플로우	0
무결성 검사없는 코드 다운로드	0
반복된 인증시도 제한 기능 부재	0
보안기능 결정에 사용되는 부적절한 입력값	0
부적절한 XML 외부개체 참조	0

부적절한 예외처리	0
부적절한 인가	0
부적절한 인증서 유효성 검증	0
부적절한 자원 해제	0
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	0
오류 상황 대응 부재	0
오류메시지 정보 노출	0
운영체제 명령어 삽입	0
위험한 형식 파일 업로드	0
잘못된 세션에 의한 데이터 정보 노출	0
적절하지 않은 난수 값 사용	0
적절한 인증없는 중요기능 허용	0
정수형 오버플로우	0
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	0
코드 삽입	0
크로스사이트 스크립트	1
크로스사이트 요청 위조	0
포맷스트링 삽입	0
하드코드된 중요정보	0
해제된 자원 사용	0

● 주요정보통신기반시설 취약점 분석·평가 기준

Reference Chapter	Issues
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	1
파일 다운로드	0

Issue Details

● [Rule Name] Form Tag without CSRF Token (High, common)

The CSRF is a vulnerability that allows an arbitrary user to send HTTP requests to arbitrary addresses with the privileges of an arbitrary user. CSRF is a client-facing attack, like XSS, that works by injecting script into a web page. The attacker can execute arbitrary functions of the web service with the privileges of the user who accessed the page containing the malicious script.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The checker finds that an anti-CSRF token has included to one of the elements.

Analyzing Results

The following elements provoke the CSRF:

```
<form action="/benchmark/BenchmarkTest01660” method="GET” id="
FormBenchmarkTest01660”>
<div><label>Please enter your details:</label></div>
<br /><div><label>Username:</label></div><div><input type="text” id="
username” name="username” /></div><div><label>Password:</label><
/div><div><input type="text” id="password” name="password” value="" /><
/div><div>&#xa0;</div><div><label>Parameter: vector <br /></label>
<input type="text” id="vector” name="vector” value="SafeText” /></div><br
/><div><input type="submit” value="Login” /></div></form>
```

Solution

To prevent the CSRF, add an anti-CSRF token to the FORM element.

The Anti-CSRF token uses "CSRFToken", "anticsrf", or "OWASP_CSRFTOKEN" as a HIDDEN field to the FORM element.

And the Anti-CSRF token is useful for Javascript.

But, for a page with XSS vulnerability, the CSRF cannot be prevented with the token.

● [Rule Name] Missing Content-Security-Policy(CSP) header (Low, common)

The Missing Content-Security-Policy(CSP) header vulnerability is caused by lacking a Content-Security-Policy header in the HTTP response. The Content-Security-Policy(CSP) is built on the Same-Origin-Policy (SOP). But in the CSP, a whitelist of trusted content sources needs to be created, and allows the browser to launch or render resources from these sources only. The attacker might use the vulnerability to attempt to execute the malicious client script on the user browser. This allows the attacker to steal sensitive information or privileges from users, and convince a user to take an unintended action. To resolve the vulnerability, you need to create the Content-Security-Policy header and prevent receiving resources from other untrusted sources.

To fix the missing Content-Security-Policy header vulnerability, you need to add the Content-Security-Policy header on your web application server. To ensure that content within all websites comes only from the same domain, excluding subdomains, use the following

```

Content-Security-Policy: default-src ''self''

```

To include subdomains, use the following

```

Content-Security-Policy: default-src ''self''*.[Hostname of the request being analyzed].

```

In addition, you can also enable resources from trusted domains to be used only on certain web page elements.

Also, if you don''t want to prevent resources from untrusted domains from being used, but only want to be informed that they are used, use the following

```

Content-Security-Policy-Report-Only: policy

```

- OWASP 2017
    - A6-Security Misconfiguration

- OWASP 2021
    - A05 Security Misconfiguration

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

The Missing Content-Security-Policy(CSP) header vulnerability is caused by lacking a Content-Security-Policy header in the HTTP response.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660?
username=sparrow8dast2text4&password=sparrow8dast2text4&vector=SafeText HTTP
/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

The following HTTP response header is sent for the HTTP request.

```
HTTP/1.1 200
Content-Length: 8
Content-Type: text/html;charset=ISO-8859-1
Date: Thu, 06 Feb 2025 04:10:55 GMT
```

As you can see in the HTTP response above, there is no Content-Security-Policy header or Content-Security-Policy-Report-Only header.

Because there is no Content-Security-Policy header, resources in untrusted domains can not be prevented from running or rendered in the browser.

Solution

To address the missing Content-Security-Policy Header vulnerability, users must add a Content-Security-Policy header in the web application server.

Do the following to receive all websites contents only within the same domain, but not on subdomains.

```
Content-Security-Policy: default-src 'self'
```

To include a subdomain, do the following.

```
Content-Security-Policy: default-src 'self'*.125.141.219.118
```

In addition, resources from trusted domains can be set to use on certain HTML elements only.

In addition, if users want to be informed that a resource in an untrusted domain has been used without blocking it, do as following.

```
Content-Security-Policy-Report-Only: policy
```

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The Missing Content-Security-Policy(CSP) header vulnerability is caused by lacking a Content-Security-Policy header in the HTTP response.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

Analyzing Results

The following HTTP response header is sent for the HTTP request.

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Length: 1070
Content-Type: text/html
Date: Thu, 06 Feb 2025 04:10:49 GMT
ETag: W/"1070-1709185617269"
Last-Modified: Thu, 29 Feb 2024 05:46:57 GMT
```

As you can see in the HTTP response above, there is no Content-Security-Policy header or Content-Security-Policy-Report-Only header.

Because there is no Content-Security-Policy header, resources in untrusted domains can not be prevented from running or rendered in the browser.

Solution

To address the missing Content-Security-Policy Header vulnerability, users must add a Content-Security-Policy header in the web application server.

Do the following to receive all websites contents only within the same domain, but not on subdomains.

```
Content-Security-Policy: default-src 'self'
```

To include a subdomain, do the following.

```
Content-Security-Policy: default-src 'self'*.125.141.219.118
```

In addition, resources from trusted domains can be set to use on certain HTML elements only.

In addition, if users want to be informed that a resource in an untrusted domain has been used without blocking it, do as following.

```
Content-Security-Policy-Report-Only: policy
```

● [Rule Name] Missing XSS Protection Header (Medium, common)

The Missing XSS Protection Header vulnerability is caused when the X-XSS-Protection header is missing or is not properly specified in an HTTP response. The X-XSS-Protection header stops loading pages when non-persistent cross-site scripting is detected. After the attacker identifies the vulnerability, he might attempt to carry out the non-persistent XSS attack. As a result, the attacker can steal user information or force the server to run unintended actions. To resolve the vulnerability, setting of the web application needs to be changed: add the X-XSS-Protection header and set its value to 0.

- OWASP 2017
  - A6-Security Misconfiguration

- OWASP 2021
  - A05 Security Misconfiguration

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

The Missing XSS Protection Header Vulnerability is caused by lacking an XSS Protection header in the HTTP response.

An X-XSS-Protection header in the HTTP response indicates the status of the ability to stop page loading when the non-persistent cross-site scripting attack is found during page loading in the Internet Explorer, Chrome, or Safari browser.

A value of 0 in the X-XSS-Protection header indicates that the cross-site scripting filter is not used.

Analyzing Results

No X-XSS-Protection header has been found in the HTTP response.

The HTTP response is shown as follow.

```
HTTP/1.1 200
Content-Length: 8
```

```
Content-Type: text/html;charset=ISO-8859-1
Date: Thu, 06 Feb 2025 04:10:55 GMT
```

Solution

The following is a way to run the X-XSS-Protection header on the web application server.

Apache(.htaccess) :

```
<IfModule mod_headers.c>
Header set X-XSS-Protection ”1; mode=block”
</IfModule>
```

PHP :

```
header(”X-XSS-Protection: 1; mode=block”);
```

Spring framework :

```
<http>
<!-- ... -->
<headers>
<xss-protection block=”true”/>
</headers>
</http>
```

Node.js :

```
app.use(function(req, res, next) {
res.header(’X-XSS-Protection’, 0);
next();
});
```

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The Missing XSS Protection Header Vulnerability is caused by lacking an XSS Protection header in the HTTP response.

An X-XSS-Protection header in the HTTP response indicates the status of the ability to stop page loading when the non-persistent cross-site scripting attack is found during page loading in the Internet Explorer, Chrome, or Safari browser.

A value of 0 in the X-XSS-Protection header indicates that the cross-site scripting filter is not used.

Analyzing Results

No X-XSS-Protection header has been found in the HTTP response.

The HTTP response is shown as follow.

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Length: 1070
Content-Type: text/html
Date: Thu, 06 Feb 2025 04:10:49 GMT
ETag: W/”1070-1709185617269”
Last-Modified: Thu, 29 Feb 2024 05:46:57 GMT
```

Solution

The following is a way to run the X-XSS-Protection header on the web application server.

Apache(.htaccess) :

```
<IfModule mod_headers.c>
Header set X-XSS-Protection ”1; mode=block”
</IfModule>
```

PHP :

```
header(”X-XSS-Protection: 1; mode=block”);
```

Spring framework :

```
<http>
<!-- ... -->
<headers>
<xss-protection block="true"/>
</headers>
</http>
```

Node.js :

```
app.use(function(req, res, next) {
res.header('X-XSS-Protection', 0);
next();
});
```

● [Rule Name] Invalid HTML (Trivial, common)

The Invalid HTML checker finds contents with (X)HTML standard violations in pages. Diverse information can be passed to users due to page handling differences of browsers with (X)HTML standard violations in pages. Comply the (X)HTML standard in pages.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request message has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for (X)HTML document.

A W3C Markup validator analysis has been performed for an HTTP response body.

Analyzing Results

An HTTP response body is an (X)HTML document.

In results of W3C Markup validator analysis, the following ''error'' has occurred.

```
Almost standards mode doctype. Expected <!DOCTYPE html>.
```

Solution

See results of W3C Markup validator analysis to modify an (X)HTML document of the HTTP response body.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request message has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for (X)HTML document.

A W3C Markup validator analysis has been performed for an HTTP response body.

Analyzing Results

An HTTP response body is an (X)HTML document.

In results of W3C Markup validator analysis, the following ''alarm'' has occurred.

```
The type attribute is unnecessary for JavaScript resources.
```

Solution

See results of W3C Markup validator analysis to modify an (X)HTML document of the HTTP response body.

● [Rule Name] Default Language Undisplayed (Trivial, common)

The Default Language Undisplayed checker finds pages without specifying their default languages. Screen readers cannot acknowledge languages to convert into voices or provide pronunciation to them when the default languages are not specified in pages. Specify the default languages of pages.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request has been passed.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for HTML document.

<html> element has been searched in an HTTP response body.

lang attribute has been searched in searched elements.

Analyzing Results

An HTTP response body is an HTML document.

<html> element has been found in an HTTP response body. The XPath of the element is shown as follow.

```
/HTML[1]
```

lang attribute has not been found in the element. Therefore, a default language of HTML document is missing.

Solution

Add lang attribute of <html> element.

● [Rule Name] Missing Label (Trivial, common)

The Missing Label checker finds labels that are not entered to entry forms. Handicapped users can have troubles to enter materials to entry forms when labels are missing. Need to provide labels for entry forms.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request has been passed.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for HTML document.

A required element has been searched in an HTTP response body.

id attribute has been searched in searched elements.

<label> element has been searched in parent elements of searched elements.

Analyzing Results

An HTTP response body is an HTML document.

<input> element required a label has been found in the HTTP response body. An XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]/DIV[3]/INPUT[1]
```

id attribute of the element is shown as follow.

```
username
```

<label> element has not been found in the parents of the elements.

<label> element with the following for attribute has not been found in the HTTP response body.

```
username
```

Therefore, a label is missing in elements unrequired of a label.

Solution

Add <label> element for a parent of <input> element. Or, add <label> element and set for attribute of the element to id attribute of <input> element.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request has been passed.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for HTML document.

A required element has been searched in an HTTP response body.

id attribute has been searched in searched elements.

<label> element has been searched in parent elements of searched elements.

Analyzing Results

An HTTP response body is an HTML document.

<input> element required a label has been found in the HTTP response body. An XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]/DIV[7]/INPUT[1]
```

id attribute of the element is shown as follow.

```
vector
```

<label> element has not been found in the parents of the elements.

<label> element with the following for attribute has not been found in the HTTP response body.

```
vector
```

Therefore, a label is missing in elements unrequired of a label.

Solution

Add <label> element for a parent of <input> element. Or, add <label> element and set for attribute of the element to id attribute of <input> element.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request has been passed.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for HTML document.

A required element has been searched in an HTTP response body.

id attribute has been searched in searched elements.

<label> element has been searched in parent elements of searched elements.

Analyzing Results

An HTTP response body is an HTML document.

<input> element required a label has been found in the HTTP response body. An XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]/DIV[8]/INPUT[1]
```

id attribute has not been found in the elements.

<label> element has not been found in the parents of the elements. Therefore, a label is missing in elements unrequired of a label.

Solution

Add <label> element for a parent of <input> element. Or, add <label> element and set for attribute of the element to id attribute of <input> element.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The following HTTP request has been passed.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

An HTTP response has been received.

The HTTP response body has been checked for HTML document.

A required element has been searched in an HTTP response body.

id attribute has been searched in searched elements.

<label> element has been searched in parent elements of searched elements.

Analyzing Results

An HTTP response body is an HTML document.

<input> element required a label has been found in the HTTP response body. An XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]/DIV[5]/INPUT[1]
```

id attribute of the element is shown as follow.

```
password
```

<label> element has not been found in the parents of the elements.

<label> element with the following for attribute has not been found in the HTTP response body.

```
password
```

Therefore, a label is missing in elements unrequired of a label.

Solution

Add <label> element for a parent of <input> element. Or, add <label> element and set for attribute of the element to id attribute of <input> element.

● [Rule Name] Missing Entry Unresponded (Trivial, common)

The Missing Entry Unresponded checker finds missing entries that does not have proper responses. Without the proper responses, users can have troubles to identify missing entries. Need to provide appropriate responses for missing entries.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

You have accessed to the analysis target page with the following process.

```
URL: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
```

The analysis target page has been checked for HTML document.

<form> element has been searched in an analysis target page.

A fillable element has been searched in child elements of searched <form> elements.

Searched <form> attribute has been set to an empty string.

<input type='submit'> element has been searched in child elements of searched <form> element.

<input type='submit'> event has occurred for searched <form> element.

A server response has been checked after the event.

Analyzing Results

The HTML document is the analysis target page.

<form> element has been found in the analysis target page. The XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]
```

3 fillable element has been found in child elements of the <form> element.

<input type='submit'> element has been found in child elements of the <form> element. The XPath of the element is shown as follow.

```
/HTML[1]/BODY[1]/FORM[1]
```

In results of <form> event on the <input type='submit'> element, DOMs between before and after the event are identical, and no message window has been popped up.

Therefore, its server does not response to a missing input.

Solution

Modify to change a DOM or pop up a message window when an empty string is entered in the <form> element.

● [Rule Name] Use of Unauthorized OPTIONS HTTP Method (Medium, common)

The OPTIONS Method causes a current web server to return a result of the available HTTP method. When the OPTIONS method is allowed in a request, the attacker can make a more effective attack against the web server. The attacker generates a request with the OPTIONS method and sends the request to the server. As a result, the attacker might receive a normal response to the request from the server, and obtain a list of allowed methods as well. The list is highly likely to be used as material for further attacks. To resolve the vulnerability, you need to take actions to forbid unnecessary requests of the OPTIONS method.

- OWASP 2017
  - A3-Sensitive Data Exposure

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

The OPTIONS Method Enable vulnerability is caused by returning a result of an OPTIONS method to unauthorized user.

The OPTIONS Method causes a current web server to return a result of the available method.

The OPTIONS method is a method that are used to find out what kind of methods are supported by the current web server.

To check an unauthorized OPTIONS method, OPTIONS method has been set and sent the HTTP request.

```
OPTIONS http://125.141.219.118:39251/benchmark/BenchmarkTest01660?
username=sparrow8dast2text4&password=sparrow8dast2text4&vector=SafeText HTTP
/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

The HTTP response for the HTTP request is shown as follow.

```
HTTP/1.1 200
Allow: GET, HEAD, POST, TRACE, OPTIONS
Content-Length: 0
Date: Thu, 06 Feb 2025 04:11:03 GMT
```

The OPTIONS request contains the following list of allowed HTTP methods:

```
GET, HEAD, POST, TRACE, OPTIONS
```

Solution

It is recommended not to allow a user to request with an OPTIONS method.

This requires proper server configuration.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The OPTIONS Method Enable vulnerability is caused by returning a result of an OPTIONS method to unauthorized user.

The OPTIONS Method causes a current web server to return a result of the available method.

The OPTIONS method is a method that are used to find out what kind of methods are supported by the current web server.

To check an unauthorized OPTIONS method, OPTIONS method has been set and sent the HTTP request.

```
OPTIONS http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.
1
Accept-Language: en-US
```

```
Empty String
```

Analyzing Results

The HTTP response for the HTTP request is shown as follow.

```
HTTP/1.1 200
Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS
Content-Length: 0
Date: Thu, 06 Feb 2025 04:11:02 GMT
```

The OPTIONS request contains the following list of allowed HTTP methods:

```
GET, HEAD, POST, PUT, DELETE, OPTIONS
```

Solution

It is recommended not to allow a user to request with an OPTIONS method.

This requires proper server configuration.

[Rule Name] Slowloris HTTP DOS (Medium, common)

The Slowloris HTTP DoS vulnerability allows an attacker to send multiple HTTP GET requests that contain incomplete headers, and seize HTTP connections available on the server. The vulnerability is caused by insufficient connection control i.e. the number of concurrent accesses or access timeout. The attacker might remove CRLF from the end of the header, and send the HTTP requests to the server with a interval in order to conduct the attack. When the attack has succeeded, the attacker runs out of the HTTP connections available on the server, causing the server to refuse responses for the requests from normal users.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

An HTTP request to be sent to the server has been configured as follow:

A newline character (" ") to denote the end of header information in an HTTP request has been omitted.

The following HTTP request has been sent to the server.

```
GET /benchmark/BenchmarkTest01660.html HTTP/1.1\r\n
Host: 125.141.219.118:39251\r\n
User-Agent: Mozilla/5.0 (Windows NT x.y; Win64; x64; rv:10.0) Gecko/20100101
Firefox/10.0\r\n
Connection: keep-alive\r\n
```

After waiting for 5000 ms, the undelivered newline character has been successfully transmitted.

Analyzing Results

The HTTP response to an HTTP request sent to the server is shown as follow.

```
HTTP/1.1 200
Accept-Ranges: bytes
ETag: W/”1070-1709185617269”
Last-Modified: Thu, 29 Feb 2024 05:46:57 GMT
Content-Type: text/html
Content-Length: 1070
Date: Thu, 06 Feb 2025 04:13:24 GMT
```

```
<!DOCTYPE html PUBLIC ”-//W3C//DTD HTML 4.01 Transitional//EN” ”http://www.w3.
org/TR/html4/loose.dtd”>
<html>
<head>
<meta http-equiv=”Content-Type” content=”text/html; charset=ISO-8859-1”>
<meta name=”insight-app-sec-validation” content=”5d884ba7-805f-4bee-a23e-
228ae4174950”>
<script src=”js/jq
```

A normal HTTP response has been received from the server.

The analysis found that the request waits for an HTTP request that does not have the last newline character in the header information to complete.

Therefore, the HTTP request can consume network resources in server for at least 5000 ms.

Solution

Ensure a single HTTP request does not occupy the session for too long by adjusting the timeout value for each HTTP request to a reasonable level.

If it''s difficult to set a timeout, web cache software can help prevent traffic failures.

● [Rule Name] Apache Tomcat Example (High, common)

The Apache Tomcat example vulnerability is caused by exposing an JSP sample pages of Apache Tomcat with the Cross-site Scripting vulnerabilities. An attacker attempts to access a directory such as /examples/ where the Apache Tomcat examples are located. This allows the attacker to access the Apahce Tomcat sample pages. As a result, attackers abuses the other critical vulnerabilities such as cross-site scripting on the Apache sample pages. To resolve the vulnerability, you need to block access to the main page of Apahce Tomcat. When an access is required, ensure to allow users with permission to access.

- OWASP 2017
  - A5-Broken Access Control

URL http://125.141.219.118:39251/examples/servlets/servlet/SessionExample

Analysis Method

2.http.tomcat_examples.access.attack.s0.o0

2.http.tomcat_examples.access.attack.s0.o1

2.http.tomcat_examples.access.attack.s0.o2

Analyzing Results

2.http.tomcat_examples.access.result.s0.o0

2.http.tomcat_examples.access.result.s0.o1

2.http.tomcat_examples.access.result.s1.o0

Solution

2.http.tomcat_examples.access.solution.s0.o0

● [Rule Name] Cross-site scripting (Critical, common)

The Cross-site Scripting vulnerability is caused by injecting malicious scripts to a web site. The attacker sends a malicious script to the server via a web element, such as text input inside a form. If the server includes that data in the response without validating it, the malicious script is executed in the browser. As a result, the attacker can steal the user''s information (session, cookies, etc.) or force the server to take unintended actions. To address this, web applications should validate and filter the values they receive from the user. This means avoiding the use of user input data in tag names, attributes, inside <script> tags, inside <style> tags, inside HTML comments, and so on, as well as performing substitutions or encoding for symbols such as ''<'', ''>'', and so on when they are used.

- 소프트웨어 보안약점 진단가이드 2021
    - 크로스사이트 스크립트

- 주요정보통신기반시설 취약점 분석·평가 기준
    - 크로스사이트 스크립팅

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

The Cross-site Scripting (XSS) vulnerability is caused by injecting a malicious script to a website.

If a web application server uses an external input value without validation, an attacker can enter a malicious script to steal information from a web application user or cause the web application to behave inappropriately.

The Non-persistent Cross-site Scripting is caused by display a malicious script entered by an attacker on a web page without further validation.

The following HTTP request with the 135abc<script>alert(1);</script>efg246 attack string has been sent to the vector parameter to check the Non-persistent Cross-site Scripting vulnerability.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660?vector=135abc%
3Cscript%3Ealert(1);%3C/script%
3Eefg246&password=sparrow8dast2text4&username=sparrow8dast2text4 HTTP/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

Non-persistent cross-site scripting returns scripts that were entered in the server, including error messages, search results, and HTTP response messages with other input data.

The following HTTP response message is sent for the HTTP request.

```
HTTP/1.1 200
Content-Length: 38
Content-Type: text/html;charset=ISO-8859-1
Date: Thu, 06 Feb 2025 04:11:03 GMT
135abc<script>alert(1);</script>efg246
```

The HTTP response above shows that it contains the <script>alert(1);</script> attack string.

Solution

The solution is shown as follow.

To eliminate the non-persistent cross-site scripting vulnerability, you must validate and filter the external data that is input from the user.

By default, an external input that entered by a user cannot be used in a tag name or a tag property name, between <script> tags, between <style> tags, between HTML comments, or in an unacceptable location.

Alternatively, an external data input from users must be encoded before using as follow.

Encoding when using external data input from HTML element user

```
& --> &amp;
< --> &lt;
> --> &gt;
" --> &quot;
' --> &#x27;
/ --> &#x2F;
```

Encoding is also required when using external data that is input from a user, such as HTML property values, URL parameters, JavaScript, and CSS values.

Java :

```
String userInput = request.getParameter(”input”);
// ## # ## # ###
if (isValidInput(userInput)) {
// ### ##
} else {
// ## ##
}
boolean isValidInput(String input) {
// ## ## ## ### #### ### #### ## ### ##
return !input.matches(”.*[<>&\”].*”);
}
```

JSP :

```
<%
String userInput = request.getParameter(”input”);
```

```
// ## # ## # ###
if (isValidInput(userInput)) {
// ### ##
} else {
// ## ##
}
%>
<%
boolean isValidInput(String input) {
// ## ## ## ### #### ### #### ## ### ##
return !input.matches(”.*[<>&\”].*”);
}
%>
```

PHP :

```
<?php
$userInput = $_POST['input'];
// ## # ## # ###
if (isValidInput($userInput)) {
// ### ##
} else {
// ## ##
}
function isValidInput($input) {
// ## ## ## ### #### ### #### ## ### ##
return !preg_match("/[<>&\"]/", $input);
}
?>
```

[Rule Name] Missing X-Content-Type-Option (Medium, common)

The Missing X-Content-Type-Option vulnerability is caused when the X-Content-Type-Options header with nosniff flag is not specified in an HTTP response. The nosniff flag prevents MIME type sniffing from happening. The MIME type sniffing is a technique used by web browsers to determine the type of MIME when it does not exist or is set to be invalid. An attacker uses the MIME type sniffing and disguises a non-executable MIME type that contains malicious code into an executable MIME type. Then, the malicious code will be executed in a web application. This allows the attacker to upload the malicious code to the web application, causing the web application to repeatedly execute the code. To resolve the vulnerability, you need to put the X-Content-Type-Options header with the value `nosniff` in the HTTP response.

- OWASP 2017
  - A6-Security Misconfiguration

- OWASP 2021
  - A05 Security Misconfiguration

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

Missing X content type option is an invalid or missing value for the X-Content-Type-Options header in the HTTP response.

If the HTTP response has X-Content-Type-Options and its value is nosniff, then MIME type sniffing is disabled.

The MIME type sniffing indicates that if a MIME type is missing or a client type is incorrectly set, the browser will scan the resource and guess the exact MIME type.

If MIME type sniffing is allowed, an attacker can pass a non-executing MIME type as an execution MIME type.

For example, HTML with cross-site scripting can be uploaded as an image file through MIME type sniffing.

If the value of X-Content-Type-Options is set to nosniff, it will be uploaded to the server only if it is a valid Content-Type.

The valid Content-Type matching is shown as follow.

```
text/css
image/*
application/javascript
application/x-javascript
application/ecmascript
```

```
application/json
text/ecmascript
text/javascript
text/json
```

Analyzing Results

No X-Content-Type-Options header has been found in the HTTP response.

The following header is added to the HTTP response.

```
HTTP/1.1 200
Content-Length: 8
Content-Type: text/html;charset=ISO-8859-1
Date: Thu, 06 Feb 2025 04:10:55 GMT
```

Solution

To avoid vulnerabilities due to MIME type sniffing, the web application server must include an X-Content-Type-Options header with a value of nosniff in the HTTP response.

Alternatively, file uploading to the web application by a user should be prevented.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

Missing X content type option is an invalid or missing value for the X-Content-Type-Options header in the HTTP response.

If the HTTP response has X-Content-Type-Options and its value is nosniff, then MIME type sniffing is disabled.

The MIME type sniffing indicates that if a MIME type is missing or a client type is incorrectly set, the browser will scan the resource and guess the exact MIME type.

If MIME type sniffing is allowed, an attacker can pass a non-executing MIME type as an execution MIME type.

For example, HTML with cross-site scripting can be uploaded as an image file through MIME type sniffing.

If the value of X-Content-Type-Options is set to nosniff, it will be uploaded to the server only if it is a valid Content-Type.

The valid Content-Type matching is shown as follow.

```
text/css
image/*
application/javascript
application/x-javascript
application/ecmascript
application/json
text/ecmascript
text/javascript
text/json
```

Analyzing Results

No X-Content-Type-Options header has been found in the HTTP response.

The following header is added to the HTTP response.

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Length: 1070
Content-Type: text/html
Date: Thu, 06 Feb 2025 04:10:49 GMT
ETag: W/”1070-1709185617269”
Last-Modified: Thu, 29 Feb 2024 05:46:57 GMT
```

Solution

To avoid vulnerabilities due to MIME type sniffing, the web application server must include an X-Content-Type-Options header with a value of nosniff in the HTTP response.

Alternatively, file uploading to the web application by a user should be prevented.

● [Rule Name] Missing X-Frame-Option (High, common)

The Missing X-Frame-Option Vulnerability is caused when the X-Frame-Options header does not exist or sets to an invalid value in an HTTP response. The X-Frame-Options header is included in an HTTP response to set the browser renders the page within frame, iframe, or object tag. An attacker attempts clickjacking by rendering a malicious site in the frame, iframe, or object tag. The clickjacking is a type of attack that tricks a user into clicking on disguised element that is different from what he or she thinks. This allows an attacker to reveal a user''s information or control a user''s computer. To resolve the vulnerability, you need to put the X-Frame-Options header with the correct value in the HTTP response.

- OWASP 2017
  - A6-Security Misconfiguration

- OWASP 2021
  - A05 Security Misconfiguration

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

The Missing X Frame Option Vulnerability is caused by an X-Frame-Options header excluded from an HTTP response.

The X-Frame-Options header is included in an HTTP response to set the browser renders the page within the <frame>, <iframe>, or <object> tags.

Use X-Frame-Options header to prevent click-jacking.

Clicking jacking is an attack method that tricks the user into clicking an embedded code or script that can be executed without the user''s knowledge and knowing that the user is clicking.

This allows an attacker to reveal a user''s information or control a user''s computer.

Analyzing Results

No X-Frame-Options header has been found in the HTTP response.

The following header is added to the HTTP response.

```
HTTP/1.1 200
Content-Length: 8
Content-Type: text/html;charset=ISO-8859-1
Date: Thu, 06 Feb 2025 04:10:55 GMT
```

Solution

The web application server must put an X-Frame-Options header with the correct value in the HTTP response.

The valid value for the X-Frame-Options header is shown as follow.

```
DENY : The page cannot be displayed in a frame, regardless of the site
attempting to do so.
SAMEORIGIN : The page can only be displayed in a frame on the same origin as
the page itself.
ALLOW-FROM uri : The page can only be displayed in a frame on the specified
origin.
```

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

The Missing X Frame Option Vulnerability is caused by an X-Frame-Options header excluded from an HTTP response.

The X-Frame-Options header is included in an HTTP response to set the browser renders the page within the <frame>, <iframe>, or <object> tags.

Use X-Frame-Options header to prevent click-jacking.

Clicking jacking is an attack method that tricks the user into clicking an embedded code or script that can be executed without the user''s knowledge and knowing that the user is clicking.

This allows an attacker to reveal a user''s information or control a user''s computer.

Analyzing Results

No X-Frame-Options header has been found in the HTTP response.

The following header is added to the HTTP response.

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Length: 1070
```

```
Content-Type: text/html
Date: Thu, 06 Feb 2025 04:10:49 GMT
ETag: W/"1070-1709185617269"
Last-Modified: Thu, 29 Feb 2024 05:46:57 GMT
```

Solution

The web application server must put an X-Frame-Options header with the correct value in the HTTP response.

The valid value for the X-Frame-Options header is shown as follow.

```
DENY : The page cannot be displayed in a frame, regardless of the site
attempting to do so.
SAMEORIGIN : The page can only be displayed in a frame on the same origin as
the page itself.
ALLOW-FROM uri : The page can only be displayed in a frame on the specified
origin.
```

● [Rule Name] Cookie attribute check (Secure) (Medium, common)

Web cookies are often a major attack vector for malicious users, so applications should use the Secure attribute to protect them.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

Testing for Cookie Attribute (Secure) checks Set-Cookie header in HTTP response to find out whether Secure attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660?
username=sparrow8dast2text4&password=sparrow8dast2text4&vector=SafeText HTTP
/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

A cookie cannot be set with the Secure in the HTTP protocol: you should change it into HTTPS.

You can use the Set-Cookie in the HTTP response to set cookie attributes or JavaScript helps you to set the attributes.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

Testing for Cookie Attribute (Secure) checks Set-Cookie header in HTTP response to find out whether Secure attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

A cookie cannot be set with the Secure in the HTTP protocol: you should change it into HTTPS.

You can use the Set-Cookie in the HTTP response to set cookie attributes or JavaScript helps you to set the attributes.

[Rule Name] Cookie attribute check (HttpOnly) (Low, common)

Web cookies are often a major attack vector for malicious users, so applications should use secure attributes to protect them.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

Testing for Cookie Attribute (HttpOnly) checks Set-Cookie header in HTTP response to find out whether HttpOnly attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660?
username=sparrow8dast2text4&password=sparrow8dast2text4&vector=SafeText HTTP
/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

You can use the Set-Cookie in the HTTP response to set cookie attributes or JavaScript helps you to set the attributes.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

Testing for Cookie Attribute (HttpOnly) checks Set-Cookie header in HTTP response to find out whether HttpOnly attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

You can use the Set-Cookie in the HTTP response to set cookie attributes or JavaScript helps you to set the attributes.

● [Rule Name] Cookie attribute check (SameSite) (Medium, common)

Web cookies are often a major attack vector for malicious users, so applications should use security attributes to protect them.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660

Analysis Method

Testing for Cookie Attribute (SameSite) checks Set-Cookie header in HTTP response to find out whether SameSite attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660?
username=sparrow8dast2text4&password=sparrow8dast2text4&vector=SafeText HTTP
/1.1
Upgrade-Insecure-Requests: 1
Referer: http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) HeadlessChrome/103.0.5060.134 Safari/537.36
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

It is recommended to set the SameSite to Lax or Strict in order to prevent the CSRF attacks.

Google Chrome updates its version to 80, enhancing the SameSite defaults to Lax from None.

If you need to set it to None, make sure to specify the Secure attribute.

URL http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html

Analysis Method

Testing for Cookie Attribute (SameSite) checks Set-Cookie header in HTTP response to find out whether SameSite attribute is specified or not.

To check this, the following HTTP request has been sent.

```
GET http://125.141.219.118:39251/benchmark/BenchmarkTest01660.html HTTP/1.1
Accept-Language: en-US
```

```
Empty String
```

Analyzing Results

The Set-Cookie does not exist in response of the HTTP request.

Solution

It is recommended to set the SameSite to Lax or Strict in order to prevent the CSRF attacks.

Google Chrome updates its version to 80, enhancing the SameSite defaults to Lax from None.

If you need to set it to None, make sure to specify the Secure attribute.

■ Excluded Issues

No issue has been excluded.

Detection Rules

Type	Risk Level	Language	Name
Web App	High	common	Form Tag without CSRF Token
Web App	Trivial	common	Absolute Path Exposure
Web App	Medium	common	Exposal of Admin Page
Web App	Medium	common	AJP Service Exposure
Web App	High	common	Application Error
Web App	Critical	common	Blind LDAP Injection
Web App	Critical	common	Blind SQL Injection
Web App	Critical	common	Blind XPath Injection
Web App	Low	common	Parameter Query Transformation
Web App	Trivial	common	Broken Skip Link
Web App	High	common	Buffer overflow
Web App	Medium	common	Poor Cache Control
Web App	Medium	common	Certificate Integrity Violation
Web App	Medium	common	Renegotiation of Client-Initiated SSL
Web App	Critical	common	Code Injection
Web App	Critical	common	Command injection
Web App	Low	common	Missing Content-Security-Policy(CSP) header
Web App	Medium	common	Missing Content Type
Web App	Medium	common	Controllable Form Action
Web App	Medium	common	Controllable Reference
Web App	Medium	common	Missing Cookie HttpOnly Flag
Web App	Medium	common	Missing Cookie Secure Flag
Web App	Medium	common	Credit Card Number Exposure
Web App	Critical	common	CRLF Injection
Web App	Medium	common	Cross-domain Script Included
Web App	Trivial	common	Deprecated Component of HTML5
Web App	Critical	common	Directory Listing
Web App	Medium	common	Driver License Number Exposure
Web App	Trivial	common	E-mail Address Exposure
Web App	Critical	common	Unverified Redirection
Web App	Medium	common	HTTP Request Forgery
Web App	Medium	common	Format String Injection

Web App	Medium	common	Missing XSS Protection Header
Web App	Medium	common	Host Header
Web App	Trivial	common	Improper Title
Web App	Trivial	common	Incompatible CSS
Web App	Trivial	common	Incompatible HTML
Web App	Trivial	common	Incompatible Javascript
Web App	Critical	common	Insufficient Session Termination
Web App	High	common	Use of Insufficient Random Value
Web App	High	common	Using Encryption Keys of Insufficient Size
Web App	Trivial	common	Invalid CSS
Web App	Trivial	common	Invalid HTML
Web App	Trivial	common	Invalid Link Text
Web App	Medium	common	JMX/RMI Service Exposure
Web App	Critical	common	Local File Inclusion
Web App	Trivial	common	Missing Substituting Texts
Web App	Trivial	common	Default Language Undisplayed
Web App	Trivial	common	Missing Label
Web App	Medium	common	Mixed Contents
Web App	Trivial	common	Non-standard Techniques
Web App	Trivial	common	Missing Entry Unresponded
Web App	Medium	common	Use of Unauthorized OPTIONS HTTP Method
Web App	High	common	Parameter Tampering
Web App	Medium	common	Passport Number Exposure
Web App	Medium	common	Password Autocomplete
Web App	Critical	common	Path Traversal
Web App	Critical	common	PHF CGI Remote Command Execution
Web App	Critical	common	Predictable Sessions
Web App	Medium	common	Individual Address Exposure
Web App	Critical	common	Remote File Inclusion
Web App	Medium	common	Social Security Number Exposure
Web App	High	common	Sensitive System Information Included in the Comment
Web App	Low	common	Sensitive Information in Error Pages
Web App	Low	common	Sensitive Information in Meta Tag
Web App	High	common	Sensitive Information in Plain Text
Web App	Critical	common	SSI Injection
Web App	Critical	common	Fixed Session

Web App	High	common	Session ID in URL
Web App	Critical	common	Session Reuse
Web App	Medium	common	Slowloris HTTP DOS
Web App	Medium	common	Slow HTTP POST
Web App	Low	common	Snoop Servlet Information Exposure
Web App	Critical	common	SQL Injection
Web App	Medium	common	Temporary File Disclosure
Web App	High	common	Apache Tomcat Example
Web App	Medium	common	Lack of Password Attempt Limit
Web App	High	common	URL Access Control Failure
Web App	Critical	common	Weak Password
Web App	Low	common	Web.xml File Exposure
Web App	Medium	common	Cross-frame Scripting
Web App	Low	common	Xitami Web Server Information Leakage
Web App	Critical	common	XPath Injection
Web App	Critical	common	Cross-site scripting
Web App	Medium	common	X-XXS-Nightmare
Web App	Medium	common	Missing X-Content-Type-Option
Web App	High	common	Missing X-Frame-Option
Web App	Medium	common	Cookie attribute check (Secure)
Web App	Low	common	Cookie attribute check (HttpOnly)
Web App	Medium	common	Cookie attribute check (SameSite)
Web App	High	common	Server information contained in HTTP response headers
Web App	Medium	common	Directory indexing
Web App	High	common	SQL Injection
Web App	High	common	XPath Injection