**웰컴저축은행 필기시험**

- 시큐어 코딩 -

부서 명 :

교육생 명 :

정답

1. (4)

2. (2)

3. (3)

4. (1)

5. (2)

6. (2)

7. (3)

8. (1)

9. (1)

10. (4)

1. 다음 설명에 들어갈 용어들로 알맞게 짝지어진 것은?

|  |
| --- |
| 올바른 입력값 검증을 위해서는 입력 데이터가 프로그램의 정당한 프로그램 입력 영역 안에 있는지 확인하기 이전에 ( A ), ( B ), ( C ) 과정을 거져야 한다. ( A )는 데이터 손실 없이 입력 데이터를 가장 간단하면서 대등한 형태로 축소하는 과정을, ( B )은 데이터 손실은 있지만 알려진 가장 간단한 형태로 변환하는 과정을, ( C )는 데이터를 받은 서브시스템의 요구사항에 맞게 데이터를 가공하는 과정을 말한다. |

1. 정규화, 새니타이즈, 규범화
2. 새니타이즈, 규범화, 정규화
3. 규범화, 새니타이즈, 정규화
4. 규범화, 정규화, 새니타이즈
5. SQL Injection 공격에 사용되는 입력값 유형으로 올바르지 않은 것은?
6. ' or 'a' = 'a
7. " %26 ipconifg -all
8. ' and db\_name() = 1 --
9. 1 and substring(user\_name(),1,1)='a' --
10. 다음 소스 코드에 존재하는 취약점으로 가장 적합한 것은?

|  |
| --- |
| <% String customerID = request.getParameter("id"); %>  :  Customer ID: <%= customerID %>  out.writeln("you searched for: "+request.getParameter("q")); |

1. SQL Injection
2. Command Injection
3. XSS
4. Open Redirect
5. 다음 설명에 해당하는 취약점으로 가장 적합한 것은?

|  |
| --- |
| 사이트가 신뢰하는 사용자(인증된 사용자)를 통해, 공격자가 원하는 명령(공격자에게는 인가되지 않은)을 대신 실행하게 한다.  공격자가 작성해 놓은 REQUEST를 통해 일어나는 악의적인 공격(form문과 JavaScript를 이용)으로 자신의 의도와는 다른 액션이 발생한다.  예를 들면 숨겨둔 공격코드를 적은 게시판 글을 관리자가 누르면 관리자의 권한으로 form문을 실행시켜 어떤 액션을 수행하는 것이다. |

1. CSRF
2. SQL Injection
3. Reflective XSS
4. Stored XSS
5. 다음 설명 중 파일 업로드 취약점의 발생 원인으로 가장 부적절한 것은 무엇입니까?
6. 업로드 파일의 종류(타입)을 제한하지 않았을 때 발생한다.
7. 업로드 파일을 저장할 때 실행 속성을 부여하지 않았을 때 발생한다.
8. 외부에서 접근할 수 있는 경로에 업로드 파일을 저장하는 경우에 발생한다.
9. 업로드 파일의 크기와 개수를 제한하지 않았을 때 발생한다.
10. 다음 소스 코드에 존재하는 취약점으로 가장 접합한 것은?

|  |
| --- |
| :  response.setContentType("text/html");  String author = request.getParameter("authorName");  if (author == null || "".equals(author)) return;  Cookie cookie = new Cookie("replidedAuthor", author);  cookie.setMaxAge(1000);  cookie.setSecure(true);  response.addCookie(cookie);  RequestDispatcher frd = request.getRequestDispatcher("test.jsp");  frd.forward(request, response);  : |

1. Open Redirect
2. HTTP 응답 분할
3. XSS
4. Command Injection
5. 다음 소스 코드에서 발생할 수 있는 취약점에 대한 설명으로 올바르지 않은 것은?

|  |
| --- |
| protected Element createContent(WebSession s) {  :  try {  username = request.getParameter("USERNAME");  password = request.getParameter("PASSWORD");  if (!"webgoat".equals(username) || !password.equals("webgoat")) {  s.setMessage("Invalid username and password entered.");  return (makeLogin(s));  }  } catch (Exception e) {  //s.setMessage(e.getMessage());  //return (makeLogin(s));  }  // authentication is good, show the content  :  } |

1. 요청 파라미터 목록에 "PASSWORD"가 누락되었을 때, 인증을 우회하게 된다.
2. NullPointerException 예외가 발생한 경우, 예외 처리 코드가 누락되어 발생한다.
3. 요청 파라미터를 GET 방식으로 전달했을 때 발생한다.
4. catch 절의 주석을 해제하는 것으로 취약점을 해결할 수 있다.
5. 다음 소스 코드에 존재하는 취약점에 대한 설명으로 올바른 것은?

|  |
| --- |
| public static int cardinality (Object obj, final Collection col) {  int count = 0;  if (col == null) {  return count;  }  Iterator it = col.iterator();  while (it.hasNext()) {  Object elt = it.next();  if ((null == obj && null == elt) || obj.equals(elt)) {  count++;  }  }  return count;  } |

1. obj가 null이고, elt가 null이 아닌 경우에 발생한다.
2. obj, elt 모두 null인 경우에 발생한다.
3. obj, elt 모두 null이 아닌 경우에 발생한다.
4. obj가 null이 아니고, elt가 null인 경우에 발생한다.
5. 다음 소스 코드에 존재하는 취약점으로 가장 적합한 것은?

|  |
| --- |
| <select id="getPerson"  parameterType="string" resultType="org.application.vo.Person">  SELET \* FROM PERSON WHERE NAME=#{name} AND PHONE LIKE '${phone}'  </select> |

1. SQL Injection
2. XPath Injection
3. XQuery Injection
4. XSS
5. 다음 소스 코드에 존재하는 취약점으로 가장 적합한 것은?

|  |
| --- |
| private String[] userRoles;  public void setUserRoles(String[] userRoles) {  this.userRoles = userRoles;  } |

1. Public 메소드로부터 반환되는 Private 배열
2. 잘못된 API 사용
3. Open Redirect
4. Private 배열에 Public 데이터 할당