

수업 내용 정리 보고서

작성일: 2026년 7월 1일

챕터 구성

- 보안의 분류 체계와 현대적 보안 패러다임
- 보안 거버넌스와 컴플라이언스 및 컨설팅 실무
- 취약점 분석·평가 자동화 스크립트 설계 및 기술 규약
- 보안 설정 실무 적용 및 운영 점검 체크리스트

보안의 분류 체계와 현대적 보안 패러다임

보안(Security)은 단순히 해킹을 막는 기술적인 행위를 넘어, 조직의 자산을 보호하기 위한 제도적, 물리적, 기술적 활동을 모두 포괄하는 광범위한 개념입니다. 실무에서는 사고 발생 시 책임 소재를 명확히 하고, 법적/규제적 요구사항을 준수하기 위해 보안을 체계적으로 분류하여 관리합니다.

보안의 세 가지 핵심 영역

보안은 목적과 수행 방식에 따라 크게 **물리적 보안**, **기술적 보안**, **관리적 보안**의 세 가지 영역으로 구분됩니다. 이 세 영역은 서로 독립적인 것이 아니라, 유기적으로 결합되어야 완전한 보안 체계를 구축할 수 있습니다.

- 물리적 보안(Physical Security):** 시설 및 장비에 대한 직접적인 접근을 통제하는 영역입니다. 출입 통제 시스템(지문 인식, 카드 키), CCTV 설치, 침입 탐지 센서 등을 통해 물리적 공간을 보호하는 활동이 이에 해당합니다.
- 기술적 보안(Technical Security):** 정보통신 기술을 이용하여 데이터와 시스템을 보호하는 영역입니다. 우리가 흔히 다루는 모의해킹, 방화벽(Firewall) 설정, 암호화, 침입 탐지 시스템(IDS/IPS) 등이 포함됩니다.
- 관리적 보안(Administrative Security):** 보안을 유지하기 위한 정책, 절차, 조직적 체계를 수립하는 영역입니다. 예를 들어, **ISMS(정보보호 관리체계)**와 같은 인증 기준을 수립하거나, 정기적인 보안 점검 후 보고서를 작성하여 보관하는 등의 활동이 포함됩니다. 이는 '증적 자료(Evidence)'를 남김으로써 보안 사고 발생 시 예방 노력을 증명하는 중요한 근거가 됩니다.

[실무 포인트: 책임과 면책]

보안 사고가 발생했을 때, 담당자가 규정된 보안 점검을 수행했는지, 그리고 그 결과를 보고서로 남겼는지에 따라 법적/행정적 **면책 여부**가 결정됩니다. 따라서 보안 담당자는 보고서 작성 및 서명 시 신중해야 하며, 모든 보안 활동은 반드시 문서화되어야 합니다.

거버넌스와 컴플라이언스(Compliance)

조직이 보안을 운영할 때, '무엇을 기준으로 지킬 것인가'에 대한 답은 거버넌스와 컴플라이언스에서 찾을 수 있습니다.

구분	정의	예시
거버넌스(Governance)	조직의 목표 달성을 위해 보안 정책과 의사결정 체계를 수립하는 상위 개념	법령(시행령), 조직 내부의 보안 내규, 주정통 가이드라인 등
컴플라이언스(Compliance)	수립된 법령이나 정책, 표준을 실제로 준수하는 활동 및 상태	ISMS 인증 취득, 주정통 항목에 따른 정기 보안 점검 수행

주요정보통신기반시설 기술적 취약점 분석·평가방법 상세가이드(이하 주정통)는 국가가 정한 보안의 최소 기준(Guideline)입니다. 주정통을 준수하지 않는다고 해서 즉시 위법이 되는 것은 아니지만, 만약 주정통 가이드를 따르지 않은 상태에서 해킹 사고가 발생한다면 관리 소홀에 대한 책임 소재가 매우 커질 수 있습니다. 반대로, 가이드라인에 따라 성실히 점검하고 조치했다면 사고 발생 시 어느 정도 책임을 경감받을 수 있는 근거가 됩니다.

글로벌 수준에서는 ISO27000 시리즈가 국제 표준 역할을 하며, 국내에서는 KISA(한국인터넷진흥원)를 통해 **ISMS-P**와 같은 인증 제도를 운영하여 기업의 보안 수준을 관리합니다.

현대적 보안 패러다임의 변화: 경계 기반 보안에서 제로 트러스트로

과거와 현재의 보안 전략은 '적(공격자)이 어디에 있는가'에 대한 가정에 따라 큰 차이를 보입니다.

- 경계 기반 보안(Perimeter-based Security):** "성벽(방화벽)만 잘 쌓으면 성 내부의 사람은 안전하다"는 전통적인 방식입니다. 네트워크를 내부와 외부로 엄격히 구분하고, 외부로부터의 침입을 막는 데 집중합니다. 하지만 일단 경계가 뚫려 공격자가 내부 네트워크에 진입하면, 내부 사용자 간의 통신은 신뢰하기 때문에 대규모 피해로 이어질 위험이 큼니다.
- 제로 트러스트(Zero Trust):** "이미 적이 내부에 침입해 있을 수 있다"는 가정 하에 운영되는 현대적 패러다임입니다. 경계의 유무와 상관없이 **"아무도 믿지 말고, 항상 검증하라(Never Trust, Always Verify)"**는 원칙을 따릅니다. 모든 접근 요청에 대해 신원, 기기 상태, 위치 등을 철저히 검증한 후 최소한의 권한만을 부여합니다.

실무자를 위한 보안 점검 및 자동화 가이드

컴플라이언스 준수를 위해 수행하는 기술적 점검은 점차 자동화되는 추세입니다. 특히 주정통과 같은 가이드를 기반으로 점검 스크립트를 작성할 때는 다음과 같은 실무적 원칙을 반드시 준수해야 합니다.

- Read-Only 원칙:** 점검 스크립트는 시스템 설정을 변경(`sed -i, echo >>` 등)하는 명령어를 포함해서는 안 됩니다. 오직 상태를 조회(`cat, grep, ls` 등)하는 명령어만으로 구성하여 운영 중인 서버의 안정성을 보장해야 합니다.
- 표준화된 출력 포맷:** 점검 결과는 통합관리시스템(TMS)에 자동으로 파싱되거나 보고서로 변환될 수 있도록 일관된 포맷을 유지해야 합니다.
 - 구분자 기반: [코드] | [항목명] | [점검결과] | [진단근거]
 - 구조화된 데이터: 시스템 연동을 위한 JSON 또는 XML 포맷
- 결과 분류의 명확성:** KISA 가이드라인에 따라 결과 상태를 **양호**, **취약**, **N/A(해당 없음)**, **검토(수동 확인 필요)** 등으로 명확히 구분해야 합니다.

학습 내용 확인 및 체크리스트

보안 전문가로서 기본 개념을 올바르게 이해했는지 다음 항목을 통해 확인해 보시기 바랍니다.

- [] 보안의 3대 요소(물리/기술/관리)를 구분하고, 각각의 대표적인 예시를 들 수 있는가?
- [] 보안 사고 발생 시 '증적 자료(보고서)'가 왜 중요한지 법적/관리적 관점에서 설명할 수 있는가?
- [] '주정통' 가이드라인 준수가 실무자에게 주는 실질적인 이점(면책 등)을 이해하고 있는가?
- [] 경계 기반 보안과 제로 트러스트의 핵심적인 차이점(가정의 차이)을 설명할 수 있는가?
- [] 보안 점검 스크립트 작성 시 시스템 변경을 금지하는 'Read-Only' 원칙이 왜 필수적인지 알고 있는가?

보안 거버넌스와 컴플라이언스 및 컨설팅 실무

보안은 단순히 해킹을 막는 기술적인 행위를 넘어, 조직의 자산을 보호하기 위한 물리적, 관리적, 기술적 요소가 유기적으로 결합된 체계적인 활동입니다. 본 장에서는 보안의 세 가지 구성 요소와 최신 보안 패러다임인 제로 트러스트(Zero-Trust), 그리고 실무에서 필수적인 컴플라이언스(Compliance)와 기술적 점검 자동화의 원칙에 대해 학습합니다.

1. 보안의 구성 요소와 패러다임의 변화

보안은 크게 세 가지 영역으로 구분됩니다. 효과적인 보안 체계를 구축하기 위해서는 어느 한 분야에 치중하기보다 이 세 가지가 균형을 이루어야 합니다.

- **물리 보안(Physical Security):** 시설 보호, 출입 통제 시스템, CCTV 설치 등 물리적 공간에 대한 접근을 통제하는 영역입니다.
- **기술 보안(Technical Security):** 방화벽, 침입 탐지 시스템(IDS), 암호화, 취약점 점검 등 기술적 수단을 통해 데이터와 시스템을 보호하는 영역입니다.
- **관리 보안(Administrative Security):** 보안 정책 수립, ISMS(정보보호 관리체계) 인증, 정기적인 보안 점검 및 보고서 작성 등 정책적·절차적 관리를 수행하는 영역입니다. 특히 점검 결과에 대한 **증적 자료(Evidence)**를 남기는 것은 보안 사고 발생 시 책임 소재를 명확히 하는 데 매우 중요합니다.

최근 보안 패러다임은 기존의 **경계 기반 보안(Perimeter-based Security)**에서 **제로 트러스트(Zero-Trust)** 모델로 변화하고 있습니다. 과거에는 "성벽(방화벽)만 잘 쌓으면 내부 사용자는 안전하다"고 가정했으나, 제로 트러스트는 "이미 적이 내부 네트워크에 침입해 있을 수 있다"는 전제하에 모든 접근 요청에 대해 철저한 검증을 수행합니다.

2. 보안 거버넌스와 컴플라이언스(Compliance)

기업이 반드시 준수해야 하는 법령과 가이드라인을 지키는 것을 컴플라이언스라고 합니다. 이는 단순히 규칙을 지키는 것을 넘어, 보안 사고 발생 시 조직의 **면책 여부를** 결정짓는 핵심적인 근거가 됩니다.

주요 가이드라인 및 인증 체계

구분	명칭/유형	특징 및 역할
국가 가이드라인	주요정보통신기반시설 기술적 취약점 분석·평가 방법 상세 가이드 (주정통)	국가가 세운 최소한의 보안 기준. 위반 시 법적 처벌보다는 사고 발생 시 관리 책임 소재를 판단하는 척도가 됨.
국제 표준	ISO 27000 Series	국제적으로 통용되는 정보보호 관리체계 표준. 주정통과 유사한 성격을 가짐.
국내 인증	ISMS-P (정보보호 및 개인정보보호 관리체계 인증)	KISA(한국인터넷진흥원)에서 시행하는 인증. 기업이 보안 관리 체계를 잘 갖추었는지 검증함.

실무 포인트: 보안 담당자는 사고 발생 시 "우리는 국가 가이드라인(주정통)이나 국제 표준(ISO 27000)을 준수하며 최선을 다했다"는 것을 증명해야 합니다. 만약 가이드라인을 준수했음에도 사고가 났다면 책임이 경감될 수 있지만, 이를 소홀히 했다면 모든 책임을 뒤집어쓸 수 있으므로 보고서와 증적 자료 관리에 주의해야 합니다.

3. 보안 컨설팅 직무의 이해

보안 컨설팅은 크게 두 가지 영역으로 나뉩니다. 신입 엔지니어가 진입하기에 적합한 분야는 기술적 역량을 바탕으로 하는 인증 컨설팅입니다.

- **인증 컨설팅:** ISMS, ISMS-P 등 기업이 보안 인증을 획득할 수 있도록 가이드하고 점검하는 업무입니다. 기술적 취약점 분석과 관리적 체계 구축을 동시에 다룹니다.
- **영업 컨설팅:** 보안 솔루션이나 서비스를 판매하기 위한 사전 영업 성격의 컨설팅입니다.

4. 기술적 취약점 점검 자동화 실무

수많은 서버와 장비를 일일이 수동으로 점검하는 것은 불가능에 가깝습니다. 따라서 주요정보통신기반시설(주정통) 가이드라인에 따른 점검 항목들은 스크립트를 통해 자동화하는 것이 일반적입니다. 이때 스크립트는 반드시 **안전성**과 **표준화**를 준수해야 합니다.

[중요] 자동화 스크립트 작성 시 5대 필수 규약

점검 스크립트는 운영 중인 시스템에 실행되므로, 시스템에 영향을 주지 않는 것이 최우선입니다.

1. **Read-Only 원칙:** 설정 파일을 수정(`sed -i, echo >>` 등)하는 명령어를 절대 포함하지 않습니다. 오직 상태를 확인하는 조회 명령어(`cat, grep, find` 등)만 사용해야 합니다.
2. **예외 처리 (Error Handling):** 파일이 없거나 명령어가 실패하더라도 스크립트가 중단되지 않아야 합니다. 에러 메시지가 결과에 섞이지 않도록 표준 에러를 리다이렉션 처리합니다.

command 2>/dev/null

3. **환경 의존성 제거 (Absolute Path):** 시스템마다 환경 변수(PATH)가 다를 수 있으므로, 명령어는 가급적 절대 경로를 사용합니다.
예: /bin/grep, /usr/bin/awk
4. **임시 파일 생성 금지:** /tmp 등에 임시 파일을 만들면 보안상 위험하고 관리가 어렵습니다. 파이프(|)와 변수를 활용하고, 불가피하게 파일을 생성할 경우 umask 077을 설정하여 권한을 엄격히 제한합니다.
5. **자원 소모 최소화:** 전체 디스크를 스캔하는 find / 같은 명령어는 서버 과부하를 일으킬 수 있습니다. 대상 경로를 명확히 지정하거나 -maxdepth 옵션으로 탐색 범위를 제한해야 합니다.

점검 결과 출력 및 분류 표준

스크립트의 결과물은 관리자가 보고서를 작성하거나 통합 관리 시스템(TMS)에 업로드하기 용이한 포맷이어야 합니다.

1) 출력 포맷 예시

- 구분자 기반 (Text):** [코드] | [항목명] | [결과] | [근거] 형태
- 구조화 데이터 (JSON):** 시스템 연동 시 사용

```
{
  "check_id": "U-01",
  "title": "root 계정 원격 접속 제한",
  "status": "양호",
  "reason": "PermitRootLogin 옵션이 no로 설정되어 있음"
}
```

2) 점검 결과 상태(Status) 분류

결과 상태	설명
양호 (Pass/Safe)	가이드라인의 기준을 완전히 만족하는 경우
취약 (Fail/Vulnerable)	취약점이 존재하여 조치가 필요한 경우
N/A (Not Applicable)	해당 서비스나 파일이 존재하지 않아 점검이 불필요한 경우
검토 (Manual Review)	자동 판단이 어려워 관리자의 육안 확인이 필요한 경우

5. Windows 보안 템플릿 관리

Windows 환경에서는 로컬 보안 정책을 관리하기 위해 secedit 도구를 사용합니다. 이를 통해 보안 설정을 추출하거나 적용할 수 있습니다.

- 설정 추출 (Export):** 현재 시스템의 보안 설정 내용을 .inf 파일로 저장합니다.
secedit /export /cfg [저장할_절대_경로].inf
- 설정 적용 (Import):** 추출된 템플릿을 시스템에 적용합니다. 단, 적용 후에는 반드시 정책을 강제로 업데이트해야 합니다.
1단계: secedit /import /cfg [파일명].inf
2단계: gpupdate /force (정책 즉시 적용)

학습 요약 및 실무 체크리스트

[실무자 관점 확인 사항]

- [] 보안 사고 발생 시 면책을 위해 **증적 자료(보고서, 로그)**를 체계적으로 관리하고 있는가?
- [] 점검 스크립트 작성 시 **시스템 설정을 변경하는 명령어(Write)**가 포함되어 있지는 않은가?
- [] 스크립트 내 명령어 사용 시 **절대 경로**를 사용하여 환경 의존성을 제거하였는가?
- [] 점검 결과가 **양호/취약/N/A/검토** 중 하나로 명확하게 분류되어 출력되는가?
- [] Windows 보안 정책 적용 후 gpupdate /force를 통해 **정책을 실제 적용**했는가?

취약점 분석·평가 자동화 스크립트 설계 및 기술 규약

보안 사고가 발생했을 때, 담당자가 사전에 점검을 수행했는지 그리고 그 결과를 어떻게 기록(증적)했는지에 따라 책임 소재와 면책 여부가 결정됩니다. 따라서 보안 점검은 단순히 취약점을 찾는 행위를 넘어, **'우리는 기준에 따라 성실히 점검했다'**는 것을 증명하는 과정입니다. 이러한 점검 과정을 효율적이고 일관되게 수행하기 위해 자동화 스크립트 설계가 필수적이며, 이때 반드시 준수해야 할 기술적 규약이 존재합니다.

1. 보안 점검의 배경과 컴플라이언스(Compliance)

보안은 크게 물리적 보안, 기술적 보안, 관리적 보안의 세 가지 영역으로 구분됩니다. 우리가 배우는 스크립트 작성과 취약점 진단은 **기술적 보안**에 해당합니다. 하

지만 기술적 조치만큼 중요한 것이 **관리적 보안**인데, 이는 정책(Policy)을 수립하고 정기적인 점검 보고서를 작성하여 보관하는 것을 의미합니다.

자동화 스크립트가 지향해야 할 기준은 주로 다음과 같은 컴플라이언스 체계를 따릅니다.

- 주요정보통신기반시설(주정통) 기술적 취약점 분석·평가 가이드:** 국가가 세운 최소한의 기술적 보안 가이드라인입니다. 이 가이드를 준수하여 점검을 수행했다는 기록은, 설령 해킹 사고가 발생하더라도 보안 담당자의 책임 소지를 경감(면책)받을 수 있는 중요한 근거가 됩니다.
- ISO27001 / ISMS-P:** 국제 표준 및 국내 인증 체계로, 기업이 보안 관리 체계를 제대로 갖추고 있는지 심사합니다. 자동화 스크립트는 이러한 인증을 위한 '증적 자료'를 생성하는 핵심 도구입니다.

2. 자동화 스크립트의 출력 포맷 설계

스크립트의 목적은 단순히 화면에 결과를 뿌리는 것이 아니라, 생성된 결과를 통합 관리 시스템(TMS)에 업로드하거나 관리자가 보고서를 작성할 수 있도록 **데이터화**하는 데 있습니다. 따라서 다음과 같은 두 가지 방식의 출력 포맷을 설계해야 합니다.

① 구분자(Delimiter) 기반 텍스트 포맷

사람이 읽기 쉽고, 엑셀(Excel)과 같은 도구로 변환하기 용이한 방식입니다. 특정 구분자(| 또는 ::)를 사용하여 항목을 나열합니다.

```
[출력 예시]
U-01 | root 계정 원격 접속 제한 | 양호 | /etc/securetty 파일 존재 및 원격 접속 제한 설정 확인됨.
U-02 | 비밀번호 복잡성 설정 | 취약 | /etc/pam.d/system-auth 파일 내 minlen 설정이 8 미만이거나 누락됨.
```

② 구조화된 데이터 포맷 (JSON/XML)

최근의 보안 솔루션들은 API를 통해 데이터를 주고받으므로, 시스템 연동을 위해 JSON과 같은 구조화된 형식을 선호합니다.

```
{
  "check_id": "U-01",
  "title": "root 계정 원격 접속 제한",
  "status": "양호",
  "reason": "PermitRootLogin 옵션이 no로 설정되어 있음",
  "raw_output": "PermitRootLogin no"
}
```

3. 점검 결과의 분류 기준 (Status)

스크립트의 로직은 KISA 가이드라인에 따라 점검 결과를 명확하게 4가지 상태로 분류하여 출력해야 합니다. 모호한 표현은 추후 데이터 통계 및 대응 시 혼란을 초래합니다.

결과 (Status)	설명
양호 (Pass / Safe)	가이드라인의 취약점 제거 기준을 완전히 만족하는 경우
취약 (Fail / Vulnerable)	취약점이 존재하여 즉시 또는 계획적인 조치가 필요한 경우
N/A (Not Applicable)	해당 서비스나 파일이 시스템에 존재하지 않아 점검이 불필요한 경우
검토 (Manual Review)	스크립트로 자동 판단이 불가능하여 관리자의 육안 확인이 필요한 경우

4. 스크립트 작성 시 5대 기술 규약

자동화 스크립트는 운영 중인 서버(Production Server)에서 **root 권한**으로 실행되는 경우가 많습니다. 작은 실수 하나가 시스템 전체를 중단시키거나 보안을 약화시킬 수 있으므로, 다음의 5가지 원칙을 반드시 준수해야 합니다.

1. 'Read-Only' 원칙 (시스템 변경 금지):

스크립트는 오직 상태를 **조회**하는 용도로만 사용해야 합니다. `cat`, `grep`, `ls`와 같은 조회 명령만 사용하고, `sed -i`(파일 수정)나 `echo >>`(설정 추가)와 같이 시스템 설정을 변경하는 명령어는 절대로 포함해서는 안 됩니다.

2. 예외 처리 및 에러 핸들링 (Error Handling):

특정 파일이 없거나 명령어가 실패하더라도 스크립트 전체가 중단되어서는 안 됩니다. 또한, 에러 메시지가 결과 데이터에 섞여 파싱 오류를 일으키지 않도록 표준 에러를 리다이렉션 처리해야 합니다.

```
# 잘못된 예: 파일이 없으면 에러 메시지가 출력됨
cat /etc/missing_file | grep "admin"

# 올바른 예: 에러를 버리고 결과만 유지 (2>/dev/null)
cat /etc/missing_file 2>/dev/null | grep "admin"
```

3. 환경 의존성 제거 (Absolute Path 사용):

서버마다 PATH 환경 변수가 다를 수 있으므로, 명령어 실행 시 가급적 절대 경로를 사용하여 실행 환경에 관계없이 동일한 동작을 보장해야 합니다.

```
# 권장 방식
/bin/grep "pattern" /etc/passwd
/usr/bin/awk '{print $1}' /etc/passwd
```

4. 임시 파일 생성 지양 및 보안 유지:

데이터 가공을 위해 /tmp 등에 임시 파일을 만드는 방식은 지양하고, 파이프라인(|)과 변수를 활용하십시오. 만약 로그 파일을 생성해야 한다면, umask 077 설정을 통해 타 사용자가 읽지 못하도록 권한을 엄격히 제한해야 합니다.

5. 자원 소모 최소화 (Performance):

전체 디스크를 스캔하는 find /와 같은 명령어는 운영 중인 서버에 과부하를 줄 수 있습니다. 점검 대상 디렉터리를 명확히 지정하거나 -maxdepth 옵션을 사용하여 탐색 범위를 제한해야 합니다.

5. [실무 예시] Windows 보안 정책 점검

윈도우 환경에서는 시스템 설정을 직접 명령어로 조회하기보다, **보안 템플릿(Security Template)**을 활용하여 정책을 관리합니다. secedit 도구를 사용하여 현재 적용된 보안 정책을 파일로 추출하고 분석할 수 있습니다.

```
# 현재 시스템의 보안 설정을 .inf 파일로 추출하는 명령어
secedit /export /cfg C:\temp\security_policy.inf
```

추출된 파일은 보안 가이드라인의 항목(예: 패스워드 복잡성, 계정 잠금 정책 등)과 비교하여 점검을 수행합니다. 만약 정책을 변경하여 적용해야 한다면 /import 명령어를 사용하되, 반드시 gpupdate /force를 통해 정책을 강제 적용해야 함을 유의해야 합니다.

학습 요약 및 실무 체크리스트

- **핵심 개념 확인:** 주정통 가이드는 법적 강제성은 없으나, 사고 발생 시 '면책 근거'로서의 가치가 매우 높다.
- **스크립트 설계 시 주의사항:**
 - 반드시 *조회(Read)* 위주의 명령어로 구성했는가?
 - 에러 메시지 제어를 위해 2>/dev/null을 적절히 사용했는가?
 - 명령어의 절대 경로를 사용하여 환경 의존성을 제거했는가?
- **결과 기록의 중요성:** 점검 결과는 단순 텍스트가 아닌, 파싱이 가능한 구조화된 형태(Delimiter 또는 JSON)로 관리해야 실무적 가치가 있다.
- **실무 포인트:** 윈도우 점검 시 secedit를 통한 정책 추출 및 템플릿 기반의 비교 분석 프로세스를 숙지한다.

보안 설정 실무 적용 및 운영 점검 체크리스트

보안 실무는 단순히 해킹 기술을 익히는 것에 그치지 않습니다. 기술적인 방어뿐만 아니라, 사고 발생 시 법적 책임을 면할 수 있는 **관리적 증거(Evidence)**을 남기고, 국가 및 국제 표준에 따른 **컴플라이언스(Compliance)**를 준수하는 과정이 필수적으로 수반됩니다. 본 장에서는 보안 관리의 3대 영역과 현대적인 보안 패러다임, 그리고 실무에서 가장 중요한 기술적 취약점 점검 자동화 가이드를 다룹니다.

1. 보안 관리의 3대 영역과 책임 소재

보안은 크게 물리적, 기술적, 관리적 보안으로 구분됩니다. 실무자는 각 영역이 어떻게 상호작용하는지 이해해야 합니다.

- **물리적 보안(Physical Security):** 시설 보호 및 침입 통제와 같이 물리적 접근을 차단하는 영역입니다.
- **기술적 보안(Technical Security):** 시스템, 네트워크, 애플리케이션의 취약점을 분석하고 방어 기술을 적용하는 영역으로, 우리가 배우는 해킹 및 방어 기술이 여기에 해당합니다.
- **관리적 보안(Administrative Security):** 보안 정책을 수립하고 이를 준수하고 있음을 증명하는 영역입니다. 대표적으로 **ISMS(정보보호 관리체계)**가 있으며, 정기적인 보안 점검 보고서를 작성하고 보관하는 등의 **증적 자료(Evidence)** 관리가 핵심입니다.

[실무 포인트: 면책과 책임]

보안 사고 발생 시, 담당자가 평상시에 보안 점검을 수행하고 보고서를 작성해 두었는지(증적 여부)에 따라 법적 책임과 면책 여부가 결정됩니다. 따라서 담당자는 점검 결과나 보고서에 서명할 때 매우 신중해야 하며, 모든 작업은 기록으로 남겨야 합니다.

2. 보안 패러다임의 변화: 제로 트러스트(Zero-Trust)

과거의 보안 모델은 성벽(방화벽)을 높게 쌓아 내부 네트워크만 보호하면 안전하다는 **경계 기반 보안(Perimeter-based Security)**이 주를 이루었습니다. 하지만 최근에는 내부자에 의한 위협이나 이미 침입한 공격자를 가정하는 **제로 트러스트(Zero-Trust)** 모델이 중요해지고 있습니다.

제로 트러스트는 "아무도 믿지 않는다"는 원칙 하에, 내부 네트워크에 접속해 있는 사용자나 기기라도 모든 접근 단계에서 철저한 검증을 거치도록 설계됩니다. 이는 클라우드 환경이나 Docker 컨테이너와 같이 경계가 모호한 환경에서 더욱 필수적인 개념입니다. *(참고: 클라우드 OS 이미지나 Docker 이미지는 이미 만들어 진 것을 가져다 쓰기 때문에, 이미지 자체에 포함된 취약점을 인지하지 못할 위험이 있으므로 주의가 필요합니다.)*

3. 컴플라이언스(Compliance)와 거버넌스(Governance)

기업은 법령과 표준에 따라 보안 수준을 유지해야 하며, 이를 준수하는 것을 컴플라이언스라고 합니다.

구분	주요 내용	비고
주요정보통신기반시설(주정통)	국가가 세운 기술적 취약점 분석·평가 가이드라인	거버넌스(Governance) 영역. 준수 시 사고 시 책임 감면 근거가 됨.
ISO27000 Series	정보보호 관리체계에 대한 국제 표준	국제적인 컴플라이언스 기준.
ISMS/PIMS	KISA에서 주관하는 국내 정보보호 인증 제도	인증 심사를 통해 기업의 보안 수준을 공식적으로 인정받음.

최근 보안 이슈가 확대됨에 따라 민간 기업에 대한 점검 범위도 넓어지는 추세이며, 이에 따라 주정통 가이드라인이나 ISMS 인증 컨설팅을 수행할 수 있는 전문 인력에 대한 수요가 높습니다.

4. 기술적 취약점 점검 자동화 실무 (스크립트 작성 규약)

수많은 서버의 설정을 일일이 수동으로 점검하는 것은 불가능하므로, **주정통 가이드라인**을 기반으로 한 자동화 스크립트 작성에 실무의 핵심입니다. 이때 스크립트는 운영 중인 시스템에 영향을 주지 않도록 매우 엄격한 규칙을 따라야 합니다.

① 필수 출력 포맷 (Output Format)

점검 결과는 나중에 엑셀 보고서로 변환하거나 통합 관리 시스템(TMS)에 업로드하기 쉬운 형태여야 합니다.

- 구분자 기반 텍스트: [코드] | [항목명] | [점검결과] | [진단근거] 형식을 권장합니다.
- 구조화된 데이터(JSON): 시스템 연동이 필요한 경우 다음과 같은 형식을 사용합니다.

```
{
  "check_id": "U-01",
  "title": "root 계정 원격 접속 제한",
  "status": "양호",
  "reason": "PermitRootLogin 옵션이 no로 설정되어 있음",
  "raw_output": "PermitRootLogin no"
}
```

② 점검 결과 분류 (Status)

결과값은 반드시 아래 4가지 기준 중 하나로 명확히 정의되어야 합니다.

- 양호 (Pass/Safe):** 가이드라인의 기준을 완전히 만족함.
- 취약 (Fail/Vulnerable):** 취약점이 존재하여 조치가 필요함.
- N/A (Not Applicable):** 해당 서비스나 파일이 없어 점검이 불필요함.
- 검토 (Manual Review):** 자동 판단이 불가능하여 관리자의 육안 확인이 필요함.

③ 스크립트 작성 시 5대 안전 규약 (중요)

스크립트 오류는 운영 서버의 장애로 직결될 수 있으므로 다음 원칙을 반드시 준수합니다.

- Read-Only 원칙:** 오직 상태를 조회하는 명령어(cat, grep, ls 등)만 사용합니다. 설정 파일을 수정하는 명령어(sed -i, echo >> 등)는 절대 금지입니다.
- 예외 처리(Error Handling):** 명령어가 실패하거나 파일이 없어도 스크립트가 중단되지 않아야 합니다. 에러 메시지가 결과에 섞이지 않도록 2>/dev/null 을 활용합니다.
- 환경 의존성 제거:** 시스템마다 환경 변수가 다를 수 있으므로 /bin/grep과 같이 **절대 경로(Absolute Path)**를 사용하는 것이 안전합니다.
- 임시 파일 생성 지양:** /tmp 등에 임시 파일을 만드는 대신 파이프라인(|)과 변수를 활용합니다. 부득이한 경우 umask 077을 설정하여 권한을 제한합니다.
- 자원 소모 최소화:** 전체 디스크를 스캔하는 find / 같은 명령은 시스템 과부하를 일으킵니다. 대상 디렉터리를 명확히 지정하거나 -maxdepth 옵션으로 탐색 깊이를 제한합니다.

5. Windows 보안 설정 및 관리 실무

Windows 환경에서는 **로컬 보안 정책**을 기반으로 한 보안 템플릿을 관리합니다. 이를 위해 **secedit** 명령어를 사용하여 설정을 내보내거나(Export) 가져올(Import) 수 있습니다.

Windows 보안 템플릿 적용 절차:

- 설정 내보내기:** 현재 시스템의 보안 설정을 파일로 저장합니다.

```
secedit /export /cfg C:\temp\security_config.inf
```

- 설정 가져오기:** 수정된 보안 템플릿을 시스템에 적용합니다.

```
secedit /import /cfg C:\temp\security_config.inf
```

3. **설정 활성화:** 가져온 설정이 즉시 반영되도록 구성 및 그룹 정책을 강제 업데이트합니다.

```
secedit /configure /db secedit.sdb /cfg C:\temp\security_config.inf
gpupdate /force
```

실무 적용 체크리스트

보안 점검 및 스크립트 작성 시 다음 항목을 최종 확인하십시오.

- **[] 관리적 증적:** 점검 결과와 보고서가 적절한 권한을 가진 담당자의 승인을 거쳐 보관되고 있는가?
- **[] 스크립트 안전성:** 점검 스크립트에 시스템 설정을 변경하거나 서비스를 재시작하는 명령어가 포함되어 있지는 않은가?
- **[] 경로 및 오류 제어:** 명령어에 절대 경로를 사용하였으며, 에러 발생 시 표준 에러를 적절히 리다이렉션 하였는가?
- **[] 결과 표준화:** 출력 포맷이 사전에 정의된 구분자나 JSON 형식을 따르며, 상태값이 '양호/취약/N/A/검토'로 명확히 구분되는가?
- **[] 리소스 관리:** 점검 명령어가 운영 중인 서비스에 과도한 CPU/Disk I/O 부하를 주지 않는 구조인가?