목차

OpenSCAP에 대한 심층 분석 보고서 3
1. 보안 콘텐츠 자동화 프로토콜(SCAP) 및 OpenSCAP 소개 3
1.1. SCAP이란 무엇인가? 목적과 목표 3
1.2. OpenSCAP 소개: 오픈소스 구현체 3
1.3. 자동화된 보안 규정 준수의 중요성 4
2. SCAP 표준의 핵심 구성 요소 4
2.1. 주요 SCAP 명세 개요 4
2.2. 주요 SCAP 구성 요소 및 설명 표 5
3. OpenSCAP 생태계: 도구 및 아키텍처 6
3.1. OpenSCAP 라이브러리: 핵심 엔진 6
3.2. oscap: 명령줄 인터페이스 6
3.3. SCAP Workbench: 그래픽 사용자 인터페이스 6
3.4. 다른 도구와의 통합 7
4. 실전 가이드: OpenSCAP을 이용한 보안 평가 7
4.1. 다양한 플랫폼에서의 설치 7
4.2. SCAP 보안 가이드(SSG) 콘텐츠 확보 및 이해 8
4.3. 스캔 수행 9
4.4. 보안 정책 맞춤 설정 (Tailoring) 9
5. OpenSCAP을 이용한 규정 준수 자동화 10
5.1. CIS(Center for Internet Security) 벤치마크 준수 10
5.2. DISA STIGs(Security Technical Implementation Guides) 요구 사항 충족 10
5.3. 기타 표준 준수 (예: FISMA, HIPAA, PCI-DSS, NIST SP 800-53) 11
6. OpenSCAP 활용의 이점 및 장점 11
7. 제한 사항 및 고려 사항 12
자원 제약 시스템에서의 메모리 소비 및 성능 12
SCAP 콘텐츠 및 프로파일의 복잡성 12
특정 프로파일의 적용 범위 격차(Coverage Gaps) 12
8. OpenSCAP 비교: 대안 살펴보기 13
8.1. OpenSCAP 대 Nessus 13
8.2. OpenSCAP 대 Qualys VMDR 13
8.3. 비교 개요 표: OpenSCAP, Nessus, Qualys VMDR 14
9. 고급 주제 및 개발 15
9.1. 개발자를 위한 OpenSCAP: 빌드, 디버깅, API 활용 15
9.2. OpenSCAP 프로젝트에 기여하기 15
10. 결론 및 향후 전망 15
10.1. OpenSCAP의 가치 제안 요약 16
10.2. 보안 자동화의 진화하는 환경과 OpenSCAP의 역할 16
참고 자료 16

OpenSCAP에 대한 심층 분석 보고서

본 문서는 Google Gemini로 생성된 보고서입니다. 생성일자 :
2025-06-09

1. 보안 콘텐츠 자동화 프로토콜(SCAP) 및 OpenSCAP 소개

정보기술(IT) 시스템의 복잡성이 증가하고 사이버 위협이 고도화됨에 따라, 조직의 보안 상태를 효과적으로 관리하고 유지하는 것은 매우 중요한 과제가 되었다. 이러한 배경에서 보안 설정 관리, 취약점 평가, 규정 준수 확인 등의 작업을 자동화하려는 노력이 지속되어 왔으며, 그 핵심에 보안 콘텐츠 자동화 프로토콜(SCAP)과 이를 구현한 OpenSCAP이 있다.

1.1. SCAP이란 무엇인가? 목적과 목표

SCAP(Security Content Automation Protocol)은 조직 내 시스템의 자동화된 취약점 관리, 측정 및 정책 준수 평가를 가능하게 하는 특정 표준을 사용하는 방법론이다.[1] SCAP의 주요 목적은 컴퓨터 보안 설정과 미국 국립표준기술연구소(NIST)의 특별 간행물 800-53(SP 800-53) 통제 프레임워크 간의 연계를 표준화하고 자동화하는 것이다.[1] 이를 통해 연방 정보 보안 관리법(FISMA)과 같은 보안 정책 준수를 자동화하고 시스템의 취약점을 지속적으로 모니터링할 수 있도록 지원한다.[1] 미국 정부의 SCAP 콘텐츠 저장소는 NIST가 관리하는 국립 취약점 데이터베이스(NVD)이다.[1] SCAP은 단순히 하나의 기술이 아니라, 다양한 프로토콜과 표준의 집합체로 구성되어 있다. 여기에는 설정 관리, 규정 준수 요구 사항, 소프트웨어 결함 또는 취약점 패치 등을 위한 여러 표준이 포함된다. 이러한 표준들의 집합은 인간과 기계 간에 효율적으로 데이터를 전달하는 수단을 제공하며, 특정 벤더에 종속되지 않는 공동체적 접근 방식을 통해 자동화된 보안 메커니즘 구현을 촉진하는 것을 목표로 한다.[1] SCAP은 소프트웨어 결함 및 보안 관련 설정 문제를 열거하는 데 널리 사용되는 여러 개방형 표준으로 구성되어 있으며, 보안 모니터링 애플리케이션은 이러한 표준을 사용하여 시스템의 취약점을 측정하고 그 결과를 평가하여 잠재적 영향을 판단한다.[1]

궁극적으로 SCAP은 NIST 위험 관리 프레임워크의 구현, 평가 및 모니터링 단계에 기여하며, NIST FISMA 구현 프로젝트의 핵심적인 부분을 형성한다.[1] NIST와 같은 정부 기관이 SCAP 개발 및 보급에 깊이 관여하고 있으며, FISMA와 같은 연방법과의 연계성[1]은 표준화되고 자동화된 사이버 보안 관행을 향한 정부 차원의 강력한 의지를 시사한다. 이러한 배경은 SCAP 기반 관행이 정부 기관과 상호 작용하거나 규제를 받는 조직들에게 사실상의 요구 사항으로 자리매김할 가능성을 내포한다. 또한, SCAP이 "개방형 표준"과 "공동체적 접근 방식[1]"을 강조하는 것은 독점적이고 폐쇄적인 보안 솔루션에서 벗어나려는 전략적 움직임을 반영한다. 이는 보안 자동화를 위한 보다 투명하고 협력적이며 잠재적으로 비용 효율적인 생태계를 조성하여, 강력한 규정 준수 도구에 대한 접근성을 높이는 데 기여한다.

1.2. OpenSCAP 소개: 오픈소스 구현체

OpenSCAP(OSCAP)은 SCAP 표준을 구현한 오픈소스 유틸리티이다.3 이는 SCAP의 구체적인 구현 예시로서, SCAP 표준의 각 구성 요소를 파싱하고 평가할 수 있는 라이브러리와 명령줄 도구(oscap)로 구성된다.[1] OpenSCAP은 주로 SCAP 보안 가이드(SSG)와 같은 보안 가이드라인을 기반으로 시스템의 보안 규정 준수 여부를 테스트하는 데 사용된다.[3] OpenSCAP의 등장은 SCAP의 개방적인 철학이 낳은 직접적인 결과물이며, 오픈소스라는 특성 덕분에 많은 비용을 들이지 않고도 자동화된 보안 점검을 구현하고자 하는 광범위한 사용자층에게 SCAP의 강력한 기능을 제공한다. OpenSCAP과 같은 견고한 오픈소스 구현체

의 가용성은 표준화된 보안 규정 준수 관행을 채택하려는 조직에게 진입 장벽을 크게 낮추는 역할을 한다. 이는 대규모 보안 예산을 가진 조직뿐만 아니라 중소기업이나 예산이 제한된 조직도 정교하고 표준화된 보안 규정 준수 점검을 구현할 수 있게 하여, 다양한 분야에 걸쳐 전반적인 보안 수준 향상에 기여할 수 있다.

1.3. 자동화된 보안 규정 준수의 중요성

오늘날 조직은 보안 위협으로부터 시스템을 보호하기 위해 지속적으로 컴퓨터 시스템과 애플리케이션을 모니터링하고, 소프트웨어의 보안 업그레이드를 통합하며, 설정 업데이트를 배포해야 한다.[1] SCAP은 이러한 취약점 관리, 측정 및 정책 준수 평가를 자동화할 수 있도록 지원한다.[1] 사이버 보안은 본질적으로 정보 관리의 문제이며, SCAP은 데이터의 수집, 이동 및 사용을 위한 표준화된 "배관(plumbing)"을 제공하여 보안 담당자가 데이터 변환이 아닌 문제 해결에 집중할 수 있도록 한다.[6] 이러한 자동화는 PCI DSS, NIST, FedRAMP, FISMA 등과 같은 규제 준수 요구 사항을 충족하는 데 핵심적인 역할을 한다.[6] 자동화된 보안 규정 준수로의 전환은 IT 환경의 복잡성 증가와 취약점 발견 및 악용 속도 가속화에 대한 직접적인 대응이다.[1] 이러한 추세는 자동화를 채택하지 않는 조직이 효과적인 위험 관리에 뒤처질 수 있음을 시사한다. 수동 프로세스는 오늘날 시스템과 위협의 규모와 역동성을 감당하기에 역부족이다. OpenSCAP이 제공하는 효과적인 보안 규정 준수 자동화는 반복적이고 수동적인 점검 작업으로부터 보안 인력을 해방시켜, 위협 탐색, 사고 대응, 전략적 보안 계획과 같은 더 높은 가치의 활동에 집중할 수 있도록 한다.[7]

2. SCAP 표준의 핵심 구성 요소

SCAP은 단일 표준이 아니라 여러 특정 표준들의 집합으로, 각각의 표준은 자동화된 보안 및 규정 준수 프로세스의 특정 측면을 담당한다. 이러한 구성 요소들의 상호 작용을 통해 SCAP은 포괄적인 보안 자동화 프레임워크를 제공한다.

2.1. 주요 SCAP 명세 개요

SCAP을 구성하는 주요 명세들은 다음과 같다[1] :

- XCCDF (Extensible Configuration Checklist Description Format): 보안 점검 목록, 벤치마크 및 설정 지침을 작성하기 위한 XML 기반 명세 언어이다. 보안 규칙과 해당 매개변수를 구조화된 방식으로 정의한다.[1] SCAP 보안 가이드(SSG) 프로파일은 종종 XCCDF 파일 형태로 제공된다.[3]
- OVAL (Open Vulnerability and Assessment Language): 개방적이고 공개적으로 사용 가능한 보안 콘텐츠를 촉진하고, 보안 도구 및 서비스 전반에 걸쳐 이 정보의 표준화된 전송을 목표로 하는 국제 정보 보안 커뮤니티 표준이다. OVAL 정의는 시스템의 특정 상태(예: 취약점 존재 여부, 특정 설정 값)를 테스트하는 데 사용된다.[1]
- CVE (Common Vulnerabilities and Exposures): 취약점 및 기타 정보 보안 노출에 대한 표준화된 이름 목록이다. 취약점에 대한 논의 시 공통된 참조점을 제공한다.[1]
- CPE (Common Platform Enumeration): 소프트웨어 애플리케이션, 운영 체제 및 하드웨어 장치를 명명하고 식별하는 표준화된 방법이다. 취약점 점검 대상을 정확하게 지정하는 데 필수적이다.[1]
- CVSS (Common Vulnerability Scoring System): 소프트웨어 취약점의 특성과 심각도를 전달하기 위한 프레임워크이다. 위험도에 따른 remediation 우선순위 결정에 도움을 준다.[1]

이 외에도 다음과 같은 관련 구성 요소들이 있다[1] :

- CCE (Common Configuration Enumeration): 시스템 설정 문제에 대한 고유 식별자를 제공

한다.
- ARF (Asset Reporting Format): 자산 및 그 관계에 대한 정보를 표현하기 위한 표준 형식이다. oscap 도구는 ARF 형식으로 결과를 생성할 수 있다.[9]
- TMSAD (Trust Model for Security Automation Data): 보안 자동화 데이터의 신뢰 모델을 정의한다.
- AID (Asset Identification): 자산을 고유하게 식별하기 위한 데이터 모델을 명시한다.
- CCSS (Common Configuration Scoring System): 설정 문제의 심각도를 점수화하는 시스템이다.

이러한 구성 요소들은 SCAP의 기본 빌딩 블록이다. 각 구성 요소의 개별적인 역할과 상호 관계를 이해하는 것은 SCAP(및 OpenSCAP)이 보안 평가 및 규정 준수를 자동화하는 방식을 파악하는 데 매우 중요하다. 각 명세는 점검 대상 정의(XCCDF, OVAL)부터 점검 대상 플랫폼 명명(CPE), 발견된 취약점 식별(CVE), 심각도 평가(CVSS)에 이르기까지 보안 자동화 프로세스의 특정 측면을 다룬다. SCAP이 개별적이지만 상호 연결된 구성 요소들의 "집합(suite)"[1] 이라는 점은 모듈성을 촉진한다. 이는 서로 다른 도구들이 특정 영역에 특화되면서도 표준화된 방식으로 데이터를 교환할 수 있게 하며, 이는 독점적이지 않은 생태계를 지향하는 SCAP의 목표 달성에 핵심적이다. 다양한 구성 요소들은 종합적으로 매우 데이터 중심적인 보안 접근 방식을 가능하게 한다. "용어 및 형식"[1]을 표준화함으로써 SCAP은 다양한 소스로부터 보안 데이터를 수집, 집계 및 분석하여 더 나은 정보에 기반한 보안 결정과 측정 가능한 보안 상태 개선을 이끌어낸다.

2.2. 주요 SCAP 구성 요소 및 설명 표

SCAP 생태계의 복잡성을 줄이고 핵심 구성 요소에 대한 이해를 돕기 위해 다음 표를 제시한다. 이 표는 각 주요 SCAP 명세의 전체 이름, 설명 및 SCAP 생태계 내에서의 주요 역할을 요약하여 제공한다.

구성 요소	전체 이름 (Full Name)	설명	SCAP 생태계에서의 주요 역할
XCCDF	Extensible Configuration Checklist Description Format (확장형 설정 점검 목록 기술 형식)	보안 점검 목록, 설정 규칙 및 벤치마크를 지정하기 위한 XML 기반 언어. "무엇을 점검할지"를 구조화한다.	보안 정책 및 기준선을 정의하고 평가 프로세스를 안내한다.
OVAL	Open Vulnerability and Assessment Language (공개 취약점 및 평가 언어)	시스템 상태(예: 취약점, 설정)에 대한 특정하고 기계 판독 가능한 테스트(정의)를 정의하기 위한 XML 기반 언어.	XCCDF에 정의된 개별 규칙에 대한 상세한 "점검 방법" 논리를 제공한다.
CVE	Common Vulnerabilities and Exposures (공통 취약점 및 노출)	공개적으로 알려진 사이버 보안 취약점에 대한 표준화된 식별자.	취약점을 고유하게 명명하여 도구 및 데이터베이스 간 일관된 참조를 가능하게 한다.
CPE	Common Platform Enumeration (공통 플랫폼 열거)	IT 플랫폼(하드웨어, OS, 애플리케이션)에 대한 표준화된 명명 시스템.	대상 시스템 및 소프트웨어를 고유하게 식별하여 점검이 올바르게 적용되도록 보장한다.
CVSS	Common Vulnerability Scoring System (공통 취약점 평가 시스템)	취약점의 심각도를 평가하는 표준화된 방법.	취약점의 잠재적 영향을 정량화하여 remediation 우선순위 결정을 돕는다.
CCE	Common Configuration Enumeration (공통 설정 열거)	시스템 설정 문제(취약점과 구별됨)에 대한 표준화된 식별자.	설정 항목을 고유하게 명명하여 일관된 설정 감사를 용이하게 한다.
ARF	Asset Reporting Format (자산 보고 형식)	자산 정보 및 규정 준수 상태를 포함한 평가 결과를 보고하기 위한 표준화된 형식.	스캔 결과의 일관되고 상호 운용 가능한 보고를 가능하게 한다.

이 표는 SCAP의 수많은 약어와 구성 요소로 인해 혼란스러울 수 있는 사용자에게 명확하고 간결한 참조점을 제공한다. 각 주요 SCAP 명세의 목적을 명확히 하고 이들이 자동화된 보안을 가능하게 하기 위해 어떻게 결합되는지 보여줌으로써, 사용자는 SCAP의 기본 빌딩 블록을 신속하게 파악할 수 있다. 이는 OpenSCAP 도구 및 프로세스에 대한 후속 논의를 훨씬 쉽게 따라갈 수 있도록 하는 기초 지식의 기반이 된다.

3. OpenSCAP 생태계: 도구 및 아키텍처

OpenSCAP은 SCAP 표준을 실제로 구현하고 활용할 수 있도록 다양한 도구와 구성 요소로 이루어진 생태계를 제공한다. 이 생태계의 핵심에는 OpenSCAP 라이브러리가 있으며, 이를 기반으로 명령줄 인터페이스(oscap)와 그래픽 사용자 인터페이스(SCAP Workbench) 등이 개발되어 사용자의 다양한 요구를 충족시킨다.

3.1. OpenSCAP 라이브러리: 핵심 엔진

OpenSCAP은 라이브러리와 명령줄 도구 모두를 대표한다.[5] 이 라이브러리 접근 방식은 기존 파일 구조를 학습하는 데 시간을 소비하는 대신 새로운 SCAP 도구를 신속하게 만들 수 있도록 한다.[5] OpenSCAP 라이브러리는 SCAP Workbench, Red Hat Satellite 통합을 위한 SCAPTimony, OpenSCAP Daemon 등에서 사용되는 핵심 빌딩 블록이다.[5] 개발자는 OpenSCAP Base C API를 활용하여 자신들의 애플리케이션을 개발할 수 있다.[11] 이 라이브러리는 SCAP 콘텐츠를 파싱, 해석 및 평가하는 기본 기능을 제공하며, API 중심적인 특성은 확장성과 통합을 용이하게 한다. C API[11]를 갖춘 핵심 라이브러리를 제공함으로써, OpenSCAP은 더 광범위한 SCAP 호환 도구 생태계 개발을 장려한다. 이는 OpenSCAP이 단순한 최종 사용자 도구 세트가 아니라, 보안 자동화 분야에서 추가적인 혁신을 구축할 수 있는 플랫폼임을 의미한다. API를 통해 다른 소프트웨어가 프로그래밍 방식으로 기능에 접근할 수 있게 되어 개발자들이 SCAP 기능을 자신들의 애플리케이션에 통합하거나 새로운 특수 SCAP 도구를 더 쉽게 만들 수 있기 때문이다. 결과적으로 라이브러리는 SCAP을 중심으로 더 넓고 다양한 생태계를 조성하여 그 범위와 유용성을 확장한다.

3.2. oscap: 명령줄 인터페이스

oscap은 OpenSCAP의 명령줄 도구로, 콘텐츠를 문서로 형식화하거나 이 콘텐츠를 기반으로 시스템을 스캔하도록 설계되었다.[5] 이 도구는 XCCDF 벤치마크와 OVAL 정의를 모두 평가하고 적절한 결과를 생성할 수 있다.[5] SCAP 1.2를 지원하며 SCAP 1.1 및 1.0과의 하위 호환성을 갖는다.[5] 일반적인 사용 사례로는 DISA STIG, NIST의 USGCB, PCI-DSS 또는 Red Hat 보안 대응팀의 콘텐츠 평가 등이 있다.[5] oscap xccdf eval은 평가 수행을 위한 핵심 명령이며[9], oscap info는 보안 정책 내 프로파일을 나열하는 데 사용될 수 있다.[9] 또한 oscap xccdf generate report는 ARF/XML 결과로부터 HTML 보고서를 생성한다.[9] 진단을 위해 oscap에는 상세(verbose) 모드와 디버그 모드도 제공된다.[12] oscap의 명령줄 인터페이스(CLI) 특성은 대규모 자동화된 보안 규정 준수에 있어 OpenSCAP의 역할을 근본적으로 뒷받침한다. GUI는 단일 시스템이나 학습에는 유용하지만, CLI는 스크립팅, CI/CD 파이프라인 통합 및 다수 시스템 전반의 규정 준수 관리에 필수적이다. oscap[5]은 스크립트 작성이 용이하고 자동화된 워크플로우(예: cron 작업, Ansible, CI/CD)에 통합될 수 있으므로, 기업 전체에 걸쳐 OpenSCAP의 규정 준수 검사 기능을 확장하는 핵심 동인이다.

3.3. SCAP Workbench: 그래픽 사용자 인터페이스

SCAP Workbench는 OpenSCAP 스캐너에 대한 간단한 인터페이스를 제공하는 그래픽 도구이다.[9] 사용자는 XCCDF 또는 SDS(Source Data Stream) 파일을 열고 로컬 또는 원격 시스템을 평가할 수 있다.[13] 이 도구는 보안 정책 및 프로파일 선택, 스캔 실행, HTML, ARF 또는 XCCDF 형식으로 결과 저장을 용이하게 한다.[9] SCAP 보안 가이드(SSG)는 종종 SCAP Workbench의 종속성으로 설치되며 Workbench와 통합된다.[9] 또한 프로파일 맞춤 설정(tailoring)을 지원한다.[13] SCAP Workbench는 명령줄 도구에 익숙하지 않거나 탐색 및 단일 시스템 스캔을 위해 시각적 인터페이스를 선호하는 사용자의 진입 장벽을 낮춘다. 학습 및 임시 평가 수행에 매우 유용하다. oscap이라는 CLI와 SCAP Workbench라는 GUI[5]를 모두 제공하는 것은 OpenSCAP을 CLI를 선호하는 숙련된 시스템 관리자부터 사용 편의성과 시각화를 선호할 수 있는 보안 분석가 또는 감사자에 이르기까지 더 광범위한 사용자에게 접근 가능하게 만들려는 노력을 보여준다. 다양한 사용자 요구에 부응함으로써 OpenSCAP은 더 폭넓은 채택을 목표로 한다.

3.4. 다른 도구와의 통합

OpenSCAP 라이브러리는 Red Hat Satellite 통합을 위한 SCAPTimony와 OpenSCAP Daemon에서 사용된다.[5] OpenShift Compliance Operator는 내부적으로 oscap xccdf eval 명령을 사용하여 OpenShift 클러스터 규정 준수를 평가한다.[10] SCAPinoculars 프로젝트는 OpenSCAP ARF 결과를 검사하고 Prometheus 메트릭을 생성하는 데 도움을 줄 수 있다.[10] 또한 OpenSCAP은 Jenkins 및 Ansible과 같은 도구와 통합될 수 있으며[7], OSCAP 출력(XCCDF)은 EM CLI를 사용하여 Enterprise Manager로 가져올 수 있다.[16] 이러한 통합은 OpenSCAP의 유연성과 독립 실행형 도구 세트로서의 역할뿐만 아니라 더 큰 보안 및 관리 프레임워크 내의 구성 요소로서의 역할을 강조한다. 특히 OpenShift Compliance Operator, Jenkins, Ansible[7]과 같은 도구와의 통합은 OpenSCAP이 현대적인 DevOps 및 DevSecOps 관행과 부합함을 보여준다. 이를 통해 보안 규정 준수를 개발 수명 주기의 초기에 포함시키고 배포 파이프라인 내에서 자동화할 수 있다. 이는 OpenSCAP이 자동화된 프로세스의 일부로 자동화된 규정 준수 검사를 활성화함으로써 DevSecOps를 채택하는 조직을 지원하는 데 유리한 위치에 있음을 나타낸다.

4. 실전 가이드: OpenSCAP을 이용한 보안 평가

OpenSCAP을 효과적으로 활용하기 위해서는 설치부터 스캔 수행, 결과 해석 및 정책 맞춤 설정에 이르기까지의 과정을 이해하는 것이 중요하다. 이 섹션에서는 OpenSCAP을 사용한 실제 보안 평가 단계를 안내한다.

4.1. 다양한 플랫폼에서의 설치

OpenSCAP을 사용하기 위한 첫 단계는 시스템에 관련 패키지를 설치하는 것이다. 설치 방법은 운영 체제 배포판에 따라 다를 수 있다. SCAP 보안 가이드(SSG)는 보안 정책 콘텐츠를 제공하므로 함께 설치하는 것이 일반적이다.

- Fedora: dnf install openscap-scanner[5] 명령으로 oscap 도구를 설치할 수 있다. SCAP Workbench를 설치하려면 yum install scap-workbench[9] 명령을 사용하며, 이는 openscap-scanner를 종속성으로 함께 설치하는 경우가 많다. 소스에서 직접 빌드하려면 다양한 개발 패키지가 필요하다.[12]
- RHEL 6/7, CentOS 6/7: yum install openscap-scanner[5] 및 yum install scap-workbench[9] 명령을 사용한다.
- RHEL 8+/CentOS 8+: Fedora와 유사하게 dnf 명령을 사용하며, 소스 빌드를 위한 의존성 패키지 목록이 있다.[12]
- Debian/Ubuntu: oscap 도구를 포함하는 libopenscap8 패키지는 apt-get install

libopenscap8[5] 명령으로 설치할 수 있다. Ubuntu에서 SCAP Workbench 및 SSG를 함께 설치하려면 sudo apt-get install -y openscap-scanner scap-security-guide oscap-anaconda-addon[17]과 같은 명령을 사용할 수 있다. 소스 빌드를 위한 의존성 패키지 목록도 제공된다.[12]

SCAP 보안 가이드(SSG) 설치:

- Fedora: dnf install scap-security-guide.[14]
- RHEL/CentOS: yum install scap-security-guide.[9]
- Debian 10+/Ubuntu 18.04+: apt install ssg-base ssg-debderived ssg-debian ssg-nondebian ssg-applications.[14]

다음 표는 주요 리눅스 배포판에서 OpenSCAP 및 관련 구성 요소를 설치하기 위한 명령어를 요약한 것이다.

배포판	oscap 스캐너 및 SSG 설치 명령어	SCAP Workbench 설치 명령어 (별도인 경우)	비고
Fedora	sudo dnf install openscap-scanner scap-security-guide	sudo dnf install scap-workbench	Workbench는 종종 openscap-scanner를 종속성으로 설치한다.
RHEL/CentOS 7 및 이전	sudo yum install openscap-scanner scap-security-guide	sudo yum install scap-workbench	유사한 종속성 관계를 가진다.
RHEL/CentOS 8 및 이후	sudo dnf install openscap-scanner scap-security-guide	sudo dnf install scap-workbench	
Ubuntu 18.04+	sudo apt update && sudo apt install openscap-scanner scap-security-guide	sudo apt install scap-workbench	Ubuntu의 scap-security-guide는 ssg- 패키지로 분리될 수 있다.[14] ([17]은 scap-security-guide를 직접 언급)
Debian 10+	sudo apt update && sudo apt install libopenscap8 scap-security-guide (또는 [14]에 따른 ssg- 패키지)	sudo apt install scap-workbench	libopenscap8은 oscap을 제공한다.[5] SSG는 특정 ssg-* 패키지가 필요할 수 있다.[14]

이 표는 사용자가 OpenSCAP 및 관련 구성 요소(예: SSG)를 사용하기 전에 설치해야 하는 과정을 단순화하여 제공한다. 주요 배포판에 대한 이러한 명령을 통합한 표는 빠르고 쉬운 참조를 제공하여, 사용자가 배포판별 지침을 찾는 데 소요되는 시간을 절약하고 좌절감을 줄여준다. 이는 가이드를 더욱 실행 가능하게 만들며, 설치를 간단하게 만들어 사용자가 OpenSCAP을 시도하도록 장려하여 보안 개선을 위한 채택 및 사용을 증가시킨다.

4.2. SCAP 보안 가이드(SSG) 콘텐츠 확보 및 이해

SCAP 보안 가이드(SSG)는 SCAP 문서 형태로 작성된 보안 정책으로, 컴퓨터 보안의 여러 영역을 다루며 모범 사례 솔루션, 상세 설명 및 remediation 스크립트를 제공한다.[14] OSCAP은 SSG 프로파일을 보안 규정 준수 테스트의 기준으로 사용할 수 있다.[4] SSG는 종종 SCAP Workbench의 종속성으로 설치되거나 별도로 설치할 수 있다(예: yum install scap-security-guide).[9] Fedora, RHEL, CentOS, Scientific Linux의 기본 정책은 SSG에 의해 제공된다.[9] 설치 후 SSG 콘텐츠는 일반적으로 /usr/share/xml/scap/ssg/content/ 디렉토리에 위치하며[9], 파일은 XCCDF, OVAL 또는 데이터스

트림(종종 *-ds.xml 형태) 형식이다.[3] SSG는 PCI DSS, STIG, USGCB와 같은 지침을 구현한다.[14] SSG 프로젝트가 DISA FSO, NSA, Red Hat과 같은 조직을 포함하는 협력적인 오픈소스 노력[14]이라는 사실은 보안 콘텐츠에 상당한 신뢰성과 품질을 부여한다. 이러한 배경은 사용자가 해당 기준선이 잘 검증되었다고 신뢰할 수 있게 하여 광범위한 채택을 장려한다. 전문가의 의견과 인정된 표준을 기반으로 하기 때문에 사용자는 규정 준수 요구에 SSG 정책을 신뢰하고 채택할 가능성이 더 높다.

4.3. 스캔 수행

4.3.1. 보안 정책 및 프로파일 선택

보안 정책은 기계가 읽을 수 있는 규칙을 포함하며, SSG는 여러 정책을 제공한다.[9] 각 정책에는 특정 보안 기준선에 대한 규칙 및 값의 하위 집합인 여러 프로파일이 있을 수 있다.[9] 사용 가능한 프로파일을 나열하려면 <datastream_file.xml>에 대해 oscap info <datastream_file.xml> 명령을 사용한다(예: oscap info /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml).[9] SCAP Workbench는 프로파일을 선택할 수 있는 콤보 박스를 제공한다.[9] 올바른 정책(예: RHEL 8용)과 프로파일(예: CIS Benchmark Level 1)을 선택하는 것은 의미 있는 스캔의 첫 번째 중요한 단계이다.

4.3.2. 평가를 위한 oscap 실행

oscap을 사용한 평가는 다음 명령 형식을 따른다: oscap xccdf eval --profile <profile_id> --results <results_file.xml> --report <report_file.html> <datastream_file.xml>.[9] 예를 들어 DISA STIG의 경우 oscap xccdf eval --profile selected_profile --results result_file --cpe cpe_dictionary disa_stig_content와 같이 사용될 수 있다.[5] 평가 프로세스는 몇 분 정도 소요될 수 있으며[9], oscap은 OVAL 정의를 직접 평가할 수도 있다: oscap oval eval....[19] 이것이 OpenSCAP 사용의 핵심 작업이며, 명령 구조와 주요 옵션을 명확하게 설명해야 한다.

4.3.3. 보고서 생성 및 해석 (HTML, ARF, XCCDF)

스캔 결과는 다양한 형식으로 저장되고 검토될 수 있다. --results-arf arf.xml 옵션은 결과를 자산 보고 형식(ARF)으로 저장하며[9], --report report.html 옵션은 사람이 읽을 수 있는 HTML 보고서를 생성한다.[9] 결과는 XCCDF 형식으로도 저장될 수 있다.[9] HTML 보고서는 점검 결과에 대한 개요와 상세 정보를 제공하며[9], SCAPinoculars와 같은 도구는 ARF 파일에서 HTML 보고서를 렌더링하고 그룹화할 수 있다.[10] 스캔 결과는 이해하기 쉽고 실행 가능해야 하며, 다양한 보고서 형식은 기계 판독 가능(ARF, XCCDF) 대 인간 판독 가능(HTML) 등 서로 다른 목적을 제공한다. 이러한 출력 형식의 유연성은 다양한 운영 워크플로우에서 OpenSCAP의 유용성을 향상시킨다.

4.4. 보안 정책 맞춤 설정 (Tailoring)

모든 환경에 완벽하게 맞는 단일 보안 프로파일은 없다. 맞춤 설정(Tailoring)은 조직이 표준 기준선을 특정 요구 사항과 위험 수용 수준에 맞게 조정하면서 원래 표준과의 연결을 유지할 수 있도록 한다. SCAP Workbench는 프로파일의 맞춤 설정을 지원하며[13], 사용자는 기존 프로파일을 복제(fork)하고 새 ID를 설정한 다음 규칙을 선택/해제하거나 값을 수정할 수 있다.[15] 변경 사항은 원본 프로파일과의 차이점만 포함하는 “ 사용자 정의 파일 ” 또는 “ 테일러링 파일 ”로 저장할 수 있다.[15] 이 접근 방식은 SCAP 보안 가이드의 업데이트된 버전에 전체 사용자 정의를 다시 수행하지 않고도 사용자 정의 파일을 적용할 수

있다는 장점이 있다.[15] OpenSCAP은 XCCDF 1.1 파일 테일러링을 위한 확장을 지원하지만, 이는 openscap 이외의 스캐너에서는 작동하지 않을 수 있다.[13]

테일러링은 엄격하고 일률적인 보안 표준과 다양한 조직의 고유한 운영 현실 사이의 간극을 메우는 중요한 기능이다. 이는 벤치마크의 정신을 준수하면서 필요한 편차를 수용할 수 있게 한다. 표준 보안 프로파일(예: SSG에서 제공)은 강력한 기준선을 제공하지만[14], 특정 조직의 요구 사항이나 기술적 제약으로 인해 표준 프로파일에서 벗어나야 할 수도 있다 ([15]은 이러한 필요성을 암시). 테일러링은 이러한 프로파일에 대한 문서화되고 통제된 수정을 허용하며[15], 변경 사항을 별도의 사용자 정의 파일로 저장하면 원래 기준선을 보존하고 새 버전에 쉽게 다시 적용할 수 있다.[15] 따라서 테일러링은 조직이 표준화된 보안과 특정 상황 사이에서 실용적이고 유지 관리 가능한 균형을 달성할 수 있도록 하며, 표준을 완전히 포기하거나 규정 준수를 잘못 나타내는 것을 방지한다.

5. OpenSCAP을 이용한 규정 준수 자동화

OpenSCAP의 강력한 기능 중 하나는 다양한 보안 표준 및 규제 요구 사항에 대한 준수 여부를 자동으로 평가하고 보고하는 능력이다. 특히 CIS(Center for Internet Security) 벤치마크와 DISA STIG(Security Technical Implementation Guides)와 같은 널리 인정받는 표준에 대한 자동화된 점검은 많은 조직에게 중요한 가치를 제공한다.

5.1. CIS(Center for Internet Security) 벤치마크 준수

CIS 벤치마크는 시스템을 안전하게 구성하기 위한 합의 기반의 모범 사례 지침으로 널리 인정받고 있다. OpenSCAP은 시스템 설정을 CIS 벤치마크와 같은 지침과 비교하여 PCI DSS, NIST, FedRAMP, FISMA 등과 같은 규제 준수를 충족하는 데 도움을 준다.[6] Red Hat은 RHEL(Red Hat Enterprise Linux) 시스템에서 CIS 규정 준수를 스캔하기 위한 콘텐츠를 포함하는 scap-security-guide 패키지를 제공한다.[18]

관리자는 oscap 도구와 SSG에서 제공하는 특정 CIS 프로파일을 사용하여 규정 준수 보고서를 생성할 수 있다.[18] 더 나아가, oscap 결과로부터 생성된 Ansible 플레이북을 사용하여 CIS 규정 준수를 위한 remediation(수정 조치)을 자동화할 수 있다.[18] OpenSCAP은 Ubuntu 서버에서도 CIS 표준(Level 1 및 Level 2 프로파일)을 배포하는 데 사용될 수 있으며[17], 일부 환경에서는 --remediate 옵션을 통해 자동 수정을 시도할 수도 있지만, 이는 운영 환경 적용 전에 반드시 테스트 환경에서 검증해야 한다.[17]

OpenSCAP이 remediation 플레이북을 생성할 수 있지만[18], 자동화된 remediation의 완전성은 다를 수 있다. 예를 들어, SSG의 RHEL 8 CIS 프로파일은 remediation 후에도 CIS 요구 사항의 100%를 다루지 못하고 약 98%의 규정 준수를 달성할 수 있다.[18] 이는 일부 규칙이 SSG 콘텐츠에 의해 다루어지지 않거나 자동화된 remediation이 불가능하거나 구현되지 않았음을 시사한다. 따라서 사용자는 100% 자동화된 remediation을 가정해서는 안 되며, 보고서를 검토하여 수동 개입 또는 위험 수용이 필요한 나머지 격차를 식별해야 한다.

5.2. DISA STIGs(Security Technical Implementation Guides) 요구 사항 충족

DISA STIG는 미국 국방부(DoD)에서 발행하는 보안 지침으로, 국방 시스템에는 의무적으로 적용되며 다른 보안 의식이 높은 조직에서도 널리 사용된다.[20] OpenSCAP은 이러한 DISA STIG를 평가할 수 있는 기능을 제공한다.[5] SCAP 보안 가이드(SSG)는 STIG를 구현하며, 예를 들어 RHEL 6용 DISA STIG는 SSG 내에 포함된 기준선 중 하나이다.[14] SSG는 DoD STIG에 대한 Red Hat의 업스트림 역할을 한다.[14]

STIG 준수 스캔은 `oscap xccdf eval --profile <STIG_profile>...`과 같은 명령을 사용하여 수행되며[5], STIG 설정은 종종 OpenSCAP/SSG와 함께 `/usr/share/xml/scap/ssg/content` 디렉토리에 사전 번들로 제공된다.[20] STIG remediation을 위한 Ansible 플레이북은 DISA의 추가 자동화 콘텐츠(Supplemental Automation Content)에서 얻을 수 있다.[20]

SSG, Red Hat, DISA FSO 및 NSA 간의 STIG 콘텐츠에 대한 협력[14]은 강력한 시너지를 의미한다. 이는 오픈소스 커뮤니티가 DoD 표준의 엄격함으로부터 혜택을 받고, DoD는 STIG 자동화를 유지하고 개선하기 위해 오픈소스 모델의 민첩성과 광범위한 입력을 활용할 수 있음을 의미한다. 이러한 협력은 STIG 자동화 콘텐츠가 강력하고 널리 사용 가능하며(SSG를 통해) 지속적으로 개선되도록 보장하여 정부 및 비정부 사용자 모두에게 혜택을 준다.

5.3. 기타 표준 준수 (예: FISMA, HIPAA, PCI-DSS, NIST SP 800-53)

OpenSCAP의 활용성은 CIS 및 STIG를 넘어 다양한 규제 및 보안 프레임워크로 확장된다. SCAP은 설정을 NIST SP 800-53 통제 항목과 연결하며[1], FISMA 규정 준수를 자동화하는 데 도움을 준다.[1] 또한 PCI-DSS 준수 여부를 평가하는 데 사용될 수 있으며 ([5], SSG가 PCI DSS를 구현한다고 언급됨[14]), Oracle Linux는 SSG를 기반으로 HIPAA 및 " 표준 시스템 보안 프로파일 " 을 위한 프로파일을 제공한다.[16] SCAP은 NIST 위험 관리 프레임워크에도 기여한다.[1]

OpenSCAP이 FISMA, HIPAA, PCI-DSS, NIST SP 800-53 등 다양한 표준[1] 전반에 걸쳐 유용하게 사용될 수 있는 것은 SCAP 보안 가이드가 기술적 점검(OVAL 정의)을 이러한 다양한 프레임워크의 특정 통제 요구 사항에 매핑하는 능력에 달려 있다. 이 매핑은 SSG가 수행하는 복잡하지만 중요한 작업이다. 다양한 규정 준수 체제에 대한 OpenSCAP의 가치는 SSG 커뮤니티가 기술적 구성을 이러한 다양한 표준의 상위 수준 요구 사항에 정확하게 매핑하는 프로파일을 만들고 유지하는 노력에 크게 의존한다. 이는 표준이 발전함에 따라 이러한 매핑을 최신 상태로 유지하기 위한 지속적인 노력이 필요함을 시사한다.

6. OpenSCAP 활용의 이점 및 장점

OpenSCAP을 도입하고 활용함으로써 조직은 다양한 측면에서 실질적인 이점을 얻을 수 있다. 이러한 이점들은 단순히 개별적인 장점들의 합을 넘어, 상호 시너지를 통해 조직의 전반적인 보안 태세 강화에 기여한다.

- 6.1. 취약점 관리 및 규정 준수 자동화: OpenSCAP은 자동화된 취약점 관리, 측정 및 정책 준수 평가를 가능하게 한다.[1] 감사 프로세스를 자동화하여 시간과 자원을 절약한다.[7] 이는 수동 작업의 부담을 줄이고 일관성을 향상시키며 더 빈번한 점검을 가능하게 하는 핵심적인 이점이다.
- 6.2. 표준화 및 상호 운용성: SCAP은 용어와 형식을 표준화하여 공동체적 접근 방식을 촉진한다.[1] 또한, SCAP은 표준화되고 벤더 중립적인 데이터 이동을 위한 " 배관 " 을 정의하며[6], SCAP 검증 애플리케이션은 상호 운용될 수 있다.[6] 표준화는 결과의 비교 가능성을 보장하고 다양한 도구가 함께 작동할 수 있도록 하여 벤더 종속성을 줄인다.
- 6.3. 비용 효율성 (오픈소스): OpenSCAP은 오픈소스이다.[4] SCAP은 비용 효율적인 솔루션 개발에 도움을 주며 [6], OpenSCAP(SCC + OVAL)과 같은 무료 도구는 비용 때문에 Nessus와 같은 유료 대안 대신 선택되기도 한다.[21] 보안 감사 수행 비용을 절감한다.[22] 무료 오픈소스라는 점은 특히 예산이 제한된 조직에게 OpenSCAP을 매우 매력적으로 만든다.
- 6.4. 향상된 보안 태세: 정기적인 점검과 취약점 탐지는 사이버 공격 및 데이터 유출 위험을 줄인다.[7] 공격자가 악용하기 전에 약점을 식별하는 데 도움을 주며[7], remediation을 위한 로드맵을 제공한다.[8] 궁극적으로 OpenSCAP은 문제를 체계적으로 식별하고 해결을 가능하게 함으로써 보안 개선에 기여한다.
- 6.5. 확장성: 중소기업과 대기업 모두에 적합하다.[7] 특히 CLI 도구인 oscap은 스크립트 기반의 대규모 배포에 핵심적인 역할을 한다. OpenSCAP은 다양한 규모의 환경에 맞게 조정될 수 있다.

- 6.6. 용이한 통합: Jenkins, Ansible과 같은 기존 시스템 및 도구와 원활하게 통합된다.[7] 이는 기존 워크플로우 및 자동화 프레임워크에 쉽게 통합될 수 있도록 한다.

OpenSCAP의 이러한 이점들은 개별적으로 작용하기보다는 서로 시너지 효과를 낸다. 예를 들어, 자동화(6.1)는 자원 절약(비용 효율성의 일부, 6.3)으로 이어지고 더 빈번한 점검을 가능하게 하여 보안 태세를 강화한다(6.4). 표준화(6.2)는 더 쉬운 통합(6.6)과 광범위한 도구 지원에 기여하여 비용 효율성을 더욱 향상시킨다. 자동화되고, 표준화되었으며, 오픈소스인 도구는 본질적으로 수동적이고, 독점적이며, 값비싼 도구보다 보안을 강화하는 데 있어 더 비용 효율적이고 확장 가능하다. 이러한 상호 연결된 이점들은 각 부분의 합보다 더 큰, 강력한 가치 제안을 형성한다.

7. 제한 사항 및 고려 사항

OpenSCAP은 강력한 자동화된 보안 규정 준수 및 취약점 평가 도구이지만, 실제 환경에서 사용할 때 고려해야 할 몇 가지 제한 사항과 과제가 존재한다. 이러한 점들을 인지하고 대비하는 것은 OpenSCAP을 효과적으로 활용하는 데 중요하다.

자원 제약 시스템에서의 메모리 소비 및 성능

OpenSCAP 스캐너는 메모리가 제한된 시스템(예: 2GB 미만 권장)에서 조기에 종료되거나 결과 파일을 생성하지 못할 수 있다.[24] “Out of memory: Killed process (oscap)” 또는 “Memory allocation failed”와 같은 오류 메시지가 발생할 수 있다.[24] 이러한 높은 메모리 소비는 스캔이 완료될 때까지 수집된 모든 결과를 메모리에 저장하는 방식, 프로파일/규칙의 복잡성, 스캔 대상 시스템의 파일/패키지 수 등 여러 요인으로 인해 발생한다.[24]

이에 대한 해결 방법으로는 OSCAP_PROBE_MEMORY_USAGE_RATIO 환경 변수 설정, OSCAP_PROBE_IGNORE_PATHS 환경 변수를 사용하여 특정 경로(예: /home 디렉토리) 제외, 재스캔 전에 remediation 적용, 더 작은 패키지 그룹 사용, oscap 명령에 대한 메모리 제한 설정 등이 있다.[24] 이는 특히 구형 하드웨어나 다수의 경량 가상 머신/컨테이너 환경에서 사용자가 인지해야 할 중요한 실제적 제한 사항이다.

SCAP 콘텐츠 및 프로파일의 복잡성

효과적인 취약점 평가는 기본 컴퓨터 인프라에 대한 상세한 지식을 필요로 한다.[8] SCAP 콘텐츠 자체(XCCDF, OVAL)는 복잡할 수 있으며, 프로파일이 구성되는 방식과 규칙이 평가되는 방식을 이해하는 데는 학습 곡선이 필요하다. 일부 프로파일에 포함된 방대한 수의 규칙은 사용자에게 부담스러울 수 있다 ([24]은 “ 평가되는 규칙의 수와 복잡성 ” 이 메모리에 영향을 미친다고 언급). OpenSCAP은 점검 실행을 자동화하지만, SCAP 콘텐츠 자체를 이해하고, 사용자 정의하며, 문제를 해결하는 것은 어려울 수 있다.

특정 프로파일의 적용 범위 격차(Coverage Gaps)

SSG의 RHEL 8 CIS 프로파일은 remediation 후에도 약 98%의 규정 준수만 달성할 수 있으며, 이는 일부 CIS 요구 사항이 다루어지지 않음을 나타낸다.[18] 보안팀은 일반적으로 100% 달성이 불가능하거나 필요하지 않을 수 있음을 인정하고 “ 성공적인 ” 규정 준수를 위한 임계값을 정의한다[18].

사용자는 특정 SSG 프로파일이 특정 벤치마크에 대해 100% 적용 범위를 제공한다고 가정해서는 안 되며, 결과를 검토하고 격차를 이해하는 것이 중요하다.

이러한 제한 사항들은 OpenSCAP의 설계와 운영 방식에 내재된 특정 트레이드오프를 반영할 수 있다.

예를 들어, 메모리 소비 문제[24]는 수집된 데이터에 대한 철저함이나 메모리 내 처리 속도를 위한 트레이드오프일 수 있다. 스캔이 끝날 때까지 모든 결과를 메모리에 저장하는 것은 복잡한 교차 규칙 분석이나 최종 보고서 생성 속도 향상을 허용할 수 있지만, 스캔 중 최대 메모리 사용량이 높아지는 대가를 치른다. 이는 다른 이점을 가질 수 있는 설계 결정의 결과이며, 일반적인 엔지니어링 트레이드오프를 강조한다. 또한, 프로파일의 적용 범위 격차[18]와 콘텐츠의 복잡성[8]은 SCAP 보안 가이드에 대한 지속적인 커뮤니티 기여와 개발의 필요성을 시사한다. 표준이 발전하고 새로운 시스템 구성이 등장함에 따라 SSG 콘텐츠는 업데이트되고 확장되어야 하며, 이는 이러한 오픈소스 보안 프로젝트의 역동적인 특성을 강조한다. 이는 정적인 솔루션이 아니라 진화하는 솔루션이며, 이러한 한계를 해결하는 것은 SSG 콘텐츠를 개선하기 위한 지속적인 커뮤니티 노력, 버그 보고 및 기여에 달려 있다.

8. OpenSCAP 비교: 대안 살펴보기

OpenSCAP의 가치를 더 명확히 이해하기 위해서는 시장의 다른 주요 보안 도구들과 비교하는 것이 유용하다. 여기서는 널리 알려진 상용 취약점 스캐너인 Nessus와 종합적인 취약점 관리 플랫폼인 Qualys VMDR을 OpenSCAP과 비교 분석한다.

8.1. OpenSCAP 대 Nessus

OpenSCAP과 Nessus는 모두 시스템 보안 평가에 사용되지만, 그 초점, 기술, 비용 모델에서 차이가 있다.

- 초점/기술: Nessus는 자체적인 기술(Nessus Attack Scripting Language 플러그인)을 사용하여 스캔 및 remediation 권장 사항을 제공한다.[21] 반면 OpenSCAP은 SCAP 표준 구현에 중점을 둔다. SCAP은 종종 보고 형식이나 DoD 규정 준수를 위해 사용되며, Nessus는 SCAP 지원이 제한적이다(예: RHEL 8 SCAP 미지원).[21] Tenable(Nessus 개발사) 감사 파일이 Nessus를 통한 SCAP 파일 스캔보다 더 완전한 스캔을 제공할 수 있다는 의견도 있다.[21]
- 보고: 한 사용자는 시스템 강화 후 OpenSCAP에서 96%의 점수를 받았지만 동일 시스템에 대해 Nessus는 72%를 보고했다고 언급했는데[25], 이는 평가 기준이나 측정 대상의 차이를 시사한다. SCAP 스캔은 종종 백분율 점수로 표시되는 설정 감사를 의미한다.[21]
- Remediation: OpenSCAP은 remediation 플레이북(예: CIS용 Ansible)을 생성할 수 있다.[18] Nessus는 remediation 권장 사항을 제공한다.[21]
- 비용 및 UI: OpenSCAP은 무료인 반면, Nessus는 사용자 친화적인 GUI를 갖춘 상용 제품이다.[21] 일부 조직은 비용 때문에 Nessus 대신 OpenSCAP(또는 SCC + OVAL)을 선택한다.[21]
- 사용 사례: SCAP(OpenSCAP)은 DoD 규정 준수에 많이 사용된다.[21] Nessus는 더 광범위한 취약점 스캐너이다.[26]

8.2. OpenSCAP 대 Qualys VMDR

OpenSCAP과 Qualys VMDR은 취약점 관리라는 공통분모를 가지지만, 범위, 제공 모델, 주요 기능에서 뚜렷한 차이를 보인다.

- 초점/기능: OpenSCAP은 기준선 적용 및 감사를 위한 SCAP 표준 구현에 중점을 둔다.[22] Qualys VMDR은 IT, OT, IoT 전반에 걸쳐 위험 기반 우선순위 지정에 중점을 둔 포괄적인 클라우드 기반 취약점 관리, 탐지 및 대응(VMDR) 플랫폼이다.[22] Qualys는 자산 검색, 패치 관리, 웹 스캐닝 등 적용 범위가 더 넓다.[22]
- 배포: 두 솔루션 모두 다양한 배포 옵션(웹, 온프레미스, OS용 에이전트)을 제공하지만[22], Qualys는 강력한 클라우드 중심 모델을 가지고 있다.
- 대상 고객: OpenSCAP은 유연하고 비용 효율적인 SCAP 규정 준수 솔루션이 필요한 관리자/감사자

를 대상으로 한다.[22] Qualys는 IT, OT, IoT 환경 전반에 걸쳐 확장 가능하고 포괄적인 취약점 및 사이버 위험 관리가 필요한 조직을 대상으로 한다.[22]

- 비용: OpenSCAP은 무료이다. Qualys VMDR은 상용 솔루션이다 (가격은 명시되지 않았으나 유료로 암시됨).[22]
- 규정 준수 대 취약점 스캐닝: OpenSCAP은 규정 준수 스캐닝(정책 준수)에 강점이 있다. Qualys는 취약점 스캐닝(악용 가능한 결함)에 강점이 있지만 규정 준수 기능도 수행한다. ([27]은 이러한 스캔 유형을 구분하는 일반적인 표를 제시하며, OpenSCAP은 규정 준수 도구로, Qualys는 취약점 도구로 분류됨).

8.3. 비교 개요 표: OpenSCAP, Nessus, Qualys VMDR

다음 표는 OpenSCAP, Nessus, Qualys VMDR의 주요 특징을 요약하여 비교한 것이다.

특징	OpenSCAP	Nessus (Tenable 제공)	Qualys VMDR
주요 초점	SCAP 기반 규정 준수 감사, 취약점 평가 (OVAL 통해)	광범위한 취약점 스캐닝, 설정 감사, 규정 준수 확인	포괄적인 취약점 관리, 탐지 및 대응 (VMDR), 사이버 위험 정량화
핵심 기술	SCAP 표준(XCCDF, OVAL, CPE, CVE)의 오픈소스 구현	독점 스캐닝 엔진, NASL 플러그인, Tenable 감사 파일	클라우드 플랫폼, 독점 스캐닝 에이전트, 광범위한 취약점 데이터베이스
라이선스	오픈소스 (무료)	상용 (Nessus Pro, Tenable.io 등)	상용 (구독 기반 클라우드 서비스)
SCAP 지원	네이티브 및 포괄적 지원	제한적 지원; SCAP 검사 실행 가능하나 네이티브 도구만큼 완전하지 않을 수 있음	규정 준수 확인 지원, SCAP 정책/결과 가져오기 가능
주요 강점	표준화, 비용 효율성, 특정 벤치마크(SSG 통한 STIG, CIS)에 강력, 스크립트 용이성 (oscap)	광범위한 취약점 커버리지, 사용자 친화적 GUI, 광범위한 플러그인 피드	확장성, 클라우드 네이티브, 전체적인 위험 관점, 광범위한 자산 커버리지 (IT/OT/IoT)
보고	HTML, ARF, XCCDF; 규칙 통과/실패, 규정 준수 점수에 초점	상세 취약점 보고서, 규정 준수 보고서, 사용자 정의 가능	클라우드 기반 대시보드, 추세 분석, 위험 우선순위 보고서, 규정 준수 보기
Remediation	remediation 스크립트 생성 가능 (예: SSG 콘텐츠로부터 Ansible)	remediation 지침 및 권장 사항 제공	우선순위 지정, 패치 관리 통합, 티켓팅 통합
일반적 사용 사례	특정 정부/산업 기준선 자동화된 규정 준수, 로컬 시스템 감사, 맞춤형 도구 체인 통합	네트워크 전반의 취약점 평가, 침투 테스트 지원, 규정 준수 보고	기업 전체의 지속적인 취약점 관리, 클라우드 보안, 위험 관리
사용자 인터페이스	CLI (oscap), GUI (SCAP Workbench)	주로 GUI (Nessus Pro, Tenable.io 콘솔)	웹 기반 클라우드 플랫폼
대상 고객	SCAP 도구가 필요한 시스템 관리자, 보안 엔지니어, 감사자, 예산 제약이 있는 조직	보안 전문가, 침투 테스터, 규정 준수팀	포괄적이고 확장 가능한 VMDR이 필요한 기업, MSSP, 조직

이러한 비교를 통해 OpenSCAP은 오픈소스이며 SCAP 중심이라는 점에서 STIG 및 CIS와 같은 특정 벤치마크에 대한 표준화되고 감사 가능한 규정 준수에 초점을 맞춘 뚜렷한 시장 지위를 확보하고 있음을 알 수 있다. Nessus 및 Qualys와 같은 상용 도구는 더 광범위한 취약점 관리 기능을 제공하며 종종 독점적인 이점을 제공하지만 비용이 발생한다. 이는 OpenSCAP이 모든 시나리오에서 이러한 도구를 직접 대체하는 것이 아니라, 특히 SCAP 준수나 비용이 가장 중요한 경우에 가치 있는 구성 요소임을 시사한다.

또한, OpenSCAP은 SCAP에 특화되어 있지만, Nessus 및 Qualys와 같은 상용 도구들은 점점 더 SCAP 지원 또는 규정 준수 모듈을 추가하고 있다.[21] 이는 포괄적인 플랫폼이 특수 기능을 흡수하려는 시장 추세를 나타낸다. 그러나 OpenSCAP과 같은 특수 도구에서의 SCAP 구현 깊이와 충실도는 범용 플랫폼의 추가 기능 모듈을 능가할 수 있다. 따라서 심층적이고 권위 있는 SCAP 구현이 필요한 사용자는 여전히 OpenSCAP을 선호할 수 있으며, " 충분히 좋은" SCAP 지원을 갖춘 광범위한 플랫폼이 필요한 사용자는 상용 제품군을 선택할 수 있다. 이는 특화된 최상의 도구와 통합 플랫폼 간의 긴장을 강조한다.

9. 고급 주제 및 개발

OpenSCAP은 단순한 사용자 도구를 넘어, 개발자들이 직접 기여하거나 라이브러리를 활용하여 새로운 솔루션을 구축할 수 있는 활발한 오픈소스 프로젝트이다. 이 섹션에서는 OpenSCAP의 개발 측면과 커뮤니티 참여 방법을 다룬다.

9.1. 개발자를 위한 OpenSCAP: 빌드, 디버깅, API 활용

OpenSCAP 프로젝트는 기여, 버그 수정 또는 OpenSCAP 라이브러리 활용을 원하는 개발자를 위한 문서를 제공한다.[12] 리눅스에서 OpenSCAP을 빌드하는 과정에는 cmake 및 make 명령 사용이 포함되며, Fedora, RHEL, Ubuntu/Debian과 같은 주요 배포판에 대한 의존성 패키지 목록이 상세히 안내되어 있다.[12]

디버깅을 위해서는 상세 모드(oscap -v), 디버그 모드 및 다양한 환경 변수(LD_LIBRARY_PATH, OSCAP_SCHEMA_PATH 등)를 활용할 수 있다.[12] OpenSCAP Base C API에 대한 문서도 제공되어[11], 개발자들은 이를 통해 OpenSCAP의 핵심 기능을 자신들의 애플리케이션에 통합할 수 있다. 코드 커버리지 생성에는 lcov, genhtml과 같은 도구가 사용된다.[12]

9.2. OpenSCAP 프로젝트에 기여하기

OpenSCAP은 오픈소스 프로젝트로, 주요 개발은 GitHub에서 이루어진다.[18] 사용자는 GitHub를 통해 이슈를 제출하거나 수정 사항을 직접 기여할 수 있다.[18] 개발자 매뉴얼은 프로젝트에 기여하고자 하는 사람들을 대상으로 작성되었다.[12]

상세한 개발자 문서[12], API 접근성[11], 그리고 명확한 기여 경로[18]의 존재는 건강하고 활동적인 오픈소스 프로젝트의 강력한 지표이다. 이는 OpenSCAP이 정체된 도구가 아니라 적극적으로 유지 관리되고, 진화하며, 커뮤니티 주도의 개선에 열려 있음을 시사한다. 이러한 활발한 개발 모델 덕분에 OpenSCAP은 새로운 보안 과제와 사용자 요구에 지속적으로 적응할 가능성이 높으며, 이는 폐쇄적이거나 비활성 프로젝트에 비해 더 미래 지향적인 선택이 될 수 있음을 의미한다.

10. 결론 및 향후 전망

OpenSCAP은 지난 수년간 보안 자동화 분야에서 중요한 역할을 수행해 왔으며, 앞으로도 그 중요성은 지속될 것으로 예상된다. 이 보고서는 OpenSCAP의 핵심 개념, 구성 요소, 실제 활용 방법, 그리고 다른 도구와의 비교를 통해 그 가치와 잠재력을 조명했다.

10.1. OpenSCAP의 가치 제안 요약

OpenSCAP의 핵심 가치는 표준화되고, 자동화되었으며, 오픈소스 기반의 보안 규정 준수 및 취약점 평가 솔루션을 제공한다는 데 있다. 이는 기술적으로는 SCAP 표준의 충실한 구현을 통해 달성되며, 실질적으로는 다음과 같은 이점을 제공한다:

- 자동화: 반복적이고 오류 발생 가능성이 높은 수동 보안 점검 작업을 자동화하여 효율성을 극대화하고 일관성을 보장한다.
- 표준화: CVE, CPE, XCCDF, OVAL 등 국제적으로 인정받는 표준을 사용하여 보안 정보의 명확성과 상호 운용성을 높인다.
- 비용 효율성: 오픈소스 라이선스로 제공되어 초기 도입 비용 부담이 적으며, 이는 예산이 제한된 조직에게도 강력한 보안 자동화 도구를 사용할 수 있는 기회를 제공한다.
- 투명성 및 커뮤니티 지원: 소스 코드가 공개되어 있어 투명성이 높고, 활발한 커뮤니티를 통해 지속적인 개선과 지원이 이루어진다.

이러한 특징들은 OpenSCAP이 단순히 기술적인 도구를 넘어, 보안 자동화를 민주화하고 더 많은 조직이 체계적인 보안 관리를 수행할 수 있도록 지원하는 중요한 역할을 하고 있음을 보여준다.

10.2. 보안 자동화의 진화하는 환경과 OpenSCAP의 역할

보안 자동화 환경은 DevSecOps, 클라우드 보안 규정 준수, 지속적인 모니터링과 같은 새로운 트렌드에 따라 끊임없이 진화하고 있다. 이러한 변화 속에서 OpenSCAP은 여전히 중요한 역할을 할 것으로 기대된다. 특히 OpenShift Compliance Operator[10]와 같은 컨테이너 환경 및 클라우드 플랫폼과의 통합은 OpenSCAP이 현대적인 IT 인프라의 보안 요구 사항에 부응하고 있음을 보여준다.

향후 OpenSCAP은 컨테이너 이미지 스캐닝, 다양한 클라우드 플랫폼에 대한 지원 확대, SCAP 데이터를 기반으로 한 더욱 정교한 분석 기능 개발 등에서 성장 잠재력을 가지고 있다. 보안이 개발 수명 주기의 초기 단계에 통합되는 "좌측 이동(Shift Left)" 패러다임과 "보안 애즈 코드(Security as Code)" 개념이 확산됨에 따라, OpenSCAP의 역할은 더욱 중요해질 것이다. 보안 정책을 코드로 정의(XCCDF)하고 이를 자동으로 검증하는 OpenSCAP의 기능은 이러한 추세와 완벽하게 부합한다.[1] 이는 OpenSCAP이 반응적인 스캔을 넘어 사전 예방적인 설정 강화 및 검증으로 나아가는 데 기여할 수 있음을 의미한다.

그러나 OpenSCAP과 SCAP 보안 가이드(SSG)와 같은 관련 콘텐츠가 직면한 가장 큰 과제이자 기회는 클라우드, 컨테이너, IoT와 같은 신기술의 빠른 발전과 이러한 플랫폼을 위한 새로운 보안 벤치마크에 발맞추는 것이다. OpenSCAP의 오픈소스 및 커뮤니티 중심적 특성[14]은 이러한 변화에 적응할 수 있는 강점(다수의 기여자를 통한 적응 능력)인 동시에, 새롭고 복잡한 콘텐츠 개발을 위한 자원 봉사 노력이나 후원 기여에 의존해야 하는 잠재적 약점이기도 하다. 따라서 OpenSCAP의 지속적인 관련성은 새로운 기술과 위협에 대한 고품질 SCAP 콘텐츠를 생성하고 유지 관리하는 커뮤니티의 능력에 달려 있을 것이다. 이는 지속적인 자원 확보라는 과제와 함께, 새로운 영역에서 개방형 표준 기반 보안을 선도할 수 있는 기회를 동시에 제시한다.

결론적으로, OpenSCAP은 조직이 복잡한 보안 환경에서 규정 준수를 유지하고 취약성을 관리하는 데 필수적인 도구로 자리매김했으며, 지속적인 발전과 커뮤니티의 노력을 통해 미래의 보안 과제 해결에도 중요한 기여를 할 것으로 기대된다.

참고 자료

- Security Content Automation Protocol - Wikipedia, 6월 9, 2025에 액세스, https://en.wikipedia.org/wiki/Security_Content_Automation_Protocol
- en.wikipedia.org, 6월 9, 2025에 액세스,

https://en.wikipedia.org/wiki/Security_Content_Automation_Protocol#:~:text=Security%20Content%20Automation%20Protocol%20(SCAP)%20checklists%20standardize%20and%20enable%20automation,800%2D53)%20controls%20framework.
- About SCAP - Oracle Help Center, 6월 9, 2025에 액세스, https://docs.oracle.com/en/operating-systems/oracle-linux/8/oscap/oscap-WhatIsSCAP.html
- docs.oracle.com, 6월 9, 2025에 액세스, https://docs.oracle.com/en/operating-systems/oracle-linux/8/oscap/oscap-WhatIsSCAP.html#:~:text=OpenSCAP%20(OSCAP)%20is%20an%20open,Linux%20to%20automate%20compliance%20testing.
- OpenSCAP Base | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/tools/openscap-base/
- Secure Configurations and the Power of SCAP, 6월 9, 2025에 액세스, https://www.cisecurity.org/insights/blog/secure-configurations-and-the-power-of-scap
- OpenSCAP: An effective Security and Compliance Audit ... - mivocloud, 6월 9, 2025에 액세스, https://mivocloud.com/blog/OpenSCAP-An-effective-Security-and-Compliance-Audit-solution
- Vulnerability Assessment | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/features/vulnerability-assessment/
- Getting Started | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/getting-started/
- How to visualize your OpenSCAP compliance reports - Red Hat Developer, 6월 9, 2025에 액세스, https://developers.redhat.com/articles/2024/02/08/how-visualize-your-openscap-compliance-reports
- Documentation | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/resources/documentation/
- openscap/docs/developer/developer.adoc at main - GitHub, 6월 9, 2025에 액세스, https://github.com/OpenSCAP/openscap/blob/main/docs/developer/developer.adoc
- SCAP Workbench User Manual, 6월 9, 2025에 액세스, https://static.open-scap.org/scap-workbench-1.1/
- SCAP Security Guide | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/security-policies/scap-security-guide/
- Customizing SCAP Security Guide for your use-case | OpenSCAP portal, 6월 9, 2025에 액세스, https://www.open-scap.org/resources/documentation/customizing-scap-security-guide-for-your-use-case/
- 24 SCAP Supported Standards - Oracle Help Center, 6월 9, 2025에 액세스, https://docs.oracle.com/en/enterprise-manager/cloud-control/enterprise-manager-cloud-control/13.5/emdbc/scap-supported-standards.html
- Automating CIS Compliance With OpenScap on Ubuntu Server - Eric Morrish, 6월 9, 2025에 액세스, https://ericmorrish.com/blog/automating-cis-compliance-with-openscap-on-ubuntu-server/
- Center for Internet Security (CIS) compliance in Red Hat Enterprise ..., 6월 9, 2025에 액세스, https://www.redhat.com/en/blog/center-internet-security-cis-compliance-red-hat-enterprise-linux-using-openscap
- Module : OpenSCAP: Using the oscap Command-Line Interface - Learn Oracle, 6월 9, 2025에 액세스, https://learn.oracle.com/ols/module/openscap-using-the-oscap-command-line-interface/37001/37155
- Efficiently automating DISA STIGS: a guide to automated STIG hardening with RKE2, 6월 9, 2025에 액세스, https://www.glasswall.com/blog/efficiently-automating-disa-stigs-a-guide-to-automated-stig-hardening-with-rke2
- Vulnerability management and SCAP : r/cybersecurity - Reddit, 6월 9, 2025에 액세스, https://www.reddit.com/r/cybersecurity/comments/1bm8pop/vulnerability_management_and_sc

ap/
- Compare OpenSCAP vs. Qualys VMDR in 2025 - Slashdot, 6월 9, 2025에 액세스, https://slashdot.org/software/comparison/OpenSCAP-vs-Qualys-VM/
- OpenSCAP vs. Qualys VMDR Comparison - SourceForge, 6월 9, 2025에 액세스, https://sourceforge.net/software/compare/OpenSCAP-vs-Qualys-VM/
- OpenSCAP memory-consumption problems - Red Hat Customer Portal, 6월 9, 2025에 액세스, https://access.redhat.com/articles/6999111
- Openscap gives better score than the nessus - Login, 6월 9, 2025에 액세스, https://tenable.my.site.com/s/question/0D5WP00000OWKOI0A5/openscap-gives-better-score-than-the-nessus?language=en_US
- 10 Vulnerability Scanning Tools to Know in 2025 - Pynt, 6월 9, 2025에 액세스, https://www.pynt.io/learning-hub/application-security/10-vulnerability-scanning-tools-to-know-in-2025
- Compliance Scan vs Vulnerability Scan: Key Differences - Invicti, 6월 9, 2025에 액세스, https://www.invicti.com/blog/web-security/compliance-scan-vs-vulnerability-scan/

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/openscap_%EA%B0%9C%EC%9A%94

Last update: **2025/06/09 03:57**