Software Security Research 릴리스 발표

Fortify 소프트웨어 보안 콘텐츠

2022 업데이트 4

2022 년 12 월 16 일 금요일

CyberRes Fortify Software Security Research 정보

Fortify Software Security Research 팀은 최첨단 연구를 Fortify Static Code Analyzer(SCA) 및 Fortify WebInspect 를 포함한 Fortify 제품 포트폴리오를 강화하는 보안 인텔리전스로 변환하는 일을 하고 있습니다. 현재 Fortify 소프트웨어 보안 컨텐츠는 30 개의 프로그래밍 언어에서 1,286 개의 취약점 범주를 지원하며 적용되는 개별 API 는 1 백만 개가 넘습니다.

Fortify Software Security Research(SSR) 팀은 Fortify Secure Coding Rulepacks(영어, 버전 2022.4.0), Fortify WebInspect SecureBase(SmartUpdate 를 통해 사용 가능) 및 Fortify Premium Content 업데이트를 즉시 사용할 수 있게 되었다는 소식을 기쁜 마음으로 알려 드립니다.

Fortify Secure Coding Rulepacks [Fortify Static Code Analyzer]

이번 릴리스에서 Fortify Secure Coding Rulepacks 는 30 개의 프로그래밍 언어에서 1,066 가지 고유 범주의 취약성을 감지하고 1 백만 개가 넘는 개별 API 를 지원합니다. 이번 릴리스에 포함되는 사항을 간략히 정리하면 다음과 같습니다.

Flask 업데이트(지원되는 버전: v2.2.x)

Python 으로 작성된 마이크로 웹 프레임워크인 Flask 를 사용할 때는 특정 기본 제공 도구 집합이나 라이브러리가 필요하지 않습니다. 간단하지만 안정적인 프레임워크인 Flask 는 대개 중소 규모 프로젝트에 적합하지만 소형 API 와 마이크로 서비스 등의 비교적 복잡한 프로젝트도 처리할 수 있습니다. 지원되는 범주에는 다음이 포함됩니다.

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- JSON Injection
- Often Misused: File Upload
- Open Redirect
- Path Manipulation
- Privacy Violation
- Server-Side Template Injection
- System Information Leak: External
- System Information Leak: Internal

Swift 용 iOS SDK 업데이트(지원되는 버전: 16)[1]

개발자는 Apple iOS SDK 에서 제공되는 프레임워크 컬렉션을 사용하여 Apple iPhone 과 iPad 장치용 모바일 응용 프로그램을 빌드할 수 있습니다. 이 릴리스에는 Swift 용 iOS SDK 지원을 위한 증분 업데이트가 포함되어 있습니다. 신규 규칙과 업데이트된 규칙을 적용하면 Swift iOS 및 iPadOS 응용 프로그램용 iOS SDK

[1] iOS SDK 16 용 새 규칙을 적용하려면 Fortify Static Code Analyzer 22.2 이상이 필요합니다.

15 와 16 내 Foundation 프레임워크의 API 적용 범위를 확장할 수 있습니다. 이러한 업데이트를 설치하면 다음과 같은 대다수 기존 취약성 범주에서 문제를 더욱 정확하게 감지할 수 있습니다.

- Insecure SSL: Overly Broad Certificate Trust
- Insecure Transport: Weak SSL Protocol
- Privacy Violation
- Resource Injection
- System Information Leak

Salesforce Apex 및 Visualforce 업데이트(지원되는 버전: v55)[2]

Salesforce Apex 는 비즈니스 트랜잭션, 데이터베이스 관리, 웹 서비스 및 Visualforce 페이지와 같은 Salesforce 응용 프로그램을 만드는 데 사용되는 프로그래밍 언어입니다. 이 업데이트를 설치하면 Database 작업, SOAP 웹 서비스, REST 웹 서비스, 핵심 Apex 시스템 API, Crypto API 및 Visualforce 페이지 구성 요소를 더욱 원활하게 지원할 수 있습니다. 새롭게 지원되는 Apex 관련 범주는 다음과 같습니다.

- Cookie Security: Cookie not Sent Over SSL
- Cookie Security: Missing SameSite Attribute
- Cookie Security: Overly Broad Path
- Cookie Security: Overly Permissive SameSite Attribute
- Cookie Security: Persistent Cookie
- Header Manipulation
- Header Manipulation: Cookies
- Insecure Transport
- Key Management: Hardcoded Encryption Key
- Log Forging (debug)
- Open Redirect
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Path Manipulation
- Privacy Violation
- Server-Site Request Forgery
- System Information Leak: External
- System Information Leak: Internal
- Weak Cryptographic Hash
- Weak Encryption: Insecure Initialization Vector

다음 지원 범주도 추가로 개선되었습니다.

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation

[2] Fortify Static Code Analyzer 22.2 이상이 필요합니다.

- Cross-Site Scripting: Reflected

기밀 검사 개선 사항

기밀 검사는 다양한 소스 코드 및 구성 파일에서 기밀을 찾는다는 개념입니다. SSR 은 이미 다양한 기밀 유형을 지원하고 있으며, SCA 는 모든 파일 유형에 기밀 검사 규칙을 적용하므로 코드 언어에 관계없이 특정 기밀을 찾을 수 있습니다. 이제는 기밀 지원 범위가 확장되어 다음과 같은 기밀도 지원됩니다.

- Credential Management: Hardcoded API Credential(하드코딩된 전달자 토큰용)
- Password Management: Hardcoded Password(SQL 서버 연결 문자열의 하드코딩된 비밀번호용)
- Password Management: Password in Comment(XML 주석의 비밀번호용)[3]

Google Guava 초기 적용 범위(지원되는 버전: v31.1)

Google 의 Java 라이브러리 집합인 Guava 에는 새 컬렉션 유형(예: multimap 및 multiset), 변경이 불가능한 컬렉션, 그래프 라이브러리, 그리고 동시성, I/O, 해싱 캐싱, 기본 형식, 문자열 등에 사용 가능한 유틸리티가 포함되어 있습니다. Guava 는 Google 및 타사의 Java 프로젝트에서 널리 사용됩니다. 지원되는 범주에는 다음이 포함됩니다.

- Null Dereference
- Path Manipulation
- Privacy Violation
- Setting Manipulation
- System Information Leak
- Unreleased Resource
- Unsafe Reflection
- Weak Cryptographic Hash
- Weak Cryptographic Hash: User-Controlled Minimum Bits
- Weak Cryptographic Hash: User-Controlled Seed
- Weak Encryption

Hot Chocolate 초기 적용 범위(지원되는 버전: 12.15.2)

Hot Chocolate 은 Microsoft .NET 플랫폼을 기반으로 빌드된 오픈 소스 GraphQL 서버입니다. 개발자는 Hot Chocolate 을 사용하여 개발 중인 응용 프로그램용 GraphQL 기반 API 를 빠르게 만들어 배포할 수 있습니다. 이번 릴리스에는 Hot Chocolate 으로 개발된 GraphQL API 에서 다음과 같은 취약성 범주 감지를 포함하여 Hot Chocolate 에 대한 초기 지원이 추가되었습니다.

---

[3] Fortify Static Code Analyzer 22.2 이상이 필요합니다.

- Cross-Site Scripting: Inter-Component Communication (Cloud)
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- GraphQL Bad Practices: Introspection Enabled
- Privacy Violation
- System Information Leak: External
- Trust Boundary Violation

Java 용 gRPC 확장 및 Python 초기 적용 범위(지원되는 버전: 1.49.1)

gRPC(Google Remote Procedure Call)는 여러 환경과 언어를 지원하는 최신 오픈 소스 고성능 RPC 프레임워크입니다. gRPC 는 부하 분산, 추적 및 인증을 지원하는 서비스를 연결합니다. 기존 JSON-over-HTTP 와 달리 gRPC 는 HTTP2 를 기반으로 하며 일반적으로 메시지에 바이너리 프로토콜 버퍼(protobuf) 형식을 사용합니다.

이번 릴리스에서는 Java gRPC 지원 범위가 확장되어 다음의 추가 범주가 포함되었습니다

- Access Control: gRPC Authentication Bypass
- Insecure SSL: Overly Broad Certificate Trust
- Log Forging
- Setting Manipulation
- Unreleased Resource: Streams

또한 Python gRPC 도 지원되므로 다음 범주가 포함되었습니다.

- Insecure Transport
- Insecure Transport: gRPC Channel Credentials
- Insecure Transport: gRPC Server Credentials
- Privacy Violation
- System Information Leak: External

클라우드 IaC(코드형 인프라)

IaC(코드형 인프라)는 다양한 수동 프로세스가 아닌 코드를 통해 컴퓨터 리소스를 관리하고 프로비저닝하는 프로세스입니다. 이번 릴리스에서는 IaC 지원이 개선되어 AWS, Azure 및 GCP 에 배포할 수 있는 Terraform 구성이 지원됩니다. 서비스의 구성과 관련된 일반적인 문제가 이제 개발자에게 보고됩니다.

Amazon AWS Terraform 구성

Terraform 은 클라우드 인프라의 구축, 변경 및 버전 관리를 위한 오픈 소스 코드형 인프라 도구입니다. Terraform 은 HCL(HashiCorp Configuration Language)이라는 자체 선언적 언어를 사용합니다. 클라우드 인프라는 원하는 상태를 설명하기 위해 구성 파일에 코드화되어 있습니다. Terraform 공급자는

Amazon Web Services(AWS) 인프라의 구성 및 관리를 지원합니다. 이번 릴리스에서는 Terraform 구성에 대한 다음 범주를 보고합니다.

- AWS Terraform Misconfiguration: Amazon API Gateway Publicly Accessible
- AWS Terraform Misconfiguration: Amazon EBS Insecure Storage
- AWS Terraform Misconfiguration: Amazon ElastiCache Insecure Transport
- AWS Terraform Misconfiguration: Amazon MQ Publicly Accessible
- AWS Terraform Misconfiguration: Amazon Neptune Publicly Accessible
- AWS Terraform Misconfiguration: Amazon RDS Insecure Storage
- AWS Terraform Misconfiguration: Amazon RDS Proxy Insecure Transport
- AWS Terraform Misconfiguration: Amazon RDS Publicly Accessible
- AWS Terraform Misconfiguration: Amazon Redshift Publicly Accessible
- AWS Terraform Misconfiguration: Amazon SNS Insecure Storage

Microsoft Azure Terraform 구성

Terraform 은 클라우드 인프라의 구축, 변경 및 버전 관리를 위한 오픈 소스 코드형 인프라 도구입니다. Terraform 은 HCL(HashiCorp Configuration Language)이라는 자체 선언적 언어를 사용합니다. 클라우드 인프라는 원하는 상태를 설명하기 위해 구성 파일에 코드화되어 있습니다. Terraform 공급자는 Microsoft Azure 인프라의 구성 및 관리를 지원합니다. 이번 릴리스에서는 Terraform 구성에 대한 다음 범주를 보고합니다.

- Azure Terraform Bad Practices: AKS Cluster Missing Host-Based Encryption
- Azure Terraform Bad Practices: Azure Disk Snapshot Missing Customer-Managed Key
- Azure Terraform Bad Practices: Azure MySQL Server Missing Infrastructure Encryption
- Azure Terraform Bad Practices: Azure PostgreSQL Server Missing Infrastructure Encryption
- Azure Terraform Bad Practices: Container Registry Missing Customer-Managed Key
- Azure Terraform Bad Practices: Cosmos DB Missing Customer-Managed Key
- Azure Terraform Bad Practices: Missing Azure Storage Infrastructure Encryption
- Azure Terraform Bad Practices: Missing SQL Database Backup Encryption
- Azure Terraform Bad Practices: Scale Set Missing Host-Based Encryption
- Azure Terraform Bad Practices: Shared Image Missing Customer-Managed Key
- Azure Terraform Bad Practices: SQL Database Missing Customer-Managed Key
- Azure Terraform Bad Practices: Storage Account Missing Customer-Managed Key
- Azure Terraform Bad Practices: Storage Encryption Scope Missing Customer-Managed Key
- Azure Terraform Bad Practices: VM Missing Host-Based Encryption
- Azure Terraform Misconfiguration: AKS Cluster Missing Customer-Managed Key
- Azure Terraform Misconfiguration: Managed Disk Missing Customer-Managed Key
- Azure Terraform Misconfiguration: Missing SQL Database Encryption
- Azure Terraform Misconfiguration: VM Storage Missing Customer-Managed Key

Google Cloud Platform(GCP) Terraform 구성

Terraform 은 클라우드 인프라의 구축, 변경 및 버전 관리를 위한 오픈 소스 코드형 인프라 도구입니다.

Terraform 은 HCL(HashiCorp Configuration Language)이라는 자체 선언적 언어를 사용합니다. 클라우드 인프라는 원하는 상태를 설명하기 위해 구성 파일에 코드화되어 있습니다. Terraform 공급자는 Google Cloud Platform 인프라의 구성 및 관리를 지원합니다. 이번 릴리스에서는 Google Cloud Platform Terraform 구성에 대한 다음 취약성 범주를 보고합니다.

- GCP Terraform Bad Practices: Apigee Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: BigQuery Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Bigtable Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Functions Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Spanner Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Filestore Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Pub/Sub Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Secret Manager Missing Customer-Managed Encryption Key
- GCP Terraform Misconfiguration: Compute Engine Missing Confidential Computing Features
- GCP Terraform Misconfiguration: Edge Cache Service Missing HTTP-to-HTTPS Redirect
- GCP Terraform Misconfiguration: Insecure App Engine Domain Transport
- GCP Terraform Misconfiguration: Insecure App Engine Transport
- GCP Terraform Misconfiguration: Insecure Cloud Function HTTP Trigger Transport
- GCP Terraform Misconfiguration: Insecure Edge Cache Service Transport
- GCP Terraform Misconfiguration: Insecure Supply Chain
- GCP Terraform Misconfiguration: URL Map Missing HTTP-to-HTTPS Redirect

기타 정정표

이번 릴리스에서는 오탐지 문제의 수를 줄이고 일관성을 위해 리팩터링하고 고객이 문제를 감사하는 역량을 향상시킬 수 있도록 리소스를 투자했습니다. 고객은 다음과 관련하여 보고된 문제를 해결하기 위해 변경된 사항도 확인할 수 있습니다.

19.x 이전 Fortify Static Code Analyzer 버전의 사용 중단:

2022.3 릴리스 발표[R3 릴리스 발표 링크]에서 언급했듯이, 해당 릴리스는 19.x 이전의 Fortify Static Code Analyzer 버전을 지원하는 규칙 팩의 마지막 릴리스였습니다. 이 릴리스에서는 19.x 이전 Fortify Static Code Analyzer 버전에서 2022.4 Rulepacks 가 로드되지 않습니다. 따라서 Rulepacks 를 다운로드하거나 Static Code Analyzer 를 버전 19.x 이상으로 업그레이드해야 합니다. 향후 릴리스에서는 Fortify Static Code Analyzer 의 마지막 4 가지 주 릴리스가 계속해서 지원됩니다.

취약성 범주에 대한 Fortify Priority Order Metadata 메타데이터 리팩토링

Micro Focus 는 사용자가 문제를 효율적으로 수정할 수 있도록 Fortify Priority Order 모델 내에서 문제 범주를 더욱 객관적으로 결정할 수 있는 메커니즘을 연구해 왔습니다. 그리고 2022 R3 릴리스 발표[R3 릴리스 발표 링크]에서 언급되었던 이러한 연구의 일환으로 모든 규칙이 적용되는 범주 검토를 시작했으며, 그 결과 업데이트가 필요한 몇 가지 영역을 확인했습니다. 이에 따라 다음의 96 개 범주와 관련된 Fortify Priority Order 메타데이터가 변경되었습니다. 이러한 변경으로 인해 기존의 문제가 기존 심각도 이하의 버킷(예: 위험, 높음, 보통, 낮음)에 표시될 수도 있습니다. 그리고 Fortify Priority Order 값과 개별 구성 요소가 변경되어 기존 필터 적용 시 추가 문제가 숨겨질 수도 있습니다.

- Android Bad Practices: Encryption Secret Held in Static Field
- Android Bad Practices: Use of Released Camera Resource
- Android Bad Practices: Use of Released Media Resource
- Android Bad Practices: Use of Released SQLite Resource
- ASP.NET Bad Practices: Non-Serializable Object Stored in Session
- Buffer Overflow
- Buffer Overflow: Signed Comparison
- Code Correctness: Arithmetic Operation on Boolean
- Code Correctness: Function Not Invoked
- Code Correctness: Incorrect Serializable Method Signature
- Code Correctness: Memory Free on Stack Variable
- Code Correctness: Negative Content-Length
- Code Correctness: Premature Thread Termination
- Code Correctness: readObject() Invokes Overridable Function
- Code Correctness: Readonly Collection Reference
- ColdFusion Bad Practices: Unauthorized Include
- Cookie Security: Cookie not Sent Over SSL
- Cross-Site Scripting: Handlebars Helper
- Cross-Site Scripting: Inter-Component Communication
- Cross-Site Scripting: Inter-Component Communication (Cloud)
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Reflected
- Cross-Site Scripting: Untrusted HTML Downloads
- Dangerous Method
- Denial of Service
- Denial of Service: Parse Double
- Denial of Service: Regular Expression
- Denial of Service: Stack Exhaustion
- Denial of Service: StringBuilder
- Double Free
- Dynamic Code Evaluation: Code Injection
- Dynamic Code Evaluation: Script Injection
- File Disclosure: Django
- File Disclosure: J2EE
- File Disclosure: Spring
- File Disclosure: Spring Webflow

- File Disclosure: Struts
- Format String: Argument Number Mismatch
- Header Manipulation: Cookies
- Header Manipulation: SMTP
- J2EE Bad Practices: Non-Serializable Object Stored in Session
- Log Forging
- Null Dereference[4]
- Often Misused: Authentication
- Often Misused: Boolean.getBoolean()
- Path Manipulation: Base Path Overwriting
- Path Manipulation: Zip Entry Overwrite
- Portability Flaw: File Separator
- Portability Flaw: Locale Dependent Comparison
- Privacy Violation
- Privacy Violation: Android Internal Storage
- Privacy Violation: BREACH
- Privacy Violation: Heap Inspection
- Privacy Violation: HTTP GET
- Privacy Violation: Image
- Privacy Violation: Keyboard Caching
- Privacy Violation: Screen Caching
- Privacy Violation: Sensitive Data Accessible From iTunes
- Privacy Violation: Shoulder Surfing
- Privacy Violation: Unobfuscated Logging
- Privilege Management: Android Disable
- Privilege Management: Missing API Permission
- Privilege Management: Missing Content Provider Permission
- Privilege Management: Missing Intent Permission
- Process Control
- Query String Injection: Android Provider
- Race Condition: PHP Design Flaw
- Race Condition: Singleton Member Field
- Race Condition: Static Database Connection
- SOQL Injection
- SOSL Injection
- System Information Leak: Overly Broad SQL Logging
- System Information Leak: PHP Errors
- System Information Leak: PHP Version
- Unreleased Resource: Cursor Snarfing
- Unsafe JNI

[4]Null 역참조 변경 사항은 Java 및 .NET 에만 적용됩니다.

- Unsafe JSNI
- Unsafe Mobile Code: Access Violation
- Unsafe Mobile Code: Database Access
- Unsafe Native Invoke
- Use After Free
- Weak Encryption: Insecure Initialization Vector
- Weak Encryption: Insecure Mode of Operation
- Weak Encryption: Insufficient Key Size
- Weak Encryption: Missing Required Step
- Weak Encryption: User-Controlled Key Size
- Weak XML Schema: Lax Processing
- Weak XML Schema: Type Any
- Weak XML Schema: Unbounded Occurrences
- Weak XML Schema: Undefined Namespace

React Bad Practices: Dangerously Set InnerHTML

이제는 React 응용 프로그램 내에서 "dangerouslySetInnerHTML" 사용 시 Bad Practice 플래그가 지정됩니다.

오탐지 개선 사항

이번 릴리스에서는 오탐지를 없애기 위한 노력이 계속되었습니다. 기타 개선 사항 외에도, 고객은 다음 영역에서 오탐지가 추가로 사라질 것으로 기대할 수 있습니다.

- *Access Control: Database* – 다른 데이터베이스 호출에서 입력이 제공될 때 Apex 에서 발생하던 문제가 제거됨
- *ASP.NET MVC Bad Practices: Model With Required Non-Nullable Property* – 데이터의 특정 특성을 적용하는 특성 사용 시 발생하는 오탐지 감소
- *Dockerfile Misconfiguration: Dependency Confusion* – 이미지를 처음부터 확장하여 최소 크기로 만들 때 발생하는 오탐지 감소
- *GraphQL Bad Practices: Introspection Enabled* – 클래스 기반 Flask 보기를 등록할 때 Flask 응용 프로그램에서 발생하는 오탐지 감소
- *Dynamic Code Evaluation: JNDI Reference Injection* – Spring Boot 프로젝트가 'log4j2.version' 속성을 Log4Shell 의 영향을 받지 않는 버전으로 설정할 때 발생하는 오탐지 감소
- *Memory Leak* – std::unique_ptr 사용 시 오탐지 감소
- *Mass Assignment: Insecure Binder Configuration* – DataContract, DataMember 또는 IgnoreDataMember 와 함께 JSONConverter 주석을 사용할 때 발생하는 오탐지 감소
- *Privacy Violation* – .NET 의 열거 값에서 발생하는 오탐지 감소
- *SQL Injection* – MyBatis 쿼리 주석에서 '$.'가 포함된 prepared 문을 사용할 때 발생하는 오탐지 감소

- *Unreleased Resource* – 컬렉션의 리소스를 번들링할 때 C/C++ 스캔에서 발생하는 오탐지 감소

잘못된 PHP 구성: magic_quotes 범주 제거됨

다음 세 가지 취약성 범주는 지원되는 PHP 버전과 더 이상 관련이 없으므로 제거되었습니다.

- PHP Misconfiguration: magic_quotes_gpc Enabled
- PHP Misconfiguration: magic_quotes_runtime Enabled
- PHP Misconfiguration: magic_quotes_sybase Enabled

결과적으로 위 범주의 모든 문제가 검사 결과에서 제거될 예정입니다.

범주 변경

오탐지 제거와 함께 범주가 통합되어야 하거나 레이블이 잘못 지정된 곳을 일부 식별했습니다. 취약성 범주 이름 변경이 발생하면 이전 검사를 새 검사와 병합할 때 검사 결과에 범주가 추가/제거됩니다.

- *Code Correctness: Class Does Not Implement equals* 가 이제 *Code Correctness: Class Does Not Implement Equivalence Method*
- *Code Correctness: Class Does Not Implement Equals* 가 이제 *Code Correctness: Class Does Not Implement Equivalence Method*
- *Code Correctness: toString on Array* 가 이제 *Code Correctness: ToString on Array*
- *Code Correctness: null Argument to equals()* 가 이제 *Code Correctness: null Argument To Equivalence Method* 로 보고됨
- *Code Correctness: null Argument to Equals()*가 이제 *Code Correctness: null Argument To Equivalence Method* 로 보고됨

또한 최신 IaC 지원의 일환으로 다음 24 개 범주가 리팩터링되어 일관성이 개선되었습니다.

- *Access Control: Azure Container Registry* 가 이제 *Azure ARM Misconfiguration: Improper Container Registry Network Access Control*
- *Access Control: Azure SQL Database* 가 이제 *Azure ARM Misconfiguration: Improper SQL Server Network Access Control*
- *Access Control: Cosmos DB* 가 이제 *Azure ARM Misconfiguration: Improper DocumentDB Network Access Control*
- *Access Control: Kubernetes Admission Controller* 가 이제 *Kubernetes Bad Practices: Improper Admission Controller Access Control*
- *Access Control: Kubernetes Image Authorization Bypass* 가 이제 *Kubernetes Misconfiguration: Image Authorization Bypass*
- *Ansible Bad Practices: CloudWatch Log Group Retention Unspecified* 가 이제 *AWS Ansible Misconfiguration: Insufficient CloudWatch Logging*

- *Ansible Bad Practices: Redshift Publicly Accessible* 가 이제 *AWS Ansible Misconfiguration: Improper Redshift Network Access Control*
- *Ansible Bad Practices: Unrestricted AWS Lambda Principal* 가 이제 *AWS Ansible Misconfiguration: Improper Lambda Access Control Policy*
- *Ansible Bad Practices: User-Bound AWS IAM Policy* 가 이제 *AWS Ansible Bad Practices: Improper IAM Access Control Policy*
- *Ansible Misconfiguration: Azure Monitor Missing Administrative Events* 가 이제 *Azure Ansible Misconfiguration: Insufficient Azure Monitor Logging*
- Azure Resource Manager Bad Practices: *Cross-Tenant Replication* 가 이제 *Azure ARM Misconfiguration:* Improper Storage Account Network Access Control
- *Azure Resource Manager Bad Practices: Remote Debugging Enabled* 가 이제 *Azure ARM Misconfiguration: Improper App Service Access Control*
- *Azure Resource Manager Bad Practices: SSH Password Authentication* 가 이제 *Azure ARM Misconfiguration: Improper Compute VM Access Control*
- *Azure Resource Manager Misconfiguration: Insecure Transport* 가 이제 *Azure ARM Misconfiguration: Insecure App Service Transport*
- *Azure Resource Manager Misconfiguration: Overly Permissive CORS Policy* 가 이제 *Azure ARM Misconfiguration: Improper CORS Policy*
- *Azure Resource Manager Misconfiguration: Security Alert Disabled* 가 이제 *Azure ARM Misconfiguration: Insufficient Microsoft Defender Monitoring*
- *Azure SQL Database Misconfiguration: Insufficient Logging* 가 이제 *Azure ARM Misconfiguration: Insufficient SQL Server Monitoring*
- *Insecure Storage: Missing EC2 AMI Encryption* 가 이제 *AWS CloudFormation Misconfiguration: Insecure EC2 AMI Storage*
- *Insecure Storage: Missing EFS Encryption* 가 이제 *AWS CloudFormation Misconfiguration: Insecure EFS Storage*
- *Insecure Storage: Missing Kinesis Stream Encryption* 가 이제 *AWS CloudFormation Misconfiguration: Insecure Kinesis Data Stream Storage*
- *Insecure Transport: Azure App Service* 가 이제 *Azure Ansible Misconfiguration: Insecure App Service Transport*
- *Kubernetes Bad Practices: API Server Publicly Accessible* 가 이제 *Azure ARM Misconfiguration: Improper AKS Network Access Control*
- *Privacy Violation: Exposed Default Value* 가 이제 *Azure ARM Misconfiguration: Hardcoded Secret*
- *Privilege Management: Overly Permissive Role* 가 이제 *Azure ARM Misconfiguration: Improper Custom Role Access Control Policy*

Fortify SecureBase [Fortify WebInspect]

Fortify SecureBase 는 SmartUpdate 를 통해 즉시 사용할 수 있는 다음 업데이트를 유도하는 정책과 수많은 취약성 검사를 통합하는 역할을 합니다.

취약점 지원

Server-Side Request Forgery[5]

응용 프로그램 서버가 설정하는 네트워크 연결을 공격자가 조작할 수 있게 되면 SSRF(Server-Side Request Forgery)가 발생합니다. 네트워크 연결은 응용 프로그램 서버의 내부 IP 에서 시작되는데, 공격자는 이 연결을 사용하여 네트워크 통제를 우회하고 정상적인 상황에서는 노출되지 않는 내부 리소스를 스캔하거나 공격할 수 있습니다. 응용 프로그램 서버가 설정하는 네트워크 연결을 공격자가 조작할 수 있게 되면 SSRF(Server-Side Request Forgery)가 발생합니다.

Expression Language Injection[6]

널리 사용되는 Apache Commons Text 라이브러리 버전 1.5~1.9 의 위험한 원격 코드 실행 취약성이 CVE-2022-42889 로 식별되었습니다. 기본 구성 사용 시에는 안전하지 않은 스크립트가 평가되고 임의 코드가 실행될 수 있습니다. 이번 릴리스에는 대상 웹 서버에서 CVE-2022-42889 취약성을 감지하는 검사 기능이 포함되어 있습니다. 이 검사 기능은 많은 수의 요청을 전송하므로 표준 정책에서는 제외되었습니다. 모두 검사 정책을 사용하거나 검사를 포함하도록 기존 정책을 사용자 지정하거나 이 검사를 실행하는 사용자 지정 정책을 만드십시오.

Insecure Transport: Weak SSL Cipher

TLS(Transport Layer Security) 및 SSL(Secure Sockets Layer) 프로토콜은 클라이언트와 웹 서버 간에 전송되는 데이터의 신뢰성, 기밀성 및 무결성을 보호할 수 있는 메커니즘을 제공합니다. 예를 들어 취약한 암호나 길이가 너무 짧은 암호화 키를 사용하는 경우 공격자가 보호 메커니즘을 무력화하여 중요한 정보를 도용하거나 수정할 수 있습니다. 이 릴리스에 포함되어 있는 새로운 검사 기능(ID: 11285)을 사용하면 심각도가 위험인 Insufficient Transport Layer Protection - Insecure

---

[5] WebInspect 21.2.0.117 이상 패치에서 사용할 수 있는 OAST 기능이 필요합니다.

[6] WebInspect 21.2.0.117 이상 패치에서 사용할 수 있는 OAST 기능이 필요합니다.

Cipher 취약성에 플래그를 지정할 수 있습니다. 안전하지 않은 암호 집합에서는 다수의 취약성이 확인되며 중요도는 낮지만 여러 가지 공격도 발생합니다.

기타 정정표

이번 릴리스에서는 오탐지 수를 더 줄이고 고객이 문제를 감사하는 역량을 향상시킬 수 있도록 리소스를 투자했습니다. 고객은 다음과 관련하여 보고된 검사 결과를 해결하기 위해 변경된 사항도 확인할 수 있습니다.

Insecure Transport: Weak SSL Cipher

이번 릴리스에는 Insufficient Transport Layer Protection – Weak Cipher 검사(11716)에 대한 개선 사항이 포함되어 있습니다. 이 릴리스를 설치하는 고객의 경우 이 검사의 심각도가 위험에서 높음으로 낮아집니다. 이와 같은 취약한 암호를 대상으로 하는 공격은 정교한 방식으로 진행되므로 차단하려면 많은 리소스가 필요하기 때문입니다. 이처럼 검사의 심각도를 낮춰 주는 이번 릴리스에는 'Insufficient Transport Layer Protection – Insecure Cipher' 검사에서 확인된 심각도가 위험인 문제에 플래그를 지정하는 새로운 검사도 도입되었습니다. PFS(Perfect Forward Secrecy)가 적용되지 않은 암호 집합은 앞으로 권장 암호에 포함되지 않을 예정입니다.

XML External Entity Injection[7]

ID 11337 로 식별되는 검사는 OAST(Out-of-band Application Security Testing) 기능을 지원하는 페이로드를 사용하도록 수정되었습니다. 이 검사가 개선되어 오탐지가 줄어들고 결과의 효율성이 높아졌으며 정확성이 개선되었습니다.

Fortify Premium Content

연구팀은 핵심 보안 인텔리전스 제품 이외에도 다양한 리소스를 구축, 확장 및 유지 관리합니다.

Fortify Taxonomy: 소프트웨어 보안 오류

[7] WebInspect 21.2.0.117 이상 패치에서 사용할 수 있는 OAST 기능이 필요합니다.

Fortify Taxonomy 사이트(https://vulncat.fortify.com)에는 새로 추가된 범주 지원에 대한 설명이 수록되어 있습니다. 마지막으로 지원되는 업데이트가 포함된 기존 사이트를 찾는 고객은 Fortify 지원 포털에서 해당 업데이트를 받을 수 있습니다.

Fortify 기술 지원 연락처

CyberRes Fortify
http://softwaresupport.softwaregrp.com/
+1 (844) 260-7219

SSR 연락처

Alexander M. Hoole
Senior Manager, Software Security Research
CyberRes Fortify
hoole@microfocus.com
+1 (650) 258-5916

Peter Blay
Manager, Software Security Research
CyberRes Fortify
peter.blay@microfocus.com

liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.