Micro Focus

Fortify 소프트웨어 보안 콘텐츠

2020 업데이트 3

2020 년 9 월 25 일 금요일

Micro Focus Fortify Software Security Research 정보

Fortify Software Security Research 팀은 최첨단 연구 결과를 보안 인텔리전스로 변환하여 Fortify Static Code Analyzer(SCA), Fortify WebInspect 및 Fortify Application Defender 를 포함한 Fortify 제품 포트폴리오를 강화합니다. 현재 Micro Focus Fortify 소프트웨어 보안 콘텐츠는 27 개의 프로그래밍 언어에서 1,032 개의 취약점 범주를 지원하며 1 백만 개 이상의 개별 API 를 지원합니다.

자세한 내용: https://software.microfocus.com/en-us/software/security-research

Fortify SSR(Software Security Research)은 Fortify Secure Coding Rulepacks(영어, 버전 2020.2.0), Fortify WebInspect SecureBase(SmartUpdate 를 통해 사용 가능) 및 Fortify Premium Content 의 업데이트를 바로 사용할 수 있다는 점을 알려 드립니다.

Micro Focus Fortify Secure Coding Rulepacks[SCA]

이 릴리스에서 Fortify Secure Coding Rulepacks 는 27 개의 프로그래밍 언어에서 815 가지 고유 범주의 취약점을 감지하고 1 백만 개가 넘는 개별 API 를 지원합니다. 이번 릴리스에 포함되는 사항을 간략히 정리하면 다음과 같습니다.

.NET System.Text.Json 지원(버전 3.1)[1]

System.Text.Json 네임스페이스는 기본 제공되며 표준을 준수하는 고속 네임스페이스로, Newtonsoft Json.NET 패키지를 대체할 수 있습니다. System.Text.Json 네임스페이스는 UTF-8 로 인코딩된 JSON 텍스트를 읽고 쓸 수 있는 유형도 제공합니다. 다양한 직렬화/역직렬화 메서드를 통한 데이터 흐름이 지원되며 다음 범주에 대한 적용 범위가 개선되었습니다.

- Cross-Site Scripting: Reflected
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Privacy Violation
- 시스템 정보 누출: External
- 시스템 정보 누출: Internal

Kotlin 코루틴 및 표준 라이브러리 지원 [2]

코루틴에 대한 지원을 포함하여 Kotlin 표준 라이브러리의 적용 범위가 확장되었습니다. 코루틴은 동시 및 비동기식 코드 개발을 위한 다양하고 유연한 API 를 제공합니다. 이번 업데이트에서는 코루틴을 사용하는 개선된 데이터 흐름 분석이 애플리케이션에 추가됩니다.

Kotlin Java 상호 운용성 및 Android 개선 사항 [3]

Java 와 Kotlin 소스 코드가 모두 포함된 프로젝트와 Kotlin 을 사용하여 개발된 Android 애플리케이션에 대한 기존 취약점 범주 전체에서 언어 지원이 개선되었습니다.

Java 14 개선 사항 [4]

Java 14 까지의 버전 변경을 지원하도록 기존 취약점 범주가 개선되었습니다.

ECMAScript 버전 업데이트 [5]

ECMAScript 버전에 대한 규칙 적용 범위가 ECMAScript 2020 까지의 모든 버전을 지원하도록 업그레이드되었습니다.

---

[1] System.Text.Json 네임스페이스는 설계상의 이유로 Newtonsoft Json.NET 패키지에서 보고된 동일한 JSON 삽입 문제에 취약하지 않습니다.

[2] select 식을 사용하는 것과 같은 실험적 코루틴 API 는 이번 업데이트에서 지원되지 않습니다.

[3] SCA 20.2 이상이 필요합니다.

[4] SCA 20.2 이상이 필요합니다.

[5] 전체 지원에는 SCA 20.2 이상이 필요합니다.

Go Logrus(버전 1.6.0)

Logrus 는 표준 라이브러리 로거와 완벽하게 호환되는 API 인 Go 를 위한 구조적 로거입니다. Logrus 는 엔터프라이즈 애플리케이션에 로깅 관련 위험을 유발할 수 있는 일반적인 로깅 패키지입니다. 지원되는 범주는 다음과 같습니다.

- 취약한 로깅 관행: 시스템 출력 스트림 사용
- Privacy Violation
- System Information Leak

Dockerfile 지원 [6]

Docker 를 사용하면 코드 및 해당하는 종속성을 샌드박스 환경(컨테이너)으로 패키징하여 모든 컴퓨팅 환경에서 실행할 수 있습니다. Dockerfile 은 컨테이너 이미지의 구성입니다. 이번 업데이트에서는 Dockerfile 에서 가장 일반적인 취약점을 해결합니다.
초기 지원에는 기존 "Password Management: Password in Configuration File" 범주와 다음의 새로운 범주가 포함됩니다.

- Dockerfile Misconfiguration: Default User Privilege
- Dockerfile Misconfiguration: Privileged Container
- Dockerfile Misconfiguration: Privileged Port
- Dockerfile Misconfiguration: Sensitive Host Discovery
- Dockerfile Misconfiguration: SSH Service

DISA STIG 4.11

컴플라이언스 영역에서 연방 고객을 지원하기 위해, Micro Focus Fortify Taxonomy 와 DISA(Defense Information Systems Agency) Application Security 및 Development STIG 버전 4.11 사이의 상관 관계가 추가되었습니다.

기타 정정표

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

17.x 이전 SCA 버전의 사용 중단 고지 사항:
사용자에게 17.x 이전 SCA 버전을 지원하는 Rulepacks 의 마지막 릴리스임을 알리는 고지 사항입니다. 다음 릴리스에서는 17.x 이전 SCA 버전에서 Rulepacks 가 로드되지 않습니다. 따라서 Rulepacks 를 다운로드하거나 SCA 버전을 업그레이드해야 합니다.
향후 릴리스에서는 SCA 의 마지막 4 가지 주 릴리스가 계속해서 지원됩니다.

매핑 사용 중단 - Security Technical Implementation Guide 버전 3.x(STIG 3.x):
Software Security Research 팀은 수년에 걸쳐 Fortify 범주를 외부 표준 및 모범 사례에 연결하는 매핑을 작성해왔습니다. 이제 지원되는 표준의 여러 버전이 매우 크게 증가하는 지점에 이르렀지만 많은 수의 이전 버전은 앞으로 사용되지 않을 것입니다.
이러한 이유로 externalmetadata.xml 파일의 새로운 XML 특성을 사용하여 버전 4.0 이전 버전의 STIG 를 "obsolete"로 표시했습니다. 이 변경 사항으로 인해 Fortify Software Security Center 및 SCA 애플리케이션 버전 20.2 부터는 이러한 이전 버전이 기본적으로 표시되지 않습니다. 이러한

[6] SCA 20.2 이상이 필요합니다.

이전 버전을 확인해야 하는 경우 해결 방법은 매핑을 변경하여 obsolete 를 "false"로 설정하거나 특성을 완전히 제거하는 것입니다.

블랙리스트 및 화이트리스트 명명:
현장의 추세에 따라 Software Security Research 팀은 Micro Focus Fortify 전체를 대표하여 "blacklist(블랙리스트)" 및 "whitelist(화이트리스트)"라는 용어를 보다 자연스럽고 의미론적으로 구체적인 용어로 바꾸기로 결정했습니다. 이에 보다 설명적인 용어인 "deny list(거부 목록)"와 "allow list(허용 목록)"를 각각 사용하기로 결정했습니다.
이 변경 사항으로 인해 다음과 같은 범주 이름도 변경되었으며 이에 따라 기존 문제가 제거되고 새로운 문제로 대체될 수 있습니다.

- "Deserialization Bad Practice: Blacklist"가 "Deserialization Bad Practice: Deny List"로 변경됨
- "Django Bad Practices: Blacklisted Attributes"가 "Django Bad Practices: Attributes in Deny List"로 변경됨
- "Setting Manipulation: User-Controlled Whitelist"가 "Setting Manipulation: User-Controlled Allow List"로 변경됨

취약한 암호화: Insecure Mode of Operation 업데이트
대칭형 암호화의 CBC(암호화 블록 체인) 모드를 사용하는 것은 더 이상 안전하지 않습니다. 이번 릴리스부터 현재 지원되는 라이브러리에 대한 CBC 사용은 Weak Encryption: Insecure Mode of Operation 으로 보고됩니다. BEAST, POODLE, Zombie POODLE, GOLDENDOODLE, LUCKY13 및 Padding Oracle 공격을 고려하여 CBC 는 현재의 알려진 암호화 연구에 기반할 때 더 이상 안전하지 않은 것으로 밝혀졌습니다.

거짓 긍정 개선 사항:
당사는 고객의 소리에 귀를 기울이고 거짓 긍정의 비율을 개선하기 위해 노력하고 있습니다. 이 릴리스에서는 거짓 긍정의 수를 줄이기 위해 다음과 같은 작업을 수행했습니다.

- Cross-Site Request Forgery
  다른 유형의 CSRF 토큰과 ASP.NET 숨겨진 필드 내부의 계정 축소
- Server-Side Request Forgery
  일부 Spring 애플리케이션에서 제거
- 취약한 로깅 관행: 시스템 출력 스트림 사용
  컨텍스트 기준이 명령줄 애플리케이션 또는 유틸리티인 경우 제거됨
- Dead Code: Unused Field
  lambdas 내부
- Privacy Violation
  Java, Scala 및 Kotlin lambda 내부의 암호와 관련된 중복 문제가 제거됨
- 이제 Java System Properties 의 기본 제공 속성이 개별적으로 처리되므로 많은 경우 거짓 긍정이 감소하고 새로운 System Information Leak 를 찾을 수 있습니다.

미리 보기:[7]
Null Dereference 및 Redundant Null Check 에 대한 규칙이 거짓 긍정 비율을 줄이도록 수정되었습니다. 특히, 내부 null 검사 기능을 처리하는 사용자 정의 규칙을 작성할 수 있는 기능이 추가되었습니다.
그러나 실제로 보편적인 이점을 제공하는지는 불명확합니다. 따라서 SCA 20.2 에서 사용할 수 있는 이러한 개선 사항은 기본적으로 해제되어 있습니다. 이러한 개선 사항을 테스트하려면 고객 지원에 문의하십시오.

[7] SCA 20.2 이상이 필요합니다.

Micro Focus Fortify SecureBase[Fortify WebInspect]

Fortify SecureBase 는 SmartUpdate 를 통해 즉시 사용할 수 있는 다음 업데이트에서 사용자를 안내하는 정책과 수천 가지의 취약점 검사를 결합합니다.

취약점 지원

취약한 암호화 서명: User-Controlled Key Source

Key Confusion 공격은 JSON Web Token(JWT)의 무결성을 위협합니다. 클라이언트에서 수신되는 토큰을 확인하기 위한 키 소스를 신중하게 선택하지 않는 JWT 구현은 공격자가 토큰 페이로드를 위조할 수 있으므로 이러한 공격에 취약합니다. 이번 릴리스에는 Key Confusion 공격에 대해 JWT 사용을 평가하는 검사가 포함됩니다.

JSON Web Token: Missing Protection Claims

"jti", "exp" 및 "aud"와 같은 보호 클레임을 구현하지 않는 JSON Web Token 은 가장 및 재생 공격에 취약할 수 있습니다. 이번 릴리스에는 검사 중에 발생한 토큰에 이러한 클레임이 누락되었는지 여부를 감지하는 다수의 검사가 포함됩니다.

Access control: Authorization Bypass

JSON Web Token(JWT)에 포함되는 정보는 HMAC 또는 RSA 알고리즘을 사용하여 디지털로 서명되므로 데이터 변조가 방지됩니다. 서명을 확인하지 않는 애플리케이션은 무단 액세스 및 가장 공격에 취약할 수 있습니다. 이번 릴리스에는 서명 누락 및 임의 서명 공격에 대해 JWT 구현을 평가하는 다수의 검사가 포함됩니다.

OAuth2: Insufficient state Parameter Entropy

엔트로피가 충분한 "state" 매개 변수 값을 사용하지 않는 OAuth2 프로토콜 구현은 가장 공격에 취약합니다. 이번 릴리스에는 state 매개 변수가 충분한 엔트로피 요구 사항을 충족하지 않는지 여부를 평가하는 검사가 포함됩니다.

OAuth2: Insufficient Refresh Token Revocation

새 액세스 토큰을 재발급하기 전에 클라이언트 ID, 클라이언트 암호 및 새로 고침 토큰 만료를 확인하지 않는 권한 부여 서버는 가장 공격에 취약할 수 있습니다. 이번 릴리스에는 클라이언트 ID 및 클라이언트 암호 없이 새로 고침 토큰을 사용하여 다수의 액세스 토큰을 생성할 수 있는지 여부를 감지하는 검사가 포함됩니다.

OAuth2: Insufficient Authorization Code Expiration

권한 부여 코드를 적시에 만료하지 않는 OAuth2 인증 서버는 도난 인증 코드에 대해 유효한 새 액세스 토큰을 계속해서 부여할 수 있으므로 가장 공격에 취약할 수 있습니다. 이번 릴리스에는 권한 부여 코드 남용을 감지하는 검사가 포함됩니다.

Cross-Site Request Forgery

OAuth2 권한 부여 흐름에서 "state" 매개 변수를 사용하지 않는 애플리케이션은 클라이언트를 가장한 무단 작업에 취약합니다. 또한 JSON Web Token(JWT)에 쿠키 저장소를 사용하는 애플리케이션은 유사한 CSRF 공격의 위험에 노출됩니다. 이번 릴리스에는 CSRF 에 취약한 JWT 및 OAuth2 인스턴스를 감지하는 다수의 검사가 포함됩니다.

취약한 암호화 서명: 불충분한 키 크기

JSON Web Token 에는 페이로드 무결성을 제공하기 위해 디지털로 서명된 정보가 포함됩니다. 약한 암호를 사용하는 HS256 과 같은 대칭형 알고리즘은 무차별 암호 대입 공격을 받아 공격자가 임의 페이로드를 생성하고 유효한 키로 서명하는 것을 허용할 수 있습니다. 이번 릴리스에는 HS256 과 같은 약한 알고리즘이 토큰 서명에 사용되는지 여부를 감지하는 검사가 포함됩니다.

SQL Injection[8]

SQLite 는 별도의 서버 프로세스 없이 SQL 쿼리 언어의 비표준 변형을 사용하여 데이터베이스에 액세스할 수 있는, 경량의 디스크 기반 데이터베이스를 제공하는 C 라이브러리입니다. 이번 릴리스에는 SQLite 라이브러리 버전 3.16.0 이상을 사용하여 웹 애플리케이션의 SQL Injection 취약점을 감지하는 향상된 WebInspect SQL Injection 검사가 포함됩니다.

컴플라이언스 보고서

DISA STIG 4.11

컴플라이언스 영역에서 연방 정부 고객을 지원하기 위해, 이번 릴리스에는 WebInspect 검사와 Defense Information Systems Agency Application Security 및 Development STIG 의 최신 버전인 버전 4.11 사이의 상관 관계가 포함됩니다.

[8] SQLite 에 대한 데이터베이스 유형 정보를 보고서 콘텐츠의 일부로 포함하려면 WebInspect 20.2 이상이 필요합니다.

정책 업데이트

DISA STIG 4.11

DISA STIG 4.11 관련 검사를 포함하도록 사용자 정의된 정책이 WebInspect SecureBase 의 기존 지원 정책 목록에 추가되었습니다.

웹 API

API 보안 평가와 관련된 검사를 포함하도록 사용자 정의된 정책이 WebInspect SecureBase 의 기존 지원 정책 목록에 추가되었습니다.

기타 정정표:

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

Insecure Transport: Weak SSL Cipher:
ID 11285 로 식별되는 검사가 수정되었습니다. 이제 CBC 암호와 함께 SSLv3 또는 TLS1.0 이 감지된 경우 CBC 모드 암호의 모든 인스턴스가 Weak SSL Protocol 의 보고서 콘텐츠 개요와 비교하여 취약한 것으로 보고됩니다. POODLE, GoldenPOODLE 및 ZombieDOODLE 과 같은 취약점을 고려하여 CBC 모드 암호화는 전송 계층 보안에 더 이상 권장되지 않습니다.

블랙리스트 및 화이트리스트 명명:
현장의 추세에 따라 Software Security Research 팀은 Micro Focus Fortify 전체를 대표하여 "blacklist(블랙리스트)" 및 "whitelist(화이트리스트)"라는 용어를 보다 자연스럽고 의미론적으로 구체적인 용어로 바꾸기로 결정했습니다. 대신, 보안 검사 콘텐츠에 "deny list(거부 목록)"와 "allow list(허용 목록)" 같은 보다 설명적인 용어를 각각 사용하기로 결정했습니다.

Micro Focus Fortify Premium Content

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

DISA STIG 4.11

새로운 상관 관계를 동반하기 위해, 이 릴리스에는 DISA STIG 4.11 을 지원하는 Fortify Software Security Center 에 대한 새로운 보고서 번들이 포함되어 있으며 Premium Content 의 Fortify 고객 지원 포털에서 다운로드할 수 있습니다.

Micro Focus Fortify Taxonomy: 소프트웨어 보안 오류

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

Fortify 기술 지원 연락처
Micro Focus Fortify
https://softwaresupport.softwaregrp.com/
+1 (844) 260-7219

SSR 연락처
Alexander M. Hoole
Software Security Research, 관리자
Micro Focus Fortify
hoole@microfocus.com
+1 (650) 258-5916

2020 년 9 월 25 일