TeamCity 서버 보안 강화하기

TeamCity는 빌드 프로세스의 중심에 있습니다. 소스 코드를 배포 가능한 아티팩트로 빌드하고 그러한 아티팩트를 배포하기도 합니다. 이 말은 곧 TeamCity가 민감한 정보에 액세스할 수 있음을 의미합니다.

TeamCity는 기본적으로 강력한 보안을 제공하지만 추가적으로 빌드 파이프라인의 보안을 강화할 수 있는 단계를 소개합니다.

일반적 조언

TeamCity 서버를 정기적으로 업데이트

TeamCity의 최신 릴리스 버전 업데이트를 정기적으로 수행할 것을 강력히 권장합니다.

새 업데이트를 사용할 수 있을 경우 TeamCity가 UI를 통해 자동으로 알려줍니다. 새 TeamCity 버전을 직접 확인하려는 경우, TeamCity 자체는 **Server Administration(서버 관리) > Updates(업데이트)**에서, 플러그인 업데이트는 **Server Administration > Plugins(플러그인)**에서 확인할 수도 있습니다.

기술적 측면에서 동일한 주요/부가 버전의 버그 수정 릴리스 간 업그레이드는 이전 버전과 호환되며(예: 2020.1.1 → 2020.1.2) 비교적 간단한 롤백을 지원합니다. 다른 모든 주요 업그레이드의 경우에도 최대한 원활하게 실행되도록 최선을 다하겠으나 간편하게 롤백하려면 백업을 꼭 하시기 바랍니다.

라이선스 측면에서도 버그 수정 릴리스 간 업그레이드가 안전합니다. 라이선스에 2020.1.1이 포함되어 있는 경우 어느 2020.1.x 버전으로든 업그레이드할 수 있습니다.

보안 알림 서비스 구독

또한 보안 알림 서비스를 구독하여 TeamCity나 다른 JetBrains 제품에 영향을 줄 수 있는 보안 문제에 관한 최신 정보를 받아보는 것이 좋습니다.

자격 증명

강력한 자격 증명을 신중하게 사용

TeamCity 서버뿐만 아니라 빌드와 연관되어 있거나 소프트웨어에서 프로덕션에 요구하는 기타 다른 모든 서비스에서 강력한 자격 증명을 사용하는 것이 좋습니다.

특히 자격 증명을 다음과 같은 곳에 보관하지 않도록 합니다.
- GitHub, GitLab 등의 저장소
- 환경 변수. 환경 변수는 타사 모니터링 시스템에 기록되거나 공유되는 경우가 많음
- 빌드 로그. 민감한 정보를 임의로 기록하지 않도록 해야 함

또한 버전 지정된 설정(Kotlin DSL 또는 XML 형식)을 사용 중이라면 자격 증명을 구성 파일에 저장하면 절대 안 됩니다. 대신 토큰을 사용하세요.

보안 데이터를 비밀번호 매개변수 유형으로 저장

비밀번호나 기타 보안 데이터를 TeamCity 설정에 저장하려면 TeamCity의 비밀번호 매개변수 유형을 사용할 것을 강력히 권장합니다. 이렇게 하면 민감한 값이 TeamCity의 웹 UI에 표시되는 일이 전혀 없고 빌드 로그에는 별표로 표시됩니다.

비밀 정보 관리 도구 사용

비밀번호 매개변수를 UI에서 가리고 저장 시 암호화하고 빌드 로그에 일반 텍스트로 노출되지 않도록 보호해도 충분히 높은 수준의 보안을 제공하지 못할 때가 종종 있습니다.

이에 대비해 HashiCorp Vault와 같은 도구를 사용해 보세요. 빌드에 사용되는 모든 민감한 자격 증명을 관리하고 교체할 수 있으며 TeamCity와도 원활하게 통합할 수 있습니다.

외부 인증 사용

LDAP 및 Windows Domain 통합부터 GitHub, GitLab 또는 기타 시스템을 통한 인증 등, 되도록 JetBrains의 외부 인증 모듈 중 하나를 사용하세요. 그런 다음 TeamCity의 기본 제공 인증을 비활성화하여 TeamCity가 내부 데이터베이스에서 해시 처리된 비밀번호를 보관하지 않도록 할 수 있습니다.

사용자 지정 암호화 키 사용

외부 시스템(VCS, 이슈 트래커 등)에서 인증하는 데 필요한 비밀번호는 스크램블된 형식으로 <TeamCity Data Directory>에 저장되고 데이터베이스에도 저장될 수 있습니다. 그러나 값은 스크램블될 뿐이므로 서버 파일 시스템이나 데이터베이스에 액세스할 수 있는 사용자가 검색할 수 있습니다.

이때 기본 스크램블링 전략 대신 사용자 지정 암호화 키 활성화를 고려할 수 있습니다. 이 경우 TeamCity는 디폴트 스크램블링 메커니즘을 사용하는 대신 고유한 사용자 지정 키를 사용하여 모든 보안 값을 암호화합니다.

권한

사전 정의된 역할 사용

TeamCity에서는 별도 설정이 필요 없는 사전 정의된 역할이 몇가지 제공됩니다.
- 시스템 관리자
- 프로젝트 관리자
- 프로젝트 개발자
- 프로젝트 뷰어

조직 구조와 일치하는 사용자 그룹을 만들고 해당 그룹에 위의 역할을 할당하세요. 그런 다음 사용자를 각 그룹에 추가하여 일상 작업에 필요한 최저 수준의 권한을 부여하세요.

또한 조금 더 권한이 필요한 사람에게는 프로젝트 관리자 역할을 즉시 할당하는 대신 추가 권한이 있는 새 역할을 만드는 것이 좋습니다. (새 역할 생성은 프로젝트별 권한을 비활성화하면 작동하지 않습니다.)

프로젝트별 승인 사용

보안을 더욱 강화하기 위해 프로젝트별 인증을 사용할 수도 있습니다. 이를 테면 개발자는 빌드 체인의 컴파일 부분에만 액세스할 수 있는 반면 DevOps는 배포 부분에 액세스하고 실행할 수 있습니다.

게스트 로그인 활성화 금지

TeamCity에 익명으로 로그인하는 것은 기본적으로 비활성화되어 있습니다. 외부 사용자가 모든 빌드 및 관련 로그 파일/아티팩트를 볼 수 있도록 하려는 경우가 아니라면 인터넷에 노출된 프로덕션 TeamCity 서버 인스턴스에서 이 옵션을 활성화하면 안 됩니다.

별도의 REST 사용자 생성

외부 스크립트 또는 프로그램에서 TeamCity의 REST API에 액세스하는 경우, 제한된 권한을 가진 별도의 사용자를 만드는 것이 좋습니다. 또한 사용자의 사용자 이름/비밀번호를 사용하여 API에 액세스하는 대신 자동 만료되는 액세스 토큰을 생성하는 것이 좋습니다.

배포 빌드 허용 제한

배포 빌드 체인에서 개인 빌드를 허용하지 않도록 하세요. 이러한 빌드를 트리거할 수 있는 개발자 수를 제한하고 해당 빌드에 대해 별도의 클린 에이전트 풀을 사용하세요.

TeamCity 서버

TeamCity 데이터 디렉터리 보호

<TeamCity Data Directory>에 대한 읽기 권한이 있는 사용자는 구성된 비밀번호를 포함하여 서버의 모든 설정에 액세스할 수 있습니다. 따라서 실제 서비스 관리자인 OS 사용자만이 디렉터리를 읽을 수 있도록 해야 합니다.

TeamCity 서버 보호

일반적으로 TeamCity 서버가 실행되는 컴퓨터에 대한 액세스를 제한하세요. 액세스 로그를 활성화하고 정기적으로 검토하세요.

모든 곳에 HTTPS 사용

TeamCity에서 HTTPS를 활성화하는 것이 좋습니다. 현재 권장 사항은 역방향 프록시(예: Nginx 또는 Apache)에서 HTTPS를 활성화하는 것입니다.

외부 데이터베이스 보호

TeamCity 서버의 데이터베이스 스키마에서 강력한 자격 증명이 있는 전용 데이터베이스 사용자 계정을 사용하세요. 데이터베이스에서 지원하는 경우 데이터베이스 암호화 사용을 고려하세요.

버전 관리

최신 Git 버전 사용

빌드 에이전트에서 항상 안정적인 최신 운영체제 및 Git 버전을 사용해야 합니다. 정기적으로 업데이트하세요.

SSH 키를 올바르게 관리

SSH 키를 사용하여 저장소에 액세스하는 경우 빌드 에이전트에 해당 키를 저장하면 안 됩니다. 대신 TeamCity의 SSH 키 관리 기능을 사용하여 키를 TeamCity 서버에 업로드하세요.

또한 알려진 호스트 검사를 비활성화하지 말고 서버에 .ssh/known_hosts 파일을 보관하고 연결 중인 모든 호스트에 대해 에이전트를 빌드해야 합니다.

전용 VCS 사용자 사용

Kotlin DSL과 같은 고급 기능을 사용하지 않거나 일반적으로 빌드 프로세스의 일부로 저장소에 커밋할 필요가 없는 경우, 쓰기 권한 없는 전용 VCS 사용자가 저장소에 연결하도록 하는 것이 좋습니다.

빌드 에이전트

클린 프로덕션 빌드 실행

에이전트에서 소스 코드가 변경되지 않도록 프로덕션 빌드에 대해 Enforcing Clean Checkout(클린 체크아웃 강제 시행) 옵션을 활성화하는 것이 좋습니다.

일회용 네트워크 보호 빌드 에이전트 사용

가능하면 일회용 빌드 에이전트를 사용하세요. 에이전트의 수명이 짧을수록 보안 침해가 발생할 가능성이 낮습니다. 또한 클라우드 에이전트에 대한 수신 네트워크 액세스를 비활성화하려면 OS 종속 방화벽 규칙을 사용해야 합니다.

다양한 프로젝트에 서로 다른 에이전트 풀 사용

하나의 컴퓨터에서 여러 에이전트를 실행하면서 Enable Clean Checkout(클린 체크아웃 활성화) 옵션을 설정하지 않으면, 손상된 에이전트 또는 신뢰할 수 없는 프로젝트가 잠재적으로 "이웃" 작업 디렉터리의 소스 코드를 수정할 수 있으니 유의하세요.

이 위험을 완화하려면 컴퓨터당 하나의 에이전트만 실행하고 서로 다른 (개인/공용) 프로젝트에 서로 다른 에이전트 풀을 사용하는 것이 좋습니다.

통합

공개 풀 리퀘스트를 무작정 빌드하지 않기

알 수 없는 사용자 또는 조직 외부의 사용자로부터 수신한 풀 리퀘스트를 빌드하는 경우, 빌드 에이전트에서 실행되는 악성 코드가 풀 리퀘스트에 포함되어 있을 수 있습니다.

공개 풀 리퀘스트 작성을 허용하지 않거나 분리되고 격리된 일회용 에이전트를 사용하세요. 또한 TeamCity에서 풀 리퀘스트 빌드를 탐지하고 보고하는 기본 제공 상태 보고서도 제공됩니다.

버전이 지정된 설정을 사용할 때 발생할 수 있는 보안 결과에 유의

버전이 지정된 설정(Kotlin DSL, XML)을 사용하고 해당 설정을 소스 코드와 동일한 저장소에 두는 경우, 악의적인 개발자가 잠재적으로 프로젝트 구성 설정을 수정하고 유출할 수 있습니다. 예를 들어 비밀번호를 인쇄하거나 파일로 전송하는 빌드 단계를 추가하여 이러한 침해 행위를 할 수 있습니다.

이를 방지하기 위한 옵션으로, 버전이 지정된 설정에 대해 제한된 사용자만 커밋할 수 있는 별도의 저장소를 사용할 수 있습니다.

타사 플러그인 사용 시 주의

플러그인을 설치할 때 그 출처를 신뢰할 수 있는지, 플러그인의 소스 코드를 사용할 수 있는지 확인하세요. 플러그인은 민감한 정보를 포함하여 TeamCity 서버의 모든 정보에 잠재적으로 액세스할 수 있습니다.

아티팩트 스토리지

익명 액세스 비활성화

빌드 아티팩트(예: S3)의 저장 위치에 관계없이 스토리지 위치에 대한 익명 액세스를 비활성화해야 합니다.

적절한 액세스 정책 사용

적절한 액세스 정책을 사용하여 아티팩트에 대한 S3 또는 기타 스토리지 위치/저장소를 보호하세요. 가능하면 암호화도 사용하세요. 저장 위치의 액세스 로그를 확인 및 모니터링하고 정기적으로 검토하세요.

민감한 데이터를 아티팩트에 두지 말 것

당연한 이야기지만 빌드의 아티팩트에 자격 증명이나 기타 민감한 정보를 일반 텍스트로 저장하면 안 됩니다.

빌드 기록 및 로그

빌드 기록 보관

프로젝트에 알맞은 정리 규칙을 지정하여 특히 중요한 배포를 수행하는 빌드의 경우 빌드 기록 및 로그를 더 오래 보관하세요. 또한 개발자에게 “빌드 제거” 권한을 부여하지 마세요. 권한을 부여하면 보관이 누락될 수 있습니다.

두 가지 방법을 사용하면 악의적인 활동이 오래 전에 발생했더라도 이를 추적할 수 있습니다.

서버 및 에이전트 로그 보관

TeamCity 서버 로그 및 빌드 에이전트 로그를 수집하여 적절하게 보안 조치된 스토리지에 보관하세요.