# خلاصه مقاله کنترل منبع: سوء استفاده از سیستم های مدیریت کد منبع

برت هاوکینز[۱]، بهاره کاوسی نژاد[۱] و سیده شکیبا انارکی فیروز[۲]

[۱] دانشگاه علم و صنعت ایران، تهران bahareh_kavousi@comp.iust.ac.ir

[۲] دانشگاه علم و صنعت ایران، تهران shakiba_anaraki@comp.iust.ac.ir

## چکیده

**سیستم‌های مدیریت کد منبع ( SCM ) نقش مهمی در سازمان‌ها دارند. این سیستم‌ها معمولاً برای مدیریت کد و ادغام با سایر سیستم‌ها در فرآیند DevOps استفاده می‌شوند. اما، این سیستم‌ها نقاط ضعفی در برابر حملات زنجیره‌ی تأمین نرم‌افزار دارند و می‌توانند به مهاجمان امکان حرکت داخلی و افزایش دسترسی در سازمان را بدهند. این گزارش فنی به بررسی سیستم‌های SCM معروف مانند GitHub Enterprise ، GitLab Enterprise و Bitbucket می‌پردازد و روش‌هایی را برای سوءاستفاده از آنها در حملات مختلف شامل بررسی، تغییر نقش کاربری، در اختیار گرفتن مخازن کد، جابه‌جایی به سیستم‌های دیگر، تقلید از کاربر و حفظ دسترسی پایدار توضیح می‌دهد. همچنین، ابزار SCMKit تیم X-Force Red برای انجام این حملات استفاده می‌شود. در پایان، راهنمایی‌های دفاعی برای حفاظت از سیستم‌های SCM نیز ارائه می‌شود.**



## کلمات کلیدی

**تجزیه و تحلیل کد منبع، بررسی وابستگی، کیفیت کد، تست امنیت، مدل سازی تهدید، نظارت مستمر، پاسخ حادثه، DevSecOps، امنیت ابری.**

## ۱ مقدمه

سیستم های مدیریت کد منبع ( SCM ) برای مدیریت کد منبع و تسهیل خط لوله DevOps برای سازمان ها یکپارچه هستند. با این حال، این سیستم ها اغلب از نظر امنیت در مقایسه با سایر سیستم های سازمانی مهم مانند Active Directory نادیده گرفته شده اند. سیستم های SCM ، مانند GitHub Enterprise ، GitLab Enterprise و Bitbucket ، به طور گسترده ای برای ذخیره و کنترل نسخه کد منبع، ادغام با ابزارهای مختلف، و امکان همکاری بین تیم های توسعه استفاده می شود.

در حالی که سیستم های SCM مزایای متعددی را ارائه می دهند، خطرات امنیتی بالقوه ای را نیز به همراه دارند. مهاجمان می‌توانند از آسیب‌پذیری‌ها در این سیستم‌ها برای راه‌اندازی حملات زنجیره تامین نرم‌افزار، دسترسی غیرمجاز، و حرکت جانبی در زیرساخت‌های سازمان استفاده کنند. بهره‌برداری از سیستم‌های SCM می‌تواند منجر به عواقب شدیدی شود، از جمله اصلاحات غیرمجاز کد، سرقت مالکیت معنوی و به خطر افتادن سایر سیستم‌های DevOps به هم پیوسته.

این مقاله با هدف روشن کردن اهمیت ایمن سازی سیستم های SCM و بررسی سناریوهای مختلف حمله ای که می تواند علیه پلتفرم های محبوب SCM انجام شود را بررسی می کند. با درک این بردارهای حمله، سازمان ها بهتر می توانند در برابر تهدیدات احتمالی دفاع کنند. علاوه بر این، جعبه ابزار حمله مدیریت کد منبع X-Force Red (SCMKit) معرفی خواهد شد که نشان می دهد چگونه می تواند این حملات را تسهیل و اجرا کند.

این مقاله موضوعات زیر را پوشش خواهد داد:

۱. پیشینه سیستم های SCM: مروری جامع بر سیستم های SCM، هدف آنها و نقش آنها در اکوسیستم DevOps.
۲. سناریوهای حمله: کاوش دقیق سناریوهای مختلف حمله، از جمله شناسایی، دستکاری نقش کاربر، تصاحب مخزن، چرخش به دیگر سیستم‌های DevOps، جعل هویت کاربر، و حفظ دسترسی مداوم.
۳. بهره برداری از پلتفرم های محبوب SCM: تجزیه و تحلیل عمیق آسیب پذیری ها و نقاط ضعف در سیستم های SCMپرکاربرد مانند GitHub Enterprise ، GitLab Enterprise و Bitbucket .

۴. مقدمه ای بر SCMKit : مقدمه ای بر SCMKit X-Force Red ، یک جعبه ابزار تخصصی که برای تسهیل و اجرای حملات علیه سیستم های SCM طراحی شده است.

۵. راهنمایی دفاعی: توصیه‌ها و بهترین شیوه‌ها برای سازمان‌ها برای محافظت از سیستم‌های SCM خود در برابر حملات احتمالی، از جمله ایمن کردن دسترسی کاربر، پیاده‌سازی مکانیزم‌های احراز هویت مناسب، نظارت بر فعالیت‌های مشکوک، و حفظ نسخه‌های نرم‌افزار به‌روز.

با درک ریسک‌ها و اجرای اقدامات امنیتی مناسب، سازمان‌ها می‌توانند از یکپارچگی، محرمانه بودن و در دسترس بودن مخازن کد منبع خود اطمینان حاصل کنند، تأثیر احتمالی آسیب‌های سیستم SCM را کاهش داده و از خطوط لوله DevOps خود محافظت کنند.

# ۲ پیشنیازها

## ۲-۱ کنترل منبع در مقابل کنترل نسخه

کنترل منبع و کنترل نسخه اصطلاحات نزدیک به هم هستند اما هدف های کمی متفاوتی دارند. کنترل منبع به طور اختصاصی روی ردیابی تغییرات در کد منبع تمرکز دارد. کنترل نسخه فراتر رفته و شامل تغییرات در فایل های باینری و انواع دیگر فایل ها می‌شود. به عنوان مثال، کنترل نسخه می تواند تغییرات را در فایل های کامپایل شده ردیابی کند، در حالی که کنترل منبع با تغییرات در کد منبع، مانند C# یا C++ ، که به تولید آن فایل های اجرایی منجر شده اند، سروکار دارد. نمونه هایی از ابزارهای مورد استفاده در این زمینه ها شامل Git برای کنترل منبع و Subversion برای کنترل نسخه است.

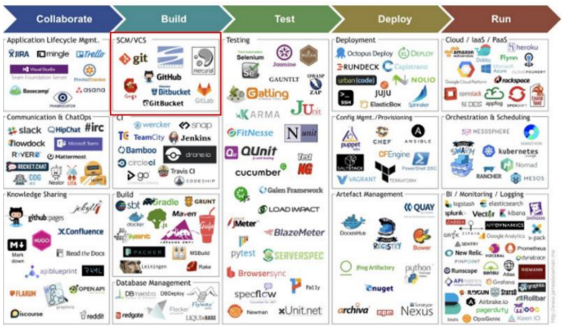
## ۲-۲ کنترل منبع در مقابل مدیریت کد منبع

کنترل منبع شامل ردیابی تغییرات در کد منبع است. برای استفاده عملی در فرآیند توسعه، از سیستم های مدیریت کد منبع ( SCM ) استفاده می‌شود. سیستم های SCM ردیابی تغییرات را در مخازن کد منبع تسهیل می‌کنند و به توسعه دهندگان در حل تعارضات هنگام ادغام همزمان کد از چندین مشارکت کننده کمک می‌کنند.

# ۳ سیستم های مدیریت کد منبع( Source Code Management Systems)

## ۳-۱ بررسی اجمالی سیستم‌های SCM

سیستم های SCM به چندین عضو تیم اجازه می دهند به طور همزمان روی همان فایل های کد منبع کار کنند، تغییرات تاریخچه فایل ها را ردیابی کرده و تعارضات را حل کنند. این سیستم ها مجهز به رابط های کاربری برای تعامل هستند و برای یکپارچگی قابل اعتماد در فرآیندهای توسعه ضروری اند. سیستم های SCM محبوب شامل GitHub Enterprise ، GitLab Enterprise و Bitbucket هستند که گزینه های میزبانی متفاوتی ارائه می دهند و از کنترل منبع Git پشتیبانی می‌کنند. آن ها همچنین با ادغام با سایر سیستم ها، لوله‌کشی DevOps را تسهیل می‌کنند.




شکل ۱: نمودار خط لوله DevOps

## ۳-۲ سیستم های SCM در لوله کشی DevOps

در لوله‌کشی DevOps ، سیستم های SCM در فاز ساخت، که تمام فازهای بعدی به کد منبع نگهداری شده در آن وابسته هستند، حیاتی هستند. پیشرفت از کد منبع به تولید شامل انتقال کد به یک سرور یکپارچه سازی مداوم ( CI ) برای تست، اسکن و استقرار برای استفاده در تولید می‌شود.

## ۳-۳ حملات زنجیره تامین نرم‌افزار

این حملات، که اخیراًمحبوب شده اند، شامل تزریق کد مخرب توسط مهاجمان در فاز تولید است. این می تواند منجر به سرایت گسترده در سازمان های متعدد شود، همانطور که توسط نقض SolarWinds نشان داده شده است. این سند به خطراتی که سیستم های SCM در چنین حملاتی با آن مواجه هستند تأکید دارد.

## ۳-۴ حرکت جانبی به سایر سیستم‌های DevOps

سیستم های SCM می توانند به عنوان نقطه دسترسی اولیه برای مهاجمان به منظور پیوست به سایر سیستم ها در چرخه حیات DevOps عمل کنند، مانند پلتفرم های CI/CD یا مخازن بسته. مثال ها شامل حرکت از سیستم SCM Bitbucket برای سوء استفاده از سیستم ساخت Jenkins یا از سیستم SCM GitLab Enterprise به سیستم بسته‌بندی Artifactory است. این سناریوها پتانسیل سیستم های SCM برای سوءاستفاده برای حملات بیشتر در زنجیره تامین نرم افزار را برجسته می‌کنند.

# ۴ Github Enterprise

در GitHub Enterprise ، استفاده از اصطلاحات " enterprise " و " organization " مهم هستند. اصطلاح " enterprise " به کل نمونه GitHub Enterprise اشاره دارد. یک یا چندین سازمان می‌توانند در یک enterprise قرار گیرند و enterprise کلیه سازمان ها را مدیریت می‌کند. لیست کاملی از اصطلاحات رایج استفاده شده در GitHub Enterprise در این منبع در دسترس است.

## ۴-۱ سطوح دسترسی و نقش‌ها

مالکان enterprise کلیه تنظیمات GitHub Enterprise ، از جمله سازمان‌ها، تنظیمات و خط‌مشی ها را مدیریت می‌کنند. اعضابخشی از enterprise هستند و می‌توانند درون سازمان ها کار کنند اما به تنظیمات سراسری enterprise دسترسی ندارند.

### ۲-۴ نقش های سازمان

مالکان سازمان تنظیمات و خط مشی‌های سازمان را کنترل می‌کنند. اعضای سازمان به پروژه های درون سازمان کمک می‌کنند. مدیران امنیت جنبه های امنیتی پروژه های سازمان را نظارت می‌کنند. مدیران برنامه GitHub ادغام و مدیریت برنامه های GitHub را انجام می‌دهند. همکاران خارجی افرادی هستند که عضو سازمان نیستند اما برای همکاری در پروژه‌های خاص دسترسی داده شده‌اند.

### ۳-۴ نقش‌های مخزن

مجوز خواندن اجازه مشاهده و کلون کردن مخازن را بدون ایجاد تغییرات می‌دهد. مجوز طبقه بندی امکان مدیریت issue و درخواست های کشیدن( pull request ) بدون دسترسی کامل را فراهم می‌کند. مجوز نوشتن اجازه افزودن تغییرات به مخزن را می‌دهد. مجوز نگهداری شامل دسترسی نوشتن به علاوه توانایی مدیریت تنظیمات مخزن است. مجوز مدیریت کنترل کامل بر مخزن را ارائه می‌دهد، از جمله توانایی حذف یا انتقال مخزن.

### ۴-۴ دامنه دسترسی توکن‌ها

توکن‌های دسترسی در GitHub Enterprise دارای "دامنه های دسترسی" هستند که میزان دسترسی به ویژگی های مختلف مانند مخازن، کلیدهای SSH و اطلاعات کاربر را تعریف می‌کنند و اطمینان از دسترسی امن و مناسب را فراهم می‌آورند.

### ۵-۴ قابلیت‌های API

API REST GitHub Enterprise امکان انجام مجموعه وسیعی از عملیات ها مانند تعامل با مخازن، مدیریت توکن های دسترسی، کلیدهای SSH و انجام وظایف مدیریتی را فراهم می‌کند تا جریان های کاری توسعه را ساده سازی و خودکار کند.

جدول ۱: جدول سناروهای جمله

| سناریوی حمله | زیر-سناریو |
|---|---|
| شناسایی | • مخزن<br>• فایل<br>• کد |
| تصاحب مخزن<br>جعل هویت کاربر | نامشخص<br>• جعل ورود کاربر<br>• جعل توکن |
| ارتقاء کاربر به ادمین سایت<br>حفظ دسترسی دائم | نامشخص<br>• توکن دسترسی شخصی<br>• توکن جعل هویت<br>• کلید SSH |
| دسترسی به کنسول مدیریت | نامشخص |

## ۵ GitLab Enterprise

### ۱-۵ اصطلاحات

یکی از اصطلاحات کلیدی که به طور مکرر در GitLab Enterprise استفاده می‌شود، «پروژه ها» است. پروژه ها می‌توانند کد را میزبانی کنند، مسائل را پیگیری کنند و می‌توانند حاوی خط لوله های CI/CD باشند.

### ۲-۵ مدل و سطوح دسترسی

پنج نقش برای کاربر در زمینه مجوزهای پروژه در دسترس است که در زیر لیست شده‌اند:

- مهمان
- گزارشگر
- توسعه‌دهنده
- نگهدارنده
- مالک

برای هر یک از پنج نقش، چندین مجوز عضو گروه در دسترس است.یک نکته قابل توجه این است که به طور پیش فرض، کاربران می‌توانند نام کاربری خود را تغییر دهند و گروه ها را ایجاد کنند. هر نقش همچنین چندین مجوز خط لوله CI/CD در دسترس دارد.

### ۵-۳ دامنه های توکن دسترسی

مجموعا هشت دامنه توکن دسترسی شخصی در GitLab Enterprise در دسترس است. لیستی از دامنه‌های مختلف و توضیحات آن‌ها در ادامه آمده است:

جدول ۲: جدول دامنه های مختلف و توضیحات آن ها

| دامنه | توضیحات |
|---|---|
| api | خواندن-نوشتن برای کل API ، شامل همه گروه‌ها و پروژه‌ها، رجیستری کانتینر و رجیستری بسته. |
| read_user | خواندن فقط برای نقاط پایانی زیر users/ . به طور اساسی، دسترسی به هر یک از درخواست‌های GET در API کاربران. |
| read_api | خواندن فقط برای کل API ، شامل همه گروه‌ها و پروژه‌ها، رجیستری کانتینر و رجیستری بسته. |
| read_repository | خواندن فقط ( pull ) برای مخزن از طریق git clone . |
| write_repository | خواندن-نوشتن ( pull, push ) برای مخزن از طریق git clone . لازم برای دسترسی به مخازن Git از طریق HTTP وقتی 2FA فعال است. |
| read_registry | خواندن فقط ( pull ) برای تصاویر رجیستری کانتینر اگر پروژه خصوصی است و احراز هویت لازم است. |
| write_registry | خواندن-نوشتن ( push ) برای تصاویر رجیستری کانتینر اگر پروژه خصوصی است و احراز هویت لازم است. (معرفی شده در GitLab 12.10 .) |
| sudo | اقدامات API به عنوان هر کاربری در سیستم (اگر کاربر تأیید شده یک مدیر باشد). |

### ۵-۴ قابلیت های API

API REST GitLab به یک کاربر اجازه می‌دهد تا چندین عملیات مانند تعامل با پروژه ها، توکن های دسترسی، کلیدهای SSH و موارد دیگر را انجام دهد. این همچنین اجازه اقدامات اداری را می‌دهد. مستندات کامل در مورد API REST در اینجا موجود است.

## ۶ Bitbucket

یک نکته در مورد Bitbucket این است که این پروژه به منظور نگهداری یک یا چندین مخزن طراحی شده است.

### ۶-۱ مدل و سطوح دسترسی

چهار سطح مجوز در Bitbucket وجود دارد که شامل مجوزهای جهانی، پروژه، مخزن، و شاخه است. یک نکته قابل توجه این است که تمام مجوزها می‌توانند در سطح کاربر یا گروه تنظیم شوند. قبل از اینکه یک کاربر بتواند به Bitbucket وارد شود، حداقل باید مجوزهایی در مجوزهای دسترسی جهانی به او اختصاص داده شده باشد.

- دسترسی جهانی ( Global ): تعیین می‌کند که چه کسی می‌تواند به Bit-bucket وارد شود، چه کسی مدیر سیستم، مدیر و غیره است.
- دسترسی پروژه ( Project ): دسترسی‌های خواندن، نوشتن و مدیریت در سطح پروژه (گروه‌های مخازن).
- دسترسی مخزن ( Repository ): دسترسی‌های خواندن، نوشتن و مدیریت بر اساس هر مخزن.
- دسترسی شاخه ( Branch ): دسترسی نوشتن ( push ) بر اساس هر شاخه.

در اینجا نقش‌های مختلفی که می‌توان از طریق دسترسی جهانی اختصاص داد، توضیح داده شده است.

در اینجا نقش‌های مختلفی که می‌توان از طریق دسترسی پروژه اختصاص داد، توضیح داده شده است.

در اینجا نقش‌های مختلفی که می‌توان از طریق دسترسی مخرن اختصاص داد، توضیح داده شده است.

API REST Bitbucket به کاربر اجازه می‌دهد تا اقدامات متعددی انجام دهد، مانند تعامل با پروژه ها، مخازن، توکن های دسترسی، کلیدهای SSH و بیشتر. مستندات کامل درباره API REST در این منبع در دسترس است.

## ۷ SCMKit

در X-Force Red ، ما خواستیم تا از قابلیت API REST موجود در سیستم های SCM رایج دیده شده در تعهدات استفاده کنیم و قابلیت های مفیدتری را در ابزاری به نام SCMKit به صورت نمونه اولیه اضافه کنیم. هدف از این ابزار ارائه آگاهی از سوء استفاده از سیستم های SCM و تشویق به شناسایی تکنیک های حمله علیه سیستم های SCM است. SCMKit به کاربر اجازه می‌دهد تا سیستم SCM و ماژول حمله مورد استفاده را مشخص کند، همراه با مشخص کردن اعتبارنامه های معتبر (نام کاربری/گذرواژه یا کلید API ) به سیستم SCM مربوطه. در حال حاضر، سیستم های SCM که SCMKit پشتیبانی می‌کند شامل GitHub Enterprise ، GitLab Enterprise و Bitbucket Server است. مهم ترین ماژول های حمله پشتیبانی شده شامل شناسایی، افزایش امتیاز و پایداری هستند. ابزار و مستندات کامل در GitHub X-Force Red موجود است.

## ۸ ملاحظات دفاعی

در مقابله با سناریوهای مختلف حملات سایبری مانند شناسایی، جعل هویت کاربر، ارتقای کاربر به مدیر، حفظ دسترسی دائمی و تغییر خطوط لوله CI/CD ، استراتژی‌های دفاعی برای GitHub Enterprise ، GitLab Enterprise و Bitbucket وجود دارد. توصیه های کلیدی در این پلتفرم ها عبارتند از:

- نظارت و ثبت: اطمینان حاصل کنید که لاگ های حیاتی برای تحلیل و تشخیص فعالیت های مخرب به سیستم SIEM شما ارسال می‌شوند. این

شامل لاگ های حسابرسی، لاگ های مدیریت، لاگ های API و لاگ های برنامه است.

- تشخیص حمله: از فیلترهای جستجوی ارائه شده برای تشخیص سناریوهای حمله خاص با تجزیه و تحلیل لاگ ها برای فعالیت های غیرعادی مانند تلاش های دسترسی غیرمجاز، جعل هویت کاربر و تغییرات در خطوط لوله CI/CD استفاده کنید.
- اقدامات پیشگیرانه: اقداماتی برای جلوگیری از حملات متداول را اجرا کنید، مانند غیرفعال کردن جعل هویت کاربر، تنظیم تاریخ انقضای خودکار برای توکن های دسترسی شخصی و کلیدهای SSH ، و نیازمندی به حداقل یک تاییدکننده برای هر کامیت کد. محدود کردن تعداد مدیران سایت یا سیستم و اجرای سیاست حداقل امتیاز نیز برای کمینه سازی سطوح حمله احتمالی توصیه می‌شود.
- بهبودهای امنیتی: فعال سازی احراز هویت چندفاکتوری ( MFA ) برای دسترسی به این سیستم های سازمانی، نیازمندی به امضای کامیت ها با استفاده از کلیدهای GPG یا گواهینامه های S/MIME ، و اطمینان از حذف به موقع شاخه های کد برای حفظ یک محیط تمیز و امن.
- دفاع از SCMKit : به طور خاص برای حملات مرتبط با SCMKit، نظارت بر امضاهای استاتیک، رشته های عامل کاربر، و هر توکن دسترسی یا کلیدهای SSH ایجاد شده با پیشوند " SCMKit- " به عنوان نشانه هایی از نفوذ پیشنهاد می‌شود.

این استراتژی ها با هدف تقویت وضعیت امنیتی سازمان‌هایی که از GitHub Enterprise، GitLab Enterprise و Bitbucket استفاده می‌کنند، با ارائه بینش های عملی برای شناسایی، پیشگیری و پاسخ مؤثر به تهدیدات سایبری طراحی شده اند و به صورت مفصل تر در مقاله تشریح شده اند.

## ۹ نتیجه گیری

سیستم های مدیریت کد منبع حاوی برخی از حساس ترین اطلاعات در سازمان ها هستند و جزء کلیدی در چرخه عمر DevOps هستند. بسته به نقش یک سازمان، به خطر افتادن این سیستم ها می تواند منجر به سازش سایر سازمان ها شود. این سیستم‌ها برای یک مهاجم ارزش بالایی دارند و نیاز به دید بیشتری از سوی جامعه امنیت اطلاعات دارند، زیرا در حال حاضر در مقایسه با سیستم‌های دیگر مانند Active Directory ، بعدا اضافه شده اند. هدف X-Force Red این است که این مقاله و تحقیق توجه بیشتری را به خود جلب کند و الهام بخش تحقیقات آینده در مورد دفاع از این سیستم های سازمانی حیاتی باشد.

## سپاس گزاری

در پایان مقاله سپاسگزاری از این افراد به جهت کمک در بهبود مقاله بعمل آمده است:

- Chris Thompson (@retBandit)
- Daniel Crowley (@dan_crowley)
- Dimitry Snezhkov (@Op_nomad)
- Patrick Fussell (@capt_red_beardz)
- Ruben Boonen (@FuzzySec)

## پیوست‌ها

جدول زیر ( ۳) سناریوهای حمله نشان داده شده در این مقاله را خلاصه می کند:

جدول ۳: جدول سناروهای جمله SCM

| سیستم SCM | سناریو |
|---|---|
| GitHub Enterprise | شناسایی |
| GitLab Enterprise | شناسایی |
| Bitbucket | شناسایی |
| GitHub Enterprise | حفظ دسترسی دائمی |
| GitLab Enterprise | حفظ دسترسی دائمی |
| Bitbucket | حفظ دسترسی دائمی |
| GitHub Enterprise | جعل هویت کاربر |
| GitLab Enterprise | جعل هویت کاربر |
| GitHub Enterprise | ارتقای کاربر به مدیر سایت |
| GitLab Enterprise | ارتقاء کاربر به نقش مدیر |
| Bitbucket | ارتقاء کاربر به نقش مدیر |
| Bitbucket | اصلاح خط لوله CI/CD |
| GitLab Enterprise | اصلاح خط لوله CI/CD |
| GitHub Enterprise | تصاحب مخزن |
| GitHub Enterprise | دسترسی به کنسول مدیریت |
| GitLab Enterprise | دسترسی SSH |

## مراجع

- برنامه جلسات توجیهی Black Hat USA 2022