اسپارا

گزارش بررسی آسیب‌پذیری Spring4Shell

مرکز عملیات امنیت اسپارا

تاریخ: ۱۴۰۱/۰۱/۱۴

مقدمه

Spring، یک فریم‌ورک کاربردی بسیار محبوب است که به توسعه دهندگان نرم‌افزار اجازه می‌دهد تا به سرعت و راحتی، برنامه‌های جاوا را با قابلیت‌های بسیار مناسب توسعه دهند. سپس این برنامه‌ها می‌توانند روی سرورهایی مانند Apache Tomcat به‌عنوان پکیج‌های مستقل به همراه وابستگی‌های (dependency) مورد نیاز قرار گرفته و استفاده شوند. در روزهای گذشته، یک آسیب پذیری روز صفر در Spring به نام «Spring4Shell» به‌صورت عمومی افشا شد که امکان اجرای کد را از راه دور و بدون احراز هویت برای مهاجمین فراهم می‌کند.

به دلیل محبوبیت این فریم‌ورک و انتشار اکسپلویت آن، اغلب محققین و شرکت‌های امنیتی در خصوص این آسیب‌پذیری، اعلام نگرانی کرده‌اند.

CVE	CVE-2022-22965
CPE	• cpe:/a:pivotal_software:spring_framework • cpe:/a:vmware:spring_framework
Affected Spring Frameworks	• < 5.2.20 • 5.3.x < 5.3.18
Risk Factor	Critical
Exploit Prerequisites	• JDK 9 or higher • Apache Tomcat as the Servlet container • Packaged as WAR • spring-webmvc or spring-webflux dependency • Spring Framework versions 5.3.0 to 5.3.17، 5.2.0 to 5.2.19، and older versions
Mitigation	Upgrade to Spring Framework version 5.2.20 or 5.3.18 or later
Patch Publication Date	3/31/2022

1. تشخیص آسیب‌پذیری

✓ بررسی از راه دور توسط پویشگر آسیب‌پذیری

پویشگرهای آسیب‌پذیری مختلف برای تشخیص این آسیب‌پذیری، پایگاه‌های خود را به‌روز کرده‌اند. به عنوان مثال Tenable، پلاگین خود را با شناسه «159374» منتشر کرده است که دارایی‌های آسیب‌پذیر را به کمک نسخه برنامه و بررسی آن با نسخه‌های آسیب‌پذیر کشف می‌کند (خود آسیب پذیری را پویش نمی‌کند).

✓ بررسی روی سرور

- بررسی نسخه استفاده شده JDK

روی سرورها دستور «Java –version» را اجرا کرده تا نسخه JDK نصب شده را مشاهده کنید. اگر نسخه 8 یا کمتر باشد، این آسیب‌پذیری شامل سرور شما نیست.

- بررسی استفاده از فریم‌ورک Spring

اگر برای نصب ابزارها در سرور از بسته‌های نرم‌افزاری با فرمت WAR× استفاده می‌کنید، مراحل ذیل را انجام دهید:

1. بسته نرم‌افزاری WAR را از حالت فشرده خارج کنید (فرمت فایل را به .zip تغییر دهید و سپس unzip کنید).

2. به دنبال یک فایل با فرمت اسمی spring-beans-×.jar، در مسیر فایل unzip شده باشید. (برای مثال spring-beans-5.3.16.jar) اگر وجود داشت به این معنی است که در سرور از این فریم‌ورک استفاده شده است.

3. اگر فایلی با فرمت اسمی spring-beans-×.jar وجود نداشت به دنبال فایلی با اسم CachedIntrospectionResuLts.class در مسیر فایل‌های unzip شده باشید. اگر وجود داشت به این معنی است که در سرور از این فریم ورک استفاده شده است.

اگر در پروژه و سرورها به صورت مستقیم و بدون واسطه از فرمت فایل JAR و بسته‌های نرم‌افزاری آن استفاده می شود، با توجه به موارد ذیل تصمیم بگیرید:

1. بسته نرم‌افزاری فایل JAR را unzip کنید.

2. به دنبال یک فایل با فرمت اسمی spring-beans-×.jar در مسیر فایل unzip شده، باشید. (برای مثال spring-beans-5.3.16.jar) اگر وجود داشت به این معنی است که در سرور از این فریم‌ورک استفاده شده است.

3. اگر فایلی با فرمت اسمی spring-beans-×.jar وجود نداشت به دنبال فایلی با اسم CachedIntrospectionResuLts.class در مسیر فایل‌های unzip شده باشید. اگر وجود داشت به این معنی است که در سرور از این فریم ورک استفاده شده است.

2. برطرف کردن آسیب‌پذیری

اولویت اول در به‌روزرسانی به آخرین نسخه از فریم‌ورک Spring یعنی 5.2.20 یا 5.3.18 یا نسخه‌های جدیدتر است. در صورتی که نمی‌توان به صورت مستقیم فریم‌ورک Spring را به نسخه‌های پیشنهاد شده ارتقا دهید، سه راه جایگزین پیشنهاد شده است:

1. برای نسخه‌های قدیمی که از فریم‌ورک Spring پشتیبانی نمی‌کردند، آپگرید شدن Apache Tomcat به نسخه 9.0.62، 10.0.20 یا 8.5.78، امنیت قابل قبولی ایجاد خواهد کرد.
2. در صورتی که آپدیت Spring یا Apache Tomcat قابل انجام نبود، Downgrade کردن جاوا به نسخه 8، یکی از راه‌های رفع این آسیب‌پذیری است.
3. یکی دیگر از راه‌های رفع این آسیب‌پذیری، استفاده از Disallowed Fieldها است که با تغییر در متغیر disallowedFields در WebDataBinder به صورت Globally انجام می‌شود.

موارد فوق به صورت کامل در لینک زیر توضیح داده شده است:

https://spring.io/blog/2022/03/31/spring-framework-rce-early-announcement#suggested-workarounds

3. شناسایی حملات مربوطه

✓ مبتنی بر شبکه

شرکت‌های مختلف تولیدکننده تجهیزات امنیت شبکه، قوانین خود را برای این آسیب‌پذیری منتشر کرده‌اند که لازم است با به‌روزرسانی پایگاه قوانین مربوطه، آنها را دریافت کنید. به‌عنوان مثال، قوانین IDS شرکت «Proofpoint» برای شناسایی فازهای مختلف این حمله آورده شده است:

```
alert http any any -> [$HOME_NET،$HTTP_SERVERS] any
(msg:"ET EXPLOIT Possible SpringCore RCE/Spring4Shell Stage 1
Pattern Set Inbound (Unassigned)"; flow:to_server،established;
http.method; content:"GET"; http.uri;
content:"pipeline.first.pattern="; fast_pattern;
classtype:attempted-admin; sid:2035674; rev:1;
metadata:attack_target Server، created_at 2022_03_31،
deployment Perimeter، deployment Internal، former_category
EXPLOIT، signature_severity Major، tag Exploit، updated_at
2022_03_31;)
```

alert http any any -> [$HOME_NET،$HTTP_SERVERS] any (msg:"ET EXPLOIT Possible SpringCore RCE/Spring4Shell Stage 2 Suffix Set Inbound (Unassigned)"; flow:to_server،established; http.method; content:"GET"; http.uri; content:"pipeline.first.suffix="; fast_pattern; classtype:attempted-admin; sid:2035675; rev:1; metadata:attack_target Server، created_at 2022_03_31، deployment Perimeter، deployment Internal، former_category EXPLOIT، signature_severity Major، tag Exploit، updated_at 2022_03_31;)
alert http any any -> [$HOME_NET،$HTTP_SERVERS] any (msg:"ET EXPLOIT Possible SpringCore RCE/Spring4Shell Stage 3 Directory Set Inbound (Unassigned)"; flow:to_server،established; http.method; content:"GET"; http.uri; content:"pipeline.first.directory="; fast_pattern; classtype:attempted-admin; sid:2035676; rev:1; metadata:attack_target Server، created_at 2022_03_31، deployment Perimeter، deployment Internal، former_category

EXPLOIT، signature_severity Major، tag Exploit، updated_at 2022_03_31;)
alert http any any -> [$HOME_NET،$HTTP_SERVERS] any (msg:"ET EXPLOIT Possible SpringCore RCE/Spring4Shell Stage 4 Prefix Set Inbound (Unassigned)"; flow:to_server،established; http.method; content:"GET"; http.uri; content:"pipeline.first.prefix="; fast_pattern; classtype:attempted-admin; sid:2035677; rev:1; metadata:attack_target Server، created_at 2022_03_31، deployment Perimeter، deployment Internal، former_category EXPLOIT، signature_severity Major، tag Exploit، updated_at 2022_03_31;)
alert http any any -> [$HOME_NET،$HTTP_SERVERS] any (msg:"ET EXPLOIT Possible SpringCore RCE/Spring4Shell Inbound (Unassigned)"; flow:to_server،established; http.method; content:"POST"; http.request_body; content:"pipeline.first.pattern="; fast_pattern; content:"pipeline.first.suffix="; content:"pipeline.first.directory=";

```
content:"pipeline.first.prefix="; classtype:attempted-admin;
sid:2035678; rev:1; metadata:attack_target Server، created_at
2022_03_31، deployment Perimeter، deployment Internal،
former_category EXPLOIT، signature_severity Major، tag Exploit،
updated_at 2022_03_31;
```

✓ مبتنی بر میزبان

قوانینی به فرمت‌های مختلف همچون سیگما و یارا (رایگان/تجاری) منتشر شده و در ادامه به دو مورد از آنها اشاره شده است که بخشی از حملات مرتبط با این آسیب‌پذیری را پوشش می‌دهد.

- https://github.com/Neo23x0/signature-base/blob/master/yara/expl_spring4shell.yar
- https://github.com/edelucia/rules/blob/main/sigma/Spring4Shell.yaml

www.spara.ir