**etcd目标主机支持RSA密钥交换修复手册**

磐基侧给的方案：<https://www.yuque.com/zhuang-9dknw/ykpyux/khuo3hmbtvshiogo?singleDoc#>

你提供的完整配置：

--tls-cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_RSA\_WITH\_CHACHA20\_POLY1305,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256

**应该修改为**（移除最后两个 RSA 密钥交换套件）：

--tls-cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_RSA\_WITH\_CHACHA20\_POLY1305,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384

# 背景

四川成都池南区资源池主机扫描出目标主机支持RSA密钥交换【原理扫描】的漏洞；etcd 目标主机支持 RSA 密钥交换意味着缺乏前向保密，存在历史流量解密风险，是过时且不推荐使用的配置，建议更新为 ECDHE 等临时密钥交换算法，以增强会话安全性和隐私保护。

# 漏洞原理及风险

RSA 密钥交换的问题：在 TLS 握手时，客户端使用服务器 RSA 公钥加密预主密钥（Pre-Master Secret），服务器用私钥解密后生成会话密钥。整个过程中，会话密钥的安全性完全依赖 RSA 私钥的保密性，无临时密钥参与，因此不具备前向保密。

etcd 的特殊性：etcd 作为分布式键值存储，存储大量敏感配置（如 Kubernetes 集群密钥、服务发现信息），若 TLS 会话被解密，可能导致集群核心数据泄露。

# 修复核心目标

禁用 RSA 密钥交换，强制启用基于 ECDHE 的临时密钥交换算法，确保：

每次 TLS 会话生成独立的临时密钥对（Ephemeral Keys）；

会话密钥仅由客户端和服务器通过 ECDHE 协商生成，不依赖 RSA 私钥加密传输；

即使 RSA 私钥泄露，历史会话密钥也无法被推导，实现前向保密。

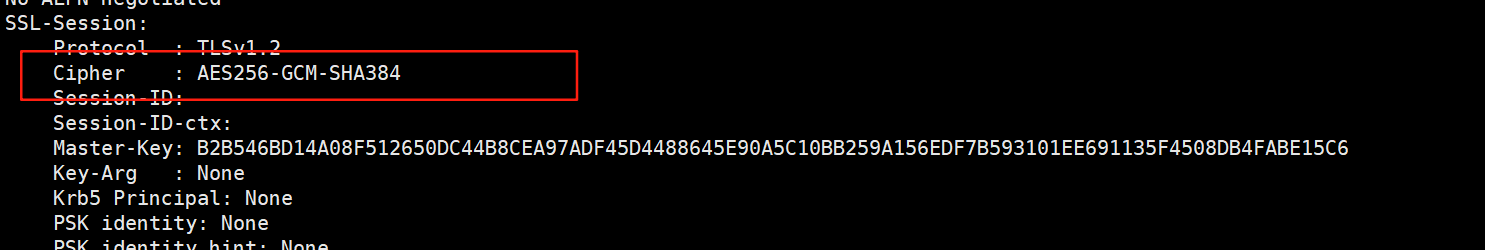
# 修复前确认

## 确认服务器是否支持RSA密钥交换方式

如果支持的话，说明存在RSA密钥交换漏洞，尝试使用 RSA 密钥交换的加密套件连接，若连接成功，说明服务器支持 RSA 密钥交换；

openssl s\_client -connect 192.168.0.119:2379 -cipher 'AES256-GCM-SHA384'

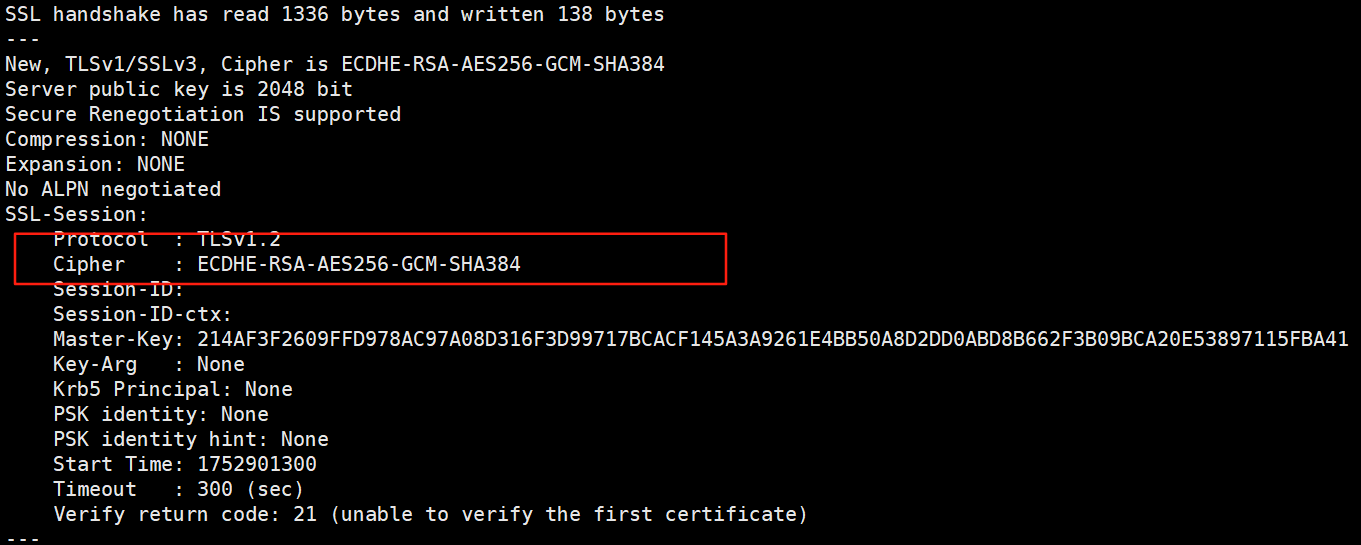
如图，SSL\_Session中指定的密钥交换连接成功：



## 确认服务器是否支持ECDH密钥交换方式

如果不支持的话，需要升级openssl才能修复此漏洞，需要放弃此次升级过程；

openssl s\_client -connect 192.168.0.119:2379 -cipher 'ECDHE-RSA-AES256-GCM-SHA384' 连接成功说明支持ECDH算法



## 确认 etcd 版本支持 ECDHE

etcd 从 v3.0 起已支持 ECDHE 密钥交换，但需确保版本 ≥ v3.3（推荐 v3.5+，包含更多安全加固）。若版本过低，先升级 etcd：

# 查看当前版本

kubectl -n kube-system exec -it etcd-node119 -- etcd --version

etcd Version: 3.5.1

Git SHA: e8732fb5f

Go Version: go1.16.3

Go OS/Arch: linux/amd64

## **建议在测试环境先测试**

运行一段时间后运行稳定没有其它问题再进行生产环境的升级；

## 升级影响

升级期间etcd会重启，api-server连接不到etcd集群，整个k8s集群将不可用，集群内的业务均不可用。

# 具体修复步骤

## 基于k8s部署的etcd

在k8s master安装节点执行即可

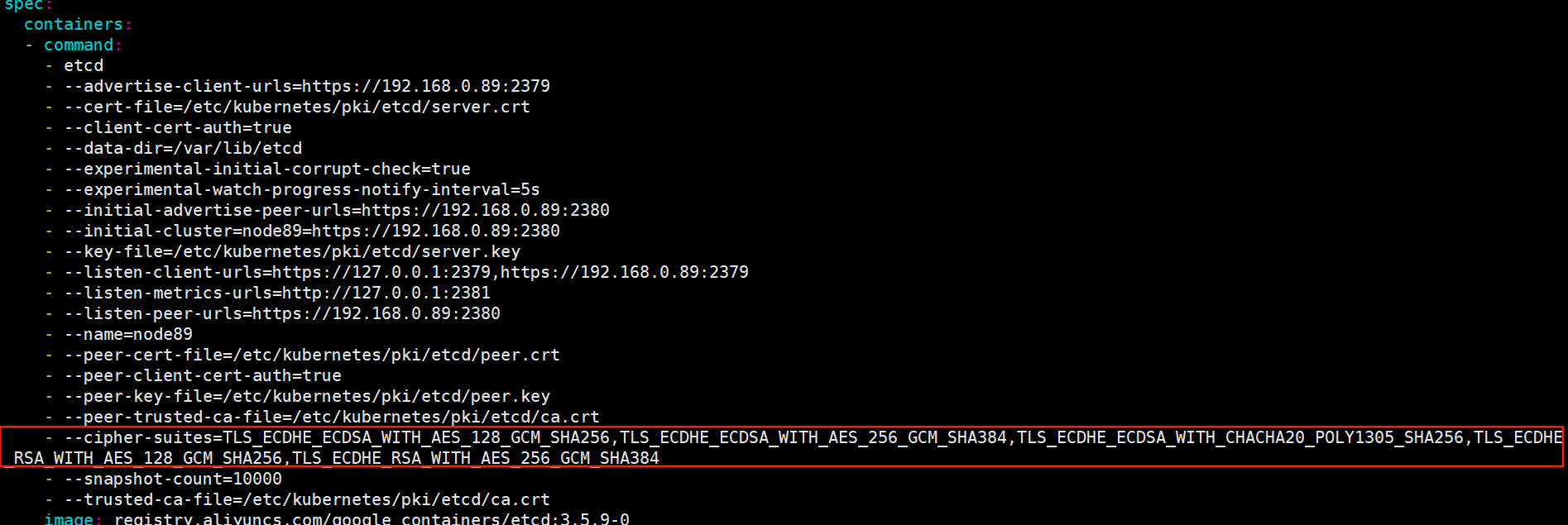
备份配置文件： cp /etc/kubernetes/manifests/etcd.yaml /data/backup/etcd.yaml.old

#注意不要备份到/etc/kubernetes/manifests目录

备份数据文件：参考etcd数据备份文档操作，这里不再提供。

修改配置：vim /etc/kubernetes/manifests/etcd.yaml，添加：

- --cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384



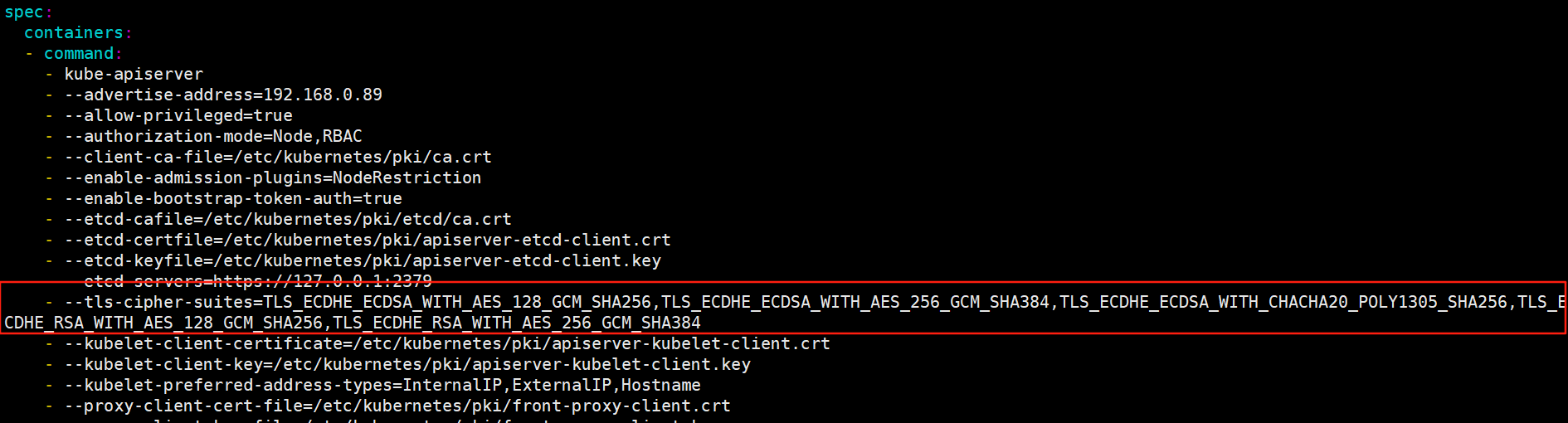
保存后etcd自动重启，过几分钟查看etcd 是否启动成功

kubectl get pods -A | grep etcd

如果通过kubectl get pods -A 很久(如超过10分钟)还没有结果，给api-server也添加对应的密钥交换方式，

vim /etc/kubernetes/manifests/kube-apiserver.yaml

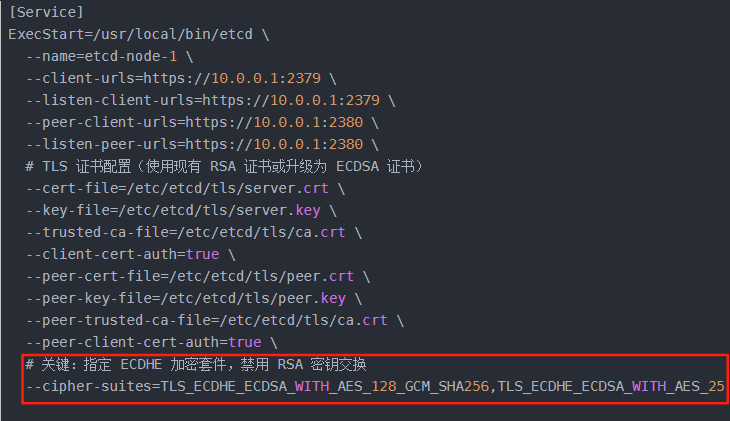
- --tls-cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384



## 基于二进制部署的etcd

在所有etcd节点执行，逐个节点依次执行，编辑 etcd 服务配置文件（如 /etc/systemd/system/etcd.service）,在ExecStart=/usr/local/bin/etcd中添加：

--cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384



systemctl daemon-reload

systemctl restart etcd

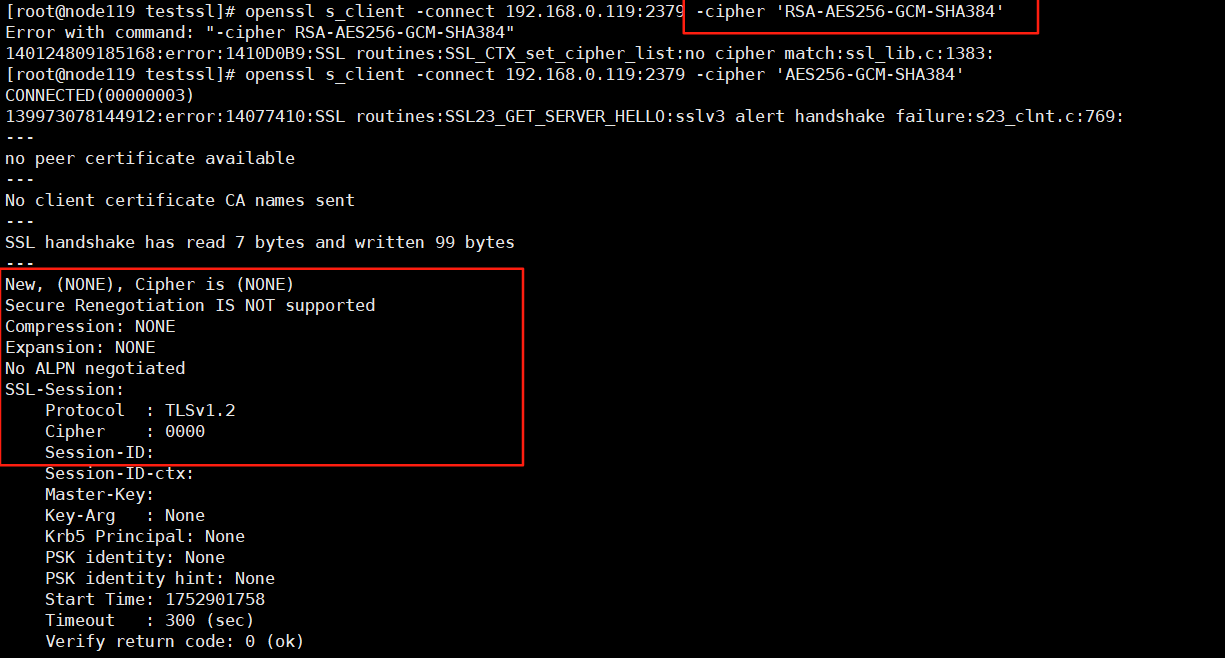
systemctl status etcd 查看etcd是否正常启动

# 验证

基于修复前确认的步骤再验证一遍确保 RSA 密钥交换已被禁用, ECDHE 密钥交换仍然可用

## openssl s\_client -connect 192.168.0.119:2379 -cipher 'AES256-GCM-SHA384'

已经不能连接，说明已禁用RSA算法



## openssl s\_client -connect 192.168.0.119:2379 -cipher 'ECDHE-RSA-AES256-GCM-SHA384'

依然可以连接，说明ECDH交换密钥正常使用

## 检查k8s集群以及pod是否正常

kubectl get nodes

kubectl get pods -A 查看apiserver以及etcd是否正常

随机挑选一些pod，查看服务是否能正常访问。

## 查看etcd是否就绪

kubectl exec -it etcd-node119 -n kube-system -- \

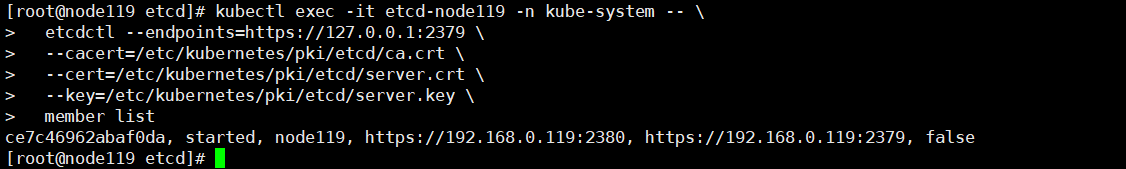
etcdctl --endpoints=https://127.0.0.1:2379 \

--cacert=/etc/kubernetes/pki/etcd/ca.crt \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key \

member list



# 回退方案

若升级后发现etcd以及apiserver等pod很久不能启动，回退只需要修改配置文件删除新增的配置即可，注意不要使用备份的配置文件拷贝。