DDoS的灾难性攻击解析与应对

Töma Gavrichenkov　　Qrator Labs公司首席技术官

2018 ISC 互联网安全大会 中国·北京

（原“中国互联网安全大会”）

目录

历史时间线

- 第一次攻击：1999年-2000年
- 2005年:微软提出STRIDE威胁模型
  - 身份欺骗
  - 篡改数据
  - 否认
  - 信息泄漏
  - 拒绝服务
  - 权限提升

[D?]DoS

“分布式攻击”与“*非*分布式攻击”的区别十分模糊。

传统意义上来说，分布式攻击有多个来源。

- *来源*是什么？是一个*IP地址*还是*一台物理机*呢？
- 如果是一台**物理机**，是一个虚拟机发起的攻击吗？

  亦或是同一台物理虚拟监视器下的多个虚拟机？

  如果这些虚拟机经常在数台物理机间迁移呢？

  假如我的电脑遭到了分布式攻击，我该如何判断攻击是单个来源还是多个来源呢？
- 假如攻击来自某个**IP地址**，那么我们该如何处理虚假流量呢？

[D?]DoS

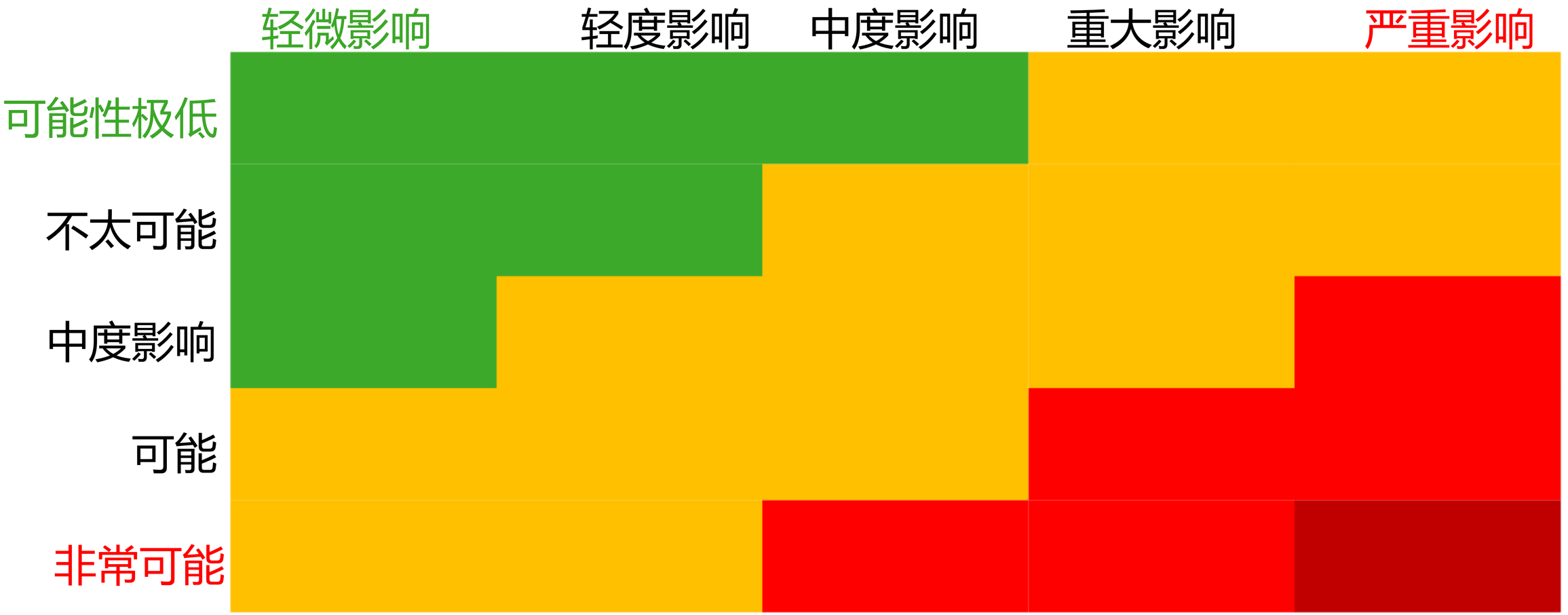
因此，需要运用不同的思维来理解：

- 拒绝服务（DoS）（正如STRIDE模型的一样）：软件中存在的**弱点**（比如像Ping of Death的空指针解引用。）
- **分布式拒绝服务（DDoS)**：消耗计算资源。

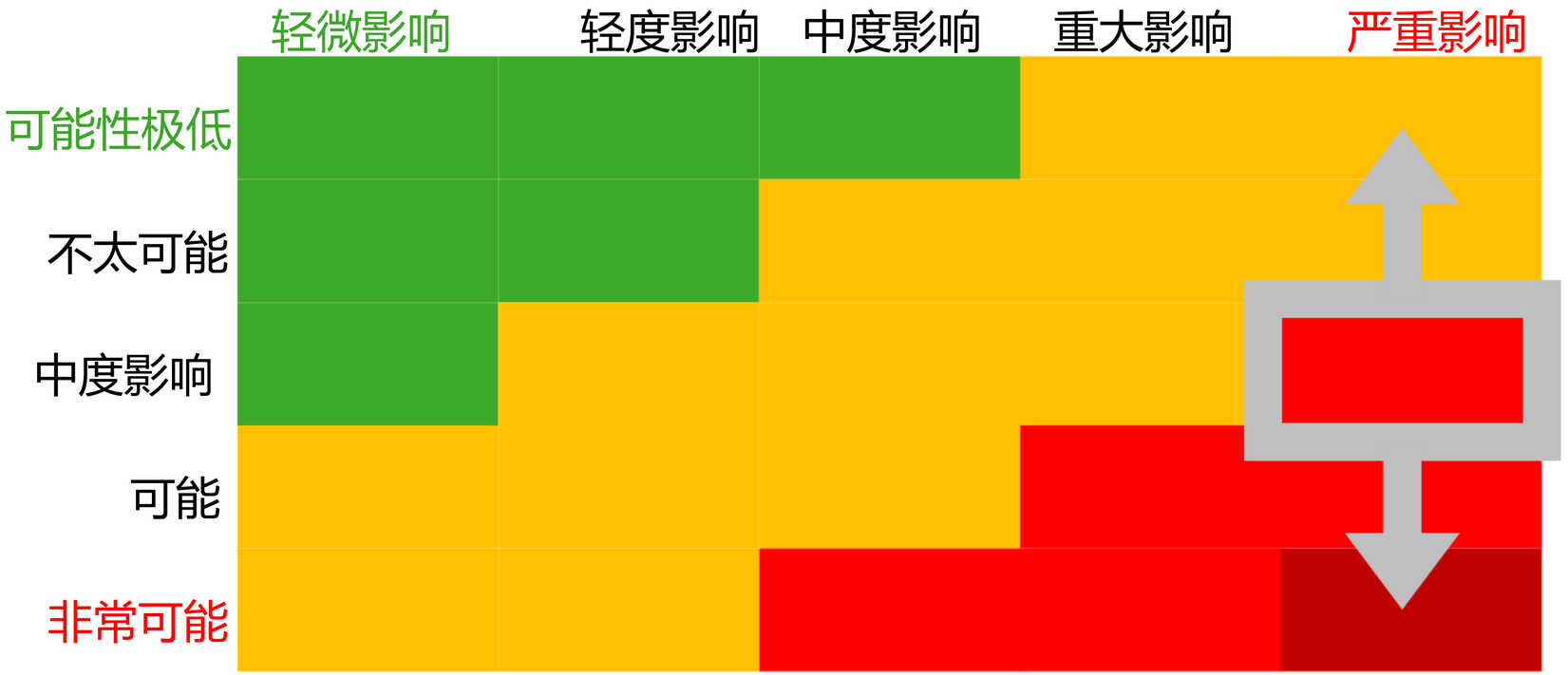
风险管理

STRIDE和其他威胁模型的基本思想是**风险评估、建模以及管理。**

可能性/影响矩阵图



可能性/影响矩阵图



2018年分布式拒绝服务（DDoS）攻击

- 影响：

严重影响

- 可能性：

？

攻击者的动机

报复
市场竞争
转移注意力
（比如窃取信息时）

防止他人访问不利于自己的信息。

大致可预测！

寻求快乐！
敲诈勒索
自我炫耀
政治目的

很难评估和控制

网络资源耗竭

- 现在，计算机网络由多层构成。
- 当至少一个网络层级停止提供服务后，用户将无法接收到相关网络资源。
- 因此，不同网络层级受到影响将形成不同的DDoS攻击：

L2 L3 普通**带宽**消耗

L4 L5 L6 超负荷使用**TCP/TLS的边缘案例**

L7 应用特有的**瓶颈问题**

攻击案例

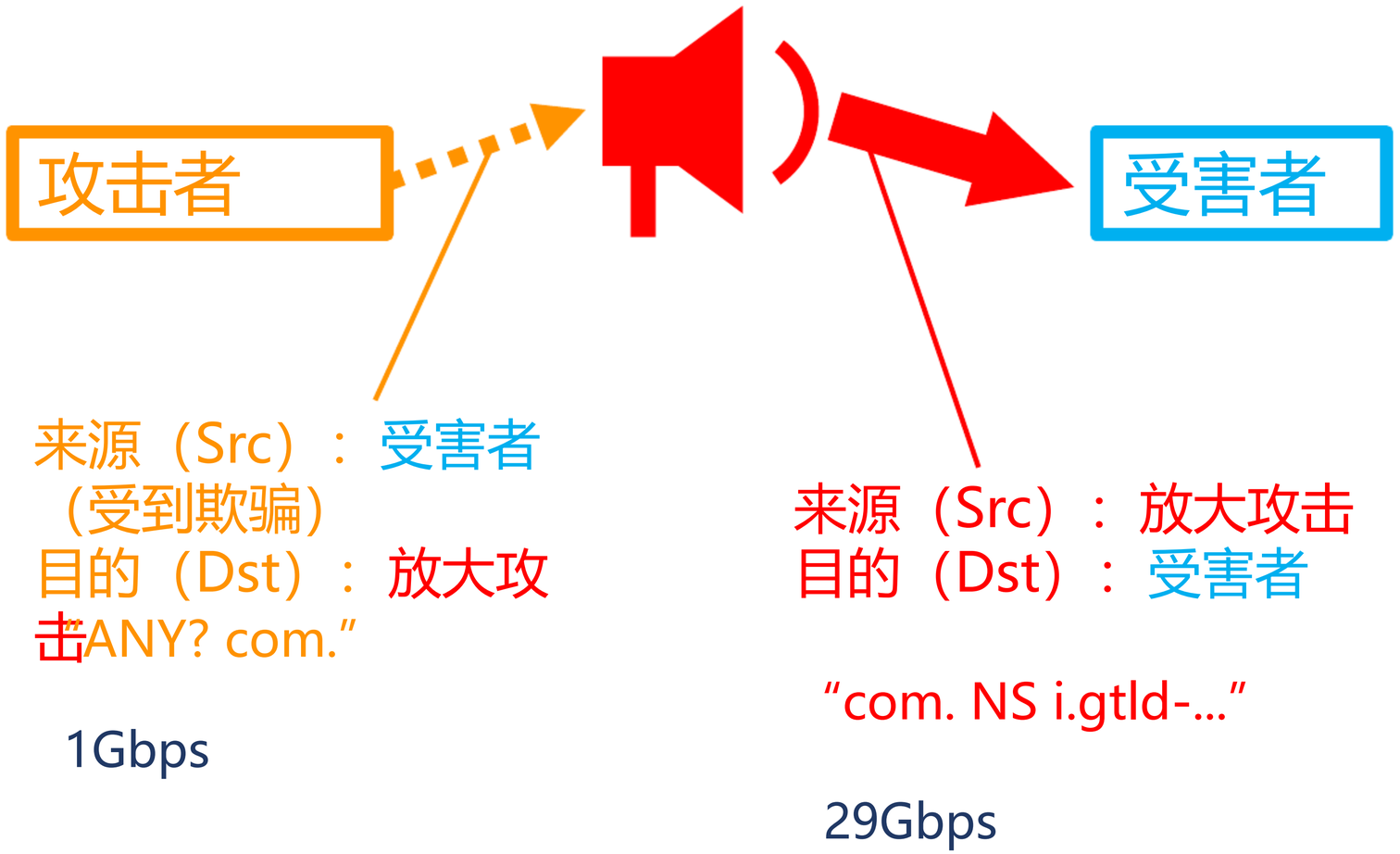
L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击**等**（简单了解即可）

- 基础设施层攻击

典型放大攻击

- 多台服务器通过互联网向客户发送的数据比接收到的多。
- UDP服务通常不会检查源IP地址。
- 这种漏洞给了DDoS放大攻击可乘之机。



脆弱的协议

- 长长的协议清单
- 大多数协议早已过时
- 诸如游戏协议类的当前协议包括：

- NTP
- DNS
- SNMP
- SSDP
- ICMP
- NetBIOS
- RIPv1
- PORTMAP
- CHARGEN
- **Quake**
- **Steam**
- Memcached
- …

脆弱的服务器

- 大多数过时的服务器最终会被更新
  - 替换
  - 或遭到淘汰。
- 因此放大器的数量呈稳定的下降趋势。
- 然而，时不时会出现新的脆弱协议。

源地址所有权的证明

例如，基于UDP的低时延的互联网传输层协议（QUIC）（由国际互联网工程任务组设计的传输层协议）：

- 初次握手包有1280字节
- 源地址验证

其他UDP协议**必须**执行类似的过程。

L2-L3 攻击缓解

在受害者看来：

- Anycast网络检验到位
- 进行清单管理，清除来路不明的通信服务器（比如UDP服务连接HTTP服务器）
- 限速控制不太重要的网络流量
- 挑战和握手（稍后会详细介绍）

在互联网服务供应商（ISP）看来：

- 抵抗典型攻击的简单启发法
- 远程触发黑洞技术（让客户自行解决问题。）

攻击案例

L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击**等**（简单了解即可）

- 基础设施层攻击

攻击案例

L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击**等**（简单了解即可）

- 基础设施层攻击

L4-L6

SYN洪水攻击、TCP连接攻击、 Sockstress攻击**等。**

- TLS攻击

攻击案例

L2-3

- 容量耗尽攻击：UDP洪水攻击、**SYN洪水攻击**、放大攻击等（简单了解即可）

- 基础设施层攻击

L4-6

SYN洪水攻击、TCP连接攻击、 Sockstress攻击等。

- TLS攻击

一次攻击可以同时影响多个网络层级。

联合攻击

- 比如，NTP放大攻击和SYN洪水攻击同时进行。

- 联合攻击的目的在于转移负责人的注意力，防止他们专心解决真正的威胁。

分批联合攻击

- 联合攻击的目的在于转移负责人的注意力，防止他们专心解决真正的威胁。

```
988     //util_strcpy(buf + util_strlen(buf), "POST /cdn-cgi/l/chk_captch
989     util_strcpy(buf + util_strlen(buf), "POST /cdn-cgi/");
990     rand_alphastr(buf + util_strlen(buf), 16);
991     util_strcpy(buf + util_strlen(buf), " HTTP/1.1\r\nUser-Agent: ");
992     util_strcpy(buf + util_strlen(buf), conn->user_agent);
993     util_strcpy(buf + util_strlen(buf), "\r\nHost: ");
994     util_strcpy(buf + util_strlen(buf), conn->domain);
                (buf + util_strlen(buf), "\r\n");
```

21:30:01.226868 IP 94.251.116.51 > 178.248.233.141:
GREv0, length 544:
IP 184.224.242.144.65323 > 167.42.221.164.80:
UDP, length 512

21:30:01.226873 IP 46.227.212.111 > 178.248.233.141:
GREv0, length 544:
IP 90.185.119.106.50021 > 179.57.238.88.80:
UDP, length 512

L4+ 攻击缓解

- SYN洪水攻击：基于3次握手过程的SYN cookies和SYN 代理（proxy）可以帮助受害者检查源IP地址。
- 其它基于数据包的洪水攻击：进行相同攻击的握手与挑战
- 剩余部分：会话分析、启发式、黑名单

在没有验证源IP地址的情况下，使用黑名单或白名单十分危险！

- 切记进行网络资产管理！

L4+ 攻击缓解

- 认为L4只是传输控制协议（TCP）的这种观点**有误。**

- 新型传输协议通过以下方式执行
  - 网络服务供应商
  - 应用程序
  - 国际互联网工程任务组

- 终端用户服务器?
- 终端用户后台?
- 互联网转接与互联网服务提供商

IPv6概要

128比特位 IP地址

- **有可能**：为地球上的每一粒沙子提供一个IP地址
- **不太可能**：在内存中存储大量条目

- 大约在10年前，将整个IPv4网络列入黑名单是很糟糕的做法。
- 除了使用IPv6，我们别无选择。

攻击案例

L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击等（简单了解即可）

- 基础设施层攻击

L4-L6

- SYN洪水攻击、TCP连接攻击、Sockstress攻击等

- TLS攻击

攻击案例

L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击等（简单了解即可）

- 基础设施层攻击

L4-L6

- SYN洪水攻击、TCP连接攻击、Sockstress攻击等

- TLS攻击

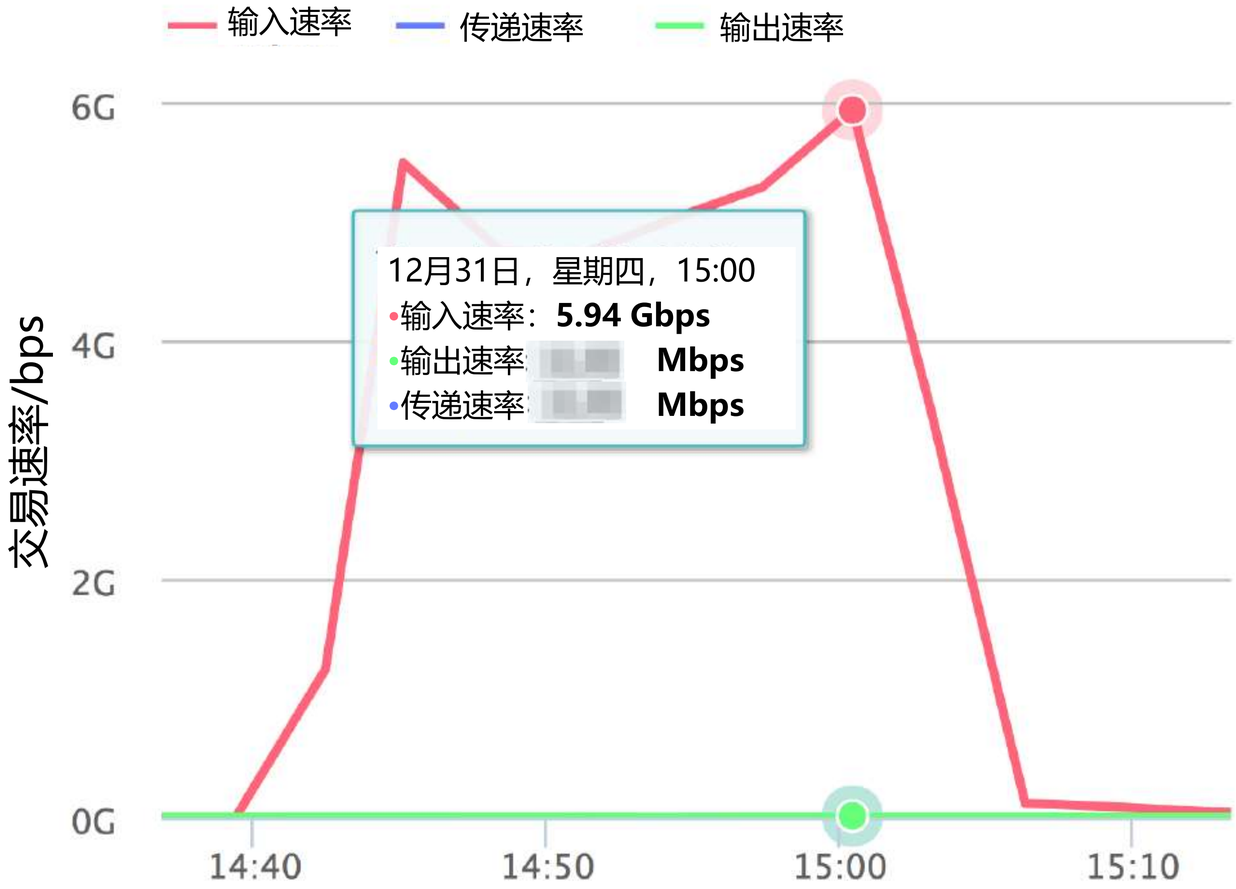
L7

- 基于应用程序的洪水攻击

WORDPRESS PINGBACK攻击

GET /whatever
用户代理：WordPress/3.9.2;
http://example.com/;
verifying pingback from
192.0.2.150

- 同时存在150000 - 170000个易受攻击的服务器
- 支持安全套接层（SSL）与传输层安全协议（TLS）



L7攻击的另一个例子：完整浏览器堆栈（FBS）

- 互联网机器人实际上可以比Wordpress机器更加聪明
- 高级僵尸网络能够使用无头浏览器（IE / Edge或Chrome）
  => *“完整浏览器堆栈”* (FBS) 僵尸网络
- 支持FBS的机器人能够完成复杂的挑战，如Javascript代码执行

L7攻击的另一个例子：完整浏览器堆栈（FBS）

如果进行正规的被动分析，验证码（CAPTCHA）将是对抗FBS的最后武器。

优势：

- 易于实施
- *通常*能够奏效

L7攻击的另一个例子：完整浏览器堆栈（FBS）

CAPTCHA弊端（1/3）：

- 需加入用户体验（UX），可能会破坏UX本身
- 破坏移动设备应用程序
- 相较于机器人，某些情况下对人来说更加困难

L7攻击的另一个例子：完整浏览器堆栈（FBS）

CAPTCHA弊端（2/3）：

- 需加入用户体验（UX），可能会破坏UX本身
- 破坏移动设备应用程序
- 相较于机器人，某些情况下对人类来说更加困难
- 并非所有机器人都是恶意的，并非所有的人都是无辜的
- CAPTCHA代理和CAPTCHA农场，例如 http://antigate.com/
- 一旦用户的计算机被感染，恶意软件将会把CAPTCHA注入该用户正在浏览的网页之中

L7攻击的另一个例子：完整浏览器堆栈（FBS）

CAPTCHA弊端（3/3）：

- 需加入用户体验（UX），可能会破坏UX本身
- 破坏移动设备应用程序
- 相较于机器人，某些情况下对人类来说更加困难
- 并非所有机器人都是恶意的，并非所有的人都是无辜的
- CAPTCHA代理和CAPTCHA农场，例如http://antigate.com/
- 一旦用户的计算机被感染，恶意软件将会将CAPTCHA注入该用户正在浏览的网页之中
- 光学字符识别（OCR）工具快速发展
- 语音识别的发展速度比OCR更加迅速
- “隐匿式安全”：使用开源机器学习工作比较容易破解开源类CAPTCHA举例：https://medium.com/@ageitgey/how-to-break-a-captcha-system-in-15- minutes-with-machine-learning-dbebb035a710

L7攻击的另一个例子：完整浏览器堆栈（FBS）

“工作量证明”，就像在客户端Javascript挖掘加密货币一样?

- 不可行性

a) 典型的僵尸网络可感染数十万台机器；
b) 典型的Web站点正在争取页面加载时间，虽然只有几*毫秒*。

L7攻击的另一个例子：完整浏览器堆栈（FBS）

- 不同于Wordpress pingback，类似攻击在大多数情况下并不会导致链接降级。
- 因而该类攻击通常不属于ISP的责任范畴

L7 攻击缓解：内容复杂

主动型：
- 超文本传输协议（HTTP）/JS 挑战
- 验证码（CAPTCHA）

被动型：
- 应用程序会话分析
- 大数据
- 关联、机器学习

监控、**事件响应**

误报和漏报

- 所有基于学习的算法并不是那么严格。
- **误报**：算法在**没有匹配项时**表现出**匹配特征**
- **漏报**：算法在**有匹配项时**表现出**无匹配特征**

- 基本上，任何算法都可调整为0%FP或0%FN
- 事实通常介于两者之间
- 攻击目的决定其均衡性

攻击案例

L2-L3

- 容量耗尽攻击：UDP洪水攻击、SYN洪水攻击、放大攻击**等**（简单了解即可）

- 基础设施层攻击

L4-L6

- SYN洪水攻击、TCP连接攻击、Sockstress攻击**等**
- TLS攻击

L7

- 基于应用程序的洪水攻击

分类如下：

- 彼此**独立** *
- 互无**遗漏**

互联网包罗万象，极其复杂

数十年以来，IT界有一道广为人知的求职面试题：
“当你在浏览器输入isc.360.cn会发生什么？”

- https://github.com/alex/what-happens-when:

目录

- 按下“g”键
- 回车键按下
- 产生中断（非USB键盘）
- （Windows）一个WM_KEYDOW键盘消息被发往应用程序
- （Mac OS X）一个KeyDown NSEvent被发往应用程序
- （GNU/Linux）Xorg服务器监听键码值
- 解析全球资源定位器（URL）
- 输入的是全球资源定位器（URL）还是搜索的关键字?

当……时发生了什么?

- 域名服务器（DNS）查询
- 使用套接字
- 传输层安全协议（TLS）握手
- 超文本传输协议（HTTP）
- 超文本传输协议（HTTP）服务器请求处理

当……时发生了什么?

- 域名服务器（DNS）查询
- IPv4/IPv6选择
- 使用套接字
- 深度包检测（DPI）
- 传输层安全协议（TLS）握手
- 证书吊销列表（CRL）/在线证书状态协议（OCSP）
- 超文本传输协议（HTTP）
- 负载均衡器
- 超文本传输协议（HTTP）服务器请求处理
- 内容分发网络（CDN）

- 应用服务器不能仅作为DDos攻击的直接目标而存在
- 每一层级都可能受到攻击，因此需要在L2-L7层实现智能防御

- **资产管理**
- **基础架构监控**
- **在条件允许的情况下，消除人为因素**

架构视图

- 网络安全并非单一的**产品**、应用或云，它是一个长期防御的**过程**
- 如果想提升应对分布式拒绝服务攻击（DDoS）的能力，需要将所有的互联网协议纳入到考虑范围内
- 相关公司（具有网络攻击防御需求）须遵循以下**原则**：
  - 更新升级（软硬件及其它）
  - 风险管理
  - 事故处理
- 多方合作与快速反应
- 联系您的计算机应急响应小组（CERT）或计算机安全事件响应小组（CSIRT）以获取建议

ISC 2018
ISC互联网安全大会

360互联网安全中心

谢 谢

2018 ISC 互联网安全大会 中国 · 北京

Internet Security Conference 2018 Beijing · China

（原“中国互联网安全大会”）