404 Not Found的知识库

最近更新日期：2019/12/27

最近一周新增：

- 对乌云漏洞库payload的整理以及Burp辅助插件
- boy-hack/wooyun-payload
- 下一座圣杯 - 2019
- 恶意样本分析资源汇总
- 从研究者视角看漏洞研究之2010年代
- 网络安全架构 | 通过安全架构提升安全性

硬实力

- 计算机理论基础
  - 计算机网络
  - 操作系统
  - 数据结构与算法
  - 数据库
- 计算机技术基础
  - 语言
  - 框架
  - 工具
  - 技术
- 底层研究
- 安全技术
  - 漏洞
  - Web安全
  - 渗透测试
  - 代码审计
  - 数据安全
  - 云安全
- 安全研究
  - APT检测
  - 恶意样本
  - Red Team
  - WAF
  - 恶意URL检测
  - 对抗机器流量
  - 异常检测
  - 图与安全
  - AI安全
- 人工智能
  - 算法体系
  - 基础知识
  - 机器学习
  - 深度学习

    - 强化学习
    - 应用领域
    - 综合素质
- 企业安全建设
    - 安全开发
    - 安全数据分析
    - 安全检测
    - 优秀开源安全项目/安全产品
    - 安全运营
    - 安全管理
    - 安全思考
    - 安全架构
    - 红蓝对抗
- 安全发展
    - 个人发展
    - 行业发展

软实力

- 职业规划
- 综合素质
    - 逻辑思维与语言表达
    - 管理
    - 思考
    - 注意事项
- 附录
    - 国内优秀技术人
    - 国外优秀技术站点
- 废弃

正文：

计算机理论基础

操作系统

- [计算机考研408全网最全!!!!!]王道计算机操作系统
- 中断与异常
- 怎样通俗的理解操作系统中内存管理分页和分段？
  粒度、信息的逻辑单位和信息的物理单位、长度不确定和长度确定、二维地址和一维地址、完整信息和内存离散分配。
- 操作系统之内核态和用户态小结
- 常见面试题整理--操作系统篇（每位开发者必备）

计算机网络

- 常见面试题整理--计算机网络篇（每位开发者必备）
  TCP和UDP的区别，TCP三次握手和四次挥手，浏览器输入URL之后的流程，HTTP协议的请求类型，GET和POST的区别，ARP地址解析协议

- 一次完整的浏览器请求流程
  页面（浏览器、HTTP）请求到响应经过的流程包括了TCP三次握手等系列流程，比如域名解析、发起TCP三次握手、发起HTTP请求、服务器响应HTTP请求，浏览器得到HTML代码、浏览器解析HTML代码，并请求HTML代码中的资源、浏览器对页面进行渲染呈现给用户。
- tcp 的可靠性到底指的是什么？ - CYS的回答 - 知乎
  TCP的可靠性是指基于不可靠的IP层在传输层提供可靠的数据传输服务，主要是指数据不会损坏或丢失，而且所有数据都是按照发送顺序进行传送。实现TCP的可靠性传输有以下机制：校验和（校验数据是否损坏）、定时器（分组丢失则重传）、序号（用于检测丢失的分组和冗余的分组）、确认（接收方告知发送方正确接收分组以及期望的下一个分组）、否定确认（接收方通知发送方未被正确接收的分组）、窗口和流水线（用于增加信道的吞吐量）。

数据结构与算法

- 算法 3：最常用的排序——快速排序
  sort and quick sort，快排的思想是挖坑填数+分治。
- 一道腾讯面试题：厉害了我的杯（学到了）
  解题方法1：二分法；解题方法2：分段查找区间法；解题方法3：基于数学方程的方法；解题方法4：动态规划法（学到了），用公式来描述就是：`w(n, k) = 1 + min{max(w(n -1, x -1), w(n, k - x))}, x in {2, 3, ……, k}`（n是杯子数，k是楼层数）
- 如何有效的写算法题
  LeetCode上的题大致分为三种类型：考察数据结构：比如链表、栈、队列、哈希表、图、Trie、二叉树等；考察基础算法：比如深度优先、广度优先、二分查找、递归等；考察基础算法思想：递归、分治、回溯搜索、贪心、动态规划。
- 浅谈什么是分治算法（学到了）
  分治思想下的全排列问题、归并排序问题、快速排序问题、汉诺塔问题。
- 2018.08求职面经
  乱序数组中第k大的数，乱序数组中的中位数：快排指针，O(N)。
- 【视频讲解】LeetCode 第 1 号问题：两数之和
- 年会抢红包策略

数据库

- 腾讯面试：一条SQL语句执行得很慢的原因有哪些？
  补充学习：数据库引擎（InnoDB支持事物处理和外键，但是慢一点、ISAM和MyISAM空间和内存使用低，插入数据快）、数据库编码（`character_set_client`、`character_set_connection`、`character_set_database`、`character_set_results`、`character_set_server`、`character_set_system`）、数据库索引(主键索引、聚集索引和非聚集索引)等基础知识点。
  一条SQL语句执行很慢的原因分为两类：1）大多数情况下正常，偶尔很慢：（1）数据库在刷新脏页，例如redo log写满了需要同步到磁盘；（2）执行的时候遇到锁，如表锁，行锁；2）一直都很慢：（1）没有用上索引：例如该字段没有索引；由于对字段进行运算、函数操作导致无法用索引；（2）数据库选错了索引，比较聚集索引到主键索引和直接全表搜索的扫描行数，有可能因为采样问题判断有误，走了全表扫描而不走索引。
- 这大概是最全的sql优化方案了

计算机技术基础

语言

- 万字长文深度解析Python装饰器
- Python3 迭代器与生成器
  Python：迭代器有两个基本的方法：iter()和next()，字符串、元组、列表等可迭代对象都可用于创建迭代器（这是因为这些类内部都实现了`__iter__()`函数，调用iter()之后，变成了一个

`list_iterator`的对象，会发现增加了`__next__()`方法，所有实现了`__iter__`和`__next__`两个方法的对象，都是迭代器），迭代器是带状态的对象，它会记录当前迭代所处的位置，以方便下次迭代的时候获取正确的元素，`__iter__`返回迭代器自身，`__next__`返回容器的下一个值。生成器：使用了yield的函数被称为生成器，调用了一个生成器函数，返回的是一个迭代器对象，生成器可以看成是迭代器。

- python 黑科技之迭代器、生成器、装饰器
- Python的高级特征你知多少？来对比看看
  Python：lamda匿名函数，功能是执行某种简单的表达式或运算，而无需完全定义函数；Map函数是一种内置的python函数，可以将函数应用于各种数据结构中的元素；Filter内置函数与Map函数类似，但是只返回应用函数返回True的元素；Itertools模块是处理迭代器的工具集合，迭代器是一种可以在for循环语句中使用的数据类型；Generator函数是一个类似迭代器的函数。
- 为什么要使用 Go 语言？Go 语言的优势在哪里？
  Go：go的优势和go的用处。go的优势主要有：静态语言，多并发，跨平台，可直接编译成机器码，丰富的标准库等。go的用处主要有服务器编程、网络编程、分布式系统、内存数据库、云平台。
- Gin实践 连载一 Golang介绍与环境安装
  Go：go的环境安装，环境安装后各个文件夹的含义；go的工作区，工作区各个文件夹的含义。
- ruby-on-rails – Ruby和JRuby有什么区别
  Ruby：Ruby是一种编程语言，我们一般说的Ruby解释器是指CRuby，CRuby在本地C语言解释器环境中运行，JRuby是一个采用纯Java实现的Ruby解释器，JRuby在Java虚拟机中运行。

框架

- Gin - 高性能 Golang Web 框架的介绍和使用
  Gin：是用Go编写的一个Web应用框架。
- spring boot与spring mvc的区别是什么？
  Spring—》Spring MVC—》Spring Boot。

工具

- spark与storm的对比
  大数据技术工具-计算类型：从实时计算模型、实时计算延迟度、吞吐量、事物机制、健壮性/容错性、动态调整并行度等方面来比较。spark streaming是准实时模型，对一个时间段内的数据收集起来，作为一个RDD，再处理，实时计算延迟度为秒级，吞吐量大，支持事物机制但不够完善，健壮性一般，不支持动态调整并行度；而storm是纯实时模型，来一条数据，处理一条数据，实时计算延迟度为毫秒级，吞吐量小，支持完善的事物机制，健壮性强，支持动态调整并行度。**应用场景**：对于storm，可以在纯实时不能忍受1秒以上延时的场景下使用；对于实时计算的功能中，要求可靠的事物机制和可靠性机制，即数据处理完全就精确，也可以考虑storm；如果还需要针对高峰低峰时间段，动态调整实时计算程序的并行度，以最大限度利用资源，也可以考虑storm；如果项目中就是纯粹的实时计算，不需要在中间执行SQL交互式查询等其他操作，用storm是较好的选择。反之如果不要求纯实时，不要求可靠的事物机制，不要求动态调整并行度，可以考虑spark streaming，spark streaming最大的优势在于处于spark生态技术栈中，从项目的宏观角度考虑，如果不仅要求实时计算，还要离线批处理、交互式查询，而且在实时计算中，也会牵扯到高延迟批处理、交互式查询等功能，那么可以用spark core开发离线批处理，spark sql开发交互式查询，用spark streaming开发实时计算，无缝整合，给系统提供高扩展性，这个特点大大增强了spark streaming的优势。两个框架擅长的细分场景不同。
- 子雨大数据之Spark入门教程(Python版)（比较重要）
- 日志采集系统flume和kafka有什么区别及联系，它们分别在什么时候使用，什么时候又可以结合？
  大数据技术工具-中间件类型：可以把kafka理解成中间件，或是cache系统，或是数据库，主要作用是维稳。可以把flume理解成日志数据的主动收集，与kafka相比，很难推动线上应用修改接口往kafka中写入数据。

- logstash 和 flume 之间的优劣，和各自所适合的场景？
  大数据技术工具-Agent类型：看需求，logstash和flume都是作为agent的存在，logstash有更多的插件，有更好的配套产品elasticsearch等，但是logstash的开发语言是ruby，运行环境是JRuby，而且传输数据可能会丢失；flume内部有机制确保传输一定量级数据不丢失的问题，flume的开发语言是Java，易于二次开发，但是不足是jvm占用内存有点大。
- Mac快捷键大全
  MAC：基础快捷键：截图、在应用程序中、文本处理、在finder中、在浏览器中；MAC启动和关机时的快捷键。
- 常用 Git 命令单
  Git：远程仓库-》本地仓库-〉暂存区-》工作区，git add .、git commit -m message、git push。
- tshark统计分析pcap包

技术

- 解码与xss(**原文中有一处错误"html实体编码后"应该是`&#x5c;&#x75;&#x37;&#x32;`产生的原因就是浏览器的html自解码**)
  浏览器技术-解码顺序：浏览器解码主要涉及到两个部分：渲染引擎和js解析器。解码顺序：在什么环境下就进行什么解码，解码顺序为：最外层的环境对应的编码最先解码。举个例子:在`<a href=javascript:alert(1)>click</a>`中alert(1)处在html->url->js环境中。
  `1、<a href=javascript:al\u65rt(1)>click</a>` 采用unicode编码e，html和url环境下都不能解码，只有在js环境下才能解码为字符e，所以不会弹窗
  `2、<a href=javascript:al%65rt(1)>click</a>` 采用url编码，在执行js前，url解码%65，所以到了js引擎启动时，看到了完整的alert(1)
  `3、<a href=javascript:alert(1)>click</a>` html实体解码先执行了
  `4、<a href=java%61script:alert(1)>click</a>` 在url解码环节，不会认为javascript是伪协议，会出现错误。
  `5、<a href=# onclick="alert('&#x27;&#x29;;alert&#x28;&#x27;2');">click</a>` htmlparser会优先于JavaScript parser执行，所以解析过程是htmlencode的字符先被解码，然后执行JavaScript事件
  浏览器解码顺序是XSS中bypass的基础。
- 数据分析与可视化：谁是安全圈的吃鸡第一人（学到了）
  数据分析与可视化：收集数据集--->观察数据集--->社群发现与社区关系--->玩家画像。
- dockerfile 和 docker-compose 的关系
  docker技术：文件和文件夹的关系。
- dockerfile 与 docker-compose的区别是什么?
  docker技术：docker-compose是编排容器的。
- 堡垒机是什么？
  堡垒机技术：为访问集群限定一个入口；方便权限控制以及监控。
- 产品的可行性需从哪几个方面分析？
  可行性分析：产品可行性分为：技术可行性、经济可行性、社会可行性，其中我关注的是技术可行性。技术可行性主要从竞争对手功能比较、技术风险及规避方法、易用性及用户使用门槛、产品环境依赖性等方面衡量。
- Nginx、Gunicorn在服务器中分别起什么作用？
  应用服务器：Nginx部署场景：负载均衡（tornado之类的框架只支持单核，所以多进程部署需要反向负载均衡。gunicorn本身就是多进程其实不需要）、静态文件支持、抗并发压力、额外的访问控制。
- 维基百科：Kerberos
  Kerberos：Kerberos的基本描述、协议内容和具体流程。
- 什么是**微服务架构**？

底层研究

- python requests库流程简析
  python requests库实现：socket->httplib->urllib->urllib3->requests。requests.get的内部调用流程：requests.get->requests()->Session.request->Session.send->adapter.send->HTTPConnectionPool(urllib3)->HTTPConnection(httplib)。
  1、socket：是TCP/IP最直接的实现，实现端到端的网络传输
  2、httplib：基于socket库，是最基础最底层的http库，主要将数据按照http协议组织，然后创建socket连接，将封装的数据发往服务端
  3、urllib：基于httplib库，主要对url的解析和编码做进一步处理
  4、urllib3：基于httplib库，相较于urllib更高级的地方在于用PoolManager实现了socket连接复用和线程安全，提高了效率
  5、requests：基于urllib3库，比urllib3更高级的是实现了Session对象，用Session对象保存一些数据状态，进一步提高了效率
- XGBoost原理和底层实现剖析（学到了）
  XGBoost：从树的分数（目标函数：损失函数（二阶展开）+正则项），树的结构（分裂决策（预排序））方面理解。
- Lightgbm 直方图优化算法深入理解
  Lightgbm：相较于预排序而言，lgb采用了直方图来处理节点分裂，寻找最优分割点。算法思想：在训练前预先把特征值转化为bin value，也就是对每个特征的取值做分段函数，将所有样本在该特征上的取值划分到某一段（bin）中，最终把特征取值从连续值转化为离散值。直方图也可以用来做差加速，计算直方图的复杂度是基于桶的个数的。

安全技术

漏洞

- 对乌云漏洞库payload的整理以及Burp辅助插件
- boy-hack/wooyun-payload
- 从研究者视角看漏洞研究之2010年代
  漏洞研究：近10年的漏洞研究现状和趋势：1、后PC时代，控制流完整性成为新的系统安全基础性防护机制。2、令人惊喜的硬件安全特性和硬件安全漏洞。3、旧瓶装新酒，移动设备的安全设计实现弯道超车。4、网络入口争夺战愈演愈烈，网络入口有：浏览器、WiFi协处理器、基带、蓝牙、路由器、即时通信设备、社交软件、邮件客户端、传统PC和服务器。5、自动化漏洞挖掘和利用仍需提高。

Web安全

- 一篇文章带你深入理解漏洞之 XXE 漏洞
  XXE漏洞：XXE的原理：调用外部实体，XXE的利用：利用通用实体、参数实体、外部实体、内部实体进行文件读取，内网主机和端口探测、内网RCE（php下需要expect扩展的支持）
- mysql无逗号的注入技巧
  注入攻击：sql注入、xml注入（一种标记语言，通过标签对数据进行结构化表示）、代码注入（eval类）、CRLF注入（\r\n）。Mysql injection：使用注释绕过空格，使用括号绕过空格，使用%20 %0a等符号替换空格；union查询下，使用join绕过逗号过滤，`select id,ip from client_ip where 1>2  union select * from  ( (select user())a JOIN  (select version())b );`使用`select case when（条件） then 代码1 else 代码2 end`绕过逗号过滤，`insert into client_ip (ip) values ('ip'+(select case when (substring((select user()) from 1 for 1)='e') then sleep(3) else 0 end));`
- SSRF漏洞利用与getshell实战（精选）
- SSRF漏洞中绕过过滤（IP限制）的几种方法总结
  SSRF：利用302跳转（xip.io、短地址、自写服务）；DNS 重绑定（绕过IP限制）；更改IP地址写法；利用解析URL所出现的问题：`http://www.baidu.com@192.168.0.1/`；通过各种非HTTP协议

- SSRF绕过方法总结
  SSRF：利用@；利用短地址；利用特殊域名xip.io；利用DNS解析（在域名上设置A记录）；利用进制转换；利用句号
- ThinkPHP 5.0.0~5.0.23 RCE 漏洞分析
- 浅析白盒审计中的字符编码及SQL注入(优秀，学到了)
  基于字符编码的注入攻击：一个gbk编码的汉字，占2个字节，一个utf-8编码的汉字，占用3个字节。宽字节注入是利用mysql的特性，mysql在使用gbk编码的时候，会认为两个字符是一个汉字（gbk下，前一个ascii码要大于128，才到汉字的范围；gb2312的编码取值范围：高位`0xA1-0xF7`，低位`0xA1-0xFE`，而`\`是`0x5c`，不在低位范围中，所以`0x5c`不是gb2312中的编码，所以不会被吃掉。把这个思路拓宽到所有的多字节编码，只要低位的范围中含有`0x5c`的编码，就可以进行宽字节注入）。防御方案一：`mysql_set_charset+mysql_real_escape_string`,考虑到连接的当前字符集。防御方案二：将`character_set_client`设置为`binary`（二进制），`SET character_set_connection=gbk, character_set_results=gbk,character_set_client=binary`。当我们的mysql接受到客户端的数据后，会认为他的编码是`character_set_client`，然后会将之将换成`character_set_connection`的编码，然后进入具体表和字段后，再转换成字段对应的编码。然后，当查询结果产生后，会从表和字段的编码，转换成`character_set_results`编码，返回给客户端。所以，我们将`character_set_client`设置成`binary`，就不存在宽字节或多字节的问题了，所有数据以二进制的形式传递，就能有效避免宽字符注入。防御过后调用iconv时也可能出现问题。使用iconv对utf-8转gbk时，利用方式是`錦'`，原因是它的utf-8编码是`0xe98ca6`，它的gbk编码是`0xe55c`，最后变成`%e5%5c%5c%27`，两个`%5c`就是`\`，正好把反斜杠转义了。使用iconv对gbk转utf-8时，利用方式直接用宽字节注入。一个gbk汉字2字节，utf-8汉字3字节，如果我们把gbk转换成utf-8，则php会每两个字节一转换。所以，如果`\'`前面的字符是奇数的话，势必会吞掉`\`，`'`逃出限制。为什么不能用`錦'`这种方式呢，根据utf-8编码规则，`\（0x0000005c）`不会出现在utf-8编码中，所以会报错。
- 客户端session导致的安全问题
- 一文洞悉DAST、SAST、IAST ——Web应用安全测试技术对比浅谈（学到了）
- 谈谈SAST/IDAST/IAST
- PHP 连接方式介绍以及如何攻击 PHP-FPM
- 一个GET请求拿到flag——XCTF 2018 Final PUBG(WEB 2) Writeup

渗透测试

- 一套实用的渗透测试岗位面试题
  代码执行函数：`eval、preg_replace+/e、assert、call_user_func、call_user_func_array、create_function`；命令执行函数：`system、exec、shell_exec、passthru、pcntl_exec、popen、proc_open`；img标签除了onerror属性外，还有其他获取管理员路径的方式吗？src指定一个远程的脚本文件，获取referer。
- 一套实用的渗透测试岗位面试题，你会吗？
- 我的面经，渗透测试

代码审计

- Java代码审计-层层推进

数据安全

- NO.27 闲扯 关于数据安全
  大数据技术、时代，**数据是很多公司最核心的资产**；传统的安全边界模糊，我们需要假设我们边界已经被渗透的同时，拥有纵深防御能力，保护信息的安全。所以在加强传统安全手段的同时，我们更应该直接把安全的重点放在数据本身上，这就是数据安全所做的工作。在做之前，有一个前提：我们要知道安全依然是为业务服务的（大部分企业安全情况下，业务>安全），所以要权衡安全性

和可用性。目前企业常用的措施主要有：数据分级、数据生命周期管理、数据脱敏&数据加密、数据防泄漏。
- 互联网企业数据安全体系建设

云安全

- 云安全，到底是什么一回事？
  云安全三大研究方向：云计算安全、安全基础设施的云化、云安全服务。在云安全未来发展趋势中也提到了数据安全协作，说明无论哪种场景，数据都是安全的重点关注对象。云安全服务可以看成厨师做饭（来自cdxy的ppt），云计算（能源）、算法（工具）、数据（原料）、工程师（厨师）、能做成什么样的饭（能提供的安全服务）

其他

- 安全资料：企业实验室、安全社区、安全团队、安全工具等

安全研究

APT检测

- APT detection based on machine learning
  APT检测模型：本篇论文提出一种APT检测模型，通过在APT生命周期的多个环节进行检测，并将各个环节告警事件进行关联，并使用机器学习训练检测模型。和我的想法略有相似，之前想过可以用图模型或者规则关联算法进行关联以此重构攻击链，但是本篇文章好像是把关联的事件集作为输入数据输入到一个预测模型中去训练。这么做的目的是要完整地描述一个APT场景下的安全事件集，降低误报率，提高准确率，避免传统APT单环节检测造成的漏报、误报的问题。但是本文也存在一些问题，比如缺少APT数据源问题，缺少安全数据一直是个难题，导致本文提出的模型未能在真实的环境中论证。

恶意样本

- 利用机器学习检测HTTP恶意外连流量（优秀）
  恶意HTTP外连流量检测：**总体思路**：**1、数据收集**，沙箱运行恶意样本，收集恶意流量，人工区分恶意流量和白流量，再根据威胁情报对恶意流量划分家族。**2、数据分析**（特征工程）：同一家族恶意外连流量的相似性，可以考虑使用聚类算法将同一家族的流量聚为一类，提取它们的共性，形成模板，再用模板检测未知流量。**3、算法：训练阶段**：提取HTTP外连流量--->提取请求头字段--->泛化--->相似度计算（**请求头中字段特异性加权再计算相似性**）--->层次聚类--->生成恶意外连流量模板（聚类中该字段并集作为该字段在模板中的值）。**检测阶段**：未知HTTP外连流量--->提取请求头字段--->泛化--->与恶意模板匹配--->判断相似度是否超过阈值（阈值确定）
- Cuckoo恶意软件自动化分析平台搭建
- Cuckoo 恶意软件分析环境
- Playing with Cuckoo
  Cuckoo沙箱：在搭建Cuckoo恶意样本分析环境的过程中遇到了很多坑，现在还印象深刻的还有**pip 换源** -i https://pypi.tuna.tsinghua.edu.cn/simple；配置agent.py到startup文件夹中；注意windows10、ubuntu16和windows7之间的网络关系，NAT和Host-Only模式。物理主机windows10装vmware，vmware装ubuntu16，ubuntu16装virtualbox和cuckoo server端，virtualbox装windows7 作为agent端。
- 恶意样本分析资源汇总

对抗机器流量

- 2018 Bad Bot Report
  对抗机器流量：安全对抗促使攻击手段进化，进入了自动化对抗的阶段，参差不齐的爬虫、撞库、

模拟器产生了大量的机器流量，这其中搜索引擎类的爬虫、自动更新的RSS订阅服务器产生了正常的机器流量，而恶意爬虫等模仿正常用户的请求产生了恶意的机器流量，模仿的程度也不同，简单点的恶意机器流量直接通过脚本产生，高级点的通过浏览器产生，比如headless browser，更高级的可以模拟鼠标移动和点击。可以根据网络环境(Amazon ISP、data centers、global hosting providers)、使用工具（机器流量的browser喜欢伪装成Chrome、Firefox、Internet explorer、Safari）、是否模仿人类交互，比如鼠标轨迹和点击来区分机器流量和正常用户流量。一旦它们发现我们尝试阻止它们，高级恶意机器流量APBs就会展现出persistent和adaptive，进行多模式转换。**防御：理解我方业务和敌方目标。抑制过时的UA/Browser；抑制知名的主机服务商；保护敏感API；根据源流量观察高低峰段（波形？）；调查该恶意机器流量的sign，即显著性标志；监控失败的登录尝试；监控未能正确验证礼品卡的失败次数；注意公开的数据泄露，以防撞库；。**

恶意URL检测

- Detecting Malicious URLs
  国内的安全算法和安全数据分析资料翻阅到了尽头，开始将矛头转向国外,追踪国外的机器学习在网络安全领域的应用的发展过程。以URL检测为例，可以衍生出很多适用场景，恶意网页检测，恶意通信活动，恶意web软件。
- Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs
  把恶意URL检测作为一种恶意网页检测的补充手段。数据：开源黑白URL样本，没什么特色；特征：词汇特征和基于主机的特征，特色一般，分析比较每子类特征，特色一般；模型：L1逻辑回归、SVM、Naive Bayes，没什么特色，分析比较每个模型，没什么特色；值得学习的是之后的结果再分析，分析了False Positive和False Negative等错误产生原因，Mismatched Data Sources，模型表现和特征表现。毕竟是十年前的论文。
- Identifying Suspicious URLs: An Application of Large-Scale Online Learning
- Exploiting Feature Covariance in High-Dimensional Online Learning

Red Team

- Red Team从0到1的实践与思考（学到了）
  Red Team的定义--->Red Team的目标（学习和利用已知真实攻击者的TTPs来攻击、评估现有防御能力的有效性以及识别防御体系的弱点并提出具体的应对方案、利用真实有效的模拟攻击来评估因为安全问题造成的潜在的业务影响）--->谁需要Red Team--->Red Team如何工作（基本构成：知识储备、基础架构、技术研究能力；工作流程：全阶段攻击模拟、分阶段攻击模拟；协作配合）--->Red Team的量化和考核（已知TTPs的覆盖率、检测率/检测时间/检测阶段、阻断率/阻断时间/阻断阶段）--->Red Team的成长与提高（仿真环境训练、漏洞分析与技术研究、外部交流与分享）
- ATT&CK APT组织TTPs总结
- ATT&CK全平台攻击技术总结
- 真实APT组织分析报告汇总

WAF

- 技术讨论 | 在HTTP协议层面绕过WAF
- 利用分块传输吊打所有WAF
- 从http协议层面和数据库层面绕过waf
- WAF攻防研究之四个层次Bypass WAF
- 对过WAF的一些认知

异常检测

- 异常检测的N种方法（学到了）
  异常检测的一大难点就是缺少ground truth，常见的方法是先用无监督方法挖掘异常样本，再用有监督模型融合多个特征挖掘更多异常。分别从时间序列（移动平均、同比和环比、STL+GESD）、统计（马氏距离、箱线图）、距离角度（KNN）、线性方法（矩阵分解和PCA降维）、分布（相对

熵KL散度、卡方检验）、树、图、行为序列、有监督模型（可以自动组合较多特征，比如GBDT）等角度检测异常。

- 机器学习-异常检测算法（一）：Isolation Forest
- 机器学习-异常检测算法（二）：Local Outlier Factor
- 机器学习-异常检测算法（三）：Principal Component Analysis
- 什么是一类支持向量机（one class SVM），是指分两类的支持向量机吗？
- 异常检测算法之IsolationForest
- 异常挖掘，Isolation Forest
- 异常检测初尝试
- 机器学习加持下的时序类数据异常智能监控
- 海量运维日志异常挖掘
- 数据预处理-异常值识别
- Abnormal Detection（异常检测）和 Supervised Learning（有监督训练）在异常检测上的应用初探
- 数据挖掘中常见的「异常检测」算法有哪些？ - 微调的回答 - 知乎
  1、介绍常见的无监督异常检测算法及实验；2、对比多种算法的检测能力；3、对比多种算法的运算开销；4、总结并归纳如何处理异常检测问题。1.1）统计与概率模型：假设分布与假设检验，一维与多维，特征独立与特征相关，欧式距离与马氏距离；线性模型：低维空间嵌入，特征向量特征空间与协方差矩阵，欧式距离与马氏距离，PCA与Soft PCA与One-Class SVM；基于相似度衡量的模型：密度、距离、夹角、划分超平面、聚类；集成异常检测与模型融合。1.2）从实验结果图的决策边界验证算法之间的联系性。2.1）模型检测效果对比，Isolation Forest和KNN表现稳定；基于距离度量的KNN等模型受数据维度影响较大。3.1）数据量和数据维度对算法开销也有影响。Isolation更适合高维空间。4.1）实验结果带来了异常检测模型选择的思路：中小数据集KNN和MCD比较稳定，中大数据集Isolation Forest稳定；模型效果与模型效率往往是对立的，比如PCA与MCD；异常检测往往是非监督的，因此稳定比忽高忽低的性能更重要；简单的模型效果也可能很好。4.2）对于一个全新的异常检测问题，可以遵循以下步骤分析：A、对数据的了解，数据的分布，异常的分布，可根据假设选择模型；B、解决的问题是否有标签，如果有，一定不能浪费；C、如果可能的话，尝试不同的算法，尤其是对数据的了解有限时；D、根据数据的特点选择算法；E、无监督异常检测模型验证结果并不容易，可采用半自动的方式，对于置信度高的放过，对置信度低的人工审核；F、异常的趋势和特征往往在不断变化，因此模型需要重训练及调整策略；G、不要完全依赖模型，尝试使用半自动化的策略：人工规则+检测模型。人工规则还是很有用的，不要尝试一步到位的使用数据策略代替现有规则。
- 梳理 | 异常检测
- Anomaly Detection Isolation Forest&Visualization
- Anomaly Detection with Time Series Forecasting

图与安全

- 图/Louvain/DGA乱谈
  图承载者拓扑信息，而拓扑信息可以看作一种特征维度，有些攻防场景有明显的拓扑特征。Louvain算法的关键点是图的边的权重，在具体的攻防场景下需要专门研究，例如在DGA场景下，域名A与B的相关性(weight)=同时访问过A和B域名的IP数量。cdxy师傅用SQL实现了这种逻辑。
- 社区发现算法 - Fast Unfolding（Louvian）算法初探
- A DGA Odyssey PDNS Driven DGA Analysis

AI安全

- 从安全视角对机器学习的部分思考
- 中科院信工所发布《深度学习系统的隐私与安全》综述论文，187篇文献总结
- Towards Privacy and Security of Deep Learning Systems: A Survey
  AI安全的攻击面：训练阶段和测试阶段的数据和模型方面，攻击有数据中毒和对抗性样本，模型提取和模型反转等。

强化学习与安全

- Deep Exploit: Fully automatic penetration test tool using Machine Learning
- Github:Deep Exploit
- Github:GyoiThon

人工智能

算法体系

- 机器学习算法集锦：从贝叶斯到深度学习及各自优缺点
  算法知识框架：主要从算法的定义、过程、代表性算法、优缺点解释回归、正则化算法、人工神经网络、深度学习||决策树算法、集成算法||支持向量机||降维算法、聚类算法||基于实例的算法||贝叶斯算法||关联规则学习算法||图模型。
  个人理解：回归系列主要基于线性回归和逻辑回归衍生，包括回归、正则化算法、人工神经网络、深度学习；树系列主要基于决策树衍生，包括决策树和基于树的集成学习算法；支持向量机属于老牌算法；降维算法和聚类算法主要基于数据的内在结构描述数据；基于实例的算法实际上并没有训练的过程，代表性算法是KNN，基于记忆的学习；贝叶斯算法利用贝叶斯定理计算输出概率；关联规则学习算法能够提取数据中变量之间的关系的最佳解释；图模型是一种概率模型，可以表示随机变量之间的条件依赖结构。
- Categories of algorithms non exhaustive (学到了)
  算法知识框架：学到了搭建自己的算法体系。

基础知识

- HTTP DATASET CSIC 2010
  安全数据集-CSIC2010：基于e-Commerce Web应用自动化生成的安全数据集，包含36000个正常请求和25000个异常请求，异常请求包括：SQL注入、缓冲区溢出、信息收集、文件泄露、CRLF注入、XSS等。
- 分类模型的性能评估——以 SAS Logistic 回归为例 (3): Lift 和 Gain
- 机器学习中非均衡数据集的处理方法？
  非均衡数据集：上采样和下采样、正负样本的惩罚权重（scikit-learn的SVM为例：class_weight: {dict,'balanced'}）、组合/集成方法（从大样本中抽取多个小样本训练模型再集成）、特征选择（小样本量具有一定规模的时候，选择显著型的特征）
- 机器学习算法中 GBDT 和 XGBOOST 的区别有哪些？
  算法比较：GBDT基分类器为CART，XGB的分类器可以是多种基分类器，比如线性分类器，这时候就相当于L1、L2正则项的逻辑回归或线性回归；传统的GBDT在优化时用到的是一阶导数，XGB则对损失函数进行了二阶泰勒公式的展开，精度变高；XGB并行处理（特征粒度的并行，对特征值进行预排序存储为block结构，在进行节点分类的时候，需要计算每个特征的增益，最终选择增益最大的那个特征去做分类，那么各个特征的增益计算就可以开多线程进行），相对于GBM速度飞跃；剪枝时，当新增分类带来负增益时，GBM会停止分裂，而XGB一直分类到指定的最大深度，然后进行后全局剪枝；从最优化的角度来看，GBDT采用的是数值优化的思维，用的最速下降法去求解Loss function的最优解，其中用CART决策树去拟合负梯度，用牛顿法求步长，而XGB用的是解析的思维，对Loss function展开到二阶近似，求得解析解，用解析解作为Gain来建立决策树，使得Loss function最优。
- SVM和logistic回归分别在什么情况下使用？
  算法使用场景-SVM和逻辑回归使用场景：需要根据特征数量和训练样本数量来确定。如果特征数相对于训练样本数已经够大了，使用线性模型就能取得不错的效果，不需要过于复杂的模型，则使用LR或线性核函数的SVM。如果训练样本足够大而特征数较小的情况下，可以通过复杂核函数的SVM来获得更好的预测性能，如果样本没有达到百万级，使用复杂核函数的SVM也不会导致运算过慢。如果训练样本特别大，使用复杂核函数的SVM已经会导致运算过慢了，因此应该考虑引入更多特征，然后使用线性SVM或者LR来构造模型。

- gbdt的残差为什么用负梯度代替？
- 欧氏距离与马氏距离
- 机器学习算法常用指标总结
- 分类模型评估之ROC-AUC曲线和PRC曲线

机器学习

- 平均数编码：针对高基数定性特征（类别特征）的数据预处理/特征工程
- Mean Encoding
- kaggle编码categorical feature总结
- Python target encoding for categorical features
- Mean (likelihood) encodings: a comprehensive study
- 如何在 Kaggle 首战中进入前 10%
- kaggle竞赛总结
- 分享一波关于做Kaggle比赛，Jdata，天池的经验，看完我这篇就够了
- 为什么在实际的kaggle比赛中，GBDT和Random Forest效果非常好？
  有监督学习-树系列算法：单模型，gradient boosting machine和deep learning是首选。gbm不需要复杂的特征工程，不需要太多时间去调参数，dl则需要比较多的时间去调网络结构。**从overfit角度理解**，两者都有overfit甚至perfect fit的能力，overfit能力越强，可塑性越强，然后我们要解决的问题就是如果把模型训练的"恰好"，比如gbm里有early_stopping功能。线性回归模型就缺乏overfit能力，如果实际数据符合线性模型的关系，那可以得到很好的结果，如果不符合的话，就需要做特征工程，可特征工程又是一个比较主观的过程。树的优势，非参数模型，gbm的overfit能力强。而random forest的perfact fit能力很差，这是因为rf的树是独立训练的，没有相互协作，虽然是非参数型模型，但是浪费了这个先天优势。
- 【总结】树类算法认知总结
  有监督学习-树类算法：分类树和回归树的区别；避免决策树过拟合的方法；随机森林怎么应用到分类和回归问题上；kaggle上为啥GBDT比RF更优；RF、GBDT、XGBoost的认知（原理、优缺点、区别、特性）。
- 快速弄懂机器学习里的集成算法：原理、框架与实战
- 时间序列数据的聚类有什么好方法？
  无监督学习-时间序列问题：传统的机器学习数据分析领域：提取特征，使用聚类算法聚集；在自然语言处理领域：为了寻找相似的新闻或是把相似的文本信息聚集到一起，可以使用word2vec把自然语言处理成向量特征，然后使用KMeans等机器学习算法来作聚类；另一种做法是使用Jaccard相似度来计算两个文本内容之间的相似性，然后使用层次聚类的方法来作聚类。常见的聚类算法：基于距离的机器学习聚类算法（KMeans）、基于相似性的机器学习聚类算法（层次聚类）。对时间序列数据进行聚类的方法：时间序列的特征构造、时间序列的相似度方法。如果使用深度学习的话，要么就提供大量的标签数据；要么就只能使用一些无监督的编码器的方法。
- 凝聚式层次聚类算法的初步理解
  无监督学习-层次聚类：算法步骤：计算邻近度矩阵--->（合并最接近的两个簇--->更新邻近度矩阵）（repeat），直到达到仅剩一个簇或达到终止条件。
- 推荐算法入门（1）相似度计算方法大全
  无监督学习-层次聚类-相似性计算：曼哈顿距离、欧式距离、切比雪夫距离、余弦相似度、皮尔逊相关系数、Jaccard系数。

深度学习

- 一组图文，读懂深度学习中的卷积网络到底怎么回事？
  卷积神经网络：卷积层参数：内核大小（卷积视野3乘3）、步幅（下采样2）、padding（填充）、输入和输出通道。卷积类型：引入扩张率参数的扩张卷积、转置卷积、可分离卷积。
- [AI识人]OpenPose：实时多人2D姿态估计 | 附视频测试及源码链接
- 使用生成对抗网络(GAN)生成DGA
- GAN_for_DGA
- 详解如何使用Keras实现Wassertein GAN

- Wasserstein GAN in Keras
- WassersteinGAN
- keras-acgan

强化学习

- 深度强化学习的弱点和局限
- 关于强化学习的局限的一些思考
  强化学习的局限性：采样效率很差、很难设计一个合适的奖励函数。

应用领域

- 全球最全？的安全数据网站（有时间得好好整理一下）
- 初探机器学习检测 PHP Webshell
- 基于机器学习的 Webshell 发现技术探索
- 网络安全即将迎来机器对抗时代？
  智能安全-智能攻击：国外已经在研究利用机器学习打造更智能的攻击工具，比如深度强化学习，就是深度学习和强化学习的结合，可以感知环境，做出最优决策，可能被应用到漏洞扫描器里，使扫描器能够自动化地入侵目标。
  个人理解：国外已有案例Deep Exploit就是利用深度强化学习结合metasploit进行自动化地渗透测试，国内还没有看到过相关公开案例。由于学习门槛高、安全本身攻击场景需要精细化操作、弱智能化机器学习导致的机器学习和安全场景结合深度不够等一系列的问题，已有的机器学习+安全的大多数研究主要集中在安全防护方面，机器学习+攻击方面的研究较少且局限，但是我相信这个场景很有潜力，或许以后就成为蓝方的攻击利器。
- 人工智能反欺诈三部曲之：设备指纹
  智能安全-业务安全-设备指纹：ip、cookie、**设备ID**；**主动式设备指纹**：使用JS或SDK从客户端抓取各种各样的设备属性值，然后组合，通过hash算法得到设备ID；**优点**：Web内或者App内准确率高。**缺点**：主动式设备指纹在Web与App之间、不同的浏览器之间，会生成不同的设备ID，无法实现跨Web和App，不同浏览器之间的设备关联；由于依赖客户端代码，指纹在反欺诈的场景中对抗性较弱。**被动式设备指纹**：从数据报文中提取设备OS、协议栈和网络状态的特征集，并结合机器学习算法识别终端设备。**优点**：弥补了主动式设备指纹的缺点。**缺点**：占用处理资源多；响应时延比主动式长。
- 风险大脑支付风险识别初赛经验分享【谋杀电冰箱-凤凰还未涅槃】
  智能安全-业务安全-风控：个人理解见：https://github.com/404notf0und/Risk-Operation-Detection/blob/master/atec.ipynb。
- 机器学习在互联网巨头公司实践
  入侵检测：机器学习和统计建模的主要区别：机器学习主要依赖数据和算法，统计建模依赖建模者对数据特征的了解。两者的优缺点：机器学习：打标数据难获取，如果采用非监督学习，则性能不足以运维；机器学习结果不可解释。所以现在机器学习在做入侵检测的时候，一般都要限定一个特定的场景。统计建模：数据预处理阶段移除正常数据的干扰（重点关注查全率，强调过正常数据的过滤能力，尽可能筛除正常数据），构建能够识别恶意可疑行为的攻击模型（重点关注precision，强调模型对异常攻击模式判断的准确性，攻击链模型），缺点是泛化能力不足、在入侵检测一些场景中，模型易被干扰。我们的最终目的：大数据场景下安全分析可运维。
- Web安全检测中机器学习的经验之谈
  Web安全：从文本分类的角度解决Web安全检测的问题。数据样本的多样性，短文本分类，词向量，句向量，文本向量。文本分类+多维度特征。**与传统方法做对比得出更好的检测方式：传统方法+机器学习：传统waf/正则规则给数据打标；传统方法先进行过滤。**
- 词嵌入来龙去脉（学到了）
  NLP：DeepNLP的核心关键：语言表示--->NLP词的表示方法类型：词的独热表示和词的分布式表示（这类方法都基于分布假说：词的语义由上下文决定，方法核心是上下文的表示以及上下文与目标词之间的关系的建模）--->NLP语言模型：统计语言模型--->词的分布式表示：基于矩阵的分布表示、基于聚类的分布表示、基于神经网络的分布表示，词嵌入--->词嵌入（word embedding是神经网络训练语言模型的副产品）--->神经网络语言模型与word2vec。

- 深入浅出讲解语言模型
  NLP：NLP统计语言模型：定义（计算一个句子的概率的模型，也就是判断一句话是否是人话的概率）、马尔科夫假设（随便一个词出现的概率只与它前面出现的有限的一个或几个词有关）、N元模型（一元语言模型unigram、二元语言模型bigram）。
- 有谁可以解释下word embedding? - YJango的回答 - 知乎
  NLP：单词表达：one hot representation、distributed representation。Word embedding：以神经网络分析one hot representation和distributed representation作为例子，证明用distributed representation表达一个单词是比较好的。word embedding就是神经网络分析distributed representation所显示的效果，降低训练所需的数据量，就是要从数据中自动学习出输入空间到distributed representation空间的映射f（相当于加入了先验知识，相同的东西不需要分别用不同的数据进行学习）。训练方法：如何自动寻找到映射f，将one hot representation转变成distributed representation呢？思想：单词意思需要放在特定的上下文中去理解，例子：`这个可爱的 泰迪 舔了我的脸` 和 `这个可爱的 京巴 舔了我的脸`，用输入单词 x 作为中心单词去预测其他单词 z 出现在其周边的可能性（**至此我才明白为什么说词嵌入是神经网络训练语言模型的副产品这句话**）。用输入单词作为中心单词去预测周边单词的方式叫skip-gram，用输入单词作为周边单词去预测中心单词的方式叫CBOW。

综合素质

- 算法工程师必须要知道的面试技能雷达图（学到了）
  个人发展-必备技术素质：算法工程师必备技术素质拆分：知识、工具、逻辑、业务。在满足最小要求的基础上，算法工程师在这四个方面的能力是相对全面的，既包括"算法"，也包括"工程"，而大数据工程师则着重"工具"，研究员则着重"知识"和"逻辑"。
  针对安全业务的算法工程师就是安全算法工程师。为了便于理解，举个例子，如果用XGBoost解决某个安全问题，那么可以由浅入深理解，把知识、工具、逻辑、业务四个方面串起来：
  1.GBDT的原理（知识）
  2.决策树节点分裂时是如何选择特征的？（知识）
  3.写出Gini Index和Information Gain的公式并举例说明（知识）
  4.分类树和回归树的区别是什么（知识）
  5.与Random Forest对比，理解什么是模型的偏差和方差（知识）
  6.XGBoost的参数调优有哪些经验（工具）
  7.XGBoost的正则化和并行化分别是如何实现的（工具）
  8.为什么解决这个安全问题会出现严重的过拟合问题（业务）
  9.如果选用一种其他模型替代XGBoost或改进XGBoost你会怎么做？为什么？（业务、逻辑、知识）。
  以上，就是以"知识"为切入点，不仅深度理解了"知识"，也深度理解了"工具"、"逻辑"、"业务"。
- [校招经验] BAT机器学习算法实习面试记录(学到了)
  个人发展-面试经验：根据面试常遇到的问题再深入理解机器学习，储备自己的算法知识库。
- 机器学习如何才能避免「只是调参数」？(学到了)
  个人发展-职业发展：**机器学习工程师分为三种：应用型（能力：保持算法全栈，即数据、建模、业务、运维、后端，重点在建模能力，流程是遇到一个指定的业务场景应该迅速知道用什么数据做特征，用什么模型，这个模型在工程上的时效性和鲁棒性，最终会不会产生业务风险等一整套链路。预期目标：锻炼得到很强的业务敏感性，快速验证提出的需求）、造轮子型（多读顶会跟上时代节奏，且拥有超强的功能能力，打造ML框架，提供给应用型机器学习工程师使用）、研究型（AI Lab，读论文+试验性复现）。个人发展：锻炼业务能力和工程能力，未来几年成长规划还是算法全栈路线，技术上独挡一面，业务上带来kpi，以后快速晋升+带队。同时保持阅读习惯，多学习新知识。**
- 做机器学习算法工程师是什么样的工作体验？
  个人发展-工作体验：业务理解、数据清洗和特征工程、持续学习（增强解决方案的判断力）、编程能力、常用工具（XGB、TensorFlow、ScikitLearn、Pandas（表格类数据或时间序列数据）、Spark、SQL、FbProphet（时间序列））
- 大三实习面经（学到了）

- 如果你是面试官，你怎么去判断一个面试者的深度学习水平？
  个人发展-心得体会：深度学习擅长处理具有局部相关性的问题和数据，在图像、语音、自然语言处理方面效果显著，因为图像是由像素构成，语音是由音位构成，语言是由单词构成，都有局部相关性，可以构造高级特征。
- 面试官如何判断面试者的机器学习水平？ - 微调的回答 - 知乎
  个人发展-心得体会：考虑方法优点和局限性，培养独立思考的能力；正确判断机器学习对业务的影响力；学会分情况讨论（比如深度学习相对于机器学习而言）；学习机器学习不能停留在"知道"的层次，要从原理级学习，甚至可以从源码级学习，知其然知其所以然，要做安全圈机器学习最6的。
- 两年美团算法大佬的个人总结与学习建议
  个人发展-心得体会：算法的基本认识（知识）、过硬的代码能力（工具）、数据处理和分析能力（业务和逻辑）、模型的积累和迁移能力（业务和逻辑）、产品能力、软实力。
- 阿里技术副总裁贾扬清：我对人工智能的一点浅见
  行业发展-AI发展：神经网络和深度学习的成功与局限，成功原因是大数据和高性能计算，局限原因是结构化的理解和小数据上的有效学习算法。**AI这个方向会怎么走？**传统的深度学习应用，比如图像、语音等，应该如何输出产品和价值？而不仅仅是停留在安防这个层面，要深入到更广阔的领域。除了语音和图像之外，如何解决更多问题？而不仅仅是停留在解决语音图像等几个领域内的问题。

企业安全建设

安全开发

- 安全扫描自动化检测平台建设(Web黑盒 中)
- 带你读神器之KunPeng源代码分析

安全数据分析

- Data-Knowledge-Action: 企业安全数据分析入门（优秀，学到了）
  综述：**1、让模型理解业务，基于业务历史行为建立异常基线，在异常的基础上检测威胁；将运营结果反馈到模型，将误报视作正常行为回流。2、安全运营可运营，降低事件调查成本，自动化信息收集与聚合。3、随着数据的积累，安全数据分析将向基于图结构的高级知识表达方式发展。（这点深表赞同）4、对场景、攻击模式、数据的认识深度，远比选择工具重要。**
- Security Data Science Learning Resources
  综述：作者的研究点也是安全数据科学，整理了一些学习方法和学习资源。学习方法主要分为三个方面：**谷歌学术、Twitter、安全会议**。谷歌学术关注知名研究者以及他们新出的文章，关注引用了你关注的文章的文章，Twitter关注细分安全领域的人群，关注安全会议以及会议议程。学习资源：书籍和课程。
- 快速搭建一个轻量级OpenSOC架构的数据分析框架（一）(学到了)
  框架：**行文思路：由粗变细（由框架到举例子（由框架到场景到实际架构））。OpenSOC介绍（框架组成和工作流程）---》构建轻量级OpenSOC（聚焦具体场景和工具及具体架构）---》搭建步骤（每一步的环境搭建及配置）---》效果展示。**
- 先知talk：从数据视角探索安全威胁
- 大数据威胁建模方法论（学到了很多）
- 安全日志维度随想
- 数据安全分析思想探索
- DataCon 2019: 1st place solution of malicious DNS traffic & DGA analysis（学到了）
  我的理解：涉及的知识点有：安全场景：DNS安全；数据处理：tshark工具的使用，MaxCompute和SQL的使用，PAI预分析和可视化；特征工程：DNS_type、src_ip维度的特征；异常检测算法：单特征3sigma检测；人工提取特征规则。
  第一小题DNS恶意流量的异常检测：个人吸收80%，整理流程无障碍，每步流程中的**细节和工具**还未完全掌握，比如DNS安全场景了解不全面、tshark的大量数据解析、统计特征的全面提取、SQL

语句做特征化；
第二小题DGA的多分类：个人吸收50%，流程搞懂了，但是对一些问题的理解还不到位，比如社区算法

安全检测

- 关于风控预警体系的搭建方案
  业务安全-风控：快速发现异常和准确定义风险。通过核心指标的变化发现异常片段及实体、通过聚类手段发现异常簇下全部实体；异常实体抽样--->无感知人工审核--->有针对性制定风险阈值
- 从传统安全转行风控领域的心路历程，兼谈黑产和风控行业趋势
  业务安全-风控：风控领域斗争日趋激烈，黑产已经从高度专业化、分工明确的团伙进化为产业化运作的公司，现在风控需要有基础安全技术支撑（传统安全），随着司法机关对黑灰产的高压打击，未来大企业会关注风控供应商的产品能力和合规合法性。
- 风控模型师面试准备--技术篇
- 风控模型实战--"魔镜杯"风控算法大赛
- 风控用户识别方法
- github:sladesha
- 多算法识别撞库刷券等异常用户
- DNS Tunnel隧道隐蔽通信实验 && 尝试复现特征向量化思维方式检测
- 企业安全建设之HIDS
- 保障IDC安全：分布式HIDS集群架构设计
- 点融开源AgentSmith HIDS--- 一套轻量级的HIDS系统
- 企业安全建设—基于Agent的HIDS系统设计的一点思路
  入侵检测-主机入侵检测系统：美团的系统性实践非常值得学习。从需求描述，产品经理提出需求->分析需求，总结产品架构要符合的特性->技术难点，分析遇到的技术挑战->架构设计与技术选型->分布式HIDS集群架构图->编程语言选择->产品实现。
- 基于统计分析的ICMP隧道检测方法与实现

安全产品

- 收集一些比较优秀的开源安全项目，以帮助甲方安全从业人员构建企业安全能力**(学到了)**
  开源安全产品：包括资产管理、安全开发、自动化代码审计、安全运维、堡垒机、HIDS、网络流量分析、蜜罐、WAF、企业云盘、钓鱼网站系统、Github监控、风控、漏洞管理、SIEM/SOC。

安全运营

- 我理解的安全运营
  公司是为产出付费，而不是为知识付费。安全运营是以解决问题为导向。安全运营的主要职责和**技能需求**：安全、研发、运维背景；**较好的沟通能力**；一定的**项目管理能力**；具备数据意识。
- 再谈安全运营
  安全运营的**Why**：**安全的风险直观化**，表象被戳破；**安全建设期已过，开始追求结果。**
  安全运营的**What和How**：抓住**主要矛盾**和次要矛盾**不放过**，尽力解决。

安全管理

- 企业安全建设技能树v1.0发布
  包括六大部分：说明、安全观、安全治理、通用技能、专业技能、优质资源。

安全思考

- 谈谈互联网企业安全的发展方向
  企业安全发展方向：由浅入深分为四个目标：1、消灭漏洞驱使，第一个目标是让工程师写出的每一行代码都是安全的，由此诞生SDL，SDL又衍生技术研究和技术产品，比如代码安全扫描工具的研究和fuzzing。2、有了SDL还无法100%安全，所以第二个目标是让所有已知、未知的攻击，都

能在第一时间发现，并迅速报警和追踪。挑战：海量数据和复杂需求 方案：超强计算能力和立体化模型。3、第三个目标是让安全成为公司的核心竞争力，深入到每个产品的特性中，能够更好地引导用户使用互联网的习惯。4、最后一个目标是能够观测到整个互联网安全趋势的变化，对未来一段时间内的风险做出预警。
在互联网公司做安全一定要有想象力，同时紧密关注其他技术领域的发展，这样就不会止步于几种漏洞的研究，而会发现有很多有趣的事情正等着去做，这是一个非常宏伟的蓝图。

- 以攻促防：企业蓝军建设思考
- 赵彦的CISO闪电战 | 两年甲方安全修炼之路（学到了）
  范围对象（公司业务、挑战及安全需求（纵深防御、自身供应链安全、赋能第三方安全））--->目标设定（当下需求设定和未来发展）--->挑战（团队全栈（知识结构和技能对口主营业务）、工程能力、管理能力）--->分解安全体系（通用领域安全建设沙盘图：研发安全、IT安全、基础设施安全、数据安全、终端安全、业务安全、隐私与安全合规）--->实现和应对（安全治理框架、业界对标（真正落地能力，demo不算有此能力）、安全研究）。**总的来说，就是全栈技术视野（努力从技能层面上升到技术视野层面）+安全管理能力。**

安全架构

- 网络安全架构 | 通过安全架构提升安全性](https://mp.weixin.qq.com/s/m90wYaEvHzfsdgnFHMGxCw)

红蓝对抗

- 【红蓝对抗】大型互联网企业安全蓝军建设（学到了）
  红蓝对抗的Why：检验企业安全防护体系；梳理风险盲点和攻防场景，为安全建设提供有价值的建议；安全价值的体现；强化业务同事的安全意识。
  红蓝对抗的What：入侵发现率；攻防场景发现率；攻击覆盖度；演戏频次/安全风险数/策略缺陷数/效率提升；攻击成本；目标达成率。
  红蓝对抗的How：仿真APT--->蓝军团队需要沉淀出一套体系化的攻击手法知识库和武器库--->ATT&&CK矩阵框架。
  红蓝对抗Do过程中的挑战：效率/收益；攻击成本量化；来自业务的挑战（红蓝对抗的核心目标是为业务保驾护航）。
  红蓝对抗的Future：多层次多范围的蓝军；蓝军的自动化渗透平台/协同作战平台；蓝军能力对外输出。

安全发展

个人发展

面试

- 有关安全的面经, 实习, etc
  面试：滴滴、百度（2）、360（2）、阿里（6）、腾讯（3）、b站、华为、同花顺、蘑菇街。总的来看，大佬们好强，选择大多是甲方安全部。我的理解：看了大佬们的面经和被问到的问题，真的是五花八门，有bin方向的，有数据安全方向的，也有安全运营方向的等等，有一些参考价值，但是因为方向不同，不能生硬照搬，还是得发挥自己的专长，先做自己小领域的领域专家。
- 2018春招安全岗实习面试总结
- 腾讯2016实习招聘-安全岗笔试题答案详细解释
  笔试：设计一个安全的web身份验证方案：前端：验证码+csrf_token+基于时间戳加密生成随机数；把身份信息传输到服务器后台，并且设置同源策略（同源网站：域名、端口、协议）；服务器端验证客户端身份后，通过随机数加密session和cookie返回客户端；客户端与服务器端建立连接。
- 大型公司安全技术岗位面试杂谈
  面试：安全技术基础--->项目细节（知识深度，在擅长的领域碾压面试官，让面试官问不出有深度

的问题）--->挑战性问题的处理思路（知识面和行业认知能力，一般也不会脱离擅长领域，需要日常多读多想）--->行业深度认知能力和职业规划

- 2019 届阿里实习生内推实况是怎样的？ - 左左薇拉vera的回答 - 知乎（学到了）
- 十面阿里，七面头条，你猜我进阿里没？
  面试：Java版优秀面经，java必备。
- 书剑恩仇录之我与阿里巴巴（太强了）

职业发展

- 安全研究者的自我修养
- 安全研究者的自我修养（续）
- 安全人员发展方向杂谈
  甲方安全发展路线：硬核技术型--->大厂实验室和安全研究岗 非硬核技术型--->互联网企业安全建设之红蓝、技术运营、安全管理
- 安全从业者存在的意义
  个人发展：目标是帮助先进生产力解决好安全问题。这其中安全问题是信任的问题（信任支撑，原点支撑），是一个研究对抗的科学（人与人的对抗），是一个概率问题（安全架构）。安全是一门应用科学，随着每个时代的不同，可以有很多不同的技术手段和工具来完成各自的安全目标，因此安全从业者应该对新技术和先进生产力保持敏感和接受度，这会带来很多新的视角和能力，包括机器智能和区块链技术等。
- 安全团队在企业中的几个身份
  团队发展：安全团队应该以服务者和协作者的身份，用专业的安全能力给出一类安全问题的解决思路和方案并解决，防止安全问题发生多次。

行业发展

安全格局

- 最新统计2005-2017年国内科研单位在国际安全顶级会议中发表文章量统计
- 从内容产出看安全领域变化
  技术格局：企鹅等互联网巨头开始进行流量封锁，对安全从业人员影响很大，爬不到数据，api又有限，只能上升到app hook了；技术上安全分析、数据挖掘、威胁情报的比重越来越重，**AI已经不仅仅是噱头了，智能安全势不可挡**；安全的职业发展方面，越来越多大佬们开始转型业务安全、数据安全。
- 网络安全行业竞争格局浅析
  市场格局：基础安全防护（传统安全防护能力），中级安全防护（海量数据建模与分析能力），高级安全防护（云端威胁情报与分析能力），中高级安全防护市场广阔。此外，全文在多处凸显了人工智能技术，智能安全开始迈入开悟之坡了吗？！半数以上的人看好智能安全，也有人不看好智能安全，未来会怎么样，让我们拭目以待！
- ZoomEye 网络空间测绘——委内瑞拉停电事件对其网络关键基础设施和重要信息系统影响

安全产品

- C端安全产品的未来之路
  C端安全产品：移动端安全产品是否会像前几天PC端安全产品一样迎来春天？PC时代windows是一家独大的完全开放的平台，这让第三方安全公司能够在平台和用户之间产生价值的空间足够的大，但在移动端，安卓开始封闭，就不好说了。传统安全软件围绕病毒和欺诈，而围绕**个人信息安全的C端安全产品**有一线生机。
- 下一座圣杯 - 2019
  API安全：**应用安全的发展**：2015年预测，数据是新中心，身份是新边界，行为是新控制，情报是新服务。基础设施演进->交付方式的改变。2015年，应用安全领域的WAF产品是良机，由市场决定。**新形势与新机遇**：微服务、Serverless、边缘计算。市场中的交付方式发生变化。**跨细分领域且跨基础设施**：API安全横跨应用安全、数据安全和身份安全三大领域。API使用场景广泛，需要产品有全面覆盖多种不同基础设施的能力。

软实力

职业规划

- 至关重要：如何做好我们的职业规划**（学到了）**
    1. 认清自己，确认方向。按照**职业规划探寻模型**来思考
    2. 收集信息，心中有数。校招/社招/JD
    3. **目标设远，步步拆解。**拆解框架：时间/目标/准备(技术能力、汇报能力|领导能力、沟通能力、团队建设能力)
    4. **盘点能力，补足短板。**盘点软实力：沟通能力、执行能力、谈判能力、情绪管理、时间管理、分解能力、汇报能力、演讲能力、协作能力、组织能力、快速学习能力、PPT撰写能力、文字总结能力、聆听技巧、同事关系、与上司之间关系，盘点短板：沟通能力、汇报能力、演讲能力、PPT撰写、文字总结能力，对短板再按多个维度分类，比如是否易于评估并得到反馈和培养难度
    5. 学会展望，调整方向。唯一不可阻挡的是时间
- 数据科学家 (Data Scientist) 的核心技能是什么？

综合素质

逻辑思维与语言表达

- 如何解决思维混乱、讲话没条理的情况？（学到了）
  结构化思维->讲话有条理。
- 哪些思维方式是你刻意训练过的？**（学到了）**
  结构化思维
  金字塔思维：结论先行，以上统下，归类分组，逻辑递进。
  金字塔结构：纵向延伸，横向分类。
  如何得出金字塔结论：归纳法，演绎推理法。实际生活中，不是每时每刻都有相关的模型套用和演绎法的，这时候就用归纳法，**自下而上进行梳理，得出结论**，比如**头脑风暴**把闪过的**碎片想法**全部写下来，再**抽象与分类**，最后得出结论。
- 厉害的人是怎么分析问题的？（学到了）
  定义问题/描述问题：问题的本质是现实和期望的落差部分；明确期望值B'，精准定位现状B，，用B--->B'这个落差，精准描述问题。
  分析问题/解决问题：不能从现状B出发，找寻一条B--->B'的路径，要透过现象看本质。方法A，现实B，期望B'，变量C。**校准期望B'，重构方法A，消除变量C。**

管理

- “我是技术总监，你干嘛总问我技术细节？”
  （快速发展期、平稳期、衰退期等业务发展时期作为时间轴）（中高层管理者）（需要掌握）（应用场景、技术基础、技术栈中的技术细节）。技术基础要扎实，技术栈了解程度深（对技术原理和细节清楚），应用场景不能浮于表面。总的来说就是一句话：**技术细节与技术深度。**
- 如何在企业中从0-1建立一个数据/商业分析部门？(**学到了**)
  部门的定位和价值——>里程碑设计——->团队搭建——->构建IT数据——->前期管理。
  定位和价值是一个部门立足公司的根本：**做报表的部门VS做战略的部门**；业务其他公司的定位和公司内其他部门的认可；一定要会放大部门的价值和一定要走高层路线。
  设立长期目标并拆解里程碑：**公司业务目标--->公司战略--->部门目标--->部门里程碑--->工作计划**；设立里程碑的技巧？**借势、共赢、取巧、筑基**；借老板势，寻找1-2个老板的痛点问题解决；寻找利益相同的部门共建共赢；取巧摘已有的“桃子”；筑基数据链路梳理、数据清洗、系统互联、数据仓库设计、数据集市设计。
  基于里程碑进行团队搭建：切忌一步到位；审慎拉帮结派；遇到人才不可错过；学会“画饼”；注意

团队文化建设。
构建公司的数据IT能力：搭建基础且通用的数据流框架：应用层、归集层、加工层、分析层、展示层；
同时根据各种数据库选型指标选择对应的数据库存储产品，数据库选型指标比如容量、水平扩展性、查询实时性、查询灵活性、写入速度、事务、数据存储、处理数据规模、列扩展性。**在搭建数据框架中需要注意的点是：需要实现公司级别的业务数据架构。**基于业务对整个公司的数据进行体系化的梳理，任何的业务变化都会体现在数据之上，实现数据充分体现业务现状的目的。**要完成这一步的关键是完成公司级别的主数据管理**：明确各项数据的业务含义和口径、明确每个数据的职责单位、打通数据链路，推动数据共享。
引领团队走向胜利：做"排长"而不要做"军长"；让合适的人做合适的事；明确规则，及时兑现。

- 26岁当上数据总监，分享第一次做Leader的心得
  团队管理方面的基本功和方法论：定策略、建团队、立规矩、拿结果。
  定策略：要明确公司高层的真实目的；对自己的团队了如指掌；管理者专精的行业知识和经验。
  建团队：避免嫉贤妒能、职场近亲、玻璃心。
  立规矩：立规矩守规矩。
  拿结果：注意吃相。
  管理中常见的误区：**做管理后放弃原来专业**（要关注行业发展方向和前沿技术）；**过度管理**（要自循环的稳定成熟团队）；**过度追求团队稳定**（衡量团队稳定的核心标准不是人员的稳定，而是团队的效率和产出是否能够有持续稳定的增长）
- 什么特质的员工容易成为管理者
  公司内部晋升管理者：天时：企业/行业所处的阶段；地利：部门/业务所处的阶段；人和：人际关系+自身能力。
  跳槽成为管理者：大公司跳槽到小公司，寻找职业突破，弊端是跳出去容易跳回来难；成为行业内有影响力的人物，被大公司挖角。大部分人都是第一种情况，在大公司的同学要多一点耐心，通过努力在公司内晋升，因为曲线救国式的跳槽已经没有市场了。

思考

- 好的研究想法从哪里来
  研究的本质是对未知领域的探索，是对开放问题的答案的追寻。"好"的定义-》区分好与不好的能力-》全面了解所在研究方向的历史和现状-》实践法/类比法/组合法。 这就好比是机器学习的训练和测试阶段，训练：全面了解所在研究方向的历史和现状，判断不同时期的研究工作的好与不好。**测试**：实践法/类比法/组合法出的idea，判断自己的研究工作好与不好。

注意事项

- **领域点-线-面体系**：点：自己focus的领域；线：上游和下游；面：大领域。不要过度focus在自己工作的领域，要有全局化的眼光，特别是自己的上游和下游。
- **日常学习点-线-面体系**：点：自己focus的安全数据分析领域；线：安全/数据分析；面：全局安全内容/行业发展/职业规划。每日专研至少一小时小领域；每日精读至少半小时/至少一篇安全/数据分析/行业发展/职业规划精品文章；每日大量浏览增量文章/存量文章。**保持学习与思考的敏感性。**

附录

国外优质技术站点

- https://resources.distilnetworks.com
  站点概况：专注于机器流量对抗与缓解。
- http://www.covert.io
  技术栈：Jason Trost，专注于安全研究、大数据、云计算、机器学习，即安全数据科学。

- http://cyberdatascientist.com
  站点概括：专注于安全数据科学，提供网络安全、统计学和AI等学习资料，并提供14个安全数据集，包括：垃圾邮件、恶意网站、恶意软件、Botnet等。**没有secrepo.com提供的资料全面。**
- https://towardsdatascience.com
  站点概括：专注于数据科学。

国内优秀技术人

- http://michael282694.com
  技术栈：michael282694，数据分析挖掘产品开发、爬虫、Java、Python。
- https://www.cnblogs.com/LittleHann
  技术栈：LittleHann，我也不知道该怎么描述，Han师傅会的太多了，C++、Java、Python、PHP、Web安全、系统安全，不过目前好像做算法多一些。
- https://feei.cn
  技术栈：FeeiCN，专注自动化漏洞发现和入侵检测防御。
- http://www.yqxiaojunjie.com
  技术栈：xiaojunjie，专注于代码审计、CTF。
- 云雷
  技术栈：云雷，阿里云存储技术专家，专注于日志分析与业务，日志计算驱动业务增长。
- https://iami.xyz
  技术栈：iami，主要研究Web安全、机器学习，喜欢Python和Go。一直偷学师傅的博客。
- https://www.cdxy.me
  技术栈：cdxy，早先主要做Web安全，CTF，代码审计，现在主要做安全研究与数据分析，初步估算技术领先我1～2年，师傅别学了。
- http://www.csuldw.com
  技术栈：csuldw，专注于机器学习、数据挖掘、人工智能。
- https://molunerfinn.com/
  技术栈：molunerfinn，专注于前端，北邮大佬，和404notfound同级。

废弃

- Efficient and Flexible Discovery of PHP Vulnerability译文
- Efficient and Flexible Discovery of PHP Application Vulnerabilities原文
- The Code Analysis Platform "Octopus"
- A Code Intelligence System：The Octopus Platform